

การปรับปรุงระบบสารสนเทศศึกษาระดับบัณฑิตศึกษาโดยวิธีเชิงวัตถุ

OBJECT-ORIENTED SOFTWARE MAINTENACE FOR  
GRADUATE STUDENT INFORMATION SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

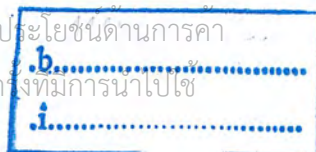
ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เลขหมู่.....ห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขทะเบียน.....

วันเดือนปี..... 55090 .....



การปรับปรุงระบบสารสนเทศนักศึกษาระดับบัณฑิตศึกษาโดยวิธีเชิงวัตถุ

OBJECT-ORIENTED SOFTWARE MAINTENACE FOR  
GRADUATE STUDENT INFORMATION SYSTEM



นางสาวณวริกา บุญเสนอ  
นายยุทธการ มิศิตา

อาจารย์ที่ปรึกษา  
ดร. วิศิษฎ์ หิรัญกิตติ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2546

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การปรับปรุงระบบสารสนเทศนักศึกษาในระดับบัณฑิตศึกษาโดยวิธีเชิงวัตถุ

OBJECT-ORIENTED SOFTWARE MAINTENANCE FOR GRADUATE  
STUDENT INFORMATION SYSTEM

คณะผู้จัดทำ นางสาวณวิภา บุญเสนอ รหัส 44015341

นายยุทธการ มิสิตา รหัส 44015343



.....อาจารย์ที่ปรึกษา

(ดร.วิศิษฎ์ ทรัพย์กิตติ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การปรับปรุงระบบสารสนเทศศึกษาระดับบัณฑิตศึกษาโดยวิธีเชิงวัตถุ

นางสาวณวิภา บุญเสนอ 44015341

นายยุทธการ มิตีลา 44015343

ดร.วิศิษฐ์ หิรัญกิตติ อาจารย์ที่ปรึกษา

ปีการศึกษา 2546

### บทคัดย่อ

ระบบสารสนเทศศึกษาระดับบัณฑิตศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ใช้อยู่ปัจจุบัน ยังประสบปัญหาในการทำงานอยู่หลายประการ เนื่องจากระดับการลงทะเบียนที่มีหลายขั้นตอน และในระบบข้อมูลทะเบียนนักเรียนนั้นมีความซับซ้อนและมีเงื่อนไขการตรวจสอบต่างๆ มากมาย อีกทั้งยังเปลี่ยนแปลงอยู่ตลอดเวลา ปัจจุบันการใช้งานยังต้องพึ่งพาการตรวจสอบโดยเจ้าหน้าที่เป็นหลัก ซึ่งอาจทำให้เกิดความผิดพลาดโดยมนุษย์ขึ้นได้ โครงการปริญญาโทนิพนธ์ของภาควิชาวิศวกรรมคอมพิวเตอร์ สจล. ในปีการศึกษา 2545 ได้มีการพัฒนาปรับปรุงระบบสารสนเทศศึกษาดังเดิมเพื่อแก้ไขปัญหาเหล่านี้ แต่ระบบที่พัฒนาขึ้นกลับยังมีข้อบกพร่อง ระบบยังทำงานได้ไม่สมบูรณ์ และการตรวจสอบเงื่อนไขบางอย่างไม่ได้นำมาพิจารณา ดังนั้นระบบดังกล่าวควรได้รับการปรับปรุงให้สมบูรณ์ยิ่งขึ้นเพื่อให้สามารถนำไปใช้งานได้จริง

ปฏิญานิพนธ์เรื่องนี้จึงได้ทำการศึกษา วิเคราะห์ ปรับปรุงและพัฒนาสารสนเทศศึกษาระดับบัณฑิตศึกษาเดิมนั้น โดยวิธีการพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-Oriented Software Development) ในการวิเคราะห์และออกแบบได้ใช้ภาษาสร้างแบบจำลอง UML และการพัฒนาระบบโดยใช้ภาษาจาวาระบบสารสนเทศศึกษาระดับบัณฑิตศึกษาที่ปรับปรุงขึ้นใหม่ประกอบด้วย 2 ส่วน คือ ส่วนที่เป็นแอปพลิเคชันพัฒนาขึ้นสำหรับเจ้าหน้าที่ทะเบียนโดยเฉพาะ เพื่อใช้ป้อนข้อมูลการลงทะเบียน เนื่องจากข้อมูลเหล่านี้มีความสำคัญและเป็นความลับ จึงออกแบบให้ส่วนนี้ใช้ได้เฉพาะบนเครื่องของผู้ใช้เฉพาะบุคคลเท่านั้น จึงได้พัฒนาเป็นแอปพลิเคชันจาวา สำหรับส่วนที่ให้นักศึกษาใช้นั้นจะเป็นเว็บแอปพลิเคชันเพื่ออำนวยความสะดวกกับนักศึกษาให้สามารถใช้บริการระบบสารสนเทศได้ในทุกสถานที่ ซึ่งในส่วนนี้ ได้พัฒนาโดยใช้เทคโนโลยี Java Servlet และ JavaServer Page (JSP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**OBJECT-ORIENTED SOFTWARE MAINTENANCE  
FOR GRADUATE STUDENT INFORMATION SYSTEM**

Monwipa Boonsaner

Yuttakarn Misila

Dr.Visit Hirankitti Advisor

Academic Year 2003

**ABSTRACT**

The graduate-student information system currently operated by the KMITL's School of Graduate Studies has suffered several problems. The system has still been incomplete and contained many errors, since it cannot cope with the complication of the current registration process which contains so many steps and those steps require many constraint checkings. Moreover, the system cannot meet with these frequently changing requirements. Therefore to overcome this problem the registrar has to perform manual checks himself/herself for the constraints that overlooked by the current system. Eventually, this can lead to human errors.

To solve this problem in 2002 a computer engineering project of the Computer Engineering Department was set up to redevelop a new graduate-student information system. The new system after the development worked well to some extends but was still incomplete. Still some constraints have not been checked and handled. To finally put it into work, it needed to be maintained and improved.

Therefore, in this year, 2003, our thesis has studied, analyzed, and maintained the 2002 information system using an object-oriented software development approach. We have employed UML (Unified Modeling Language) for the re-analysis and re-design phrases, and Java technology for the implementation and maintenance phrases. The new system we have developed consists of 2 subsystems. The first subsystem is a Java application; this is operated only by the registrar who uses the application to input and maintain the data used in the registration process. Since this kind of data must be maintained securely the system to manage it should be an application installed on an authorized user's machine. The second subsystem is a Java web application running on a web server. We use this web application to provide the information service to the students using any web browsers. To make the system services widely available to the any KMITL graduate student, the subsystem serving this purpose is inevitably a web application. This second subsystem has been developed using Java Servlet and JavaServer Page (JSP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี เนื่องจากทางคณะผู้จัดทำได้รับความร่วมมือ และความอนุเคราะห์จากบุคคลต่าง ๆ หลายฝ่าย ทั้งทางด้านการทำงาน กำลังใจ ทางคณะผู้จัดทำขอขอบพระขอบคุณท่านเหล่านั้นมา ณ โอกาสนี้

ดร.วิศิษฎ์ ทรัพย์กิตติ อาจารย์ที่ปรึกษาที่ได้ให้ความกรุณาช่วยเหลือ ให้คำปรึกษาที่เป็นประโยชน์ ชี้แจงข้อผิดพลาด และเสียสละเวลาดูแลเอาใจใส่ในการทำงานของพวกเรามาโดยตลอด

คณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ได้ให้ความรู้ทั้งทางด้านวิชาการและด้านการใช้ชีวิตในสังคมแก่พวกเรา

และต้องขอขอบพระคุณบุคคลที่มีความสำคัญที่สุดที่ทำให้มีวันนี้ได้ ก็คือ บิดา และ มารดาอันเป็นที่เคารพและรักยิ่งซึ่งได้อบรมเลี้ยงดูมาเป็นอย่างดี พร้อมทั้งได้ให้โอกาสในการศึกษาที่ดีที่สุด และให้กำลังใจในการทำงานเสมอมา คณะผู้จัดทำขอระลึกในพระคุณ และขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปตาราง	VI
สารบัญรูปภาพ	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีการดำเนินงาน	3
บทที่ 2 การพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุและยูเอ็มแอล	4
2.1 บทนำ	4
2.2 หลักการเชิงวัตถุ	5
2.3 ข้อดีของหลักการเชิงวัตถุ	7
2.4 การพัฒนาซอฟต์แวร์ด้วยวิธีเชิงวัตถุ	8
2.5 ภาษา Unified Modeling Language (UML)	11
บทที่ 3 เทคโนโลยีจาวาที่ใช้ในการพัฒนาระบบ	18
3.1 จาวาเซิร์ฟเล็ต (Java Servlet)	18
3.2 ภาษาเจเอสพี (Java Server Page : JSP)	20
บทที่ 4 เครื่องมือที่ใช้ในการพัฒนาฐานข้อมูล	23
4.1 เครื่องมือที่ใช้ในการพัฒนาฐานข้อมูล IBM DB2	23
4.2 Java Database Connectivity (JDBC)	23
บทที่ 5 การวิเคราะห์และออกแบบระบบทะเบียนนักศึกษาระดับบัณฑิตศึกษา	28
5.1 ฝ่ายทะเบียนนักศึกษาของบัณฑิตวิทยาลัย	28
5.2 ปัญหาที่ผู้ผู้ต้องการแก้ไข	29
5.3 การแก้ไขปัญหาระบบเดิม	29
5.4 ปัญหาที่เหลือและปัญหาที่เกิดจากระบบที่ได้พัฒนามาแล้ว	30
5.5 การวิเคราะห์ระบบเพื่อปรับปรุง	32
5.6 การออกแบบระบบโดยใช้ยูเอ็มแอล	33
5.7 การออกแบบการทำงานของฟังก์ชันต่าง ๆ	37

## สารบัญ (ต่อ)

	หน้าที่
5.8 การออกแบบรายละเอียดของคลาสต่าง ๆ	40
5.9 การออกแบบโครงสร้างของระบบ	41
5.10 การออกแบบระบบ เพื่อสร้างระบบสารสนเทศ	41
5.11 การออกแบบฐานข้อมูล	42
บทที่ 6 การเขียนโปรแกรมเพื่อสร้างระบบ	44
6.1 การสร้างฐานข้อมูล	44
6.2 การเขียนโปรแกรมในส่วนของแอปพลิเคชัน	46
6.3 การเขียนโปรแกรมในส่วนของเว็บแอปพลิเคชัน	49
บทที่ 7 การแสดงผลการทำงานของระบบ	50
7.1 ส่วนของแอปพลิเคชันสำหรับเจ้าหน้าที่	50
7.2 ส่วนของเว็บแอปพลิเคชันสำหรับนักศึกษา	56
บทที่ 8 บทสรุปและบทวิจารณ์	61
8.1 บทสรุป	61
8.2 บทวิจารณ์	62
8.3 ปัญหาที่พบในการดำเนินงาน	63
8.4 ข้อเสนอแนะสำหรับระบบทะเบียนนักศึกษา ของบัณฑิตวิทยาลัย	63
8.5 ข้อเสนอแนะในการนำไปพัฒนาต่อ	64
ภาคผนวก ก	65
ภาคผนวก ข	105
บรรณานุกรม	

## สารบัญตาราง

ตารางที่ 3-1 เปรียบเทียบเทคโนโลยี JSP กับเทคโนโลยีอื่น

หน้าที่

20



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ

	หน้าที่
รูปที่ 2-1 Best Practices of Software Engineering	9
รูปที่ 2-2 แผนภาพในการพัฒนาเป็นรอบ (Develop Software Iteratively)	10
รูปที่ 2-3 สถาปัตยกรรมของ View	12
รูปที่ 2-4 แสดงไดอะแกรมต่างๆ ในโมเดลของ UML	13
รูปที่ 2-5 ตัวอย่างแผนภาพ Use Case ในระบบลงทะเบียน	13
รูปที่ 2-6 แสดงตัวอย่างแผนภาพคอลลaboraชัน (Collaboration Diagram)	14
รูปที่ 2-7 แสดงตัวอย่างแผนภาพลำดับการทำงาน (Sequence Diagram)	15
รูปที่ 2-8 แสดงตัวอย่างแผนภาพสถานะ (State Diagram)	15
รูปที่ 2-9 แสดงตัวอย่างแผนภาพกิจกรรม (Activity Diagram)	16
รูปที่ 2-10 แสดงตัวอย่างแผนภาพคอมโพเนนต์ (Component Diagram)	16
รูปที่ 2-11 แสดงตัวอย่างดีพลอยเมนต์ไดอะแกรม	17
รูปที่ 3-1 รูปอธิบายการทำงานของ JSP	22
รูปที่ 4-1 แสดงลักษณะการทำงานของ JDBC	23
รูปที่ 4-2 แสดงรูปแบบ 2-Tier	24
รูปที่ 4-3 แสดงรูปแบบ 3-Tier	25
รูปที่ 4-4 ระดับการเชื่อมต่อของ JDBC API	25
รูปที่ 4-5 ลำดับการเชื่อมต่อเข้าฐานข้อมูลโดยใช้ JDBC	26
รูปที่ 5-1 Usecase ในส่วนของนักศึกษา	34
รูปที่ 5-2 Usecase ในส่วนของเจ้าหน้าที่	35
รูปที่ 5-3 Activity Diagram ของระบบการลงทะเบียน	38
รูปที่ 5-4 Sequence Diagram ของยูสเคสการลงทะเบียนลาพัก	39
รูปที่ 5-5 Class Diagram ของระบบ	40
รูปที่ 5-6 Deployment Diagram ของระบบ	41
รูปที่ 5-7 การแบ่งการทำงานของระบบสารสนเทศ	42
รูปที่ 5-8 อีอาร์ไดอะแกรมของระบบ	43
รูปที่ 6-1 การสร้างฐานข้อมูลบน DB2 โดยใช้โปรแกรม Control Center	44
รูปที่ 6-2 การใช้ Control Center เพื่อการสร้างตาราง	45
รูปที่ 6-3 การใช้ Control Center เพื่อเพิ่มหรือเปลี่ยนแปลงข้อมูลในตาราง	45
รูปที่ 6-4 รูปส่วนอินเทอร์เฟซของ MainPanel	47
รูปที่ 6-5 รูปส่วนอินเทอร์เฟซของ รายละเอียดข้อมูลทั่วไป	47
รูปที่ 6-6 ตัวอย่างผลลัพธ์จากการแปลงจาก XSL-FO เป็น PDF โดยใช้ FOP	48

## สารบัญรูปภาพ (ต่อ)

	หน้าที่
รูปที่ 6-7 ตัวอย่างยูสเซอร์อินเตอร์เฟซสำหรับส่วนเว็บแอปพลิเคชัน	49
รูปที่ 7-1 แสดงหน้าจอการล็อกอินเพื่อเข้าสู่ระบบ	50
รูปที่ 7-2 รูปแสดงข้อมูลที่อยู่ในฐานข้อมูล	51
รูปที่ 7-3 การจัดการเกี่ยวกับข้อมูลทั่วไป	52
รูปที่ 7-4 แสดงข้อมูลในตาราง FACULTY	52
รูปที่ 7-5 การจัดการเกี่ยวกับโครงสร้างหลักสูตร	53
รูปที่ 7-6 แสดงข้อมูลในตาราง MAJOR	53
รูปที่ 7-7 การเลือกภาคการศึกษาและปีการศึกษาก่อนทำการกำหนดเวลา	54
รูปที่ 7-8 การกำหนดเวลาสำหรับการลงทะเบียนต่าง ๆ	54
รูปที่ 7-9 แสดงข้อมูลในตาราง APPOINTMENT	55
รูปที่ 7-10 การลงทะเบียนปกติ	56
รูปที่ 7-11 การแสดงช่วงเวลาซึ่งเมนูจะเปลี่ยนแปลงตาม	56
รูปที่ 7-12 แสดงให้เลือกลงทะเบียนตามรายวิชาที่เปิดสอนตามหลักสูตร	57
รูปที่ 7-13 แสดงรายวิชาที่เปิดสอนตามหลักสูตร	57
รูปที่ 7-14 แสดงรายละเอียดรายวิชาที่เปิดสอนตามหลักสูตร	58
รูปที่ 7-15 แสดงระบบตรวจสอบวิชาที่ต้องเรียนวิชาบังคับก่อน	59
รูปที่ 7-16 แสดงรายวิชาที่ลงทะเบียน	59
รูป 7-17 แสดงค่าใช้จ่ายตามหลักสูตร	60
รูปที่ 7-18 แสดงข้อความการลงทะเบียนสมบูรณ์	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มา

การพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ(Object-Oriented Software Engineering) นับเป็นกระบวนการในการพัฒนาโปรแกรมที่ประสบความสำเร็จอย่างยิ่ง อันเนื่องมาจากเป็นวิธีการที่ได้อธิบายกระบวนการ วิธีการไว้อย่างละเอียดและเป็นลำดับขั้นตอนชัดเจน ทำให้สามารถนำไปปฏิบัติตามได้ง่าย และยังมีรูปแบบที่เป็นมาตรฐาน ทำให้สามารถนำมาใช้เป็นตัวในการสื่อสารให้เกิดความเข้าใจในแนวทางเดียวกันในการทำงานเป็นทีม หรือการสื่อสารกับบุคคลอื่นที่เกี่ยวข้องกับระบบที่จะพัฒนาได้อีกด้วย

ในการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ นั้น ได้นำเอาภาษาสัญลักษณ์เข้ามาใช้ในการสื่อความหมายว่าระบบทำงานอย่างไร แต่ละวัตถุทำงานและเชื่อมต่อกันอย่างไร ภาษาสัญลักษณ์ดังกล่าวมีชื่อว่า ยูเอ็มเอล (Unified Modeling Language: UML) ซึ่งอยู่เราใช้ยูเอ็มเอลแสดงถึงมุมมองต่าง ๆ ของระบบออกมาในรูปของไดอะแกรม(Diagram) ซึ่งมีหลายไดอะแกรม แต่ละไดอะแกรมก็จะแสดงมุมมองแต่ละมุมมองของระบบออกมา อีกทั้งการพัฒนาโปรแกรมเชิงวัตถุยังมีโปรแกรมที่ใช้เป็น เครื่องมือช่วยในการพัฒนาโปรแกรม(CASE Tools) อยู่มาก จึงทำให้เพิ่มความสะดวกและมาตรฐานในการพัฒนาโปรแกรมขึ้นอย่างมาก

จากเหตุผลข้างต้นการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ นั้นจึงได้รับการนิยมนำมาใช้ในการพัฒนาโปรแกรมในระดับต่าง ๆ มากขึ้น แต่สำหรับในประเทศไทยนั้น การพัฒนาโปรแกรมคอมพิวเตอร์ในภาคอุตสาหกรรมนั้นยังมีไม่มากนัก จึงส่งผลให้วิธีการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุถูกนำไปใช้น้อยเมื่อเทียบเชิงปริมาณ บุคลากรที่มีความรู้ความเข้าใจในด้านดังกล่าวก็มีน้อยด้วยเช่นกัน ดังนั้นในอนาคตความต้องการผู้ที่มีความรู้ความเข้าใจในด้านนี้ก็จะเพิ่มมากขึ้นเรื่อย ๆ ตามความเต็มโตของอุตสาหกรรมซอฟต์แวร์

แต่ถึงแม้ว่าการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ นั้นจะมีหลักการที่เป็นมาตรฐาน แต่การที่จะเรียนรู้และทำความเข้าใจจนสามารถนำวิธีดังกล่าวไปใช้อย่างถูกต้องนั้น ไม่ใช่เป็นเรื่องที่ง่ายเลย เพราะเรามักจะคุ้นเคยกับการเขียนโปรแกรมขนาดเล็ก คุ้นเคยกับการทำงานที่ลงมือทำเลย เมื่อพบปัญหาแล้วจึงค่อย ๆ แก้ปัญหานั้น ๆ แต่ในการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ นั้นจะต้องมองภาพของระบบทั้งหมดให้ได้ แล้วมองแยกย่อยระบบออกเป็นวัตถุเป็นชิ้น ๆ ต้องหาความสัมพันธ์ระหว่างวัตถุ การสื่อสารระหว่างวัตถุ และวิเคราะห์ไกลไปจนถึงปัญหาที่อาจเกิดขึ้นได้และทางแก้ปัญหานั้น ซึ่งนั่นก็หมายความว่าผู้ออกแบบระบบจะต้องมีการวิเคราะห์และวางแผนอย่างดียิ่งเสียก่อนที่จะดำเนินการเขียนโปรแกรมนั้นจริง ๆ

และที่อยากอีกอย่างหนึ่งก็คือความเข้าใจอย่างถูกต้องในการใช้สัญลักษณ์ต่าง ๆ ในยูเอ็มเอลเพราะเราอาจใช้อย่างผิด ๆ ทำให้การสื่อความหมายไม่ถูกต้อง และอาจไม่ก่อให้เกิดประโยชน์ใด ๆ เลย เพราะในการใช้ยูเอ็มเอลนั้นเพียงการเรียนรู้แต่ไม่ได้ลงมือทำอย่างจริงจังแล้วนั้น ไม่อาจทำให้เข้าใจในยูเอ็มเอลได้อย่างแน่นอน ถึงแม้จะเข้าใจก็อาจเข้าใจเพียงบางแง่มุมเท่านั้นเอง ดังนั้นการที่ได้ลงมือปฏิบัติจริงในการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ นั้น จึงก่อให้เกิดประโยชน์อย่างมากกับผู้ศึกษา ทั้งก่อให้เกิดความเข้าใจและ

เอกสารนี้เห็นข้อเด่น ข้อด้อย และข้อควรระวังในการใช้วิธีนี้ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกประการหนึ่งก็คือ ระบบสารสนเทศศึกษาระดับบัณฑิตศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่มีอยู่เดิมนั้น ได้ประสบปัญหาในดำเนินงานอยู่หลายด้าน ไม่ว่าจะเป็นเรื่องของความเร็วและความสะดวกที่มีไม่มากเท่าที่ควร อีกทั้งในระบบข้อมูลทะเบียนนักศึกษา ระดับบัณฑิตศึกษานั้นมีความซับซ้อน และมีเงื่อนไขอยู่มาก ระดับความถูกต้องและการตรวจสอบกระบวนการต่าง ๆ ยังทำโดยใช้เจ้าหน้าที่ดำเนินการเป็นหลักจึงอาจมีข้อผิดพลาดและข้อสับสนเกิดขึ้นได้ง่าย

การวิเคราะห์และพัฒนาระบบดังกล่าวให้มีความสามารถมากขึ้นและเพิ่มความสะดวกให้กับผู้ใช้ระบบคือนักศึกษา เจ้าหน้าที่ทะเบียนและอาจารย์นับว่ามีประโยชน์อย่างยิ่ง จึงได้มีการพัฒนาระบบสารสนเทศศึกษาระดับบัณฑิตศึกษาโดยวิธีการเชิงวัตถุขึ้น

## 1.2 วัตถุประสงค์

1. เพื่อนำวิธีการทางวิศวกรรมซอฟต์แวร์เชิงวัตถุไปศึกษา พัฒนาและปรับปรุงระบบทะเบียนนักศึกษาในระดับบัณฑิตศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังให้มีความสามารถยิ่งขึ้น เพิ่มความสะดวกและความถูกต้องในการทำงานมากขึ้น
2. เพื่อศึกษาและทำความเข้าใจระบบทะเบียนนักศึกษาในระดับบัณฑิตศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังเดิมและระบบที่ได้พัฒนามาแล้วว่าเป็นอย่างไร และสามารถปรับปรุงพัฒนาให้ดีขึ้น โดยไม่ทำให้เกิดผลกระทบต่อผู้ปฏิบัติงาน
3. เพื่อศึกษาและทำความเข้าใจวิธีการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุว่ามีกระบวนการอย่างไร ต้องวิเคราะห์และดำเนินการอย่างไรจึงจะถูกต้อง และทำให้ได้โปรแกรมที่มีความถูกต้องและมีประโยชน์มากที่สุด
4. เพื่อศึกษาและทำความเข้าใจในการใช้ประโยชน์จากเครื่องมือช่วยในพัฒนาโปรแกรม(CASE Tools) ว่ามีตัวใดบ้าง ใช้งานในกระบวนการใดบ้าง ใช้อย่างไร มีความสามารถใดบ้าง และ ใช้เครื่องมือเหล่านี้ทำการพัฒนาระบบให้สามารถนำมาใช้งานได้

## 1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้จะเป็นการศึกษาและทำความเข้าใจการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ และการใช้ประโยชน์จากเครื่องมือช่วยในการพัฒนาโปรแกรม แล้วนำความรู้ดังกล่าวมาช่วยในวิเคราะห์และปรับปรุงระบบสารสนเทศทะเบียนนักศึกษาในระดับบัณฑิตศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังให้มีความสามารถและความสะดวกเพิ่มมากขึ้น และทำการแก้ไขปัญหาจากระบบที่ได้พัฒนามาแล้วให้มีความถูกต้อง เพิ่มเติมงานบางส่วนเช่น ส่วนของอาจารย์ที่ได้มีการติดต่อกับระบบ คือการส่งเกรด การเสนอรายวิชา การตรวจสอบการลงทะเบียนของนักศึกษา ส่วนการลงทะเบียนลาพัก การลงทะเบียนรักษาสุขภาพ และได้แก้ไขปรับปรุงงานบางส่วนที่ยังไม่ถูกต้องเช่นการตรวจสอบเงื่อนไขที่มีเพิ่มขึ้นมาในการลงทะเบียนปกติเช่น เรื่องเงินสนับสนุน และการลงทะเบียนล่วงหน้าเช่นเรื่องเวลาเรียนชนกัน เวลาสอบชนกัน นอกจากนี้ในส่วนของฐานข้อมูลต้องทำการปรับปรุงใหม่

แต่อย่างไรก็ตามขอบเขตที่จะดำเนินการนั้นจะครอบคลุมงานที่เกี่ยวข้องกับงานทะเบียนของเอกศานักศึกษาเป็นหลักไม่ว่าจะเป็นการลงทะเบียน การเปลี่ยนแก้ไขรายการลงทะเบียน ข้อมูลประจำวันนักศึกษาเกรดค่าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นักศึกษา ตารางเรียน ตารางสอบ การดำเนินการตามคำร้องต่าง ๆ การจัดการใบทรานสคริปต์และอื่น ๆ ที่เกี่ยวข้อง โดยจัดทำ Prototype ในส่วนที่ได้พัฒนาให้กับเจ้าหน้าที่ได้ทำการทดลองใช้งาน

อีกประการหนึ่งก็คือในการพัฒนาระบบดังกล่าวนั้นจะพยายามไม่ทำการเปลี่ยนแปลงลักษณะสำคัญของระบบเดิม เพื่อไม่ให้กระทบกับตัวระบบเดิม และระบบอื่นที่เกี่ยวข้องมากจนเกินไป ดังนั้นจึงมีงานบางอย่างที่อาจมีความยุ่งยากมากกว่าที่ควรจะเป็น แต่ถึงอย่างนั้นก็ยังเป็นการเพิ่มความสะดวกและถูกต้องให้มากขึ้นในระดับที่น่าพอใจ และเอื้อต่อการนำไปพัฒนาต่อในอนาคต

#### 1.4 วิธีการดำเนินงาน

งานวิจัยในโครงการนี้จะเริ่มด้วยการศึกษาทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งแบ่งเป็น 2 ส่วนหลัก ๆ นั่นก็คือ เรื่องของกระบวนการพัฒนาโปรแกรมเชิงวัตถุ ยูเอ็มแอล(Unified Modeling Language: UML) เพื่อนำความรู้ดังกล่าวมาใช้ในการวิเคราะห์ศึกษาระบบสารสนเทศนักศึกษาทั้งระบบเดิมและระบบที่ได้พัฒนาแล้วในระดับบัณฑิตศึกษา โดยการเข้าไปทำการสัมภาษณ์เจ้าหน้าที่ปฏิบัติงานอยู่ในระบบเดิมและคณะผู้จัดทำที่ได้พัฒนาระบบสารสนเทศศึกษามาแล้ว รวมทั้งอาจารย์และนักศึกษาในระดับบัณฑิตศึกษาอีกด้วย จากนั้นทำการศึกษาระบบที่ได้พัฒนามาแล้วนำความรู้ดังกล่าวมาวิเคราะห์เปรียบเทียบหาข้อบกพร่องและงานที่ยังขาดไปเพื่อทำการปรับปรุงและพัฒนาระบบเพิ่ม และนำมาปรับปรุงยูเอ็มแอลใหม่เพื่ออธิบายมุมมองต่าง ๆ ของระบบอย่างถูกต้อง

ทฤษฎีอีกส่วนหนึ่งที่ต้องทำการศึกษาก็คือเรื่องของเครื่องมือที่ใช้ในการพัฒนาโปรแกรม ไม่ว่าจะเป็น IBM DB2 ซึ่งเป็นโปรแกรมจัดการฐานข้อมูล และโปรแกรมที่ใช้ในการเขียนโปรแกรม หลังจากที่เราได้โมเดลของระบบแล้วสิ่งที่เราต้องทำการออกไปก็คือเรื่องของฐานข้อมูล ซึ่งเราก็จะนำข้อมูลจากคลาสต่าง ๆ ที่ได้เราออกแบบไว้แล้วนั้น นำมาแปลงให้เป็นโครงสร้างตารางข้อมูลต่อไป

## บทที่ 2

# การพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุและยูเอ็มแอล

### 2.1 บทนำ

ในราวๆปี 1990 มีวิธีที่ใช้ Model ระบบมากมาย แต่มีอยู่ 3 วิธีที่เป็นที่นิยมใช้ในการ model ระบบคือ OMT (Rumbaugh), Booch และ OOSE (Jacobson) โดยในแต่ละวิธี จะมีข้อดีข้อเสียต่างกัน วิธีแบบจำลองรูปภาพเชิงวัตถุ (Object Modeling Technology: OMT) นั้นจะมีจุดแข็งในการวิเคราะห์ระบบได้ง่าย แต่พื้นที่ออกแบบนั้นจะใช้ยาก ส่วนวิธี Booch นั้นจะมีจุดแข็งในการออกแบบ แต่การวิเคราะห์จะยาก และวิธีสุดท้ายวิธีแบบจำลองรูปภาพเชิงวัตถุทางวิศวกรรมซอฟต์แวร์ (Object Oriented Software Engineering: OOSE) จะมีจุดแข็งในคุณสมบัติของการวิเคราะห์แต่มีจุดอ่อนในจุดต่างๆอีก เมื่อเวลาผ่านไป การใช้ วิธีนำเสนอรายละเอียดงานที่ต่างกัน ทำให้เกิดความสับสน ดังนั้นจึงจำเป็นต้องมีวิธีที่มาตรฐานที่สามารถเข้าใจได้ง่าย และยังคงอยู่บนพื้นฐานของวิธีเดิมที่มีอยู่ UML จึงได้ถือกำเนิดขึ้นมาจากการร่วมมือกันของทั้ง Booch ,Jacobson ,Rumbaugh รวมถึง คนอื่น โดยได้นำข้อดีของแต่ละวิธีมารวมกัน

Unified Model Language หรือ UML เป็นภาษาโมเดลที่นำเอาสัญลักษณ์มาช่วยในการเพิ่มประสิทธิภาพในส่วนของ การวิเคราะห์(Analysis) และออกแบบ(Design) ซึ่งมีประโยชน์ในหลายด้าน โดยจะเป็นรูปแบบของออปเจกต์โอเรียนเท็ด ซึ่งการออกแบบจะทำให้เราสามารถวิเคราะห์ส่วนประกอบโดยรวมของระบบและยังสามารถที่จะรองรับภาษาที่จะใช้ในการพัฒนาได้หลากหลายภาษามาก และยังหลีกเลี่ยงความซ้ำซ้อนและเฉพาะเจาะจงเกินไป ซึ่งการออกแบบโดยวิธี UML นี้สามารถที่จะแสดงออกมาในรูปแบบของไดอะแกรม(Diagram)ต่างๆ

แอปพลิเคชันที่พัฒนาขึ้นนั้นต่างก็มีขนาด ความซับซ้อน และการกระจายที่มากขึ้น การพัฒนาซอฟต์แวร์ให้สามารถทำงานได้ทั้งเร็ว และถูกต้องนั้น จึงจำเป็นต้องมีการพัฒนาร่วมกันเป็นทีม ซึ่งความยากอย่างหนึ่งของทีมพัฒนาซอฟต์แวร์ก็คือการสื่อสารระหว่างนักพัฒนาในทีม เนื่องจากเมื่อซอฟต์แวร์มีขนาดใหญ่ นักพัฒนาในทีมก็มีจำนวนมากขึ้นตามขนาดของซอฟต์แวร์ และแต่ละท่านก็มีความเชี่ยวชาญในด้านต่างๆกัน นอกจากนั้นเทคโนโลยีในการพัฒนาก็มีการเปลี่ยนแปลงที่รวดเร็วมาก

นอกจากนั้น ปัญหาของการพัฒนาซอฟต์แวร์ ส่วนหนึ่งเกิดจากการที่นักพัฒนาไม่สามารถจัดการกับความต้องการที่เปลี่ยนแปลงของแต่ละ โมดูล รวมถึงซอฟต์แวร์ที่พัฒนาขึ้นนั้นไม่สามารถทำงานร่วมกันได้ และยังดูแลรักษาและเปลี่ยนแปลงได้ยาก นักพัฒนาในทีมไม่สามารถติดตามได้ว่าใครแก้ไขอะไร เมื่อใด ที่ไหน และทำไม

สาเหตุสำคัญที่ทำให้เกิดปัญหาในการพัฒนาซอฟต์แวร์ก็คือ ไม่มีระบบจัดการความต้องการที่ดีเพียงพอ มีการสื่อสารที่ไม่รัดกุมและกำกวม สถาปัตยกรรมของซอฟต์แวร์ไม่มั่นคง

ดังนั้น การจัดการกระบวนการในการพัฒนาจึงจำเป็นต้องมีการพัฒนาซอฟต์แวร์ให้ถูกต้อง และมีคุณภาพตามที่ต้องการ จึงจำเป็นต้องศึกษาการพัฒนากระบวนกรด้วยวิธีเชิงวัตถุ, กระบวนการพัฒนาซอฟต์แวร์ที่ดี, ส่วนประกอบของยูเอ็มแอล เพื่อการพัฒนากระบวนกรซอฟต์แวร์ที่มีคุณภาพตามความต้องการมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และใช้เฉพาะเพื่อการเรียนการสอนเท่านั้น ไม่สามารถนำไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 หลักการเชิงวัตถุ

หลักการเชิงวัตถุเป็นการมองการพัฒนาาระบบเหมือนกับการมองโลกแห่งความเป็นจริง ก็คือมองสิ่งต่างๆ เป็นวัตถุหรือออบเจกต์(Object) ซึ่งแต่ละวัตถุจะมีคุณสมบัติและการทำงานเฉพาะตัว บางวัตถุก็มีความสัมพันธ์กับวัตถุอื่นๆ และถ้าวัตถุที่มีคุณลักษณะบางประการคล้ายๆ กันเราก็จะจัดกลุ่มให้แก่วัตถุเหล่านั้น โดยหลักการเชิงวัตถุแล้วประกอบด้วย ดังนี้

### 2.2.1 ลำดับชั้นของคลาส (Class Hierarchy)

- Inheritance คือการสืบทอดคุณสมบัติเป็นการสร้างคลาสใหม่ขึ้นมา โดยมีพื้นฐาน มาจากคลาสเดิม แต่จะมีข้อมูล (Data) หรือเมธอด ที่พิเศษเป็นของตัวเองเพิ่มขึ้นมาจากคลาสพื้นฐานเดิม ซึ่งก่อให้เกิดการแตกสายพันธุ์ของวัตถุ ขึ้นเป็นวัตถุใหม่ที่มีคุณสมบัติของวัตถุเดิมซ่อนอยู่ไม่มากนักน้อยนั่นเอง

ปัญหาสำคัญของการสืบทอดคุณสมบัติ อยู่ตรงที่ว่า ในการสร้างคลาสขึ้นมาใหม่แต่ละครั้งนั้น คลาสเหล่านั้นมีอะไรที่เป็นพื้นฐานเหมือนกันและมีคุณสมบัติอะไรที่แตกต่างกัน นักพัฒนาจะจำแนกคลาสเหล่านั้นได้อย่างไร จึงจะเหมาะสมกับพฤติกรรมการทำงานของมัน คำตอบก็คือ จะต้องเริ่มต้นมองที่ขอบบนของลำดับชั้นของคลาสก่อน คลาสที่อยู่สูงที่สุดจะต้องเป็นคลาสที่มีคุณสมบัติพื้นฐาน (Simplest) ของคลาสต่างๆ ให้มากที่สุด แล้วค่อยๆ แบ่งลำดับชั้นออกไปตามคุณสมบัติอื่นๆ ที่รองๆ ลงไป โดยเมื่อคุณสมบัติอย่างหนึ่งที่กำหนดลงในคลาสพื้นฐานแล้ว ทุกๆ คลาสที่สืบทอดคุณสมบัติภายใต้คลาสนั้นจะได้รับคุณสมบัติติดตามไปด้วยอัตโนมัติ

- Multiple Inheritance หรือ MI เป็นการสร้างคลาสขึ้นมาโดยการสืบทอดคุณสมบัติ มาจากคลาสพื้นฐานที่มีมากกว่า 1 ตัว ทำให้ภาษาในลักษณะของการเขียน โปรแกรมเชิงวัตถุมีความยืดหยุ่นในการทำงานเพิ่มมากขึ้นอีกด้วย

### 2.2.2 แพคเกจ (Packages)

เป็นการรวมคลาสที่มีความสัมพันธ์ใกล้ชิดกันเข้าไว้ในแพคเกจ เพื่อประโยชน์สำหรับระบบที่มีคลาสเป็นจำนวนมาก ในระบบที่มีความซับซ้อนเราอาจจะสร้างแพคเกจขึ้นมาก่อน แล้วจึงบรรจุคลาสและความสัมพันธ์เข้าไว้ในแพคเกจ

### 2.2.3 เอ็นแคปซูลชันและการซ่อนข้อมูล (Encapsulation and Information Hiding)

เอ็นแคปซูลชัน คือ การรวมกันของโครงสร้างข้อมูล (Data Structure) กับฟังก์ชันที่เกี่ยวข้องหรือเรียกใช้ข้อมูลนั้น (เรียกว่าเมธอด หรือ Action) เกิดเป็นวัตถุตัวใหม่ที่สามารถทำการซ่อนข้อมูลจากภายนอกได้ ซึ่งจะเรียกวัตถุนั้นว่า คลาส นั่นเอง

ในการพัฒนาโดยวิธีแบบโครงสร้างในสมัยก่อนนั้น ส่วนที่เป็นโครงสร้างข้อมูลและฟังก์ชันที่เรียกใช้ข้อมูลนั้นจะถูกสร้างขึ้นแยกจากกัน นั่นคือจะมองข้อมูลและฟังก์ชันเป็น โมดูลเดี่ยวๆ ไม่มีความสัมพันธ์กันแต่อย่างใด ข้อมูลชุดหนึ่งๆ อาจมีฟังก์ชันหลายๆ ตัวมาเรียกใช้งานโดยตรง ซึ่งแต่ละฟังก์ชันอาจจัดการกับโครงสร้างของข้อมูลไม่เหมือนกัน การเข้าถึงข้อมูลโดยตรงของฟังก์ชันเหล่านี้จะนำไปสู่ปัญหาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอ็นแคปซูลชั้นก้ำวามาในจุดนี้ ผลจากการที่นำโครงสร้างข้อมูลและฟังก์ชันที่ใช้งานข้อมูลนั้นมารวมกันสร้างความสัมพันธ์ให้กันและกัน จะทำให้ข้อมูลมีความมั่นคงขึ้น โดยการจะเข้ามาใช้ข้อมูลจะต้องเรียกผ่าน เมธอดเท่านั้น ผลก็คือ ไม่มีการเข้าถึงข้อมูลโดยตรงอีกต่อไป ซึ่งนั่นก็คือ เป็นการทำ Data Hiding นั่นเอง

ข้อดีอีกประการหนึ่งของเอ็นแคปซูลชั้น ก็คือ เราสามารถแบ่งระดับของการเข้าถึงข้อมูลได้โดยตัวของเมธอดของวัตถุเอง

## 2.2.4 โพลิมอร์ฟิซึม (Polymorphism)

โพลิมอร์ฟิซึม หมายถึงการที่คลาสหนึ่งๆ สามารถเปลี่ยนไปได้หลายรูปร่างขึ้นอยู่กับสภาพแวดล้อมหรือสถานการณ์ในขณะนั้น

โดยจะเป็นการยอมให้ผู้พัฒนาใช้เมธอดชื่อเดียวกัน แต่การทำงานไม่เหมือนกัน จำแนกไปตามชนิดของข้อมูลหรือเมสเสจ(Message) ที่เมธอดจะได้รับขณะทำงาน เมื่อได้รับข้อมูลที่ต้องการแล้วก็จะไปเลือกเมธอดที่เหมาะสมที่สุดกับข้อมูลนั้นๆ ไปทำงานเองต่อไป

ความสามารถอีกอย่างหนึ่งของโพลิมอร์ฟิซึมก็คือ สามารถกำหนดการดำเนินการใหม่ให้กับตัวดำเนินการหรือเมธอดได้ เรียกว่าการทำ "Overloading" ยังผลให้เราสามารถเรียกชื่อเมธอดหรือตัวดำเนินการนั้น มาใช้งานกับวัตถุต่างๆ ในกรณีที่แตกต่างกัน ขึ้นอยู่กับว่าในขณะนั้น โปรแกรมกำลังทำงานอยู่กับวัตถุอะไร

## 2.2.5 ความสัมพันธ์

- Generalization เป็นความสัมพันธ์แบบการให้กำเนิดคลาสร้อย (Subclass) จากคลาสมแม่ (Superclass) โดยคลาสร้อยจะสืบทอดแอตทริบิวต์, โอเปอเรชันและความสัมพันธ์ทั้งหมดจากคลาสมแม่ ทั้งของตัวเองและจากคลาสมแม่ของคลาสมแม่ขึ้นไปเรื่อยๆ คลาสร้อยอาจมีการเพิ่มแอตทริบิวต์และ โอเปอเรชันในคลาสมของตัวเอง ความสัมพันธ์แบบนี้อาจถูกเรียกได้ว่า is-a หรือ kind-of
- Aggregation เป็นความสัมพันธ์แบบเป็นส่วนประกอบหรือที่เรียกว่า Part-of คำว่าเป็นส่วนประกอบบางครั้งอาจมีความหมายที่ต่างกัน เช่น เครื่องยนต์และล้อเป็นส่วนประกอบของรถยนต์ในขณะที่เรากำลังขับรถอยู่ แต่ถ้ารถยนต์คันนั้นอยู่ในระหว่างการซ่อมที่อู่ซ่อมรถ นั่นคือเครื่องยนต์และล้อก็ไม่ได้เป็นส่วนประกอบของรถยนต์แล้ว
- Association เป็นความสัมพันธ์แบบสองทิศทาง ซึ่งใช้ในการติดต่อกันระหว่างคลาสม ความสัมพันธ์แบบ แอสโซซิเอชันระหว่างคลาสมหมายถึง การเชื่อมต่อระหว่างออบเจกต์ในคลาสมทั้งสอง ที่ปลายของเส้นความสัมพันธ์จะมีการระบุขอบเขตเพื่อบอกว่าคลาสนั้นถูกมองว่าเป็นอย่างไรจากคลาสมอื่น
- Constraints เป็นการกำหนดข้อบังคับของความสัมพัทธ์ระหว่างเอนติตี้ โดยเอนติตี้จะหมายถึง ออบเจกต์, คลาสม, บทบาท, แอตทริบิวต์, การเชื่อมต่อ (Link) และความสัมพันธ์ ข้อบังคับนี้จะเป็นการจำกัดค่าที่เอนติตี้สามารถเป็นไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 ข้อดีของหลักการเชิงวัตถุ

### 2.3.1 มุมมองของวัตถุจะเป็นการมองในระดับสูง (Higher level of abstraction)

วิธีการวิเคราะห์ระบบแบบ โครงสร้างนั้นมีข้อจำกัดในการแยกแยะเอกลักษณ์ และการซ่อนรายละเอียด อย่างไรก็ตามการวิเคราะห์และพัฒนาระบบแบบ โครงสร้างนี้ จะแยกแยะ โครงสร้างและการทำงานของระบบออกจากกัน เพื่อให้เกิดประสิทธิภาพในการทำงาน ตัวอย่างเช่น เซนเซอร์จะได้รับการ โมเดลให้เป็นส่วนหนึ่งของข้อมูลการอ่านหรือการกระทำอื่นๆ กับเซนเซอร์จะถือว่าเป็นโอเปอเรชัน (Operation) แต่การ โมเดลนี้ไม่ได้แสดงความสัมพันธ์ระหว่างข้อมูลและ โอเปอเรชัน

ส่วน โมเดลของการพัฒนาเชิงวัตถุ นั้นจะแสดงความสัมพันธ์ระหว่างข้อมูลกับ โอเปอเรชันที่กระทำกับข้อมูลอย่างใกล้ชิด เนื่องจาก โมเดลนี้จำลองมาจากโลกจริงๆ การแยกแยะเอกลักษณ์ของการพัฒนาเชิงวัตถุ นั้น ให้ความเข้าใจและความสามารถที่ดีกว่า รวมทั้งมีการตั้งชื่อสำหรับวัตถุ นั้นมาจากชื่อที่มีอยู่จริงในโดเมนของปัญหา มุมมองของวัตถุจะเป็นการมองในระดับสูงและใกล้กับ โดเมนของปัญหา มากกว่า

### 2.3.2 Seamless transition among different phases of software development

การพัฒนาซอฟต์แวร์นั้นต้องการวิธีการ (Methodology) ที่แตกต่างกันในแต่ละช่วงของการพัฒนา การ เปลี่ยนแปลงจากช่วงการทำงานหนึ่งไปช่วงอื่นๆ ในแต่ละ Process จะมีความซับซ้อน และยังไม่มีความสอดคล้องกัน และยิ่งช้าด้วย และยังทำให้ขนาดของงานใหญ่ขึ้น และมีโอกาสที่จะเกิดความผิดพลาดขึ้นด้วย แต่วิธีการของเชิงวัตถุ นั้นจะใช้ภาษาเดียวกันในการติดต่อระหว่างการพัฒนาวิเคราะห์ การออกแบบ การเขียนโปรแกรม และการออกแบบฐานข้อมูลด้วย ทำให้ลดระดับของความซับซ้อนลง และช่วยให้การเข้าใจระบบก็ชัดเจนด้วย

### 2.3.3 สนับสนุนเทคนิคการเขียนโปรแกรมที่ดี (Encouragement of good programming techniques)

เป็นการสนับสนุนเทคนิคการเขียนโปรแกรมที่ดี เพราะการเขียน โปรแกรมเชิงวัตถุ นั้นจะเป็นการเขียนในลักษณะของคลาส ซึ่งบอกว่าคลาสนี้ทำอะไร และทำอย่างไร และคุณสมบัติของระบบนั้นสามารถที่จะรวมคลาสเข้าด้วยกันเป็นระบบย่อยได้ และการเปลี่ยนแปลงคลาสหนึ่งก็จะมีผลกระทบต่อคลาสอื่นๆ ด้วย อย่างไรก็ตามภาษาต่างๆ เช่น C++, Smalltalk หรือ Java ก็มีส่วนที่สามารถรองรับการออกแบบเชิงวัตถุด้วย เพื่อง่ายต่อการสร้าง และการนำ Code มาใช้ใหม่กับแนวคิดของคลาสและการสืบทอด (Inheritance )

### 2.3.4 มีคุณสมบัติที่สนับสนุนการนำกลับมาใช้ใหม่ (Promotion of Reusability)

ระบบแบบ โครงสร้างมีข้อจำกัดในการนำสิ่งที่พัฒนาไปแล้วกลับมาใช้ใหม่ แต่ในความเป็นจริงแล้ว ถ้าผู้พัฒนาระบบมีคอมไพเลอร์ อยู่แล้วก็ควรจะใช้ได้เลย แต่ระบบแบบ โครงสร้างนี้ผู้ที่จะนำมาใช้ต้องทำการเปลี่ยนแปลงซอร์สโค้ด (Source Code) เพื่อให้เหมาะกับความต้องการของระบบใหม่เพื่อที่จะใช้งานได้ แต่การพัฒนาเชิงวัตถุ นั้นมีคุณสมบัติที่สนับสนุนการนำกลับมาใช้ใหม่โดยตรงคือ Generalization และ Refinement โดย Generalization คือการสืบทอดคุณสมบัติโดยจะได้คุณสมบัติเดิมทุกประการและยังสามารถเพิ่มและขยายคุณสมบัติได้ด้วย โดยไม่ต้องมีการเปลี่ยนแปลงซอร์สโค้ดเก่าเลย ลักษณะการนำมาใช้ใหม่นี้เรียกว่า การเขียนโปรแกรมแบบแตกต่าง ส่วน Refinement นั้นก็คล้ายๆ กับ Generalization แต่จะยอมให้นักพัฒนาใช้วัตถุที่ไม่สมบูรณ์โดยการเพิ่มส่วนที่หายไป ตามที่ตนเองต้องการ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.5 สามารถปรับเปลี่ยนขนาดของระบบได้ง่าย (Promotion of Scalability)

ระบบโครงสร้างมีข้อจำกัดในการปรับเปลี่ยนขนาดของระบบ ทั้งนี้เนื่องจากระบบแบบโครงสร้างมีการแยกแยะเอกลักษณ์และการซ่อนรายละเอียดที่ไม่ดี ทำให้ระบบแบบโครงสร้างมีความซับซ้อนมากขึ้นเมื่อมีปัญหาที่เกิดขึ้นกับระบบมากขึ้น แต่ระบบที่พัฒนาด้วยวิธีเชิงวัตถุ นั้น จะแยกแยะเอกลักษณ์และการซ่อนรายละเอียดส่งที่ดีส่งผลให้คอมโพเนนต์แต่ละส่วนแยกออกจากกันอย่างชัดเจน และการใช้เครื่องหมาย (Notation) เดียวกันตลอดการพัฒนา ทำให้สามารถเปลี่ยนจากการวิเคราะห์เป็นการออกแบบได้ง่าย การปรับเปลี่ยนระบบให้มีขนาดใหญ่ขึ้นก็สามารถทำได้ง่ายตามด้วย

### 2.3.6 มีความน่าเชื่อถือและควบคุมได้ (Promotion of Reliability and Safety)

เนื่องจากการพัฒนาเชิงวัตถุมีการแยกแยะเอกลักษณ์และการซ่อนรายละเอียดที่ดีกว่าทำให้การสื่อสารระหว่างวัตถุที่ไม่เกี่ยวข้องกันจำกัดอยู่ด้วยอินเทอร์เฟซ ที่ผู้พัฒนาได้กำหนดไว้เท่านั้น ส่งผลให้ระบบที่พัฒนาด้วยวิธีเชิงวัตถุมีความน่าเชื่อถือได้มากกว่าเพราะว่าผู้พัฒนาสามารถควบคุมการติดต่อระหว่างคอมโพเนนต์ได้ และยังสามารถเพิ่มเงื่อนไขก่อนหน้าและท้าย (Pre and Post Condition) เพื่อให้ระบบสามารถทำงานได้อย่างถูกต้อง

### 2.3.7 วัตถุสนับสนุนการทำงานพร้อมกันในตัวเอง (Promotion of Concurrency)

การทำงานไปพร้อมๆกัน (Concurrency) เป็นรูปแบบการทำงานของสิ่งที่มีตัวตนในโลกแห่งความเป็นจริง และยังเป็นคุณสมบัติสำคัญของคอมพิวเตอร์ยุคใหม่ด้วย การพัฒนาแบบโครงสร้างไม่มีสัญลักษณ์สำหรับการทำงานพร้อมกัน การจัดการงาน (Task) หรือการซิงโครไนซ์ (Synchronization) ระหว่างงานดังนั้นโครงสร้างที่สำคัญของระบบอาจจะไม่สามารถพัฒนาขึ้นได้โดยการใช้รูปแบบพัฒนาแบบโครงสร้างมาตรฐานได้ แต่การพัฒนาระบบด้วยวิธีเชิงวัตถุสนับสนุนการทำงานพร้อมกันในตัวเอง และรายละเอียดของงาน การทำ Synchronization ระหว่างงานก็สามารถแสดงออกมาในแผนภาพ (Statecharts) ของวัตถุที่ทำงานแบบแอคทีฟ (Active Object) และการส่งแอสเซจ (Message) ระหว่างวัตถุ แผนภาพเหล่านี้เป็นเครื่องมือที่มีความเหมาะสมสำหรับการสร้างระบบที่มีเงื่อนไขทางด้านประสิทธิภาพ

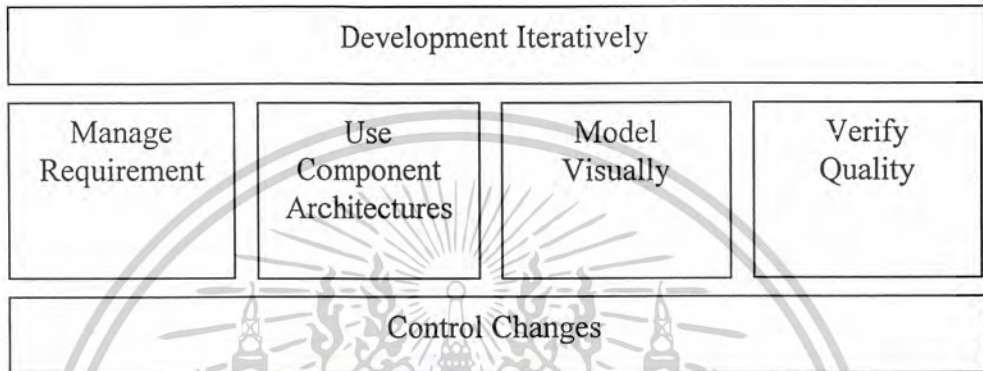
## 2.4 การพัฒนาซอฟต์แวร์ด้วยวิธีเชิงวัตถุ

การพัฒนาซอฟต์แวร์ด้วยวิธีเชิงวัตถุเป็นการพัฒนาที่ช่วยลดความยุ่งยากซับซ้อนของระบบใหญ่ๆได้ โดยผู้พัฒนาจะมองระบบและส่วนประกอบหรือระบบย่อย (Sub System) เป็นวัตถุ (Object) เนื่องจากสิ่งต่างๆ ที่มีอยู่ในโลกความเป็นจริงนั้นก็คือได้ว่าเป็นวัตถุได้อยู่แล้ว และด้วยการมองระบบเป็นวัตถุทำให้ผู้พัฒนาระบบสามารถมองระบบได้ชัดเจนมากยิ่งขึ้น ส่งผลให้การวิเคราะห์และออกแบบระบบทำได้ง่ายและมีความถูกต้องมากขึ้น

การพัฒนาซอฟต์แวร์ด้วยวิธีเชิงวัตถุ นั้นยังมีส่วนเกี่ยวข้องกับทุกขั้นตอนของการพัฒนาระบบ ทั้งการรวบรวมความต้องการ (Requirement gathering), การวิเคราะห์ (analysis), การออกแบบ (design), การเขียนโปรแกรม (coding), และการทดสอบ (testing) ซึ่งเป็นการแก้ปัญหาของการพัฒนาระบบสมัยก่อนแบบที่เรียกว่า การพัฒนาแบบโครงสร้าง (Structural) ซึ่งเป็นการบอกเพียงว่าระบบจะมีลักษณะการทำงานอย่างไรแต่ไม่ได้บอกถึงวิธีการพัฒนาระบบหรือรูปแบบการเขียนโปรแกรมเลย จุดนี้เองทำให้หลายๆครั้ง ระบบที่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาขึ้นมีปัญหาเนื่องจากการตรวจสอบความถูกต้องของระบบ (Validate) ระบบในเชิงพฤติกรรม (Behavior) นั้นทำได้ยาก

วิธีการพัฒนาระบบเชิงวัตุนั้นมีมากมายหลายวิธี โดยแต่ละวิธีก็มีจุดเด่นและจุดด้อย แตกต่างกันไป โดยผู้พัฒนาระบบจะใช้วิธีการที่มีประสิทธิภาพที่ดีที่สุด เพื่อระบบที่ดีที่สุด โดยจะใช้วิธีผสมนำข้อดีของแต่ละวิธีมาใช้นั้นคือ Unified Modeling Language (UML) ที่นำข้อดีของ Rumbaugh et al., Booch และ Jacobson มาใช้ และยังสามารถใช้เทคนิคต่างๆที่เกิดขึ้นมาพัฒนาด้วยเช่น Patterns และ Frameworks เป็นต้น

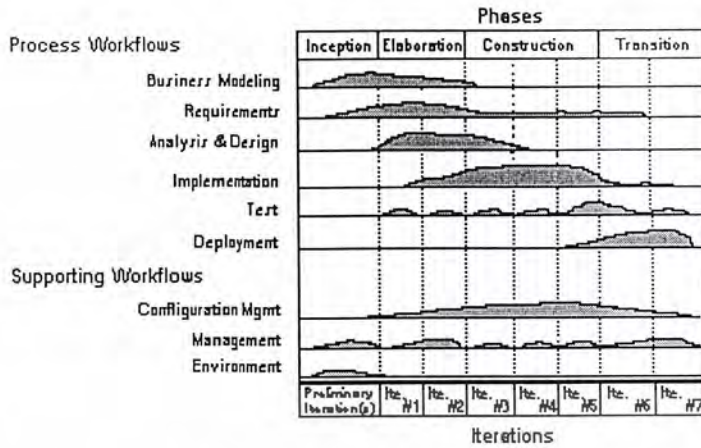


รูปที่ 2-1 Best Practices of Software Engineering

จากรูปเป็นวิธีปฏิบัติที่ดีในการพัฒนาซอฟต์แวร์ ซึ่งจะเห็นได้ว่าส่วนนี้จะนำไปใช้กับส่วนของการจัดการความต้องการ (Manage Requirement) , การใช้สถาปัตยกรรมคอมโพเนนต์ (Use Component Architecture) , การโมเดลซอฟต์แวร์เป็นภาพให้เห็น (Model Visually) และ การตรวจสอบคุณภาพซอฟต์แวร์ (Verify Quality) ดังนั้นการพัฒนาแบบวนรอบนี้ ถือเป็นกระบวนการที่สำคัญกระบวนการหนึ่งทีเดียว คุณลักษณะของการพัฒนาวนรอบคือ นักพัฒนาจะสามารถวิเคราะห์ และแก้ปัญหาที่มีความเสี่ยงสูง ก่อนที่จะมีการลงทุนพัฒนาระบบ การพัฒนาแบบวนรอบสามารถได้รับความต้องการอย่างต่อเนื่องจากผู้ใช้นอกจากนั้น การทดสอบและการรวมระบบจะค่อยเป็นค่อยไปอย่างต่อเนื่อง การมี milestone ทำให้นักพัฒนาทราบว่าตอนนี้การพัฒนากำลังเน้นไปที่จุดใด

วิธีปฏิบัติที่ 1 : การพัฒนาซอฟต์แวร์เป็นรอบ ( Develop Software Iteratively) ทำให้ทีมพัฒนาได้รับความคิดเห็นจากผู้ใช้อยู่เป็นระยะ ทำให้ผู้ใช้และผู้พัฒนาสามารถทำความเข้าใจความต้องการที่แท้จริงได้ถูกต้องและครบถ้วนมากยิ่งขึ้น นอกจากนี้ยังช่วยแสดงให้เห็นความไม่เข้าใจระบบร่วมกันระหว่างผู้ใช้กับทีมพัฒนาในรอบก่อนหน้าทำให้อรอบต่อไปสามารถแก้ไขการสื่อสารที่ไม่ตรงกันหรือมีความกำกวมได้ดีขึ้น จากรูปที่ 2-3 แสดงให้เห็นถึงการประยุกต์ใช้การพัฒนาแบบวนรอบ สำหรับในการพัฒนารอบแรกนั้นจะให้เวลากับกระบวนการ Business Modeling มากที่สุด ในรอบต่อไปก็จะให้เวลาการศึกษา Requirement มากที่สุด แต่ในทุกรอบการทำงานก็จะทำในกระบวนการทุกกระบวนการ แต่จะมากขึ้นน้อยแตกต่างกันไปตามรอบของการพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-2 แผนภาพในการพัฒนาเป็นรอบ (Develop Software Iteratively)

วิธีปฏิบัติที่ 2 : การจัดการความต้องการ (Manage Requirement) ความต้องการนั้นเป็นสิ่งที่มีความสำคัญต่อกระบวนการซอฟต์แวร์ทั้งหมด ซึ่งความต้องการส่วนใหญ่เป็นความต้องการที่มีความเปลี่ยนแปลงได้สูง ดังนั้นทีมพัฒนาจำเป็นต้องตระหนักไว้ว่าตลอดช่วงเวลาการพัฒนานั้นความต้องการสามารถเปลี่ยนแปลง ได้อยู่เสมอ ในการจัดการความต้องการจึงจำเป็นต้องดำเนินการอย่างต่อเนื่องเช่นกัน

วิธีปฏิบัติที่ 3 : การใช้สถาปัตยกรรมคอมโพเนนต์ (Use Component – Based Architecture) สถาปัตยกรรมซอฟต์แวร์ เป็น โครงสร้างหลักของระบบซอฟต์แวร์และเป็นส่วนหนึ่งที่ได้จากการออกแบบระบบซึ่งเกี่ยวข้องกับ โครงสร้างและพฤติกรรมของระบบในด้านของการใช้งาน หน้าที่การทำงาน ประสิทธิภาพ และเงื่อนไขของเทคโนโลยี กระบวนการนี้ จะทำให้สถาปัตยกรรมซอฟต์แวร์มีพื้นฐานอยู่บนหลักการของ Component เป็นหลัก ซึ่งจะทำให้สถาปัตยกรรมที่ได้นั้นมีความยืดหยุ่นและทำให้สามารถดูแลรักษา เพิ่มเติมระบบได้ง่าย และยังสามารถนำกลับมาใช้ได้

วิธีปฏิบัติที่ 4 : การ โมเดลซอฟต์แวร์เป็นภาพที่มองเห็นได้ (Visually Model Software) เป็นการนำเอาโมเดลที่เป็นรูปภาพมาช่วย ซึ่งทำให้สามารถเก็บ โครงสร้างและพฤติกรรมของสถาปัตยกรรม แสดงส่วนประกอบในระบบ หรือแสดงรายละเอียดของงาน โดยมีการดูแลรักษาความตรงกันของการออกแบบและการพัฒนา และยังช่วยลดความกำกวมในการสื่อสาร ส่วนนี้จะเป็นส่วนที่จะจะพุดถึง UML กัน ซึ่งรายละเอียดจะกล่าวต่อไป

วิธีปฏิบัติที่ 5 : การตรวจสอบคุณภาพซอฟต์แวร์ (Verify Software Quality) การพัฒนาแบบวนรอบนั้นทำให้นักพัฒนาสามารถทดสอบระบบได้ดียิ่งขึ้น ทั้งนี้แทนที่จะทำการทดสอบระบบครั้งใหญ่ครั้งเดียวก่อนที่จะส่งให้กับผู้ใช้ ด้วยการพัฒนาแบบวนรอบจะทำให้มีการทยอยทดสอบในแต่ละรอบ ส่งผลให้ระบบในรอบต่อไปมีคุณภาพมากยิ่งขึ้น ในการพัฒนาแต่ละรอบซอฟต์แวร์จะได้รับการพัฒนาเพิ่มขึ้น มีการประกอบและทดสอบส่วนที่ได้พัฒนาเพิ่มขึ้น

วิธีปฏิบัติ 6 : การควบคุมความเปลี่ยนแปลงของซอฟต์แวร์ (Control Change Software) หลักการในการจัดการความเปลี่ยนแปลงก็คือ จะต้องมีการแยกสถาปัตยกรรมของระบบออกเป็นระบบย่อยๆ และมีการกำหนดความรับผิดชอบของระบบย่อยให้กับทีมพัฒนาหนึ่ง มีการจัดตั้งพื้นที่ทำงานสำหรับนักพัฒนาแต่ละคนไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีการกำหนดการรวมกันของแต่ละพื้นที่ทำงาน มีการกำหนดวิธีการควบคุมความเปลี่ยนแปลง มีการบันทึกความเปลี่ยนแปลงในแต่ละรุ่น มีการทดสอบในแต่ละรอบของการพัฒนาระบบ ระบบที่ถูกต้องนั้น จะมีการจัดการความต้องการความต้องการพื้นฐาน จากนั้นจึงเปรียบเทียบ ความเปลี่ยนแปลงที่เกิดขึ้นโดยใช้สถาปัตยกรรมคอมโพเน้น นักพัฒนาสามารถที่จะตรวจสอบรุ่นของ คอมโพเน้นที่จะใช้โมเดลด้วยรูปภาพ ทำให้สามารถตรวจสอบความเปลี่ยนแปลงได้ง่าย ผลที่ได้จากการตรวจสอบจะต้องสามารถติดตามได้ในแต่ละรุ่นของระบบ จากนั้นจะพัฒนามารอบต่อไป

จากรายละเอียดที่ได้กล่าวข้างต้น การสื่อสารระหว่างนักพัฒนา นับเป็นปัญหาหนึ่งในการพัฒนาระบบซอฟต์แวร์ใหญ่ๆ ดังนั้นในกระบวนการการพัฒนาซอฟต์แวร์ที่ดังนั้นจึงนำเอา แบบจำลองรูปภาพ(Visual Modeling) มาใช้ในการแสดงรายละเอียดของงาน แทนที่จะใช้ข้อความในการอธิบายเหมือนระบบในอดีต ซึ่งการใช้ข้อความนั้นอาจจะทำให้ความหมายนั้นกำกวม ต่างกับรูปภาพที่แทนคอมโพเน้น ในระบบจริงๆ ที่สามารถสื่อความหมายได้ชัดเจน

## 2.5 ภาษา Unified Modeling Language (UML)

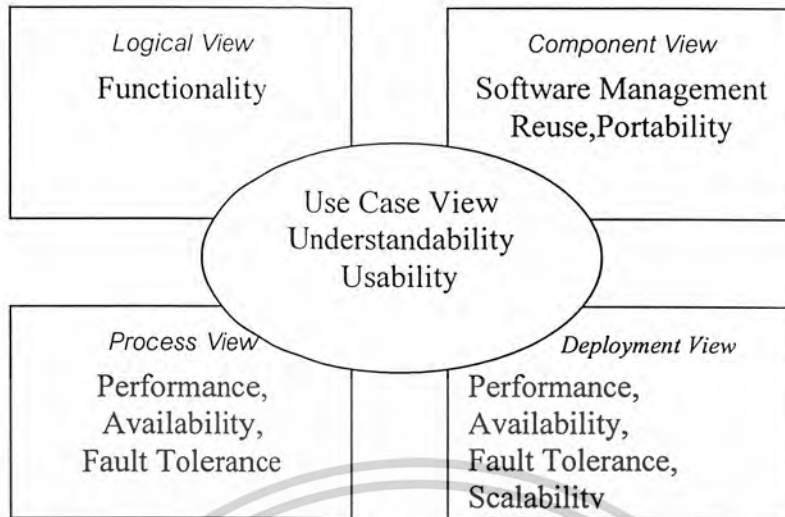
Unified Modeling Language(UML) เป็นภาษาในการกำหนด (Specifying) , แสดงภาพ (Visualizing), สร้างและบันทึกเอกสารเกี่ยวกับสิ่งที่ได้จากการพัฒนาซอฟต์แวร์ รวมทั้งยังมีความสามารถในการโมเดลเชิงธุรกิจและระบบอื่นๆ ที่ไม่ทำงานบนซอฟต์แวร์อีกด้วย ภาษายูเอ็มแอลนั้น ได้นำเสนอชุดของวิธีปฏิบัติทางด้านวิศวกรรมซอฟต์แวร์ที่ดี ซึ่งวิธีปฏิบัติเหล่านี้จะสามารถช่วยให้การ โมเดลและพัฒนาระบบขนาดใหญ่ที่มีความซับซ้อนสามารถทำได้อย่างมีประสิทธิภาพภายใต้เงื่อนไขของเวลาและงบประมาณที่กำหนดไว้

ภาษายูเอ็มแอลนั้นได้รับการพัฒนาขึ้น โดยบริษัท Rational Software และผู้ที่เกี่ยวข้องในอุตสาหกรรม ภาษา UML นั้นเป็นภาษาที่ได้รับการสืบทอดมาจากภาษาในการ โมเดลซอฟต์แวร์ที่ประกอบด้วย Booch, OOSE/Jacobson, OMT และวิธีอื่นๆ รวมกัน โดยมีการนำข้อดีของแต่ละวิธีและแต่ละภาษามาประกอบเป็นโครงสร้างหลักของภาษายูเอ็มแอล นอกจากนี้ภาษายูเอ็มแอลยังช่วยกำหนดระเบียบปฏิบัติในการพัฒนาดังเช่นที่กล่าวมาข้างต้น ซึ่ง ยูเอ็มแอลประกอบด้วยหลายๆ ส่วนดังนี้

### 2.5.1 วิว (View)

ในการออกแบบระบบที่มีขนาดใหญ่และมีความซับซ้อนมากๆนั้นจะทำให้ผู้ออกแบบระบบไม่สามารถที่จะออกแบบระบบได้ครบถ้วน ดังนั้นจึงต้องมีการมองระบบเป็นมุมมองต่างๆ เพื่อทำให้ง่ายในการออกแบบ เช่น มุมมองด้านฟังก์ชันนอล , นอนฟังก์ชันนอล(Nonfunctional), มุมมองขององค์กรเป็นต้น ดังนั้นระบบจึงมี View ที่ต่างๆ กันซึ่งแต่ละ view จะแสดงมุมมองเฉพาะของระบบซึ่งอธิบายรวมกันเป็นระบบที่สมบูรณ์ ซึ่งจะประกอบด้วย View ต่างๆ ดังนี้

1. Use-case View อธิบายการทำงานต่างๆ ของระบบที่ถูกมองจากภายนอกหรือผู้ใช้ระบบ Use-case view ซึ่งอธิบายโดยยูสเคสไดอะแกรม(use-case diagram) เป็นมุมมองสำหรับลูกค้า ,ผู้ออกแบบ ,ผู้พัฒนาระบบ และผู้ทดสอบระบบ



รูปที่ 2-3 สถาปัตยกรรมของ View

2. Logical view อธิบายการทำงานต่างๆ ที่ถูกออกแบบไว้ภายในระบบ ว่าระบบจะมีบริการอะไรให้กับผู้ใช้งาน โดยจะแสดงโครงสร้างแบบสถิต (Static) เช่น คลาส, ออบเจกต์, ความสัมพันธ์และการทำงานร่วมกันแบบไดนามิก (dynamic collaboration) ซึ่งจะเกิดขึ้นเมื่อออบเจกต์ ส่งเมสเสจระหว่างกันในการทำงาน

โครงสร้างแบบสถิตจะอธิบายโดยใช้ คลาสไดอะแกรม (Class diagram) และออบเจกต์ไดอะแกรม (object diagram) ส่วนการทำงานร่วมกันแบบไดนามิกจะอธิบายโดยใช้ สเตตไดอะแกรม (state diagram) และ แอคติวิตีไดอะแกรม (activity diagram)

3. Component View อธิบายการสร้างและสร้างความขึ้นต่อกันของโมดูล (Module) ที่ใช้ในการพัฒนาระบบ โดยใช้คอมโพเนนต์ไดอะแกรม (Component diagram) ในการอธิบาย

4. Deployment View อธิบายการจัดวางระบบให้เหมาะสมในด้านกายภาพ (Physical) แสดงด้วยคอมพิวเตอร์และ โหนด (nodes) ต่างๆ เพื่อให้ระบบมีเสถียรภาพมากขึ้น โดยใช้ดีพลอยเม้นต์ไดอะแกรม (deployment diagram) ในการอธิบาย

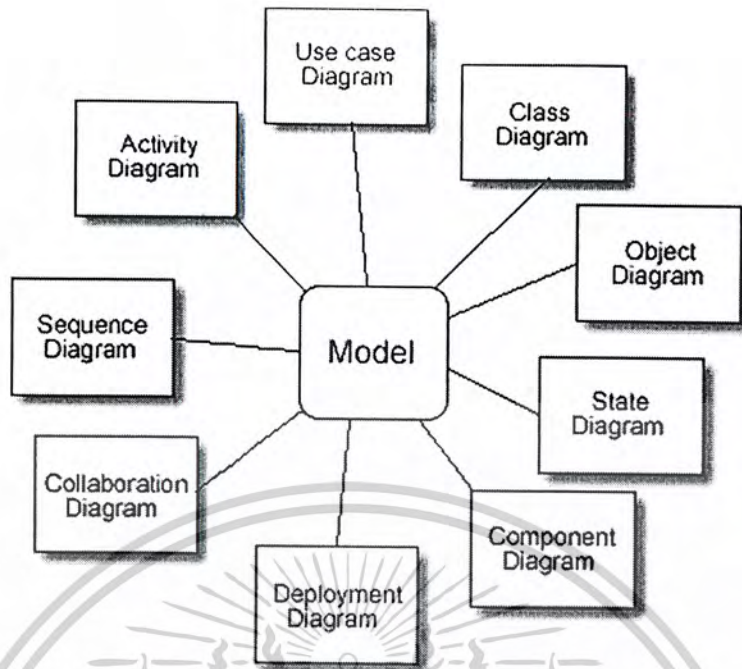
5. Process View แสดงการทำงานร่วมกันและการติดต่อกันของส่วนต่างๆ ในระบบ

### 2.5.2 ไดอะแกรม (Diagram)

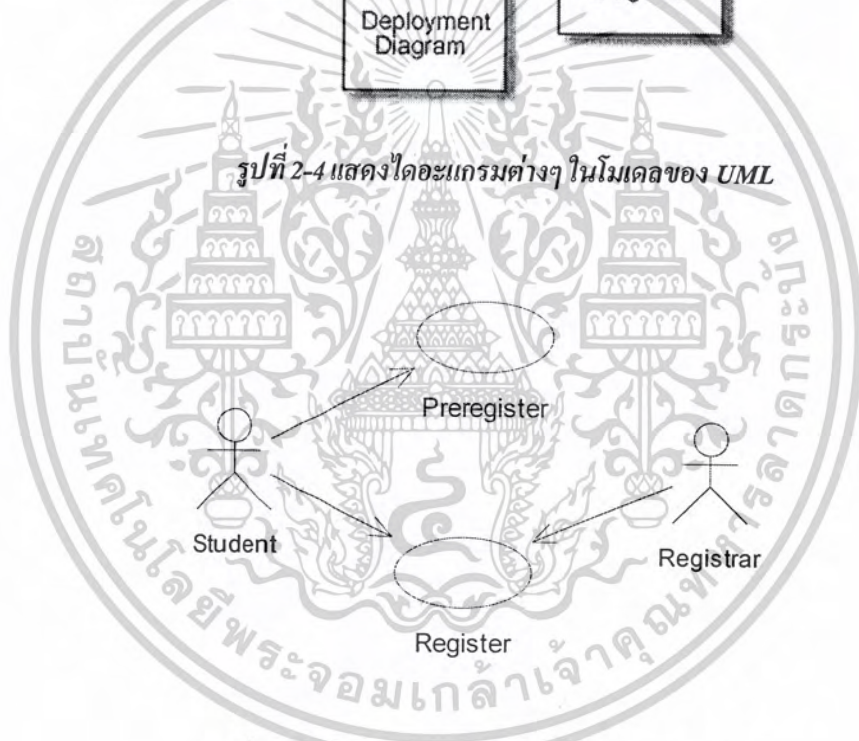
เป็นไดอะแกรมซึ่งแสดงสัญลักษณ์ที่ถูกจัดเรียงเพื่อ อธิบายระบบในมุมมองต่างๆ ซึ่งในระบบหนึ่งๆ จะประกอบด้วยหลายๆ ไดอะแกรม และในแต่ละไดอะแกรมยังสามารถมองในหลายๆ มุมมองได้ UML มีไดอะแกรมที่ต่างกันอยู่ ดังนี้

1. Use Case Diagram ยูสเคสไดอะแกรมจะแสดงความสัมพันธ์ระหว่างผู้ใช้งาน (User) กับระบบ โดยใช้แอ็กเตอร์ (actors) แทนผู้ใช้งานและแอ็กเตอร์จะต้องติดต่อกับระบบโดยผ่าน ยูสเคส (Use case) ต่างๆ ซึ่งยูสเคสใน Use case diagram ก็คือการทำงานต่างๆ ของระบบที่ผู้ใช้ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-4 แสดงไดอะแกรมต่างๆ ในโมเดลของ UML



รูปที่ 2-5 ตัวอย่างแผนภาพ Use Case ในระบบลงทะเบียน

แเอ็กเตอร์ไม่ใช่ส่วนประกอบของระบบ แต่จะเป็นส่วนที่ใช้ติดต่อกับระบบ ซึ่งอาจจะเป็นเพียงการป้อนข้อมูลเข้าไปในระบบหรือเป็นเพียงการรับข้อมูลจากระบบ หรืออาจจะเป็นทั้งสองอย่างก็ได้ ในระบบหนึ่งๆ จะต้องมีแเอ็กเตอร์ซึ่งจะต้องทำการกำหนดคุณลักษณะของแเอ็กเตอร์ (Identify actor) ให้ได้ก่อนดังนี้

- ใครเป็นผู้ใช้งานระบบ?
- ใครที่มีความเหมาะสมที่จะใช้งานระบบ?
- ใครที่มีส่วนสนับสนุนข้อมูลในระบบ ใช้ข้อมูลในระบบและแก้ไขข้อมูลในระบบ?
- ใครที่สนับสนุนการบำรุงรักษาระบบ (Maintenance)?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

● ระบบมีการใช้งานกับภายนอกหรือไม่?  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Class Diagram คลาสไดอะแกรมเป็นไดอะแกรมที่มีความสำคัญมากที่สุดในการวิเคราะห์และออกแบบโดยใช้วิธีเชิงวัตถุ เนื่องจากคลาสไดอะแกรมจะแสดงโครงสร้างของวัตถุและคลาสที่มีในระบบ รวมทั้งแสดงความสัมพันธ์ด้วย คลาสไดอะแกรมจะเป็นโครงสร้างหลักของระบบและยังใช้ในการแยกย่อยรายละเอียด (Decomposite) ด้วยวิธีเชิงวัตถุ

สัญลักษณ์ความสัมพันธ์ต่างๆ ของคลาสไดอะแกรม มีดังนี้

1 หมายถึงจะมีออบเจกต์ในคลาสไดอะแกรมได้หนึ่งออบเจกต์เท่านั้น

0..1 หมายถึงจะมีออบเจกต์ในคลาสไดอะแกรมได้แก่หนึ่งอาจจะไม่มีก็ได้

M..N หมายถึงจะมีออบเจกต์ในคลาสไดอะแกรมได้ตั้งแต่ M ถึง N (เมื่อ M และ N เป็นจำนวนเต็มบวก)

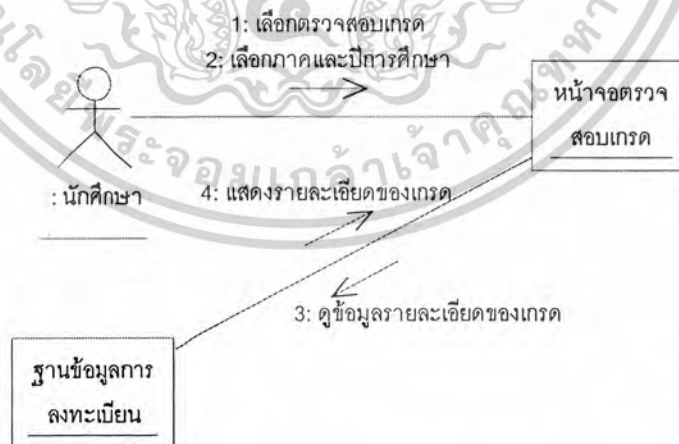
\* หมายถึงจะมีออบเจกต์ในคลาสไดอะแกรมได้ตั้งแต่ศูนย์ขึ้นไป

0..\* หมายถึงจะมีออบเจกต์ในคลาสไดอะแกรมได้ตั้งแต่ศูนย์ขึ้นไป

1..\* หมายถึงจะมีออบเจกต์ในคลาสไดอะแกรมได้ตั้งแต่หนึ่งขึ้นไป

3. Object Diagram ออบเจกต์ไดอะแกรมจะได้มาจากคลาสไดอะแกรม ซึ่งจะเป็นการแสดงการเชื่อมต่อระหว่างอินชแทนซ์ (Instances) ต่างๆ ที่อยู่ภายในคลาสไดอะแกรมอย่างละเอียด เพื่อใช้เป็นตัวอย่างสำหรับคลาสไดอะแกรมที่มีความซับซ้อนโดยจะนำมาแทนเป็น ออบเจกต์จริงๆ และแสดงความสัมพันธ์ต่างๆ ให้เห็น โดยสัญลักษณ์ของอินชแทนซ์ในออบเจกต์ไดอะแกรม จะใช้การขีดเส้นใต้เป็นสัญลักษณ์

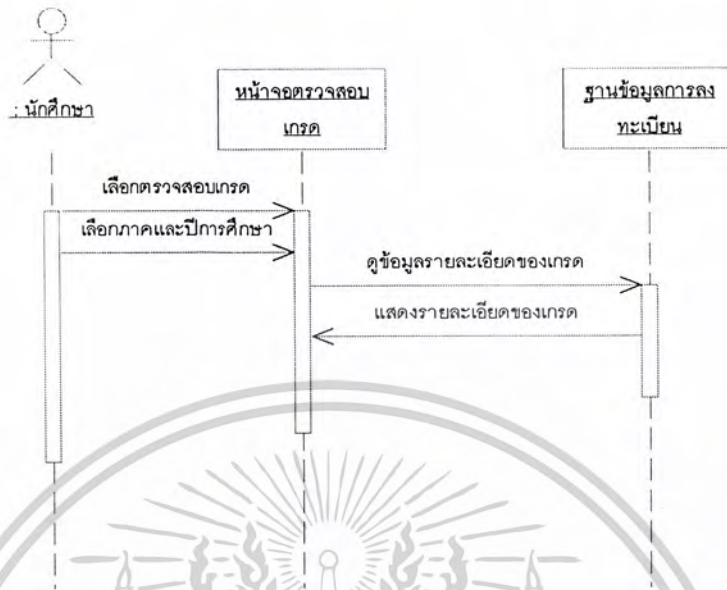
4. Collaboration Diagram คีอแลโบริชั่นไดอะแกรมใช้อธิบายการทำงานร่วมกันของออบเจกต์ที่มีการส่งแอสเซสระหว่างกัน เพื่อบอกถึงลำดับขั้นตอนการทำงานของระบบ โดยใช้สัญลักษณ์ ลูกศรเป็นตัวกำหนดทิศทางซึ่ง คีอแลโบริชั่นไดอะแกรมจะเป็นส่วนอธิบายการทำงานของออบเจกต์ ไดอะแกรม



รูปที่ 2-6 แสดงตัวอย่างแผนภาพคีอแลโบริชั่น (Collaboration Diagram)

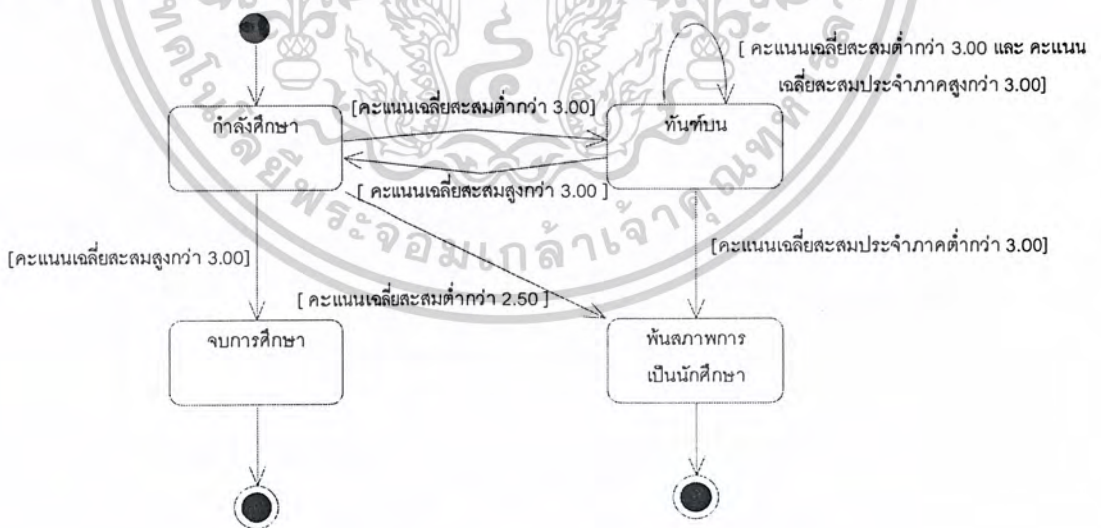
5. Sequence Diagram ซีควีนซ์ไดอะแกรมแสดงการติดต่อโต้ตอบระหว่างออบเจกต์ ที่เน้นไปที่ลำดับก่อนหลังและเวลาที่มีความสำคัญในการเปลี่ยนแปลง จะมีการส่งแอสเซส เพื่อให้ออบเจกต์ ต่างๆ ที่อยู่ในระบบเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานโดยใช้สัญลักษณ์ที่เรียกว่า ออบเจกต์เมสเสจส ซีควีนซ์ชาต (Object message sequence chart) ในการแสดงดังรูป



รูปที่ 2-7 แสดงตัวอย่างแผนภาพลำดับการทำงาน (Sequence Diagram)

6. State Diagram สเตตไดอะแกรมใช้อธิบายคลาสต่างๆ ในระบบโดยจะแสดงทุกๆ สถานะที่เป็นไปได้และเหตุการณ์ทำให้ออบเจกต์เหล่านั้นเกิดการเปลี่ยนแปลง โดยเหตุการณ์ที่ทำให้เกิดการเปลี่ยนแปลงอาจเกิดจากออบเจกต์อื่นส่งเมสเสจมา การเปลี่ยนแปลงสถานะเรียกว่า ทรานซิชัน (Transitions)



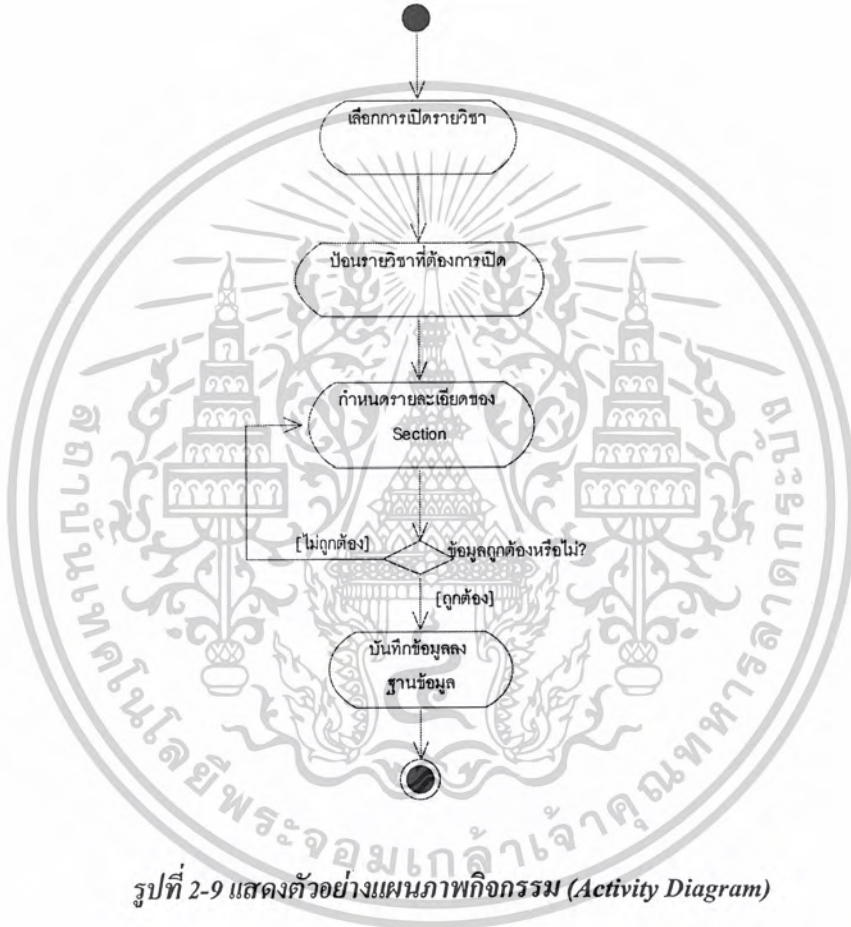
รูปที่ 2-8 แสดงตัวอย่างแผนภาพสถานะ (State Diagram)

7. Activity diagram แอกติวิตีไดอะแกรมแสดงลำดับการไหลของกิจกรรม(activity) ต่างๆ โดยจะอธิบายกิจกรรมต่างๆ ในลักษณะของการกระทำในไดอะแกรม จะแสดงเป็นสถานะการกระทำ (Action State)

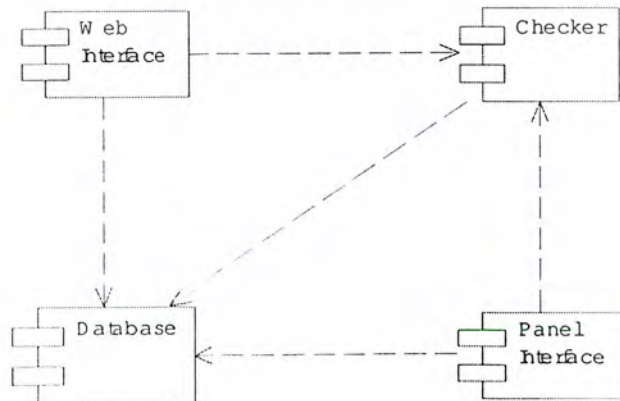
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งสถานะจะเปลี่ยนไปเมื่อเกิดการกระทำอย่างใดอย่างหนึ่งเกิดขึ้นตามเงื่อนไขหรือการตัดสินใจที่กำหนดไว้ เพื่อควบคุมการไหลของกิจกรรมรวมถึงสามารถมีเมสเสจที่รับ – ส่งระหว่างแต่ละกิจกรรม

8. Component diagram คอมโพเนนท์ไดอะแกรมแสดงโครงสร้างทางกายภาพของโค้ดในรูปคอมโพเนนท์ของโค้ด (Code) อาจเป็นส่วนประกอบของ ซอสเมื่อเกิดการเปลี่ยนแปลงของ คอมโพเนนท์ หนึ่งจะมีผลต่อคอมโพเนนท์อื่นๆ อย่างไร ซึ่งช่วยในการ โปรแกรม โค้ด (source code) คอมโพเนนท์จะประกอบด้วย ข้อมูลเกี่ยวกับ ลอจิกัล คลาส (logical class) ของคอมโพเนนท์ในไดอะแกรม มีการแสดงความสัมพันธ์หรือ ความพึ่งพากันของคอมโพเนนท์ให้ช่วยในการวิเคราะห์ว่า

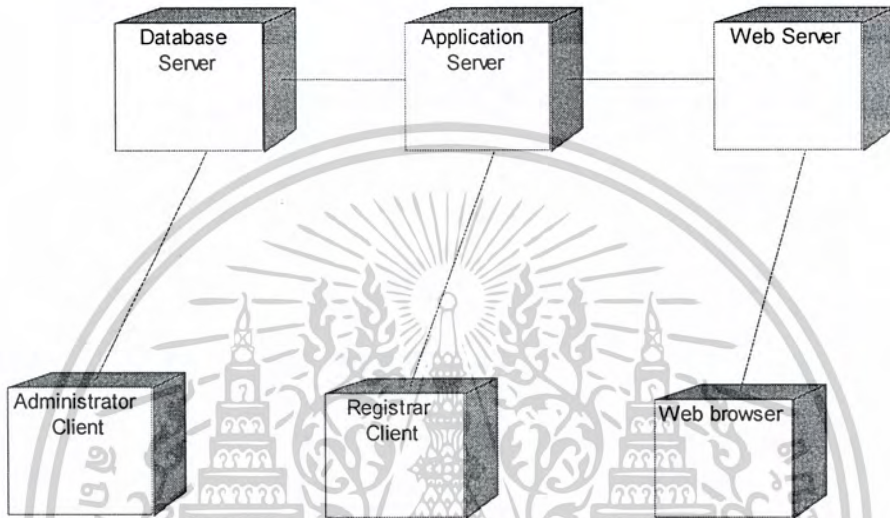


รูปที่ 2-9 แสดงตัวอย่างแผนภาพกิจกรรม (Activity Diagram)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 2-10 แสดงตัวอย่างแผนภาพคอมโพเนนท์ (Component Diagram)  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. Deployment diagram คือพลอยเมนที่ไดอะแกรมแสดงสถาปัตยกรรมทางกายภาพของส่วนประกอบต่างๆ ของฮาร์ดแวร์(hardware) ซึ่งถูกเรียกว่า โหนด(Node) ในระบบซึ่งสามารถแสดงเป็น โหนดและการเชื่อมต่อระหว่างชนิดของการเชื่อมต่อ นอกจากนี้ภายในโหนด ยังสามารถมีคอมโพเนนต์ หรือออบเจกต์ที่สามารถปฏิบัติกับโหนดเพื่อแสดงว่าโปรแกรมส่วนใดถูกปฏิบัติบนโหนดใดและความสัมพันธ์ระหว่างคอมโพเนนต์ที่อยู่บนโหนด ซึ่งโหนดจะถูกแทนด้วยสัญลักษณ์ของสี่เหลี่ยมลูกบาศก์ดังรูป



รูปที่ 2-11 แสดงตัวอย่างดีพลอยเมนไดอะแกรม

## บทที่ 3

# เทคโนโลยีจาวาที่ใช้ในการพัฒนาระบบ

### 3.1 จาวาเซิร์ฟเลต (Java Servlet)

Java Servlet คือเทคโนโลยี ที่เกิดขึ้นมาจาก Java เพื่อใช้งานทางด้านการสร้าง โปรแกรมในรูปแบบ CGI สำหรับทำงานบนเว็บเซิร์ฟเวอร์ เพื่อให้ผลการแสดงเว็บผู้ใช้งาน มีการเปลี่ยนแปลงตามการใช้งาน (On The Fly) ซึ่งเป็นเทคโนโลยีที่สืบเนื่องมาจากเหตุผลดังนี้

- ความต้องการให้หน้าเว็บเปลี่ยนตามข้อมูลที่ผู้อ่านส่งให้ เช่น เว็บที่ให้บริการระบบค้นหา (Search Engine) ทั้งหลาย ที่ได้รับข้อมูลจากผู้ใช้ คือข้อความที่ต้องการค้นหา และแสดงผลลัพธ์หน้าเว็บ ตามข้อมูลที่ค้นหาเจอแก่ผู้ใช้
- ความต้องการให้หน้าเว็บเปลี่ยนตามข้อมูลที่มีการเปลี่ยนแปลงบ่อยครั้ง เช่น เว็บที่เกี่ยวข้องกับการรายงานข่าว หรือสภาพอากาศ จะเห็นว่าข้อมูลที่รายงานสู่ผู้ใช้ ดังกล่าว มีการเปลี่ยนแปลงอยู่ตลอดเวลา อาจเป็นรายชั่วโมง หรือต่อนาทีก็ได้
- ความต้องการให้หน้าเว็บเปลี่ยนตามระบบฐานข้อมูล เช่น ระบบควบคุมคลังสินค้า ที่อาศัยการเก็บข้อมูลในระบบฐานข้อมูล สามารถแสดงผลจำนวนสินค้า ตามข้อมูลที่มีอยู่จริงในฐานข้อมูล ออกสู่ผู้ใช้บนระบบเว็บ

#### 3.1.1 ข้อแตกต่างของ Java Servlet เมื่อเทียบกับ CGI Servlet

Java Servlet เป็นเทคโนโลยี ที่มีรูปแบบการทำงานเหมือนกับ เทคโนโลยี CGI ที่มีการใช้งานอยู่ แต่เนื่องจาก Java Servlet ถูกพัฒนามาจากเทคโนโลยีจาวาอีกที ดังนั้น ต่อไปนี้คือข้อเปรียบเทียบซึ่งแสดงให้เห็นถึงความแตกต่างระหว่าง Java Servlet และ CGI

- ประสิทธิภาพ การทำงานของ CGI จะมีการสร้าง โปรเซส (Process) ใหม่ทุกครั้ง ที่มีการเรียกใช้งานจากผู้ใช้งาน ดังนั้นหากมีการเรียกใช้งาน จำนวนหลายครั้งในช่วงเวลาเดียวกัน ก็จะมีจำนวนโปรเซสเกิดขึ้นเท่ากับจำนวนครั้งที่เรียกใช้งาน โปรเซสจะมีขนาดและรูปแบบการทำงาน ที่เหมือนกัน ทำให้กินเวลาในการทำงานตามจำนวนโปรเซส แต่สำหรับ Java Servlet แล้ว ระบบเวอร์ชวลแมชชีน (Java Virtual Machine) จะถูกดึงขึ้นมาทำงานก่อน เพื่อทำให้เกิดโปรเซสตามลักษณะการเรียกใช้ของผู้ใช้เช่นกัน แต่การทำงานของ Java Servlet จะเป็นรูปแบบเธรด(Thread) แทนที่จะเรียกว่าโปรเซส ซึ่งแต่ละเธรดขนาดการทำงานเล็กกว่าเดิม และกินระยะเวลาการทำงานน้อยกว่า จึงทำให้รูปแบบของ Java Servlet มีประสิทธิภาพกว่านั่นเอง
- ความสะดวก หากคุณสามารถเขียนโปรแกรม ด้วยภาษา C, C++ หรือ Java อยู่แล้ว จะเกิดความสะดวกและรวดเร็วในการสร้าง Java Servlet มากกว่า แทนที่จะเริ่มศึกษาภาษาคำสั่งใหม่ๆ และ

Java Servlet ก็มีความสามารถเพียงพอสำหรับการใช้งานบนเว็บอยู่แล้ว เช่น การรับข้อมูลจากเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟอร์ม, การอ่านและตั้งค่า HTTP Header, การทำงานกับ Cookie, การทำงานแบบ Session และอื่นๆอีกมาก

- ความสามารถ Java Servlet มีข้อดีที่เห็นได้ชัดคือ สามารถติดต่อกับเว็บเซิร์ฟเวอร์ได้โดยตรง ในขณะที่ CGI ปกติทำไม่ได้ (ยกเว้นรุ่นใหม่ๆ) และ Java Servlet ยังสามารถแชร์ข้อมูลการทำงานร่วมกับโปรแกรมอื่นๆได้ด้วย
- การโยกย้าย เนื่องจาก Java Servlet ถูกสร้างมาจาก Java ซึ่งเป็นเทคโนโลยี ที่สามารถถูกใช้งาน ณ. ที่ใดก็ได้ โดยไม่จำกัดรูปแบบ ดังนั้น หากพัฒนา Java Servlet เพื่อใช้งานแล้ว สามารถที่จะโยกย้ายผลผลิตไปยัง ระบบอื่นๆได้ โดยไม่ต้องปรับปรุงหรือเปลี่ยนแปลง
- ไม่แพง ในปัจจุบันเว็บเซิร์ฟเวอร์ที่สนับสนุนการทำงาน Java Servlet มีอยู่มากมาย ทั้งที่เป็นแบบฟรี(ดาวน์โหลด) และซื้อ-ขายกัน นอกจากนี้ ในอนาคตแผนการผลิตเว็บเซิร์ฟเวอร์อื่นๆ ยังมีการสนับสนุนการใช้งาน Java Servlet อีกด้วย

### 3.1.2 โครงสร้างของ Java Servlet

การรับส่งข้อมูลโดยใช้เว็บ ผ่านทางแท็กฟอร์ม <FORM ...> มีรูปแบบการส่งอยู่ 2 ลักษณะคือ GET และ POST โดยที่การส่งแบบ GET เป็นรูปแบบการส่งข้อมูล ผ่านทาง URL (ช่องพิมพ์ชื่อเว็บหรือเว็บแอดเดรส) ซึ่งผู้ใช้อาจจะพิมพ์ได้เอง โดยตรง เช่น <http://www.javacentrix.com?user=1245&pwd=4444> หรือ ฟอร์มที่แท็ก กำหนดแอททริบิวต์ METHOD เป็น GET เช่น <FORM METHOD="GET" ... เป็นต้น (ตัวอย่างที่เป็นตัวหนาคือการส่งข้อมูลผ่านฟอร์มในแบบ GET)

สำหรับการส่งข้อมูลผ่านฟอร์มในแบบ POST คือการกำหนดแท็กฟอร์มให้มีแอททริบิวต์ METHOD เป็น POST เช่น <FORM METHOD="POST" ... เป็นต้น

### 3.1.3 การใช้ Java Servlet ส่งข้อมูลรูปแบบ HTML

แน่นอนว่าการสร้าง Servlet ขึ้นมาใช้งาน ไม่ได้มุ่งให้ส่งตัวอักษรเหมือนกับตัวอย่างที่ยกมา แต่ความต้องการคือ ให้สามารถส่งข้อมูลเว็บ ซึ่งก็คือข้อมูลในรูปแบบ HTML นั่นเอง ดังนั้นการส่งข้อมูล HTML ที่สำคัญ จำเป็นต้องส่งข้อมูลบอกลักษณะของรูปแบบ ให้กับผู้ใช้ (ซึ่งเป็นข้อมูลที่เบราว์เซอร์ได้รับ) ว่าเป็นข้อมูลรูปแบบ HTML ซึ่งเรียกกันว่า Content-Type แบบ text/html การส่งข้อมูลบอกเบราว์เซอร์นี้ ทำได้โดยการเรียกใช้เมธอด setContentType ของคลาส HttpServletResponse แล้วส่งข้อความ text/html เป็นอาร์กิวเมนต์ไป

### 3.1.4 การอ่านข้อมูลจากเว็บ

สำหรับการใช้งานเว็บผ่านเบราว์เซอร์ โดยทั่วไปแล้ว ปกติในช่องที่ใช้กำหนดข้อมูล URL จะประกอบด้วยชื่อเซิร์ฟเวอร์ ชื่อโดเมนทอร์รี่ และชื่อไฟล์เว็บเพจที่ต้องการดู เช่น หากข้อมูล URL คือ <http://www.javacentrix.com/tutorials/api/index.htm> นั้นหมายถึงต้องการอ่านเว็บเพจที่ชื่อ index.htm จากโดเมนทอร์รี่ tutorials/api จากเว็บเซิร์ฟเวอร์ที่ชื่อ www.javacentrix.com

สำหรับ URL นั้นนอกจาก จะใช้ชี้ตำแหน่งไฟล์ที่ต้องการเรียกดูแล้ว ยังสามารถใช้สำหรับส่งข้อมูลให้กับระบบเซิร์ฟเวอร์ เพื่อใช้ในการทำงาน หรือดำเนินการอย่างใดอย่างหนึ่ง ลักษณะ URL ประเภทนี้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากชื่อไฟล์แล้ว จะตามด้วยเครื่องหมายคำถาม ? และข้อมูลที่ต้องการส่งไปให้เซิร์ฟเวอร์ ซึ่งเรียกโดยทั่วไปว่าการส่งข้อมูลแบบ GET โดยมีรูปแบบคือ URL?Parameter1=Data1&Parameter2=Data2

จากรูปแบบการส่งข้อมูลผ่าน URL โดย Parameter1 และ Parameter2 คือชื่อพารามิเตอร์ที่ถูกเซิร์ฟเวอร์อ้างอิงได้ ในขณะที่ Data1 และ Data2 คือข้อมูลที่อยู่ในพารามิเตอร์แต่ละตัว หากต้องการส่งข้อมูลมากกว่า 2 ตัว ก็จะเพิ่มต่อท้ายไปได้เรื่อยๆ เช่น

URL?Parameter1=Data1&Parameter2=Data2&Parameter3=Data3&Parameter4=Data4

### 3.2 ภาษาเจเอสพี (JavaServer Page : JSP)

JavaServer Page หรือที่เรียกกันง่าย ๆ ว่า JSP เป็นเทคโนโลยีที่เป็นผลรวม ของการนำเอาหลักการสร้างเว็บแบบสแตติกด้วย HTML มารวมกับการสร้างเว็บแบบไดนามิก ก่อให้เกิดการแสดงผลเว็บ ที่มีการเปลี่ยนแปลงข้อมูลได้ตามการใช้งานของผู้ใช้ เห็นได้ว่าหากพูดถึงหลักการนำเสนอเว็บแล้ว เป็นหลักการที่ทำให้เว็บมีการเปลี่ยนแปลงข้อมูล เช่นเดียวกับ ที่ใช้โปรแกรมจำพวก CGI ทั่วไป แต่ในความหมายของ JSP แล้ว เป็นขั้นตอนและวิธีการสร้างที่ทำงานร่วมกับคำสั่ง HTML และเก็บบันทึกไว้ในไฟล์เดียวกัน แต่ในขณะที่ CGI เป็นโปรแกรมที่ผลิต HTML ให้เมื่อถูกสั่งให้ทำงาน

```
<HTML><HEAD><TITLE>Show JSP Page</TITLE></HEAD>
<BODY>
<% out.println("Hello Java Server Page"); %>
</BODY>
</HTML>
```

จากตัวอย่างด้านบนเป็นเพียงส่วนหนึ่งของ JSP ที่ถูกใส่ไว้ร่วมกับคำสั่ง HTML สังเกตจากตัวหนาที่อยู่ในแท็กพิเศษ <% ... %> ซึ่งเป็นคำสั่งในรูปแบบ JSP ทำให้ส่วนดังกล่าวสร้างข้อมูลในแบบไดนามิก ไว้ในเว็บเพจตามคำสั่ง ไฟล์ดังกล่าวจะบันทึกไว้ในนามสกุล .jsp จาก HTML เดิม

รูปแบบดังกล่าวไม่สามารถทำได้สำเร็จตามเทคโนโลยี JSP หากไฟล์ตัวอย่างด้านบนไม่ ได้ถูกเก็บไว้ในเว็บเซิร์ฟเวอร์ที่สนับสนุนการทำงานของ JSP ข้อความที่เป็นไดนามิกนั้น ถูกสำรวจเพื่อประมวลผล และใส่ลงในตำแหน่งดังกล่าว ด้วยความสามารถของ เซิร์ฟเวอร์ที่สนับสนุน JSP ก่อนส่งกลับไปให้ผู้ใช้งาน โดยผู้ใช้งานจะได้รับข้อความ Hello Java Server Page เท่านั้น โดยไม่เห็นคำสั่ง JSP เลย

#### 3.2.1 ข้อแตกต่างของ JSP เมื่อเทียบกับเทคโนโลยีอื่นๆ

JSP ไม่ได้เป็นเทคโนโลยีเดียว ในปัจจุบันที่สามารถทำให้เว็บ แสดงข้อมูลในแบบไดนามิกได้ มีเทคโนโลยีอื่นๆที่ทำงานในลักษณะนี้หลายแบบ เนื้อหาส่วนนี้ เป็นการแนะนำให้ผู้อ่านได้เห็นข้อแตกต่างบางประการ เมื่อเทียบกับ JSP

- Active Server Page(ASP) ASP เป็นเทคโนโลยีที่เหมือนกับ JSP แต่เป็นเทคโนโลยีที่พัฒนามาจาก บริษัทไมโครซอฟต์ ผู้ผลิตระบบปฏิบัติการวินโดวส์ที่รู้จักกัน โดยทั่วไป ลักษณะ JSP มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในองค์กรเท่านั้น เมื่อคุณอยู่ที่หน้าเว็บไซต์ของบริษัทฯ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาษา VB หรือภาษาใดๆ ซึ่งเป็นข้อกำหนดเฉพาะของ ไมโครซอฟต์ ประการที่สอง JSP ซึ่งเป็นผลพวงมาจากประการแรกคือ สามารถโยกย้ายการทำงาน ไปใช้งานระบบปฏิบัติการใดๆก็ได้ ที่มีการใช้งานกันอยู่ในปัจจุบัน โดยไม่ต้องแก้ไขหรือเปลี่ยนแปลงใดๆ ในขณะที่ ASP ก็โยกย้ายได้เช่นเดียวกัน แต่ต้องอยู่ในระบบปฏิบัติการที่ ไมโครซอฟต์กำหนดขึ้นเท่านั้น

- Servlet JSP สามารถทำงานได้เช่นเดียวกับที่ Java Servlet ทำได้ แต่มีจุดเด่นมากกว่าที่ JSP มีความสะดวกในการสร้าง และเปลี่ยนแปลง มากกว่า เพราะทำโดยตรงที่ไฟล์ HTML มากกว่าการที่ต้องลงมือเขียนคำสั่งภาษาจาวาเหมือน Java Servlet ซึ่งต้องนำไปคอมไพล์ ก่อนการใช้งาน
- Server-Side Includes(SSI) SSI คือรูปแบบการสร้างไดนามิกเว็บเช่นเดียวกัน แต่รูปแบบการทำงานคือการนำเอาข้อมูลที่มีอยู่แล้ว บนเซิร์ฟเวอร์นำมาประกอบใส่ ในเว็บเท่านั้น ต่างจาก JSP ซึ่งสามารถนำข้อมูลมาใส่ แต่มีรูปแบบในการทำงาน มีรูปแบบในการประมวลผล หรือการเรียกโปรแกรมภายนอก เช่น Java Servlet มาช่วยในการทำให้ข้อมูล มีรูปแบบต่างๆ เช่น ดึงข้อมูลมาจากระบบฐานข้อมูล (Database) นอกจากข้อมูลที่มีอยู่บนเซิร์ฟเวอร์เช่นที่ใช้ใน SSI เท่านั้น
- JavaScript เป็นการทำเนื้อหาไดนามิกให้กับเว็บ แต่รูปแบบการทำงานเกิดขึ้นของ JavaScript เกิดจากการประมวลผลและดึงข้อมูล ที่มีอยู่บนเครื่องผู้ใช้ หรือไคลเอนต์เท่านั้น มารวบรวมเนื้อหาในเว็บ ต่างกับ JSP ที่เป็นการทำไดนามิก แต่ข้อมูลถูกสร้าง และดึงมาจากระบบเซิร์ฟเวอร์ ที่มีหลากหลายมากกว่า และทำให้เนื้อข้อมูลเป็นเนื้อเดียวกัน เมื่อผู้ใช้เรียกดู แต่ JavaScript เนื้อข้อมูลเป็นของเครื่องผู้ใช้เอง ซึ่งผู้ใช้ต่างกัน(ต่างเครื่อง) ก็จะให้ข้อมูลที่คล้ายกัน
- Static HTML หรือสแตติก HTML แน่นนอนข้อนี้ คงเห็นได้ชัดเจนที่ว่า สแตติก HTML ให้เนื้อข้อมูลที่คงที่ตลอดเวลา ไม่ว่าจะถูกเรียกดูในเวลาใดๆ (ยกเว้นมีคนมาแก้ไขคำสั่ง HTML) ในขณะที่ JSP ทำให้เนื้อข้อมูลมีการเปลี่ยนแปลงตามการใช้งาน หรือเครื่องผู้ใช้ โดยไม่ต้องเปลี่ยนแปลงแก้ไขคำสั่ง JSP

	CGI/Perl	Mod_Perl	ASP	JSP
เว็บเซิร์ฟเวอร์	Any Web server	Apache Web Server	Microsoft IIS or Personal Web Server	Any Web server, including Apache, Netscape, IIS today
โยกย้ายการทำงาน	No	No	No	Yes
ลักษณะนำกลับมาใช้	No	No	No	Yes
สคริปต์หรือภาษา	C, Perl	Perl	VBScript, JScript	Java
ปกป้องการใช้งานหน่วยความจำ	Yes	No	No	Yes
สนับสนุนการทำงานร่วมในหลายโปรเซส	No	Yes	Yes	Yes

ตารางที่ 3-1 เปรียบเทียบเทคโนโลยี JSP กับเทคโนโลยีอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น เมื่อผู้ดูเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

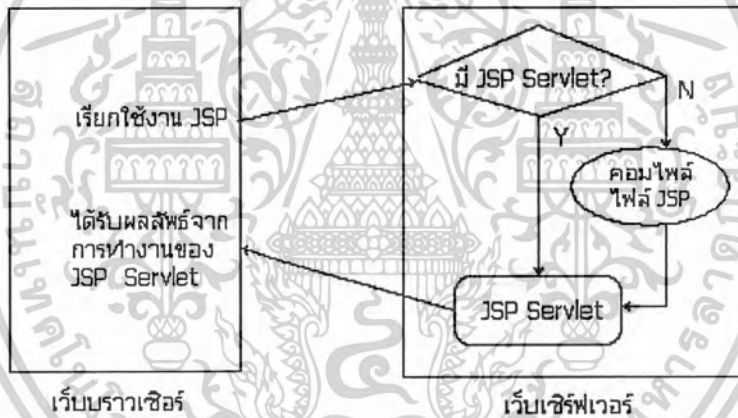
### 3.2.2 หลักการทำงานของ JSP และ ไตรรกะที่ฟ

JSP (Java Server Page)เป็นผลพวงมาจาก เทคโนโลยีของ Java Servlet ซึ่งเป็นหลักการสร้างระบบ การประมวลผลเว็บ ที่ทำงาน โดยอาศัยทรัพยากร (เมโมรี, ซีพียู, ดิสก์) ของเว็บเซิร์ฟเวอร์ ก่อนจะส่งผลลัพธ์สู่ ผู้ใช้ ผ่านระบบเน็ตเวิร์คหรืออินเทอร์เน็ต โดยผู้สร้าง JSP สามารถสร้างโดยการ แทรกคำสั่ง JSP ไว้ในชุดคำสั่ง HTML ที่ใช้สร้างเว็บ และบันทึกไว้ในไฟล์นามสกุล JSP จากนั้นเก็บไว้ที่เซิร์ฟเวอร์ ที่สนับสนุนการทำงานของ JSP

### 3.2.3 JSP Servlet

JSP Servlet คือ Servlet ที่เกิดขึ้นจากชุดคำสั่ง JSP ที่ถูกคอมไพล์ไปเป็น JSP Servlet โดยขบวนการนี้ เกิดโดยอัตโนมัติภายในเว็บเซิร์ฟเวอร์ที่สนับสนุน การทำงานของ JSP และ JSP Servlet นี้เองที่เป็นส่วนที่เกิด การประมวลผลบนเซิร์ฟเวอร์ ดังนั้นไฟล์นามสกุล JSP ที่ถูกเรียกใช้งาน จะต้องถูกคอมไพล์ไปเป็น JSP Servlet ก่อนในครั้งแรกของการเรียกใช้งาน หรือเมื่อมีการเปลี่ยนแปลงชุดคำสั่ง

### 3.2.4 การทำงานของ JSP



รูปที่ 3-1 รูปอธิบายการทำงานของ JSP

เมื่อเว็บเบราว์เซอร์ มีการร้องขอโดยผู้ใช้เพื่อเรียกข้อมูล JSP ที่อยู่บนเซิร์ฟเวอร์ ซึ่งก็หมายถึงไฟล์ที่มี นามสกุล JSP เว็บเซิร์ฟเวอร์จะทำการตรวจสอบว่า ไฟล์นั้นมีการคอมไพล์ไปเป็น JSP Servlet แล้วหรือยัง ถ้า ยังไฟล์ JSP จะถูกคอมไพล์ โดยตัวเว็บเซิร์ฟเวอร์ ไปเป็น JSP Servlet แต่ถ้าไฟล์ถูกคอมไพล์แล้ว (ซึ่งเกิดขึ้น จากการใช้งานในครั้งแรก) เว็บเซิร์ฟเวอร์จะนำ JSP Servlet มาใช้งาน โดยการเรียกเพื่อประมวลผล ได้ผลลัพธ์ ใดๆก็จะเป็นข้อมูล ซึ่งเว็บเซิร์ฟเวอร์ จะส่งกลับไปให้ผู้เรียกใช้งาน และการประมวลผลของ JSP Servlet นี้เอง ที่ก่อให้เกิด ข้อมูลเว็บในแบบไดนามิก

## บทที่ 4

# เครื่องมือที่ใช้ในการพัฒนาฐานข้อมูล

### 4.1 เครื่องมือที่ใช้ในการพัฒนาฐานข้อมูล IBM DB2

ในบทนี้จะขอกกล่าวถึงเครื่องมือที่ใช้ในการพัฒนาฐานข้อมูลซึ่งมี 2 โปรแกรมด้วยกันที่สัมพันธ์กันนั้นก็คือ IBM DB2 ที่ใช้ในการพัฒนาฐานข้อมูลและ JBuilder ในการพัฒนาโปรแกรม

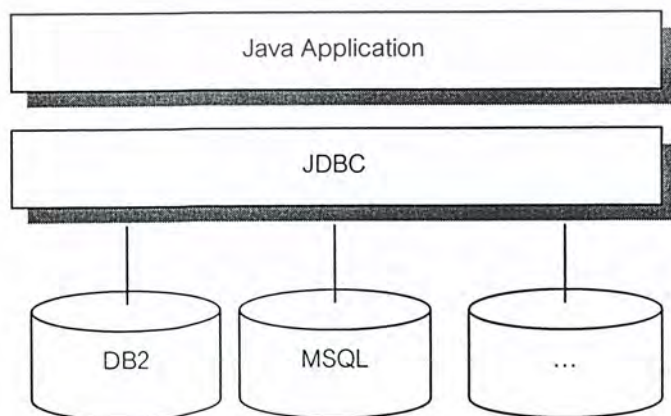
IBM DB2 เป็น ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management) ระบบหนึ่งซึ่งได้รับการอนุเคราะห์จากบริษัทไอบีเอ็ม ประเทศไทย โดยทางบริษัทได้ให้ชุดผลิตภัณฑ์ของ DB2 ประกอบไปด้วย DB2 Client Tool, DB2 UDB และ DB2 Personal Edition สำหรับโครงการนี้นำเอา DB2 มาใช้ในการพัฒนา และ ทดสอบโครงการ โดย DB server จะใช้ DB2 Universal Database (UDB) for Unix ในการจัดการฐานข้อมูลสารสนเทศนักศึกษา และใช้ DB2 Client Tool for Window ในการทดสอบโปรแกรม

DB2 Universal Database (UDB) เป็น RDBMS ที่สนับสนุนความสามารถในการจัดการ RDBMS ผ่านเว็บอย่างเต็มรูปแบบ โดยโครงการนี้ใช้ version 7.2 ซึ่งมีความสามารถใหม่ๆ เช่น

- สนับสนุน XML โดย DB2 ขอมให้ XML เป็นชนิดข้อมูลในคอลัมน์ได้
- สามารถทำงานร่วมกับ โปรแกรมต่างๆ เช่น The DB2 Software Developer's Kit (SDK), XML, JBuilder ฯลฯ

### 4.2 Java Database Connectivity (JDBC)

JDBC (Java Database Connectivity) ถูกพัฒนาโดย JavaSoft Department ของบริษัท Sun Microsystems ซึ่งก็คือฟังก์ชันมาตรฐานหรือ Java Application Programming Interface (API) สำหรับการเชื่อมต่อกับระบบฐานข้อมูลสัมพันธ์ (Relation Database) ใน Library ของ JDBC ได้มีส่วนที่ติดต่อกับฐานข้อมูลและได้นำคำสั่ง SQL ไปประมวลผลและควบคุมการทำงานของฐานข้อมูลเชิงสัมพันธ์ ดังรูป 4.1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในวงที่อาจารย์จะเห็นว่าไม่เหมาะสมให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 4-1 แสดงลักษณะการทำงานของ JDBC  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JDBC ถูกสร้างขึ้นมาในระดับของการเชื่อมต่อกับฐานข้อมูลในรูปแบบที่คล้ายกับ ODBC (ของทางบริษัทไมโครซอฟท์) หลักการของทั้งสองมาตรฐานตั้งอยู่บนฐานเดียวกันคือ X/Open SQL Call-Level Interface ของระบบ X-windows และ JDBC driver ต้องเข้ากันกับระดับมาตรฐานในการเข้าถึง SQL (ANSI SQL Entry Level Standards) และต้องผ่าน Conformance test ซึ่ง JavaSoft เป็นผู้กำหนดขึ้น

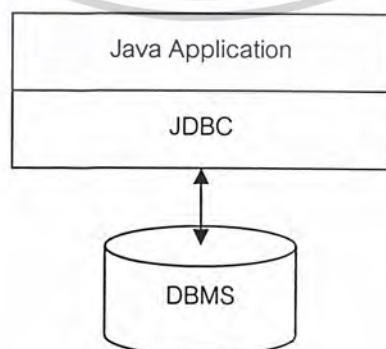
#### 4.2.1 รูปแบบการเชื่อมต่อฐานข้อมูลของ JDBC

โดย JDBC API นั้น สนับสนุนการเชื่อมต่อฐานข้อมูลแบบ 2-Tier และ 3-Tier

1. รูปแบบ 2-Tier จาวาแอปพลิเคชันหรือจาวาแอปพลิเคชันจะทำการติดต่อกับฐานข้อมูลโดยตรงจึงมีความจำเป็นที่โปรแกรมจาวาต้องการ JDBC Driver พิเศษที่สามารถสื่อสารกับระบบจัดการฐานข้อมูลชนิดนั้นได้ รูปแบบการเชื่อมต่อแสดงรูปที่ 4-2

คำสั่งในการเรียกค้นข้อมูลในการเรียกค้นข้อมูลในรูปแบบของภาษา SQL จะถูกส่งจากผู้ใช้ไปสู่ฐานข้อมูลหลังจากนั้นการประมวลผลของระบบจัดการฐานข้อมูลก็จะถูกส่งกลับมาให้ผู้ใช้ ฐานข้อมูลดังกล่าวนี้ส่วนมากจะถูกติดตั้งอยู่บนคอมพิวเตอร์อีกเครื่อง โดยผู้ใช้สามารถเชื่อมต่อผ่านระบบเน็ตเวิร์ค รูปแบบ 2-tier นี้ใช้หลักการทำงานเช่นเดียวกับรูปแบบไคลเอนต์/เซิร์ฟเวอร์ที่เรารู้จักกันดี โดยที่เครื่องคอมพิวเตอร์ของผู้ใช้คือไคลเอนต์และเครื่องคอมพิวเตอร์ที่ให้บริการฐานข้อมูลคือเซิร์ฟเวอร์ เครื่องข่ายคอมพิวเตอร์ที่นิยมใช้รูปแบบ 2-Tier มักจะเป็นเครือข่ายอินทราเน็ต (Intranet) สำหรับดำเนินธุรกรรมภายในองค์กร

2. รูปแบบ 3-Tier คำสั่งต่างๆจากผู้ใช้จะถูกส่งไปให้กับ Tier กลางหรือตัวกลาง หลังจากนั้น Tier กลางจะแปลงคำสั่งเหล่านั้นให้เป็นภาษา SQL เพื่อส่งให้กับระบบฐานข้อมูลเพื่อทำการประมวลผลและทำการส่งผลลัพธ์คืนให้กับ Tier กลางและส่งให้ผู้ใช้ในที่สุด หลักการทำงานเช่นนี้มักจะพบในเครือข่ายอินเทอร์เน็ตซึ่งประกอบด้วยเครื่องคอมพิวเตอร์หลากหลายชนิดและเว็บเซิร์ฟเวอร์ (Web Server) ซึ่งทำหน้าที่เป็น Tier กลางในการจัดการให้ระบบคอมพิวเตอร์ทั้งไคลเอนต์และเซิร์ฟเวอร์ฐานข้อมูลสามารถติดต่อกันได้ การปรับเปลี่ยนระบบไคลเอนต์และเซิร์ฟเวอร์ หรือแม้กระทั่งการเปลี่ยนตัวฐานข้อมูลตัวใหม่จะไม่มีผลกระทบซึ่งกันและกันเกิดขึ้น หรือแม้กระทั่งการเปลี่ยนตัวฐานข้อมูลตัวใหม่จะไม่มีผลกระทบซึ่งกันและกันเกิดขึ้น แสดงดังรูป 4.3

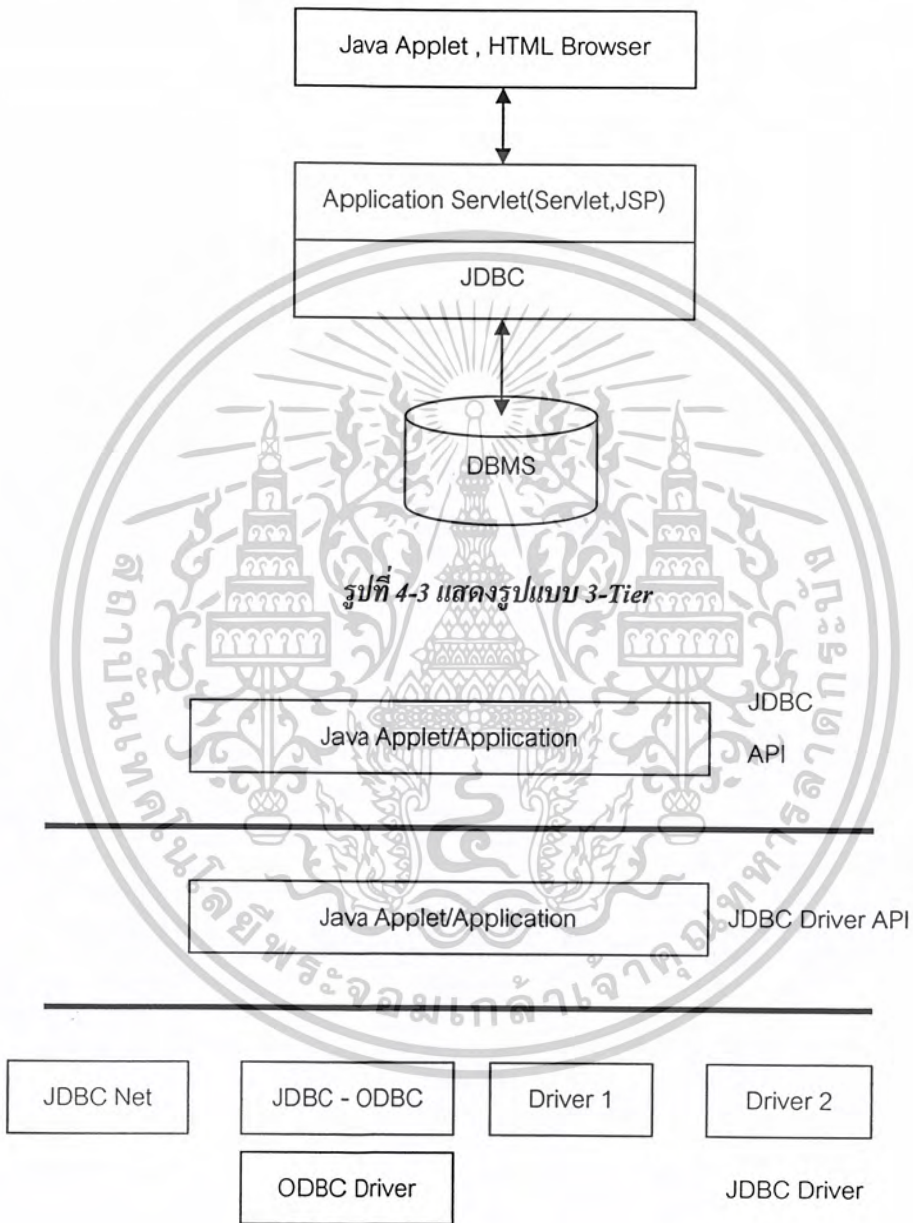


รูปที่ 4-2 แสดงรูปแบบ 2-Tier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.2.2 โครงสร้างของ JDBC

โครงสร้างการเชื่อมต่อภายใน JDBC ประกอบด้วย 3 ระดับ คือ JDBC API, JDBC Driver API และ JDBC Driver ดังรูปที่ 4.4 ระดับบนสุด เป็น JDBC API เป็นระดับฟังก์ชัน API ที่อำนวยความสะดวกให้แก่โปรแกรมประยุกต์ ระดับกลาง JDBC Driver (มี Driver ที่ต่างกันอยู่ 4 ชนิด) ที่เหมาะสม



รูปที่ 4-4 ระดับการเชื่อมต่อของ JDBC API

### 4.2.3 รูปแบบของ JDBC Driver

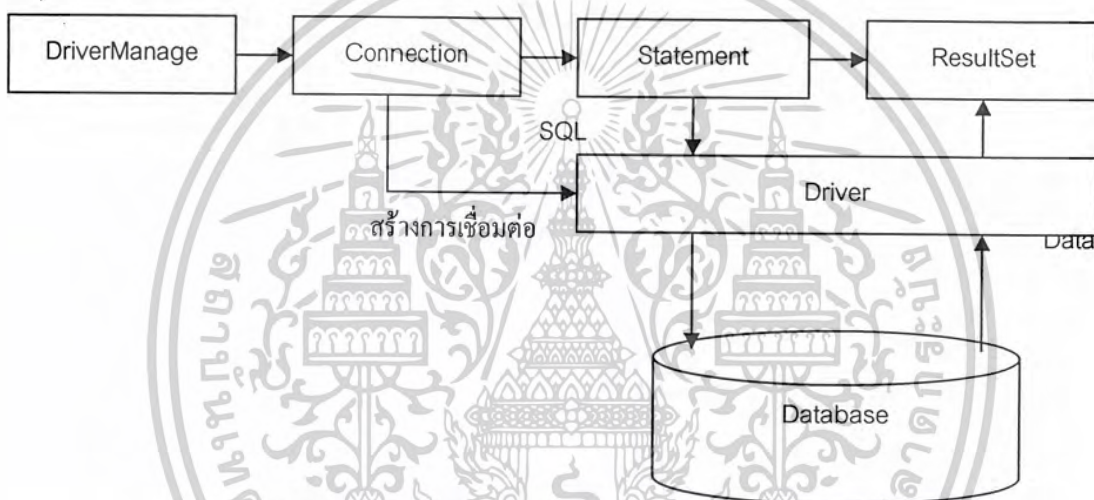
รูปแบบของ JDBC Driver หรือส่วนที่อยู่เบื้องหลังการทำงานของฟังก์ชัน API ต่างๆ ของ JDBC ถูกแบ่งออกเป็นทั้งหมด 4 ชนิดคือ

- JDBC-ODBC Bridge
- เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Native-API(Party-Java) Driver
- Net-Protocol(All-Java) Driver
- Nativer-Protocol(All-Java)

ในที่นี้จะขอล่าถึงแค่ JDBC/ODBC Bridge ถูกพัฒนาโดย JavaSoft และ InterSolv ทำหน้าที่เป็นตัวกลางในการเข้าถึงฐานข้อมูลได้โดยผ่านทางการทำงานของ ODBC โดยนำข้อดีของ ODBC-enabled data source ที่มีข้อดีอยู่โดยทั่วไปอย่างมากมา ผังไคลเอนต์ที่เป็นจาวาแอปพลิเคชันหรือจาวาแอปพลิเคชันจะถูกเขียนโดยใช้ JDBC API Bridge จะทำการแปลงการเรียกใช้ JDBC ไปยัง ODBC Driver ที่เหมาะสมสำหรับฐานข้อมูล ข้อดีของ Bridge ทำให้แอปพลิเคชันสามารถติดต่อกับฐานข้อมูลได้ง่ายจากผู้ผลิตที่มากโดยสามารถเลือกได้จาก ODBC Driver ที่เหมาะสม

4.2.4 ขั้นตอนการติดต่อกับระบบจัดการฐานข้อมูล



รูปที่ 4-5 ลำดับการเชื่อมต่อเข้าฐานข้อมูลโดยใช้ JDBC

ขั้นตอนการติดต่อกับระบบจัดการฐานข้อมูล (DBMS:ในที่นี้คือ IBM DB2) ด้วยภาษาจาวา สำหรับการเชื่อมต่อกับระบบจัดการฐานข้อมูลด้วยภาษาจาวานั้นจะต้องประกอบด้วยขั้นตอนที่มากพอสมควร โดยแสดงเป็นขั้นตอนดังนี้

- เรียกใช้คลาสที่จำเป็น เช่น  
`import java.sql.*;`
- เลือกใช้ Driver ของ DB2  
`Class.forName("COM.ibm.db2.jdbc.net.DB2Driver").newInstance();`
- สร้างออบเจกต์ Connection  
`Connection dbconn;`
- กำหนดแหล่งของข้อมูลภายในระบบจัดการฐานข้อมูล  
`dbconn = DriverManager.getConnection(url, userid, passwd);`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สร้างออบเจกต์ Statement  
Statement statement = dbconn.createStatement();
- ประมวลผลคำสั่ง SQL โดยใช้คำสั่งออบเจกต์ Statement  
statement.executeQuery("SELECT \* FROM STUDENT");
- รับผลจากการประมวลผลคำสั่งด้วยออบเจกต์ Resultset  
Resultset res = statement.executeQuery(query);
- ปิดออบเจกต์ Resultset  
res.close()
- ปิดออบเจกต์ Statement  
statement.close();
- ปิดออบเจกต์ Connection  
dbcon.close();



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# การวิเคราะห์และออกแบบระบบทะเบียน

## นักศึกษาระดับบัณฑิตศึกษา

ในบทนี้จะได้กล่าวถึงขั้นตอนการวิเคราะห์และปรับปรุงระบบทะเบียนนักศึกษาระดับบัณฑิตศึกษา โดยเริ่มด้วยการอธิบายถึงหน้าที่ความรับผิดชอบ การดูแลงานด้านทะเบียนนักศึกษาของฝ่ายทะเบียนนักศึกษา บัณฑิตวิทยาลัย และวิเคราะห์ถึงข้อเสีย และจุดบกพร่องต่าง ๆ ของระบบทะเบียนนักศึกษาที่ใช้อยู่ในปัจจุบัน และที่พัฒนามาแล้ว หลังจากนั้นก็จะกล่าวถึงขั้นตอนการวิเคราะห์และการออกแบบ โดยการนำเอายูเอ็มแอล เข้ามาอธิบายลักษณะของระบบ ดังนี้

### 5.1 ฝ่ายทะเบียนนักศึกษาของบัณฑิตวิทยาลัย

ฝ่ายทะเบียนนักศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เป็นหน่วยงานที่รับผิดชอบงานในเรื่องของทะเบียนนักศึกษา ในระดับบัณฑิตศึกษาทุกชั้นตอนนับตั้งแต่ การรับนักศึกษาใหม่ การขึ้นทะเบียนนักศึกษาใหม่ การลงทะเบียน การแก้ไขข้อมูลการลงทะเบียน การเปิด-ปิด รายวิชา การกรอกข้อมูลเกรด การเพิ่มแก้ไขข้อมูลหลักสูตร การดำเนินการตามคำร้องต่าง ๆ ของนักศึกษา จนถึงเรื่องของการจบการศึกษา

การทำงานของฝ่ายทะเบียนในปัจจุบันนั้น มีระบบสารสนเทศของสำนักวิจัยและบริการคอมพิวเตอร์ รองรับการทำงานอยู่ ซึ่งนับว่าเป็นระบบที่ได้อำนวยความสะดวกให้มากพอสมควร แต่ก็ยังคงพบปัญหาในการทำงานอยู่เช่นกัน ในที่นี้ขอยกตัวอย่างหนึ่งในงานที่ฝ่ายทะเบียนต้องรับผิดชอบ นั่นก็คือการลงทะเบียนเรียน ของนักศึกษา แต่เดิมนั้นใช้รหัสแท่งเข้าช่วย โดยนักศึกษาจะต้องคัดแถบรหัสแท่งของรายวิชาที่ต้องการจะ ลงทะเบียน มาแนบลงบนเอกสารการลงทะเบียนแล้วนำเอกสารดังกล่าวไปให้อาจารย์ที่ปรึกษาทำการ ตรวจสอบ และเห็นชอบ หลังจากให้อาจารย์ที่ปรึกษาได้ลงนามเห็นชอบแล้วถึงจะนำเอกสารดังกล่าวมาให้ เจ้าหน้าที่ทะเบียน เพื่อเอาข้อมูลลงสู่ระบบ แล้วออกจะออกเอกสารการชำระเงินเพื่อให้นักศึกษานำเงิน ไปชำระ ที่ธนาคารที่ได้กำหนดไว้ ก็จะสิ้นสุดขั้นตอนการลงทะเบียน ต่อมาในปีการศึกษา 2544 ได้ทำการปรับระบบให้ นักศึกษาสามารถทำการลงทะเบียนผ่านระบบอินเทอร์เน็ตได้ โดยมีการยกเลิกการตรวจสอบของอาจารย์ที่ ปรึกษาไป หลังจากให้นำระบบดังกล่าวเข้ามาใช้ก็ยังคงพบปัญหาหลายอย่างอันเนื่องมาจากเกิดความสับสนใน ขั้นตอนการปฏิบัติ รวมถึงระบบที่นำมาใช้ไม่ได้มีการตรวจสอบข้อกำหนดในการลงทะเบียนอย่างครบถ้วน อย่างเช่นการตรวจสอบรายวิชาที่จำเป็นต้องเรียนมาก่อน (Prerequisite) เป็นต้น จากตัวอย่างการลงทะเบียน เรียนของนักศึกษานับเป็นงานหนึ่งในหลายงานที่ฝ่ายทะเบียนต้องรับผิดชอบ ซึ่งคงไม่สามารถกล่าวถึง รายละเอียดของงานทุกงานได้ในที่นี้

ในการดำเนินงานของฝ่ายทะเบียน และการตรวจสอบ เงื่อนไขที่ระบบไม่ได้รองรับไว้นั้น จะต้องใช้ บุคคลเป็นผู้ดำเนินการเป็นส่วนใหญ่ และยังมีขั้นตอนการปฏิบัติงานบางอย่างที่ยังไม่ชัดเจน รวมถึงกฎ เอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อบังคับต่าง ๆ ที่อาจล้าสมัยอยู่บ้าง ทำให้เกิดความสับสนในการปฏิบัติ ซึ่งเมื่อเกิดปัญหาขึ้นก็จะมีการแก้ปัญหาเป็นกรณี ๆ ไปซึ่งอาจก่อให้เกิดปัญหานั้นซ้ำอีกได้ในภายหลัง

## 5.2 ปัญหาที่ผู้ใช้ต้องการแก้ไข

ก่อนที่จะดำเนินการพัฒนาโครงการนี้ ผู้พัฒนาโครงการได้เข้าไปศึกษาทำความเข้าใจถึงการทำงานเดิมของระบบทะเบียนนักศึกษาในระดับบัณฑิตศึกษา รวมทั้งสัมภาษณ์เจ้าหน้าที่ทะเบียนบัณฑิตวิทยาลัย ซึ่งทำให้ได้ทราบถึงจุดอ่อนของระบบเดิมอยู่หลายประการคือ

- ในขั้นตอนการลงทะเบียนของนักศึกษาระดับบัณฑิตศึกษานั้นมีรายละเอียดหลายขั้นตอน ตั้งแต่การเข้าดูรายวิชาที่เปิดสอนร่วมกับการปรึกษาอาจารย์ที่ปรึกษาหลังจากที่ได้ข้อตกลงในการลงทะเบียน แล้วนักศึกษาต้องทำการตัดกระดาษรหัสแท่ง รายวิชาที่ต้องการพร้อมมาดำเนินการลงทะเบียนที่ฝ่ายทะเบียนซึ่งนับว่าเป็นขั้นตอนที่ซับซ้อน
  - รายวิชาที่เปิดสอนในระดับบัณฑิตศึกษานั้น เปิดเพื่อให้นักศึกษาสามารถเลือกลงทะเบียน ดังนั้นอาจมีบางรายวิชาที่นักศึกษาให้ความสนใจลงทะเบียนน้อย ทำให้รายวิชานั้นต้องปิดลงไปสร้างความเดือดร้อนให้กับนักศึกษาที่ลงทะเบียนวิชานั้น ๆ อยู่
  - ระบบเดิมยังไม่ได้มีการตรวจสอบเงื่อนไขการลงทะเบียนอย่างจริงจัง อย่างเช่นการตรวจสอบรายวิชาที่จำเป็นต้องเรียนมาก่อน(Prerequisite) จำนวนหน่วยกิตรวม และในบางวิชาที่จำกัดจำนวนผู้ที่ลงทะเบียนยังไม่สามารถจำกัดได้ ทำให้เกิดการลงทะเบียนที่ผิดเงื่อนไขได้ง่าย
  - ในระบบเดิมยังไม่สามารถตรวจสอบรายวิชาที่นักศึกษาได้ลงทะเบียนไปแล้วได้ จึงทำให้เกิดปัญหาในการลงทะเบียนซ้ำซ้อนในรายวิชาเดียวกัน และนักศึกษาที่ถูกตัดชั้นไว้ยังสามารถทำการลงทะเบียนได้เหมือนนักศึกษาทั่วไป
  - ประเภทนักศึกษาที่มีหลายประเภทเช่นนักศึกษาสามัญ นักศึกษาทดลองเรียน ซึ่งมีระเบียบปฏิบัติบางอย่างที่แตกต่างกัน
  - นักศึกษาต้องไปติดต่อกับฝ่ายทะเบียนบ่อยครั้ง ไม่ว่าจะเป็นการยื่นคำร้องต่าง ๆ การเพิ่มแก้ไขการลงทะเบียน การตรวจสอบข้อมูลเกี่ยวกับการลงทะเบียนและเกรด
  - เจ้าหน้าที่ทะเบียนต้องทำงานซ้ำซ้อนเช่น การกรอกเกรดนักศึกษา ต้องส่งแบบฟอร์มให้อาจารย์เป็นผู้กรอกแล้วนำมากรอกซ้ำอีกครั้ง ซึ่งอาจเกิดข้อผิดพลาดขึ้นและทำให้เสียเวลาการทำงาน
- นั่นคือตัวอย่างของปัญหาที่เกิดขึ้นในระบบเดิม ปัญหาเหล่านี้ก่อให้เกิดความสับสนในการทำงาน จึงเป็นการเหมาะสมที่จะพัฒนาระบบให้มีประสิทธิภาพมากขึ้น

## 5.3 การแก้ไขปัญหาระบบเดิม

เนื่องจากปัญหาของระบบเดิมข้างต้น จึงได้มีการพัฒนาระบบสารสนเทศนักศึกษาขึ้นด้วยวิธีการเชิงวัตถุ โดยมีผู้พัฒนามาแล้วทางผู้พัฒนาต่อจึงได้ทำการศึกษาวิเคราะห์ระบบและสัมภาษณ์ผู้พัฒนามาแล้วนั้นทำให้พบว่าความสามารถของระบบที่ได้พัฒนามาแล้วมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนของแอปพลิเคชันได้ทำการพัฒนาในส่วนของการจัดการข้อมูลหลักสูตร นักศึกษา หน่วยงานการศึกษา ระยะเวลาการลงทะเบียน การจัดการรายวิชา จัดการค่าใช้จ่ายต่างๆ ลงทะเบียนเรียน เพิ่ม ถอน เปลี่ยนรายวิชา ป้อนเกรดในแต่ละรายวิชาและจัดการ ใบทรานสคริปต์
- ในส่วนของเว็บแอปพลิเคชันได้ทำการพัฒนาในส่วนของการลงทะเบียนล่วงหน้า ลงทะเบียน ปกติ ตรวจสอบรายละเอียดรายวิชา ตรวจสอบข้อมูลการลงทะเบียน ตรวจสอบเกรด
- ระบบที่พัฒนาขึ้นสามารถทำการตรวจสอบเงื่อนไขได้ต่างๆ เช่น
  - ตรวจสอบรายวิชาที่นักศึกษาได้ลงทะเบียนไปแล้ว จึงทำให้ไม่เกิดปัญหาในการลงทะเบียนซ้ำซ้อนในรายวิชาเดียวกัน
  - ตรวจสอบจำนวนผู้ลงทะเบียน เนื่องจากมีบางคณะจำกัดจำนวนผู้ลงทะเบียนแต่ละวิชา โดยถ้ามีผู้ลงทะเบียนครบแล้ว ผู้ที่ลงทะเบียนทีหลังจะไม่สามารถลงทะเบียนได้
  - ตรวจสอบรายวิชาที่จำเป็นต้องเรียนมาก่อน (Prerequisite)
  - ตรวจสอบรายวิชาที่ถูกปิดไป โดยถ้ารายวิชาถูกปิดไป จะไม่สามารถทำการลงทะเบียนรายวิชาดังกล่าวได้
  - ตรวจสอบจำนวนรวมหน่วยกิต สูงสุด , ต่ำสุด ที่นักศึกษาสามารถทำการลงทะเบียนได้
  - สามารถตรวจสอบ ว่านักศึกษาถูกทำทัณฑ์บน หรือ พ้นสภาพ หรือไม่

ข้างต้นแล้ว

#### 5.4 ปัญหาที่เหลือและปัญหาที่เกิดจากระบบที่ได้พัฒนามาแล้ว

หลังจากที่ได้ศึกษาระบบทะเบียนนักศึกษาระดับบัณฑิตศึกษาเดิมแล้ว จึงทำการศึกษาระบบทะเบียนนักศึกษาระดับบัณฑิตศึกษาจากระบบที่ได้พัฒนามาแล้ว และทำการวิเคราะห์เปรียบเทียบทำให้เห็นข้อบกพร่องและงานที่ยังขาดไป แยกเป็นส่วน ๆ ดังนี้

##### 5.4.1 ส่วนการลงทะเบียน

เป็นส่วนที่ต้องเพิ่มและแก้ไขเนื่องจากระบบที่พัฒนามาแล้วยังมีปัญหาอยู่ คือ

- การลงทะเบียนลาพัก เป็นส่วนที่ต้องทำเพิ่ม เนื่องจากในระบบที่ได้พัฒนามาแล้วไม่มีส่วนการทำงานนี้
- การลงทะเบียนรักษาสภาพ เป็นส่วนที่ต้องทำเพิ่ม เนื่องจากในระบบที่ได้พัฒนามาแล้วไม่มีส่วนการทำงานนี้
- ขึ้นแบบฟอร์มคำร้องต่างๆ
  - แบบฟอร์มคำร้องทั่วไป
  - แบบฟอร์มคำร้องของ TRANSCRIPT

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แบบฟอร์มคำร้องขอหนังสือรับรอง
- แบบฟอร์มคำร้องขอลาพักการศึกษา
- แบบฟอร์มคำร้องขอลาออก
- แบบฟอร์มคำร้องขอปรับสภาพเป็นนักศึกษาสามัญ
- แบบฟอร์มคำร้องขอขยายเวลาการศึกษา
- แบบฟอร์มคำร้องขอลงทะเบียนวิชา (Audits)
- แบบฟอร์มคำร้องขอถอนวิชาเรียน (เกินกำหนด)
- แบบฟอร์มคำร้องขอถอนเงินประกันของเสียหาย

ทั้งหมดนี้เป็นแบบฟอร์มที่ยังไม่ได้ทำในระบบที่ได้พัฒนามาแล้ว

● ตัวอย่างขั้นตอนการลงทะเบียน เป็นส่วนที่ต้องทำเพิ่ม เพื่อให้นักศึกษาสามารถดูขั้นตอนในการลงทะเบียนได้

● การตรวจสอบเงื่อนไขสำหรับการลงทะเบียนล่วงหน้า ส่วนที่ต้องทำเพิ่ม คือ ตรวจสอบเวลาเรียน และเวลาสอบว่ามีการชนกันหรือไม่ในวิชาที่นักศึกษาต้องการลงทะเบียน

● การตรวจสอบเงื่อนไขสำหรับการลงทะเบียนปกติ ส่วนที่ต้องทำเพิ่ม คือ การตรวจสอบกรณี นักศึกษาได้รับการยกเว้นต่าง ๆ เช่น ได้รับการยกเว้นค่าน่วยกิต ยกเว้นเงินสนับสนุน ยกเว้นทั้งหมด ยกเว้น หน่วยกิต/เงินสนับสนุน

● การตรวจสอบเงื่อนไขกรณีสถานะของนักศึกษา ในระบบเดิมไม่ได้พิจารณา กล่าวคือสถานะ นักศึกษาเก่าและนักศึกษาใหม่ เพื่อใช้ในการคำนวณค่าธรรมเนียมซึ่งแต่ละสถานะจะมีค่าใช้จ่ายไม่เท่ากัน

● การตรวจสอบเงื่อนไขค่าใช้จ่ายการจ่ายเงินอุดหนุนสมทบ กล่าวคือ ปริญญาโท ตามหลักสูตร ระดับมหาบัณฑิตคือ 2 ปี ต้องจ่ายเป็นเวลา 4 ภาคการศึกษา และปริญญาเอก หลักสูตรระดับดุษฎีบัณฑิตต้อง จ่ายเป็นเวลา 6 ภาคการศึกษา แต่ระบบที่พัฒนามาแล้วนั้นไม่ได้พิจารณาในส่วนเงื่อนไขนี้เลย ทำให้การ คำนวณค่าใช้จ่ายการลงทะเบียนผิด เช่น กรณีที่นักศึกษาใช้ช่วงเวลาการเรียนเกินตามหลักสูตร ระบบที่พัฒนา มาแล้วนั้นให้นักศึกษาจ่ายทุกภาคการศึกษาที่เรียน ซึ่งตามระเบียบแล้วต้องจ่ายเพียงตามหลักสูตรที่เรียนเท่านั้น

● ประเภทกลุ่มการเรียนมีเพิ่ม คือ ภาคพิเศษ(เสาร์-อาทิตย์), ภาคพิเศษ(นอกเวลาราชการ) และ หลักสูตรนานาชาติ ซึ่งตามระบบเดิมนั้นมีเพียง ประเภทกลุ่มการเรียน ภาคปกติและภาคสมทบเท่านั้น

#### 5.4.2 ส่วนการทำงานของอาจารย์ที่ติดต่อกับระบบ

เป็นส่วนที่ต้องเพิ่มเติมเนื่องจากระบบที่พัฒนามายังไม่การทำงานในส่วนนี้ คือ

● การตรวจสอบการลงทะเบียนของนักศึกษา เพื่อให้อาจารย์ที่ปรึกษาสามารถตรวจสอบการ ลงทะเบียนของนักศึกษาผ่านทางเว็บแอปพลิเคชัน และให้อาจารย์สามารถเสนอแนะวิชาที่ควรจะลงหรือไม่ ลงทะเบียน เพื่อให้นักศึกษาเป็นแนวทางในการลงทะเบียนปกติ

● การส่งเกรด เพื่อเพิ่มความสะดวกให้แก่อาจารย์และเจ้าหน้าที่ในการกรอกข้อมูลเกรดของนักศึกษา ซึ่งจากเดิมอาจารย์จะต้องรับแบบฟอร์มเพื่อกรอกเกรดของนักศึกษาแต่ละคน แล้วส่งให้เจ้าหน้าที่บัณฑิต วิทยาลัยในการเก็บลงฐานข้อมูล ซึ่งเป็นการทำงานที่ซ้ำซ้อนและมีข้อผิดพลาดบ่อยครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.4.3 ส่วนอื่น ๆ

- การจัดการใบทรานสคริปต์ ในระบบที่ได้พัฒนามาแล้วยังคำนวณผิดเนื่องจากคิดเงื่อนไขไม่ครบ คือ ไม่มี non-credit (Nc.) เป็นวิชาเรียนปรับพื้นฐาน เกรดที่ได้จะต้องไม่ต่ำกว่า C+ ถ้านักศึกษาได้คะแนนต่ำกว่า C+ นักศึกษาต้องลงทะเบียนใหม่ โดยรายวิชาปรับพื้นฐานนี้ ไม่นำไปคิดค่าระดับคะแนนเฉลี่ยสะสม และไม่นับจำนวนหน่วยกิตรวม

- ข้อมูลนักศึกษาในส่วนข้อมูลวิทยานิพนธ์ เนื่องจากไม่มีการเก็บรายละเอียดเกี่ยวกับวิทยานิพนธ์ที่นักศึกษาทำ

- ประเภทกลุ่มการเรียน มีส่วนเพิ่มเติม คือ กลุ่มการเรียน ภาคพิเศษ (เสาร์-อาทิตย์), ภาคพิเศษ (นอกเวลาราชการ)

- การจัดการเซคชั่น ในส่วนนี้ระบบเดิมที่พัฒนามาแล้วนั้นไม่ได้มีส่วนการทำงานนี้

- การจัดการใบทรานสคริปต์ สำหรับในระบบที่พัฒนาขึ้นมาั้นยังเป็นเพียงแค่การแสดงผลออกมาเป็น HTML ยังไม่สามารถพิมพ์รายงานออกมาได้

### 5.5 การวิเคราะห์ระบบเพื่อปรับปรุง

ในการวิเคราะห์ระบบนั้น นอกจากเข้าไปศึกษาระบบเดิมและระบบที่พัฒนามาแล้วนั้น การหาถึงความต้องการของผู้ใช้ระบบนับว่าเป็นสิ่งสำคัญอีกประการหนึ่งในการพัฒนาระบบใด ๆ ขึ้นมา ทางผู้พัฒนาจึงได้ไปสัมภาษณ์เจ้าหน้าที่ทะเบียนบัณฑิตวิทยาลัยเพื่อหาความต้องการเพิ่มเติมเพื่อปรับปรุงระบบให้มีความสมบูรณ์ขึ้น ทำให้ทราบว่าเงื่อนไขต่างๆ มีการเปลี่ยนแปลงไปและทำให้พบความต้องการของผู้ใช้เพิ่มขึ้น แล้วนำข้อมูลที่ได้มาทำความเข้าใจและวิเคราะห์ให้ได้ทั้งกระบวนการทำงานเดิมและแนวที่จะปรับปรุงให้ได้ตามความต้องการของผู้ใช้ นอกจากนี้ผู้พัฒนาโครงการยังได้พยายามเลือกใช้เทคโนโลยีที่เหมาะสมและไม่ต้องลงทุนเพิ่มเติมมากนัก

หลังจากที่ได้ทำการวิเคราะห์ระบบอย่างละเอียด ทำให้ทราบถึงปัญหาของระบบทั้งระบบเดิมและระบบที่ได้พัฒนามาแล้วดังที่กล่าวมาแล้วข้างต้น และนอกจากนี้ยังได้ทราบความต้องการของผู้ใช้เพิ่มเติม ซึ่งพอที่จะสรุปเป็นแนวทางในการปรับปรุงและออกแบบพัฒนาระบบต่อไปดังนี้

- ให้นักศึกษาสามารถทำกิจกรรมและเข้าถึงข้อมูลต่าง ๆ ที่เกี่ยวกับงานทะเบียนได้ โดยที่ไม่ต้องมาติดต่อฝ่ายทะเบียน หรือมาติดต่อให้น้อยที่สุด เพื่อเพิ่มความสะดวกให้นักศึกษาเอง รวมถึงลดภาระงานของเจ้าหน้าที่ นั่นก็คือให้นักศึกษาสามารถทำกิจกรรมเหล่านั้นผ่านทางอินเทอร์เน็ต
- ลดภาระการตรวจสอบเงื่อนไขต่าง ๆ ด้วยการสร้างให้โปรแกรมสามารถตรวจสอบเงื่อนไขเหล่านั้นให้ครอบคลุมและถูกต้องมากที่สุดที่เพิ่มเติมขึ้นมา เช่น การตรวจสอบเงื่อนไขค่าใช้จ่าย การจ่ายเงินอุดหนุนสมทบ, การตรวจสอบเงื่อนไขเวลาเรียนและเวลาสอบว่ามีกรณีชนกันหรือไม่ ในวิชาที่นักศึกษาต้องการลงทะเบียน จึงสามารถทำการลงทะเบียนรายวิชาได้ ซึ่งเราต้องทำการศึกษาเงื่อนไขเหล่านั้นให้ชัดเจนและออกแบบเรื่องของโปรแกรมและฐานข้อมูลให้สามารถสนับสนุนการตรวจสอบดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การเพิ่ม แก้ไขข้อมูล ต่าง ๆ นั้น เจ้าหน้าที่ที่จะต้องสามารถทำได้อย่างครบถ้วนและสะดวก เพราะข้อมูลที่เกี่ยวกับงานทะเบียนนั้นค่อนข้างซับซ้อน ยกตัวอย่างเช่นข้อมูลของหลักสูตรซึ่งมีข้อมูลย่อย ๆ ประกอบอยู่หลายอย่าง ระบบจึงควรเก็บข้อมูลให้ครอบคลุมให้ได้มากที่สุด
- ในส่วนของการลงทะเบียนควรจะให้มีการลงทะเบียนล่วงหน้าก่อน เพื่อจะเป็นการโหวตจำนวนผู้ที่ลงทะเบียนในรายวิชานั้นๆ เพื่อที่จะได้ทำการเปิดสอนและให้ลงทะเบียนจริงเฉพาะวิชาที่มีผู้ลงทะเบียนมากพอที่จะเปิดสอนได้
- ระบบควรมีความเป็นอัตโนมัติในการแสดงข้อมูลต่าง ๆ ให้กับผู้ใช้ยกตัวอย่างเช่นในการตรวจสอบช่วงเวลาการลงทะเบียน ระบบควรสามารถตรวจสอบได้เอง ไม่ต้องให้เจ้าหน้าที่ทำการตั้งค่าเอง

### 5.6 การออกแบบระบบโดยใช้ยูเอ็มแอล

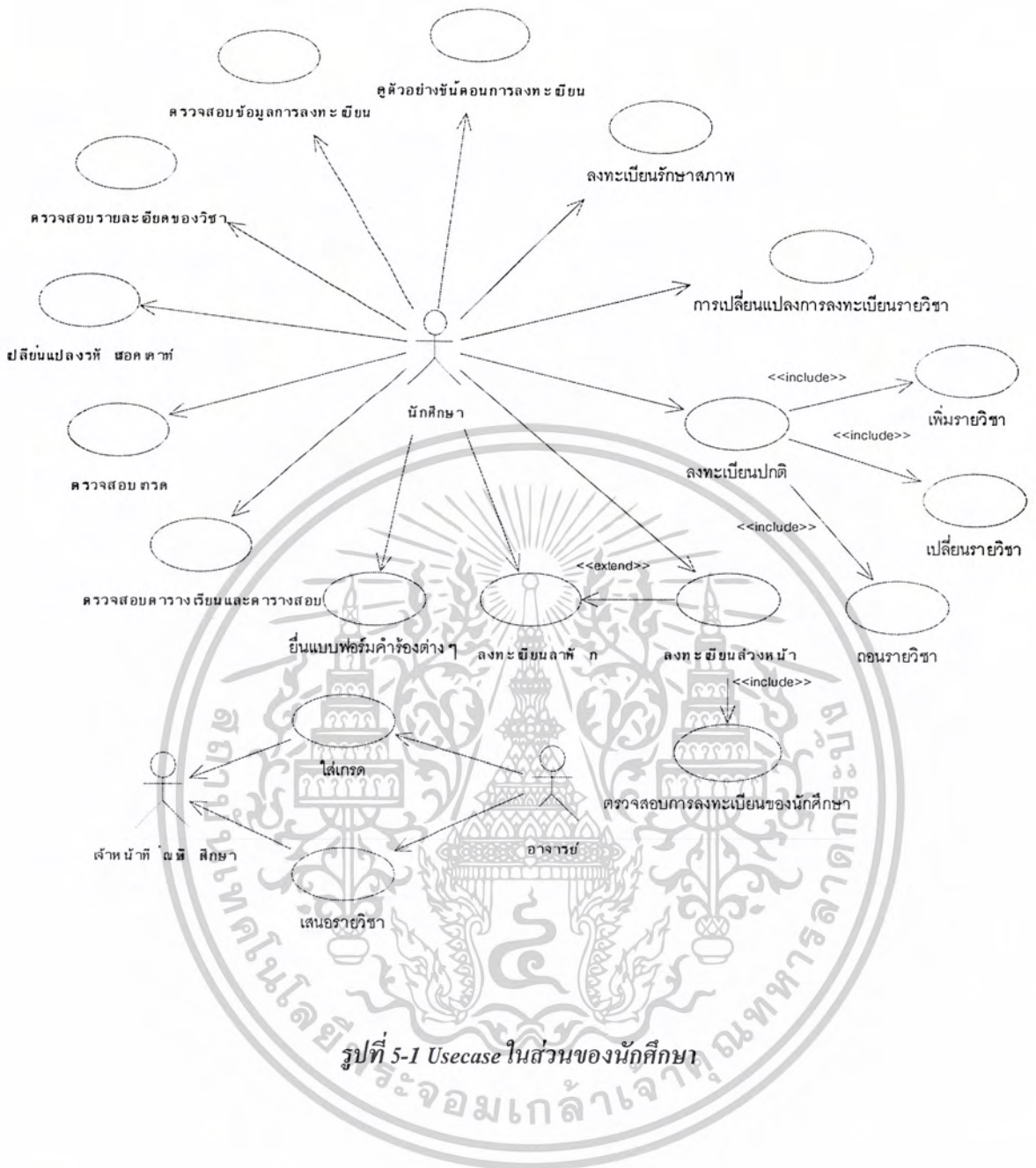
หลังจากที่เราได้วิเคราะห์หาความต้องการของผู้ใช้และศึกษาระบบเดิมพอสมควรแล้วเราต้องนำข้อมูลที่ได้มาทำการวิเคราะห์และปรับปรุงระบบโดยใช้ยูเอ็มแอล แต่ถึงอย่างไรก็ตามในการออกแบบระบบนี้เราใช้กระบวนการทางวิศวกรรมซอฟต์แวร์แบบวนรอบ (Iteration Software Process) ซึ่งเป็นกระบวนการที่ทำงานเป็นวงรอบ ในแต่ละรอบก็จะมีการปรับปรุงในแต่ละขั้นจนจนสุด ดังนั้นขั้นตอนที่อธิบายต่อไปนี้จะมีการปรับปรุงอยู่ตลอดเวลา และในส่วนนี้จะกล่าวถึงไดอะแกรมต่าง ๆ จึงควรศึกษาในบทที่ 2 ประกอบด้วย

การหาว่ามีใครบ้างที่ต้องใช้ระบบนี้ซึ่งเราเรียกว่าแอกเตอร์ (Actor) และหาว่าแต่ละคนทำอะไรกับระบบบ้างหรือใช้ฟังก์ชันใดของระบบบ้าง เมื่อได้แอกเตอร์และฟังก์ชันการทำงานแล้วนำมาเขียนเป็นยูสเคสไดอะแกรม (Use Case Diagram) ซึ่งเราถือว่าแต่ละฟังก์ชันการทำงานของระบบคือหนึ่งยูสเคส (Use case)

จากรูปที่ 5-1 และ 5-2 นั่นคือ ยูสเคสไดอะแกรมในส่วนของนักศึกษา และเจ้าหน้าที่ ซึ่งจะเห็นได้ว่าประกอบด้วย 5 แอกเตอร์ซึ่งมีรายละเอียดของแต่ละแอกเตอร์ดังนี้

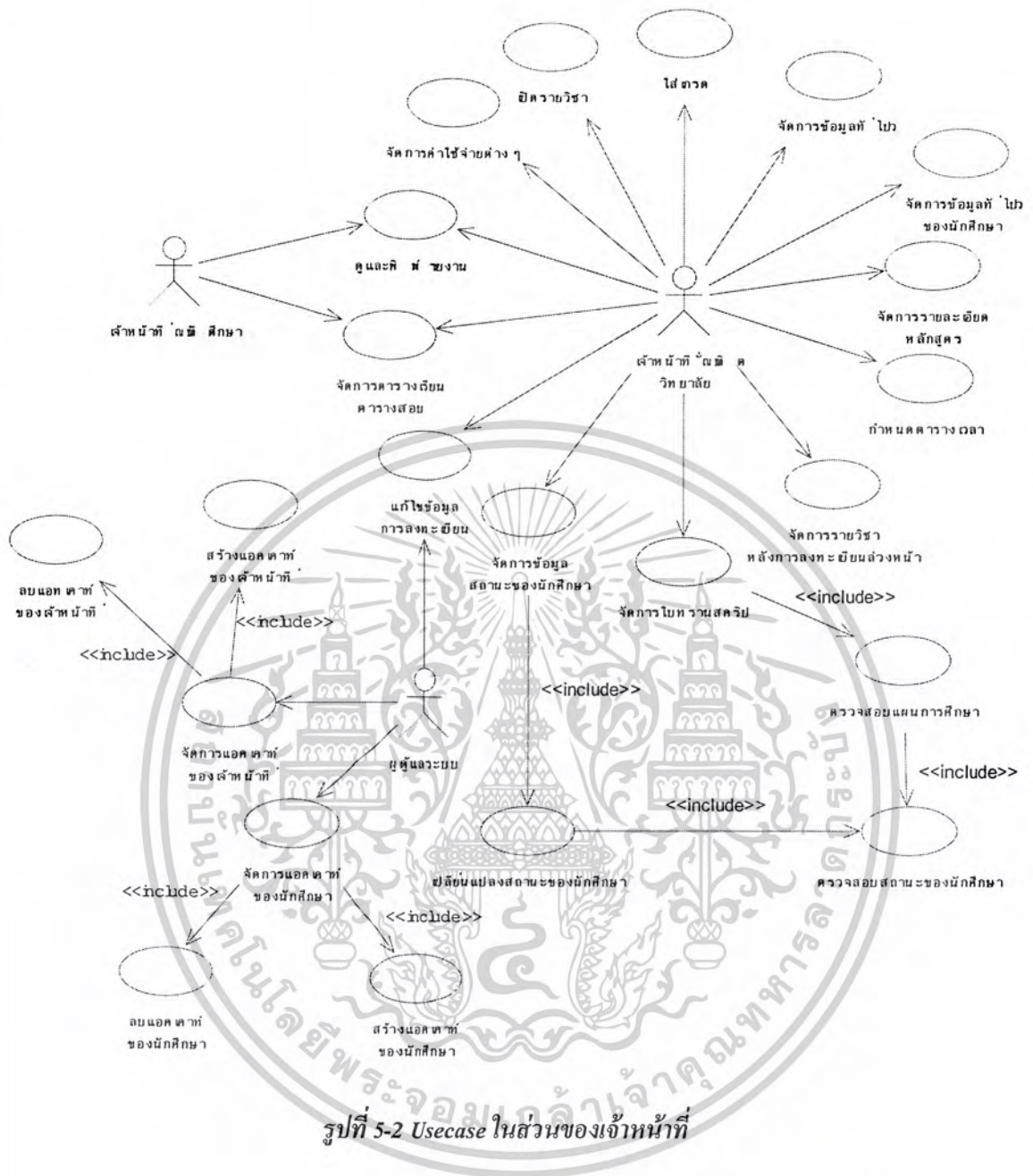
- นักศึกษา ก็คือ ผู้ใช้ระบบ ซึ่ง ได้แก่ นักศึกษาในระดับบัณฑิตศึกษา ซึ่งสามารถที่จะใช้งานระบบผ่านทางอินเทอร์เน็ต
- เจ้าหน้าที่บัณฑิตวิทยาลัย ก็คือ เจ้าหน้าที่ทะเบียนของบัณฑิตวิทยาลัย ซึ่งจะเป็นผู้ที่ดูแล เพิ่ม แก้ไขข้อมูลของระบบ การทำงานกับระบบนั้นจะทำผ่านทางโปรแกรมที่จะพัฒนาขึ้น
- เจ้าหน้าที่บัณฑิตศึกษา ก็คือ เจ้าหน้าที่งานบัณฑิตศึกษาในส่วนคณะต่าง ๆ ก็จะสามารถในการเพิ่ม แก้ไข ข้อมูลบางอย่างของระบบ เพื่อเป็นการแบ่งภาระงานของเจ้าหน้าที่ทะเบียนของบัณฑิตวิทยาลัย รวมถึงการตัดตอนกระบวนการทำงาน ทำให้มีโอกาสเกิดข้อผิดพลาดน้อยลง
- อาจารย์ ก็คือ อาจารย์ซึ่ง อาจจะไม่ได้ใช้งานระบบโดยตรง แต่ก็มีส่วนเกี่ยวข้องกับระบบในงานบางงานเช่นการกรอกเกรดของนักศึกษา ตรวจสอบการลงทะเบียนของนักศึกษา เป็นต้น
- ผู้ดูแลระบบ คือผู้ดูแลระบบซึ่งจะเป็นผู้ที่จัดการเกี่ยวกับระบบโดยตรงกับฐานข้อมูลทำให้สามารถจัดการกับข้อมูลได้ทั้งหมด รวมถึงเป็นผู้ที่ดูแลเกี่ยวกับความปลอดภัยของระบบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-1 Usecase ในส่วนของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-2 Usecase ในส่วนของเจ้าหน้าที่

จากคำอธิบายข้างต้น ฟังก์ชันการทำงานต่าง ๆ ของระบบที่แอดเดอรสามารถใช้ได้นั้นก็จะประกอบไปด้วย 3 ส่วนนั้นคือ

- ส่วนที่เป็นแอปพลิเคชันซึ่งเป็นโปรแกรมที่พัฒนาด้วยภาษาจาวาเพื่อให้เจ้าหน้าที่ทะเบียนทั้งส่วนกลางและส่วนคณะได้ใช้
- ส่วนที่เป็นเว็บแอปพลิเคชัน ซึ่งเป็นส่วนที่ให้นักศึกษาใช้ในการติดต่อกับระบบ
- ส่วนที่เป็นเอกสาร หรือ โปรแกรมพื้นฐานที่มีอยู่แล้ว ซึ่งไม่ต้องทำการพัฒนาอย่างเช่นระบบงานในส่วนของผู้ดูแลระบบที่ใช้โปรแกรมจัดการฐานข้อมูล เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากในส่วนของแอดเดอเรอร์แล้ว จะขออธิบายถึงฟังก์ชันการทำงานต่าง ๆ ของระบบนั้นก็คือยูสเคส แต่ละยูสเคสนั้นเอง มีรายละเอียดดังนี้

- ตรวจสอบตารางเรียนและตารางสอบ : เป็นส่วนที่ให้นักศึกษาจะเข้าไปดูตารางเรียนและตารางสอบ
- ตรวจสอบรายละเอียดของรายวิชา : เป็นส่วนที่ให้นักศึกษาดูรายละเอียดของวิชาที่เปิดสอน และข้อมูลต่าง ๆ ของรายวิชานั้น ๆ
- ลงทะเบียนล่วงหน้า : เป็นส่วนที่ให้นักศึกษาทำการลงทะเบียนล่วงหน้า
- ลงทะเบียนปกติ : เป็นส่วนที่ให้นักศึกษาลงทะเบียนจริง
- จัดการรายวิชา : เป็นส่วนที่ให้นักศึกษาเพิ่ม เปลี่ยน ถอน รายวิชา
- ตรวจสอบข้อมูลการลงทะเบียน : เป็นส่วนที่ให้นักศึกษาดูผลการลงทะเบียน
- ตรวจสอบเกรด : เป็นส่วนที่ให้นักศึกษาดูผลการศึกษา
- ขึ้นแบบฟอร์มคำร้องต่างๆ : เป็นส่วนที่ให้นักศึกษาสามารถกรอกคำร้องและพิมพ์ออกมา เพื่อนำไปดำเนินการต่อไป ซึ่งนักศึกษาไม่จำเป็นต้องกรอกข้อมูลทั้งหมด เพราะข้อมูลบางอย่างจะถูกอ่านขึ้นมาจากฐานข้อมูลให้
- ใส่เกรด : เป็นส่วนที่ให้อาจารย์และเจ้าหน้าที่กรอกเกรดเข้าสู่ระบบ
- เปลี่ยนแปลงข้อมูลทั่วไปของนักศึกษา : เป็นส่วนที่ให้เจ้าหน้าที่เพิ่มแก้ไขข้อมูลนักศึกษา
- จัดการรายละเอียดหลักสูตร : เป็นส่วนที่ให้เจ้าหน้าที่เพิ่มแก้ไขข้อมูลหลักสูตร
- จัดการรายละเอียดองค์กร : เป็นส่วนที่ให้เจ้าหน้าที่เพิ่มแก้ไขข้อมูลหน่วยงานการศึกษา
- จัดการตารางเรียนและตารางสอบ : เป็นส่วนที่ให้เจ้าหน้าที่เพิ่มแก้ไขข้อมูลตารางเรียนและตารางสอบ
- เปิดรายวิชา : เป็นส่วนที่ให้เจ้าหน้าที่ เพิ่ม แก้ไข ข้อมูลการเปิดปิดรายวิชา
- กำหนดตารางเวลา : เป็นส่วนที่ให้เจ้าหน้าที่ เพิ่ม แก้ไข ข้อมูลกำหนดระยะเวลาต่าง ๆ เช่น ระยะเวลาการลงทะเบียน
- จัดการรายวิชาหลังการลงทะเบียนล่วงหน้า : เป็นส่วนที่ให้เจ้าหน้าที่สั่งให้ระบบประเมินว่ามีรายวิชาใดที่มีผู้ลงทะเบียนน้อยกว่าการลงทะเบียนล่วงหน้า และให้สิทธิเจ้าหน้าที่ในการปิดรายวิชานั้น
- จัดการค่าใช้จ่ายต่างๆ : เป็นส่วนให้เจ้าหน้าที่ จัดการค่าใช้จ่ายต่างๆ เช่น ค่าธรรมเนียมการศึกษา
- จัดการใบทรานสคริป : เป็นส่วนให้เจ้าหน้าที่ ทำการ พิมพ์ใบทรานสคริปให้แก่นักศึกษา
- ดูและพิมพ์รายงาน : เป็นส่วนที่ให้เจ้าหน้าที่ดู และพิมพ์รายงานต่าง ๆ จากระบบ
- แก้ไขข้อมูลการลงทะเบียน : เป็นส่วนที่ให้เจ้าหน้าที่และผู้ดูแลระบบแก้ไขข้อมูลการลงทะเบียนที่อาจเกิดความผิดพลาดขึ้นได้
- จัดการแอดเดอเรอร์ของนักศึกษา : ให้ผู้ดูแลระบบสร้างหรือลบ รหัสผ่านให้กับนักศึกษา
- จัดการแอดเดอเรอร์ของเจ้าหน้าที่ : ให้ผู้ดูแลระบบสร้างหรือลบ รหัสผ่านให้กับเจ้าหน้าที่ทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนักศึกษาเห็นว่าไม่เหมาะสมในการนำ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.7 การออกแบบการทำงานของฟังก์ชันต่าง ๆ

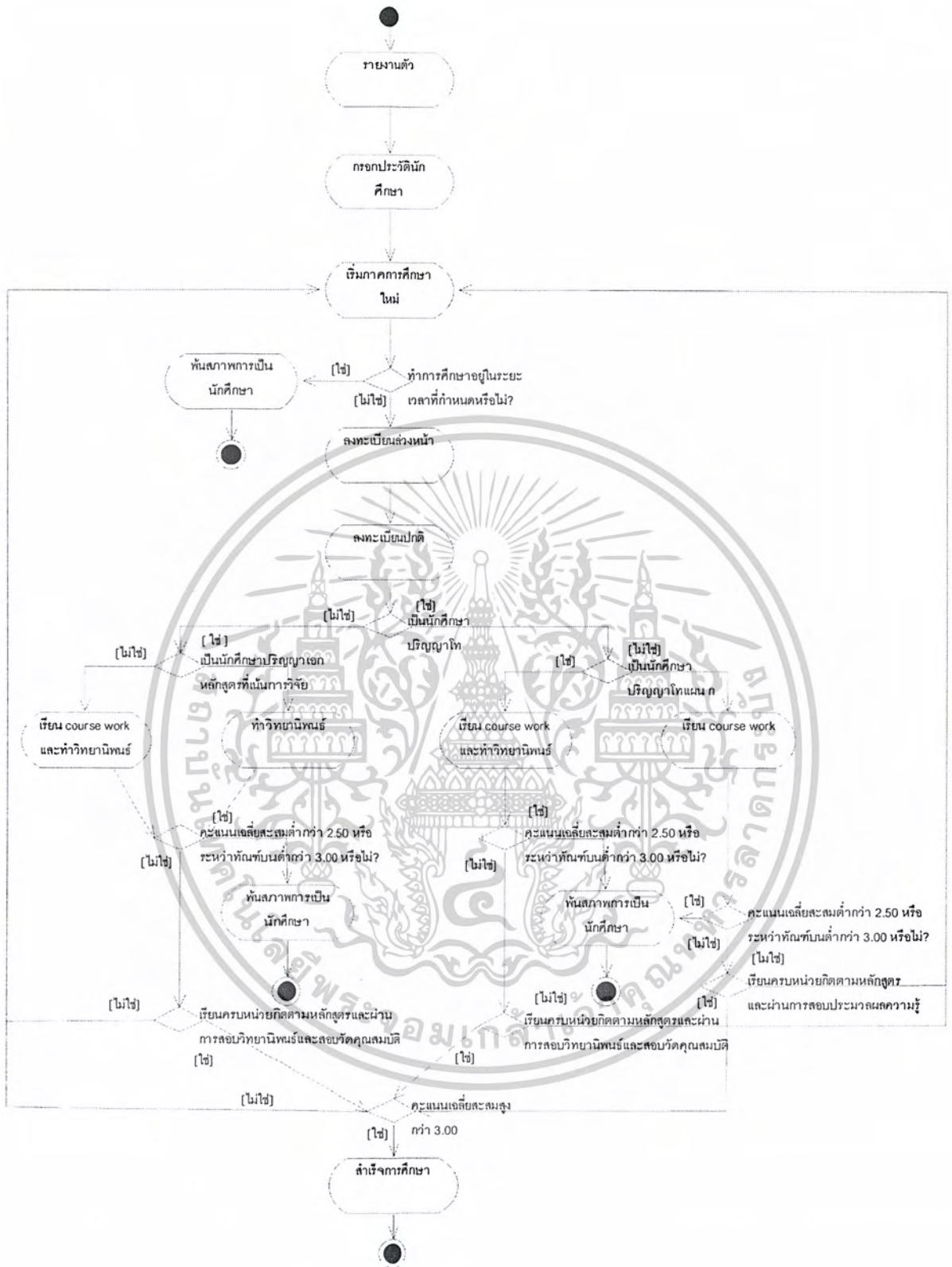
จะเห็นได้ว่าหลังจากขั้นตอนที่ผ่านมาระบบจะได้ฟังก์ชันการทำงานต่าง ๆ ของระบบทั้งหมด ซึ่งเราสามารถทำการแยกออกแบบทีละฟังก์ชัน ซึ่งทำให้เราสามารถออกแบบได้ง่ายขึ้น สิ่งแรกที่เราจะต้องทำก็คือการอธิบายว่าในแต่ละฟังก์ชันมีกิจกรรมใดเกิดขึ้นบ้าง ซึ่งเราใช้แอกทิวิตีไดอะแกรมในการอธิบายสิ่งดังกล่าว

จากรูปที่ 5-3 จะเป็นแอกทิวิตีไดอะแกรมของระบบการลงทะเบียน ซึ่งจะเริ่มตั้งแต่เมื่อนักศึกษารายงานตัว เจ้าหน้าที่จะทำการกรอกประวัติของนักศึกษา เมื่อถึงภาคการศึกษาใหม่ระบบจะทำการตรวจสอบว่านักศึกษานั้นได้พ้นสภาพนักศึกษาหรือไม่ นักศึกษาที่พ้นสภาพนักศึกษาจะไม่สามารถลงทะเบียนได้นอกจากนั้นจะมีการตรวจสอบกรณีที่นักศึกษามีคะแนนเฉลี่ยสะสมน้อยกว่า 2.50 หรือ ระหว่างพักตัวต่ำกว่า 3.00 ซึ่งถ้าคะแนนเฉลี่ยของนักศึกษาไม่ผ่านเกณฑ์นี้ นักศึกษาจะพ้นสภาพการเป็นนักศึกษา นักศึกษาจะสำเร็จการศึกษาได้ก็ต่อเมื่อนักศึกษาได้เรียนครบหน่วยกิตตามหลักสูตร ผ่านการสอบวิทยานิพนธ์ในกรณีที่นักศึกษาเรียนหลักสูตรที่เน้นการวิจัย และได้ผ่านการสอบวัดคุณสมบัติหรือสอบประมวลผลความรู้ ตามหลักสูตรที่กำหนด

สิ่งที่ต้องทำต่อไปก็คือการอธิบายว่าในแต่ละฟังก์ชันการทำงานนั้น มีคลาสใดบ้าง และมีการทำงานสื่อสารกันอย่างไร ซึ่งเราใช้ซีควเนต์ไดอะแกรม(Sequence Diagram) เป็นตัวอธิบายสิ่งดังกล่าว ซึ่งแต่ละยูสเคสอาจมีมากกว่าหนึ่งซีควเนต์ไดอะแกรม หรืออาจไม่มีก็ได้

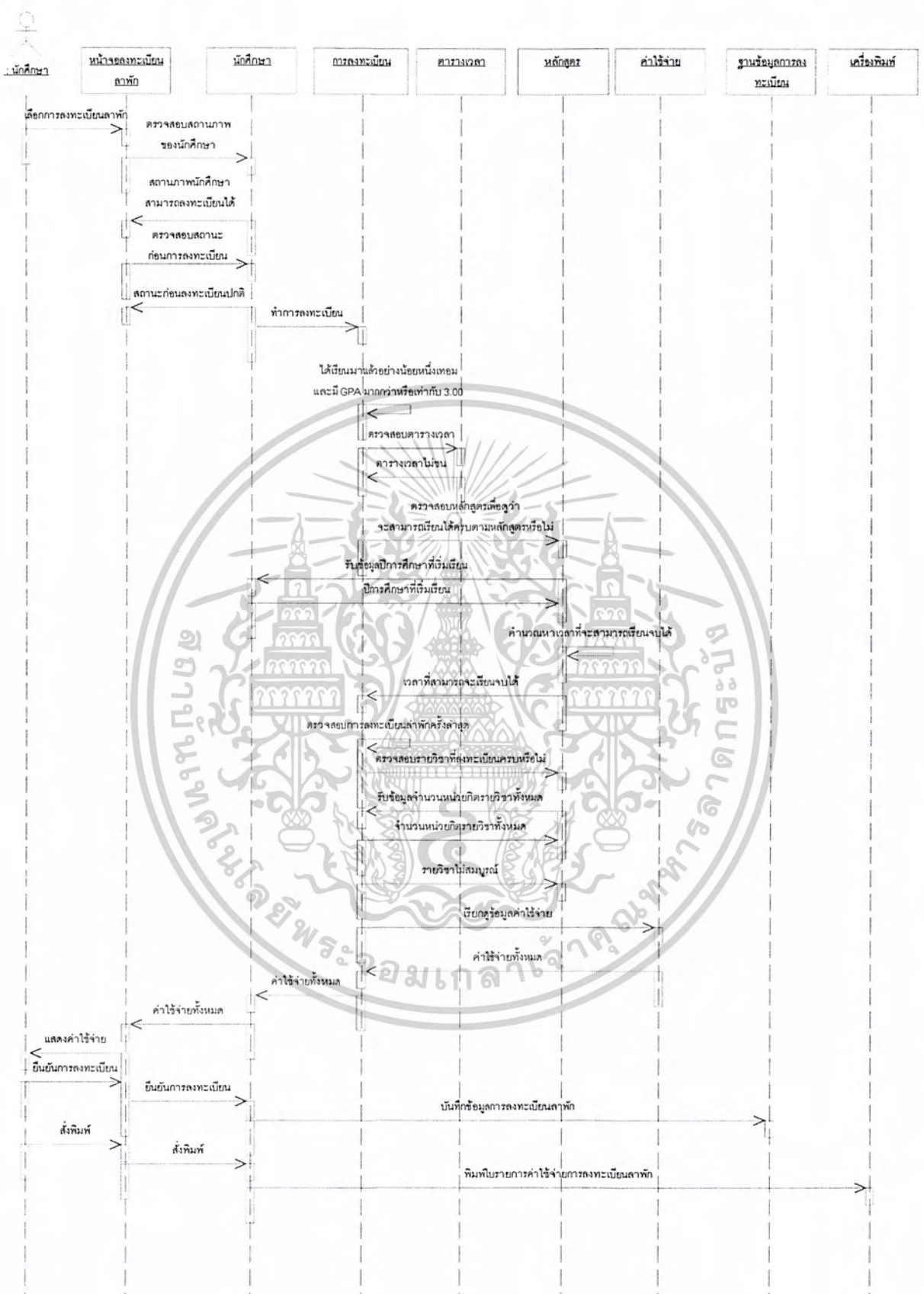
จากรูป 5-4 ก็เป็นตัวอย่างของซีควเนต์ไดอะแกรมของยูสเคสการลงทะเบียนลาพัก ซึ่งจะเห็นได้ว่าการส่งข้อมูลกันระหว่างคลาสต่าง ๆ ทำให้เราสามารถเข้าใจและง่ายในการนำไปเขียนเป็นโปรแกรมต่อไป

ในการการออกแบบระบบในส่วนของการเขียนไดอะแกรมต่าง ๆ เหล่านี้ควรคำนึงถึงการนำไปเขียนเป็นโปรแกรมได้ นั่นหมายความว่าแต่ละคลาสที่ได้จากการเขียนไดอะแกรมก็จะนำไปเขียนเป็นคลาสจริง ๆ ดังนั้นการสกัดเพื่อหาคลาสเหล่านี้จึงต้องทำอย่างรอบคอบและคำนึงถึงขนาดและหน้าที่ของแต่ละคลาสด้วย



รูปที่ 5-3 Activity Diagram ของระบบการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

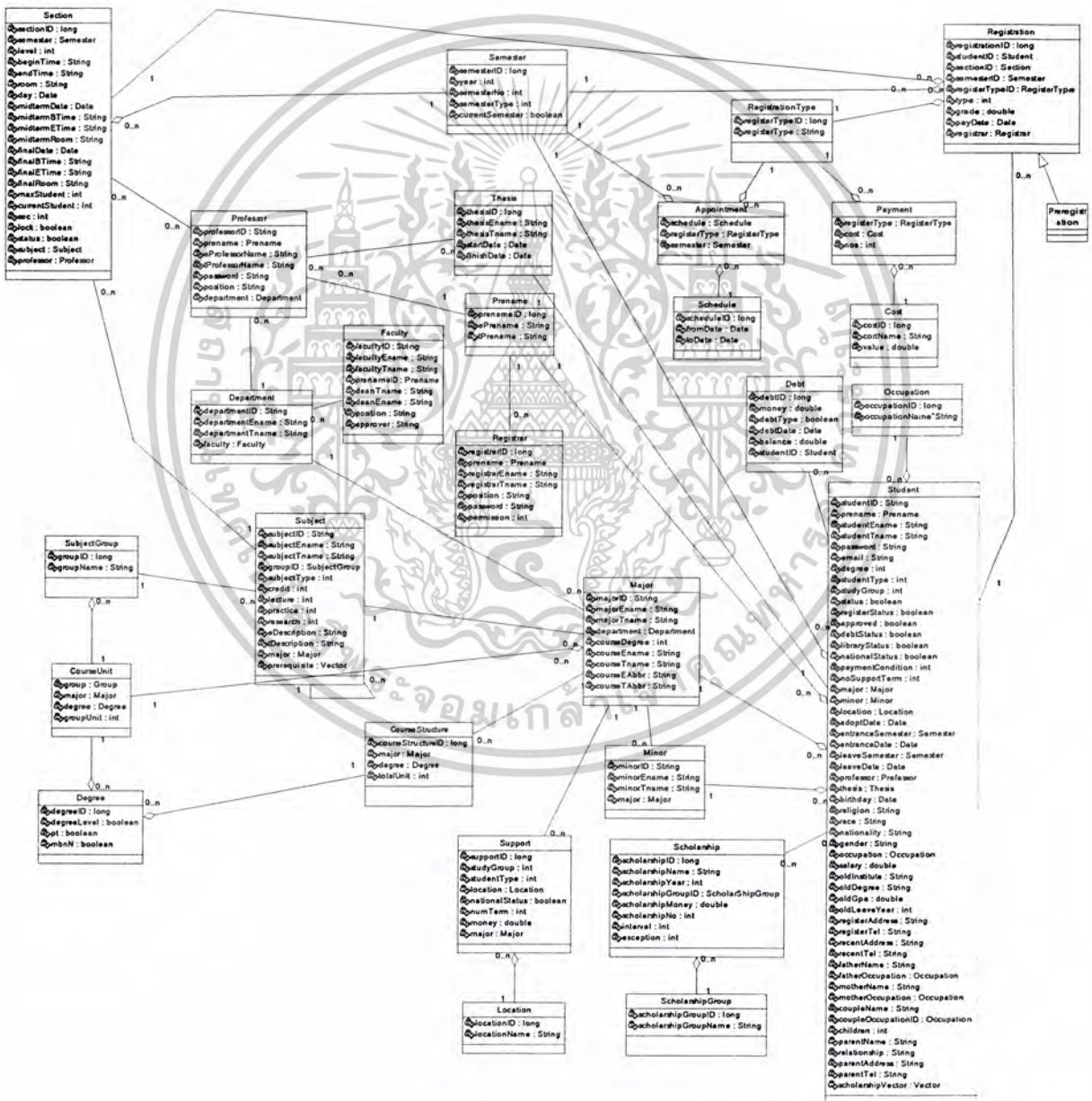


**รูปที่ 5-4 Sequence Diagram ของ ยุทธศาสตร์ ลงทะเบียนลาพัก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.8 การออกแบบรายละเอียดของคลาสต่าง ๆ

มาถึงขั้นตอนนี้เราจะได้คลาสที่ใช้ในระบบอย่างครบถ้วน เราจะนำคลาสเหล่านี้มากำหนดความสัมพันธ์ระหว่างกัน รวมถึงอาจจัดคลาสให้เป็นกลุ่มแล้วเก็บไว้ในแพ็คเกจ(Package) แล้วกำหนดความสัมพันธ์ระหว่างแพ็คเกจด้วยก็ได้ แล้วปรับแต่งซีควเนต์ไดอะแกรมเพื่อให้การส่งข้อมูลระหว่างคลาสอยู่ในรูปของการเรียกใช้เมธอด(Method) ให้ได้มากที่สุด กำหนดคุณสมบัติของคลาสแต่ละคลาส กำหนดแอตทริบิวต์(Attribute) และเมธอดของคลาสให้ครบถ้วน แล้วนำไปเขียนเป็นคลาสไดอะแกรม เขียนความสัมพันธ์ในรูปของการสืบทอดคุณสมบัติของคลาส ดูรายละเอียดที่คลาสของระบบว่ามีคลาสใดที่แยกกันได้ รวมกันได้ หรือกำจัดออกได้ เพื่อลดความซ้ำซ้อนของคลาส จากรูปที่ 5-5 เป็นคลาสไดอะแกรมของระบบ

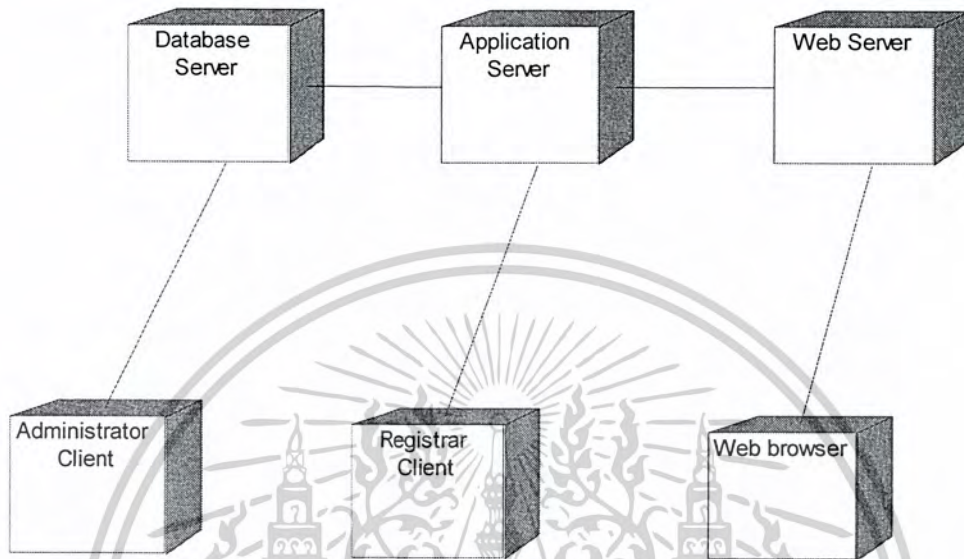


รูปที่ 5-5 Class Diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.9 การออกแบบโครงสร้างของระบบ

เขียนดีพอยเมนต์ไดอะแกรม(Deployment Diagram) ซึ่งจะต้องดูว่าระบบของเราประกอบด้วย โครงสร้างของระบบทางกายภาพอย่างไร



รูปที่ 5-6 Deployment Diagram ของระบบ

ที่ได้กล่าวมาทั้งหมดนี้คือขั้นตอนการวิเคราะห์และออกแบบระบบ ซึ่งได้ยกเอาไดอะแกรมบาง ไดอะแกรมมาเพื่ออธิบายขั้นตอนการทำงาน ส่วนไดอะแกรมทั้งหมดในโครงการนี้จะอยู่ในส่วนของ ภาคผนวก ก ท้ายเล่ม

## 5.10 การออกแบบระบบ เพื่อสร้างระบบสารสนเทศ

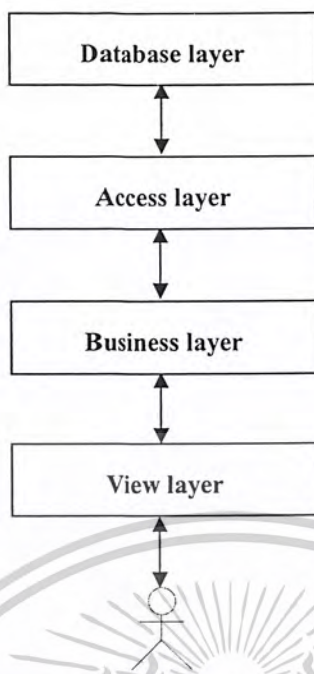
การออกแบบระบบทางผู้พัฒนาต่อได้แบ่ง การทำงาน โดยภาพรวม เป็น 3 เลเยอร์ ด้วยกัน คือ

5.8.1 View layer เป็นส่วนของไคลเอนต์(Client) สำหรับผู้ใช้(User) โดยจะแบ่งเป็น 2 ส่วน ก็คือ ส่วนที่เป็นแอปพลิเคชัน (Application) ให้สำหรับเจ้าหน้าที่ทะเบียน และส่วนที่เป็นเว็บแอปพลิเคชัน (Web Application) สำหรับนักศึกษา

5.8.2 Business layer ทำหน้าที่กำหนดหน้าที่การทำงานหลัก ของระบบ เป็นการทำงาน เกี่ยวกับระบบทั้งหมด ซึ่งทำงานในลักษณะ เว็บเซิร์ฟเวอร์(Web Server)และ โปรแกรมประยุกต์

5.8.3 Access layer ทำหน้าที่ เป็นส่วนที่ใช้สำหรับติดต่อกับระบบจัดการฐานข้อมูล(DBMS Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-7 การแบ่งการทำงานของระบบสารสนเทศ

### 5.11 การออกแบบฐานข้อมูล

ในการออกแบบระบบฐานข้อมูล ต้องทำการศึกษาคุณสมบัติและความสัมพันธ์ระหว่างข้อมูลที่มีอยู่ในระบบ เพื่อให้ได้มาซึ่งโครงสร้างพื้นฐานของฐานข้อมูล สำหรับโครงการนี้การวิเคราะห์หาความสัมพันธ์ของข้อมูลจะมีความยุ่งยาก เนื่องจากมีการขึ้นต่อกันระหว่างแอตทริบิวต์เหล่านี้เป็นจำนวนมากด้วย ดังนั้นในทางปฏิบัติการวิเคราะห์ระบบจึงกระทำในลักษณะของภาพรวม ด้วยการมองจากระดับบน (top-level) นั่นคือการใช้ อี-อาร์ไดอะแกรม ซึ่งเป็นการแสดงภาพรวมของระบบในลักษณะของรูปภาพทำให้ง่ายต่อความเข้าใจ และเป็นสื่อกลางในการติดต่อสื่อสารระหว่างกลุ่มบุคคลที่เกี่ยวข้องทั้งหมดทำให้เข้าใจลักษณะของข้อมูลและความสัมพันธ์ระหว่างข้อมูลให้เป็นไปอย่างถูกต้องสมบูรณ์ตรงกัน

จากรูปที่ 5-8 เป็นอี-อาร์ไดอะแกรมของระบบ ซึ่งจากอี-อาร์ไดอะแกรมของระบบนี้สามารถแปลงให้เป็นตารางในฐานข้อมูลได้ โดยคลาสหนึ่งๆ ในคลาสไดอะแกรมจะมีลักษณะเหมือนกับเอนทิตีในอี-อาร์ไดอะแกรม ทำให้สามารถที่จะแปลงจากคลาสไดอะแกรมให้กลายเป็นอี-อาร์ไดอะแกรม ซึ่งจากอี-อาร์ไดอะแกรมของระบบก็จะสามารถที่จะนำไปทำนอร์มอลไลซ์เซชัน (Normalization) ตามกระบวนการในโมเดลของฐานข้อมูลแบบสัมพันธ์และจะได้ตารางออกมา



## บทที่ 6

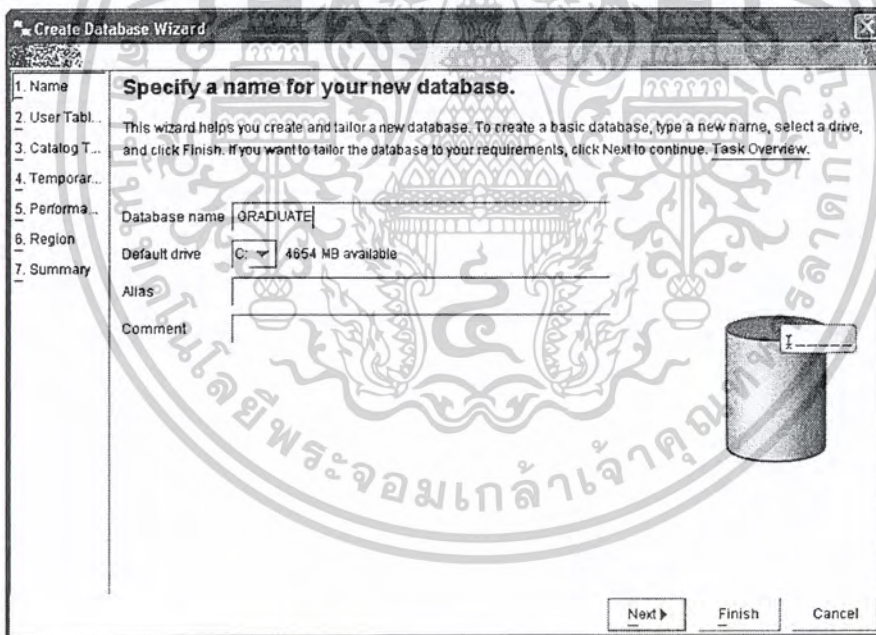
# การเขียนโปรแกรมเพื่อสร้างระบบ

ในส่วนของบทนี้จะได้กล่าวถึงขั้นตอนการเขียนโปรแกรมเพื่อสร้างระบบ ซึ่งเราจะเริ่มตั้งแต่การสร้างในส่วนของฐานข้อมูลขึ้นมาก่อน แล้วจึงทำการเขียนโปรแกรมซึ่งก็แบ่งออกเป็น 2 ส่วนก็คือส่วนที่เป็นแอปพลิเคชัน ให้สำหรับเจ้าหน้าที่ทะเบียน และส่วนที่เป็นเว็บแอปพลิเคชันสำหรับนักศึกษา ซึ่งมีรายละเอียดดังนี้

### 6.1 การสร้างฐานข้อมูล

ในการสร้างฐานข้อมูลนั้นเราอ้างอิงจากการนำเอาอี-อาร์ไอโอะแอมไปทำนอมนอลไลซ์เซชัน (Normalization) ตามกระบวนการในโมเดลของฐานข้อมูลแบบสัมพันธ์และจะได้ตารางออกมาสร้างเป็นโครงสร้างตารางฐานข้อมูล และสร้างเป็นตัวฐานข้อมูลใน DB2 ซึ่งมีขั้นตอนดังนี้

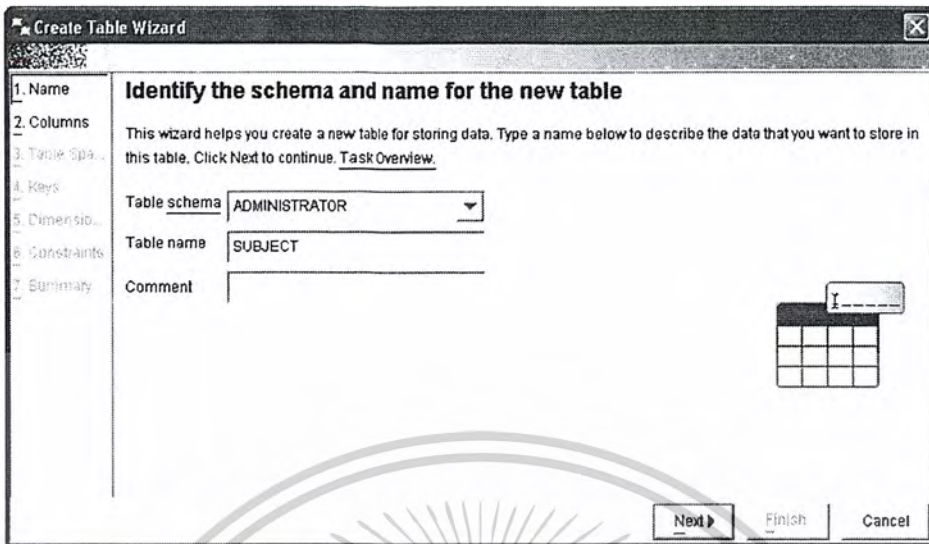
1. ทำการสร้างฐานข้อมูลบน DB2 ขึ้นโดยใช้ โปรแกรม Control Center



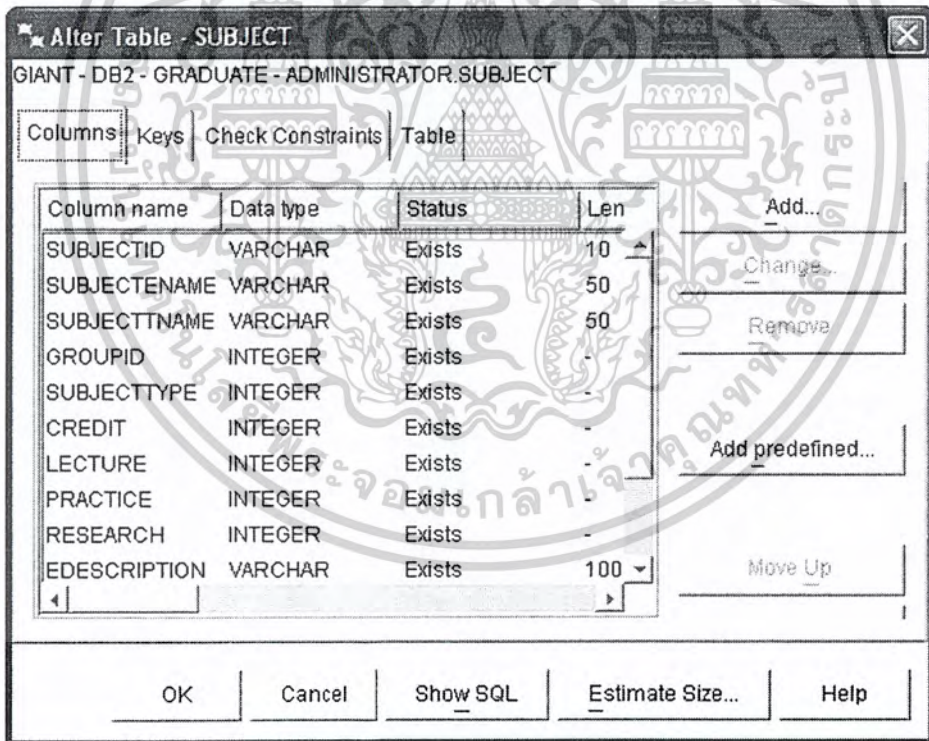
รูปที่ 6-1 การสร้างฐานข้อมูลบน DB2 โดยใช้โปรแกรม Control Center

2. ทำการสร้างตารางแต่ละตาราง บนฐานข้อมูลที่เราสร้างขึ้นโดยใช้โปรแกรม Control Center
3. ทำการเพิ่มข้อมูลแต่ละฟิลด์เข้าไป ซึ่งสามารถที่จะกำหนดคัลักษณะได้ว่าจะให้เป็นข้อมูลใด
4. ในการติดต่อกับฐานข้อมูลต่าง ๆ เราก็คจะใช้โปรแกรม Command Center ซึ่งสามารถทำงานด้วยภาษาเอสคิวแอล(SQL)ได้ และ โปรแกรม Control Center ซึ่งจะเป็นตัวจัดการในการบริหารฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-2 การใช้ Control Center เพื่อการสร้างตาราง



รูปที่ 6-3 การใช้ Control Center เพื่อเพิ่มหรือเปลี่ยนแปลงข้อมูลในตาราง

หลังจากนี้เราก็จะมีฐานข้อมูลอยู่บนเครื่องเซิร์ฟเวอร์ที่เราสามารถติดต่อเข้ามาใช้งานได้จากเครื่องคอมพิวเตอร์อื่นได้โดยผ่านทางโปรโตคอลที่ซีพีทีบีไอพี(TCP/IP Protocol)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.2 การเขียนโปรแกรมในส่วนของแอปพลิเคชัน

แอปพลิเคชันที่สร้างขึ้นนี้จากข้างต้นที่เคยกล่าวแล้วว่าใช้ภาษาจาวา ซึ่งเป็นภาษาที่มีข้อดีหลายประการและรองรับการพัฒนาโปรแกรมแบบ Object Oriented ได้เป็นอย่างดี รวมถึงใช้โปรแกรม JBuilder เป็นเครื่องมือในการพัฒนา ในการเขียนโปรแกรมนั้นก่อนข้างมีรายละเอียดมากมาย ผู้พัฒนาจะพยายามทำให้เนื้อหาง่ายที่สุดในการทำความเข้าใจโดยการแบ่งเป็นข้อ ๆ ตามลำดับขั้นตอนดังนี้

1. ออกแบบยูสเซอร์อินเทอร์เฟซ(User Interface) โดยแบ่งโปรแกรมออกเป็นส่วน ๆ ตามยูสเคส ในแต่ละยูสเคสก็นำมาแบ่งเป็นส่วนย่อย ๆ อีกตามซีเควนซ์ไดอะแกรม โดยให้แต่ละส่วนเป็นพาเนล(Panel)

2. หลังที่ทำการแบ่งเป็นพาเนลได้แล้วนั้น แต่ละพาเนลก็หมายถึงหนึ่งคลาสนั่นเอง เราจึงสามารถแยกแต่ละพาเนลไปเขียนโปรแกรม อย่างอิสระ แล้วจึงค่อยนำมารวมกันในภายหลัง

3. ในการเขียนโปรแกรมติดต่อกับฐานข้อมูลนั้นเราใช้ JDBC เป็นตัวเชื่อมต่อให้โดยมีการระบุถึงเซิร์ฟเวอร์ฐานข้อมูลด้วยหมายเลขไอพี(IP Address)

4. หลังจากขั้นตอนนี้เราสามารถแยกแต่ละส่วนมาเขียนโปรแกรมได้ ซึ่งแยกไปตามคลาสต่าง ๆ

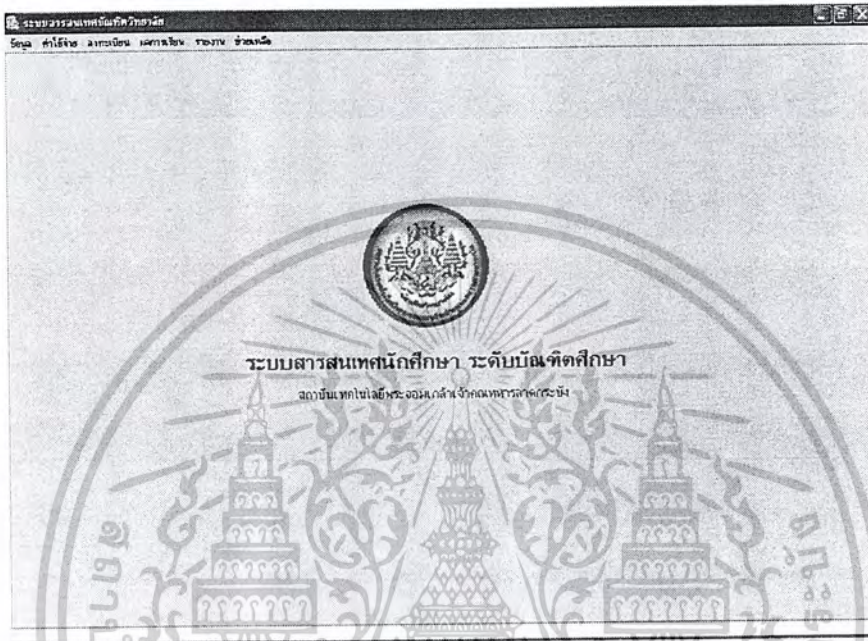
```
try
{
    Class.forName("COM.ibm.db2.jdbc.net.DB2Driver").newInstance();
    con = DriverManager.getConnection(url, uname, passwd);
}
catch(ClassNotFoundException cnfex)
{
    System.err.println("Failed to load JDBC/ODBC driver.");
    cnfex.printStackTrace();
}
catch(SQLException sqlx)
{
    System.err.println("Unable to connect");
    sqlx.printStackTrace();
}
catch(Exception ex)
{
    ex.printStackTrace();
}
```

### ตัวอย่างโค้ดการติดต่อฐานข้อมูลด้วย JDBC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. สิ่งหนึ่งที่ต้องคำนึงในการเขียนโปรแกรมก็คือข้อกำหนดต่าง ๆ ที่ได้ทำการศึกษาจากระบบจริงซึ่งเราต้องเขียนโปรแกรมให้มีการตรวจสอบตรงตามข้อกำหนดเหล่านั้นด้วย เช่น การตรวจสอบกรณีตารางเรียนหรือตารางสอบชนกัน

6. เมื่อเราเขียนโปรแกรมแต่ละส่วนเสร็จก็จะนำมาประกอบในส่วนของพาเนลหลัก โดยเราได้ออกแบบอินเตอร์เฟซให้เป็นในลักษณะเมนูบาร์



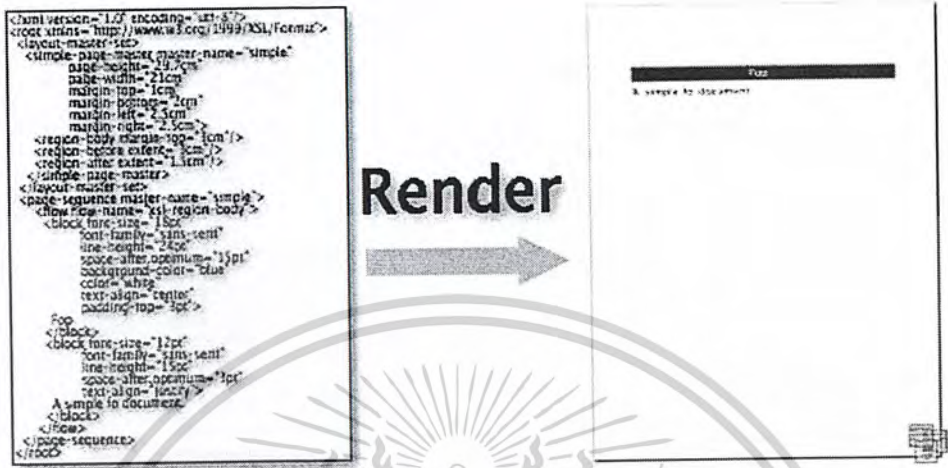
รูปที่ 6-4 รูปส่วนอินเตอร์เฟซของ MainPanel

รายละเอียดคณะ	
รหัสคณะ:	002
ชื่อคณะ [ท]:	วิศวกรรมศาสตร์
ชื่อคณะ [อ]:	Engineering
คณะ:	
ตำแหน่ง [ท]:	คิงเคอร์
ชื่อ-นามสกุล [ท]:	วิศวกรรมศาสตร์
ตำแหน่ง [อ]:	King
ชื่อ-นามสกุล [อ]:	TestDeanEnome
ตำแหน่ง:	คณะ
ผู้อำนวยการคณะ [ท]:	stasdat

รูปที่ 6-5 รูปส่วนอินเตอร์เฟซของ รายละเอียดข้อมูลทั่วไป

จากขั้นตอนข้างต้นอาจไม่สามารถอธิบายลงในรายละเอียดทั้งหมดได้ซึ่งสามารถดูในรายละเอียดจากเอกสารโค้ด (Source Code) ได้ ซึ่งงานนี้ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ส่วนการเขียนโปรแกรมเพื่อพิมพ์รายงานใบทรานสคริปต์นั้น เราจะใช้ Formatting Objects Processor (FOP) ซึ่งเป็นตัวจัดรูปแบบการพิมพ์โดยวิธีการเขียนโปรแกรมจะเขียนให้อยู่ในรูปแบบ XSL-FO จากนั้นจึงใช้ FOP แปลงรูปแบบโค้ดที่ได้ให้เป็น PDF



รูปที่ 6-6 ตัวอย่างผลลัพธ์จากการแปลงจาก XSL-FO เป็น PDF โดยใช้ FOP

```
<?xml version="1.0"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="content"
      page-width="210mm" page-height="297mm">
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="content">
    <fo:flow flow-name="xsl-region-body">
      <fo:table table-layout="fixed" width="100%">
        <fo:table-column column-width="proportional-column-width(1)"/>
        <fo:table-body>
          <fo:table-row height="297mm">
            <fo:table-cell display-align="center">
              <fo:block text-align="center">
                <fo:external-graphic src="fop.jpg"/>
              </fo:block>
            </fo:table-cell>
          </fo:table-row>
        </fo:table-body>
      </fo:table>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

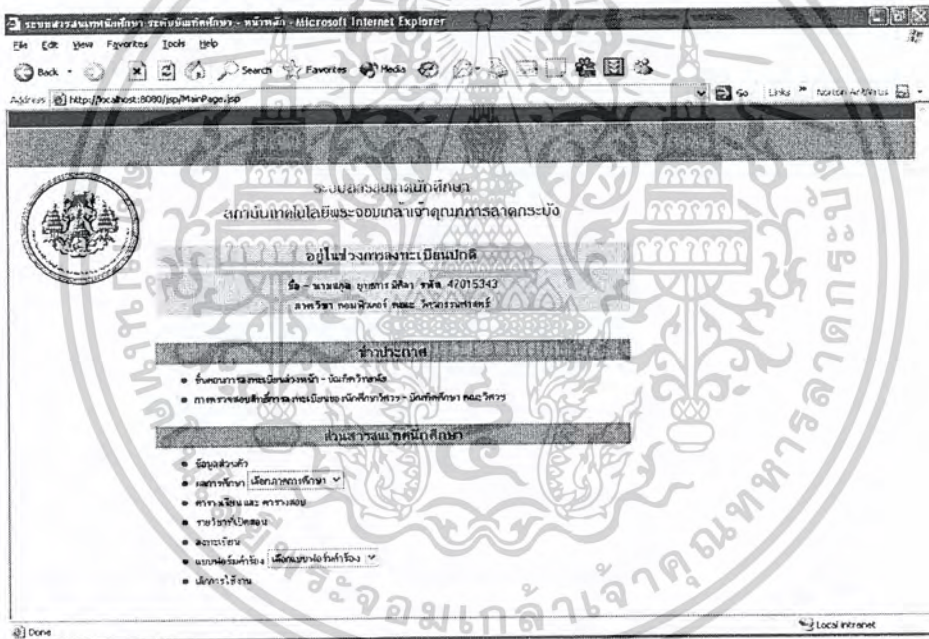
#### ตัวอย่างโค้ดการเขียน XSL-FO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3 การเขียนโปรแกรมในส่วนของเว็บแอปพลิเคชัน

การเขียนโปรแกรมในส่วนนี้จะแตกต่างจากการเขียนโปรแกรมในส่วนแอปพลิเคชันเพราะว่า การแสดงผลในส่วนนี้จะอยู่บนเว็บเบราว์เซอร์(Web Browser) ดังนั้นการเขียนโปรแกรมแบบนี้จึงอยู่บนสิ่งแวดล้อมอินเทอร์เน็ตโดยใช้เทคโนโลยีของจาวาเซิร์ฟเลทและภาษาเจเอสพี ซึ่งได้กล่าวถึงทฤษฎีทั้งสองเรื่องนี้แล้วในบทที่ 3 แล้ว

1. เนื่องจากโปรแกรมในส่วนนี้จะอยู่บนเว็บเบราว์เซอร์ เราจึงต้องมี เว็บเซิร์ฟเวอร์(WebServer) ที่รองรับเทคโนโลยีของ Servlet และ JSP ซึ่งในโครงการนี้เราใช้ Apache TOMCAT
2. เราสามารถเขียนโปรแกรมของเราทั้งในส่วนคลาสที่เป็น Servlet และ JSP ไว้ในโฟลเดอร์(Folder) ที่ Apache TOMCAT สร้างไว้ให้ เราก็จะสามารถทดลองโปรแกรมที่เราเขียนได้โดยผ่านทางเว็บเบราว์เซอร์บนเครื่องเราเอง
3. ขั้นตอนต่อไปก็คือการออกแบบยูสเซอร์อินเตอร์เฟซในแต่ละส่วนซึ่งในที่นี้เราใช้ Macromedia Dreamweaver ในการออกแบบ แล้วก็สามารถนำไปเขียนโปรแกรม โดยแยกจากกันเพื่อทดลองทีละส่วนได้



รูปที่ 6-7 ตัวอย่างยูสเซอร์อินเตอร์เฟซสำหรับส่วนเว็บแอปพลิเคชัน

4. ในขั้นตอนนี้เราสามารถเขียนคลาสต่าง ๆ ขึ้นมาใช้งานได้ตามปกติ การติดต่อกับฐานข้อมูลก็ทำคล้ายกับการติดต่อในส่วนแอปพลิเคชัน ซึ่งสามารถใช้มอดจูลจากส่วนแอปพลิเคชันมาช่วยในการจัดการได้

5. นอกจากใช้ JBuilder แล้วในการเขียน JSP ซึ่งเป็นการทำงานแบบ Server Side Included อย่างหนึ่ง จำเป็นต้องสร้างโครงร่างของหน้า html ขึ้นมาก่อนซึ่งในที่นี้เราใช้ Macromedia Dreamweaver เข้ามาช่วยเสริมการทำงานในส่วนนี้ด้วย

6. นำส่วนต่าง ๆ มาประกอบกันให้สามารถทำงานเชื่อมต่อกันได้ด้วยการกำหนดลิงก์(Link) ให้เชื่อมโยงถึงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 7

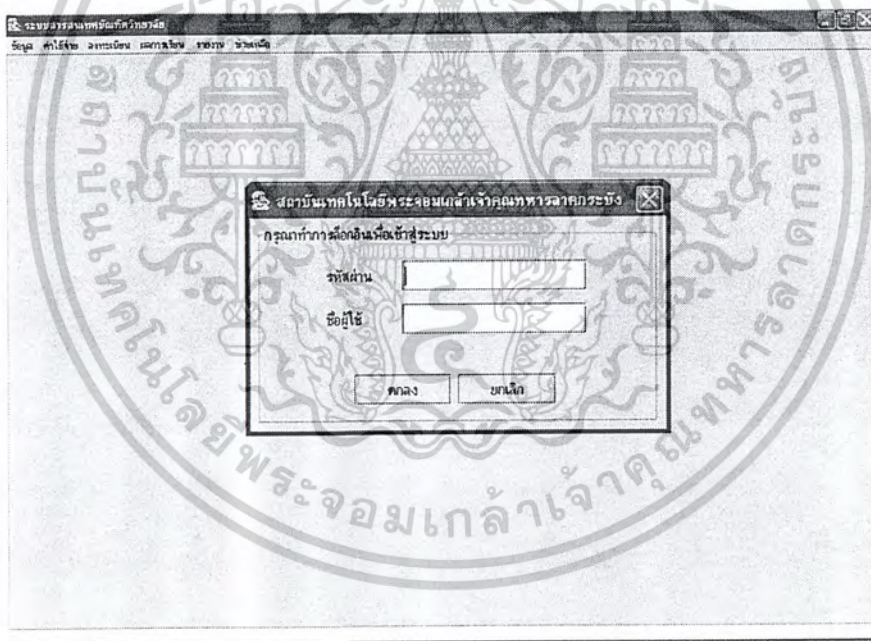
### การแสดงผลการทำงานจากระบบ

กล่าวถึงการทดสอบระบบ เป็นการแสดงผลการทำงาน โดยแบ่งออกเป็น 2 ส่วนก็คือส่วนที่เป็นแอปพลิเคชัน ให้สำหรับเจ้าหน้าที่ทะเบียน และส่วนที่เป็นเว็บแอปพลิเคชันสำหรับนักศึกษา ซึ่งมีรายละเอียดดังนี้

#### 7.1 ส่วนของแอปพลิเคชันสำหรับเจ้าหน้าที่

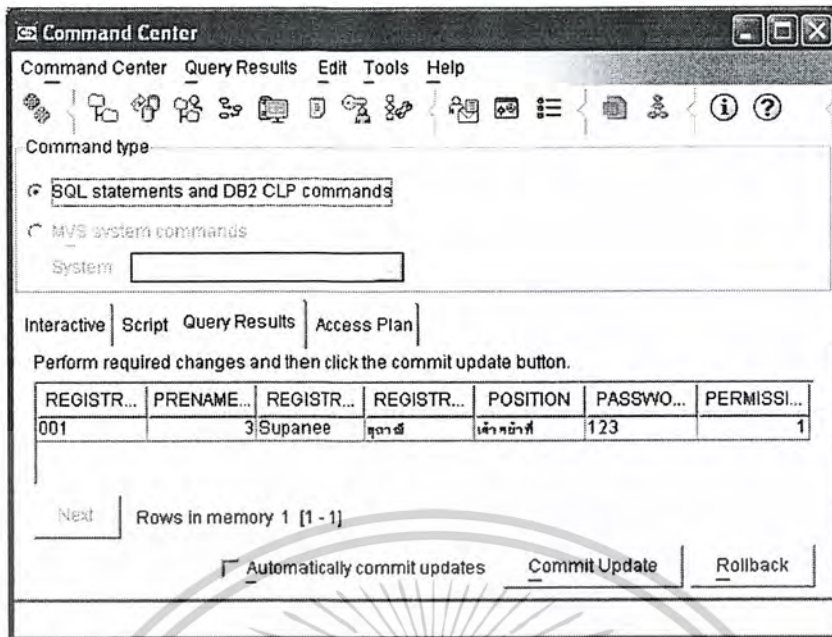
สำหรับการทดสอบแสดงผลการทำงานนั้น จะแสดงเพียงฟังก์ชันการทำงานหลักๆ ของโปรแกรม โดยมีรายละเอียดการทำงานดังนี้

1. ก่อนที่มีการเข้าใช้งานระบบต้องทำการล็อกอินเข้าสู่ระบบก่อน ดังรูป โดย ชื่อผู้ใช้ และ รหัสผ่าน ได้จากตาราง Registrar ในฐานข้อมูลของ DB2



รูปที่ 7-1 แสดงหน้าจอการล็อกอินเพื่อเข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-2 รูปแสดงข้อมูลที่อยู่ในฐานข้อมูล

2. ฟังก์ชันการทำงานต่างๆ ของระบบสามารถเลือกเข้าไปทำงานกับระบบได้ที่เมนูบาร์ ซึ่งเราสามารถออกแบบให้เป็นเมนูบาร์เพื่อความสะดวกแก่เจ้าหน้าที่ในการใช้งาน โดยแบ่งเป็นการทำงานดังนี้คือ

- เมนูข้อมูลทั่วไป โดยแบ่งเป็นเมนูย่อยคือ ข้อมูลทั่วไป, โครงสร้างหลักสูตร, นักศึกษา, ทุนการศึกษา, อาจารย์, เจ้าหน้าที่ และอื่นๆ
- เมนูค่าใช้จ่าย เป็นฟังก์ชันการทำงานเกี่ยวกับข้อมูลค่าใช้จ่าย แบ่งเป็นเมนูย่อยคือ ค่าธรรมเนียมการศึกษาและเงินสนับสนุนหรือเงินอุดหนุนการศึกษา
- เมนูลงทะเบียน แบ่งเป็นเมนูย่อยคือ กำหนดเวลา, ลงทะเบียน, เพิ่มรายวิชา, เปลี่ยนรายวิชาและถอนรายวิชา
- เมนูผลการเรียน โดยมีฟังก์ชันการทำงานการกรอกเกรดนักศึกษาตามรหัสนักศึกษาและรายวิชา และมีการทำงานในการพิมพ์ใบทรานสคริปต์
- เมนुरายงาน เป็นการแสดงผลรายงานต่างๆ
- เมนูช่วยเหลือ แสดงรายละเอียดเกี่ยวกับ โปรแกรม

การทดสอบการทำงานจะแสดงเพียงฟังก์ชันการทำงานหลักๆ ของโปรแกรม โดยมีรายละเอียดการทำงานดังนี้

#### 7.1.1 ฟังก์ชันการทำงานการจัดการเกี่ยวกับข้อมูลทั่วไป

เป็นการทำงานเกี่ยวกับข้อมูลรายละเอียดคณะ, ภาควิชา, สาขาและแขนง โดยสามารถที่จะเพิ่ม,ลบ และแก้ไขข้อมูลได้ ดังรูป ซึ่งทางผู้พัฒนาได้ทำเป็นทริโมเดลเพื่อความสะดวกและเข้าใจได้ง่ายยิ่งขึ้นในการจัดการข้อมูล เมื่อต้องการทำการเพิ่มข้อมูลใดๆ ต้องให้สถานะแอดที่พีที่ระดับสูงกว่าที่ต้องการเพิ่ม เช่น ถ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการเพิ่มข้อมูลคณะ ต้องให้สถานะของข้อมูลทั่วไปที่ต้องการเพิ่มคณะนั้นแอคทีฟ จากนั้นกดปุ่มเพื่อทำการเพิ่มข้อมูลได้ตามต้องการ เป็นต้น และข้อมูลที่เพิ่มเข้าไปนั้นก็ถูกเพิ่มเข้าไปในทรีโมเดลตามระดับ

รูปที่ 7-3 การจัดการเกี่ยวกับข้อมูลทั่วไป

เมื่อทำงานเกี่ยวกับฟังก์ชันการทำงานของส่วนข้อมูลทั่วไปนี้ จะมีผลต่อฐานข้อมูลในระบบคือ ตาราง FACULTY, DEPARTMENT, MAJOR และ MINOR ดังรูปที่ 7-4 จะเห็นข้อมูลที่เพิ่มเข้าไปในฐานข้อมูลคือ คณะวิศวกรรมศาสตร์ เมื่อลองเทียบกับข้อมูลในตาราง FACULTY ใน Command Center ของ DB2

FACULTYID	FACULTY...	FACULTY...	DEANNAME	DEANENAME	POSITION	APPROVER
001	Architecture	...	4 สมศักดิ์ รุ่งเรือง	Somsak Rungraung	เจ้าพนักงานทะเบียน	สุปราณี สมพร...
002	Engineering	วิศวกรรมศาสตร์	4 บุญวัฒน์ อัครชัย	Boonwat Attachoo	เจ้าพนักงานทะเบียน	สุปราณี สมพร...

รูปที่ 7-4 แสดงข้อมูลในตาราง FACULTY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7.1.2 ฟังก์ชันการทำงานโครงสร้างหลักสูตร

เป็นการทำงานเกี่ยวกับการจัดการหลักสูตรและรายวิชา รวมถึงเซกชันด้วย การทำงานคือก่อนที่จะทำการจัดการข้อมูลต่างๆ ต้องทำการเลือกคณะ, ภาควิชา และสาขา ที่ต้องการจัดการ เมื่อทำการเลือกแล้วจะสามารถเข้าไปทำงานจัดการหลักสูตรและรายวิชาได้ดังรูปที่ 7-5

รูปที่ 7-5 การจัดการเกี่ยวกับโครงสร้างหลักสูตร

เมื่อทำงานเกี่ยวกับฟังก์ชันการทำงานของ โครงสร้างหลักสูตรนี้ จะมีผลต่อฐานข้อมูลในระบบคือ ตาราง MAJOR ดังรูปที่ 7-5 จะเห็นข้อมูลหลักสูตรที่เพิ่มเข้าไปในฐานข้อมูลจะเป็นข้อมูลหลักสูตรสำหรับ สาขาวิศวกรรมคอมพิวเตอร์ เมื่อลองเทียบกับข้อมูลในตาราง MAJOR ใน Command Center ของ DB2

MAJORID	MAJOREN...	MAJORNAME	DEPARTMENTID	MASTERENAME	MASTERTNAME	MASTEREAE
001	Computer...	วิศวกรรมศาสตร์	002	Master of Engineering (Computer Engin...	วิศวกรรมศาสตรมหาบัณฑิต (วิ...	M Eng (Computer

รูปที่ 7-6 แสดงข้อมูลในตาราง MAJOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 7.1.3 ฟังก์ชันการทำงานการกำหนดเวลา

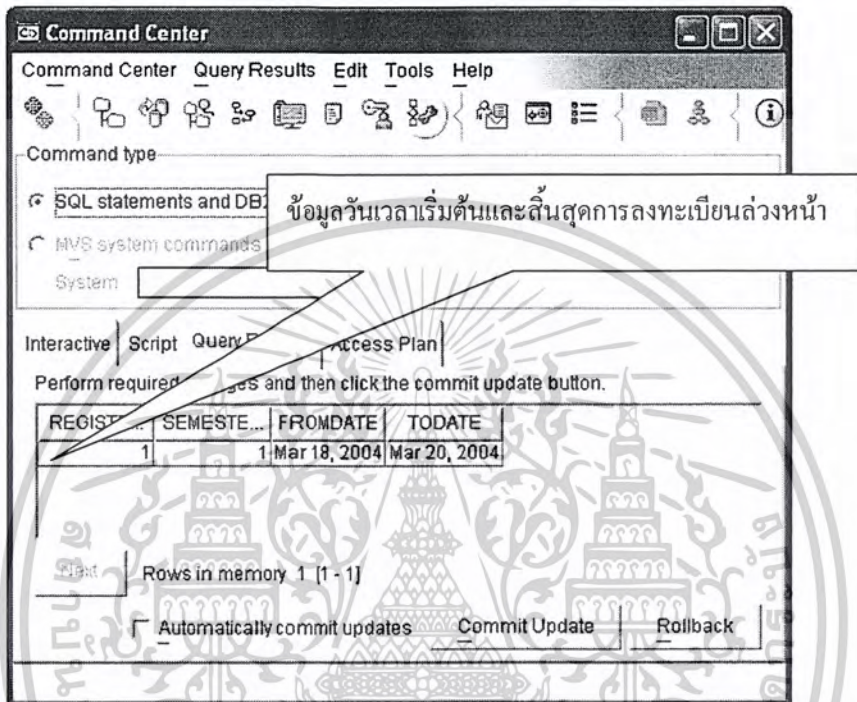
เป็นการทำงานเกี่ยวกับการจัดการการกำหนดเวลา การทำงานคือก่อนที่จะทำการจัดการข้อมูลต่างๆ ต้องทำการเลือกภาคการศึกษาและปีการศึกษาที่ต้องการกำหนดเวลาดังรูปที่ 7-7 เมื่อทำการเลือกแล้วจะสามารถเข้าไปทำงานการกำหนดเวลาได้ดังรูปที่ 7-8

รูปที่ 7-7 การเลือกภาคการศึกษาและปีการศึกษาก่อนทำการกำหนดเวลา

รูปที่ 7-8 การกำหนดเวลาสำหรับการลงทะเบียนต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำงานเกี่ยวกับฟังก์ชันการทำงานของกาหนดเวลานี้ จะมีผลต่อฐานข้อมูลในระบบคือ ตาราง APPOINTMENT ดังรูปที่ 7-9 จะเห็นข้อมูลหลักสูตรที่เพิ่มเข้าไปในฐานข้อมูลคือรายละเอียดของวันเวลา เริ่มต้นและสิ้นสุดการลงทะเบียนล่วงหน้า เมื่อลองเทียบกับข้อมูลในตาราง APPOINTMENT ใน Command Center ของ DB2

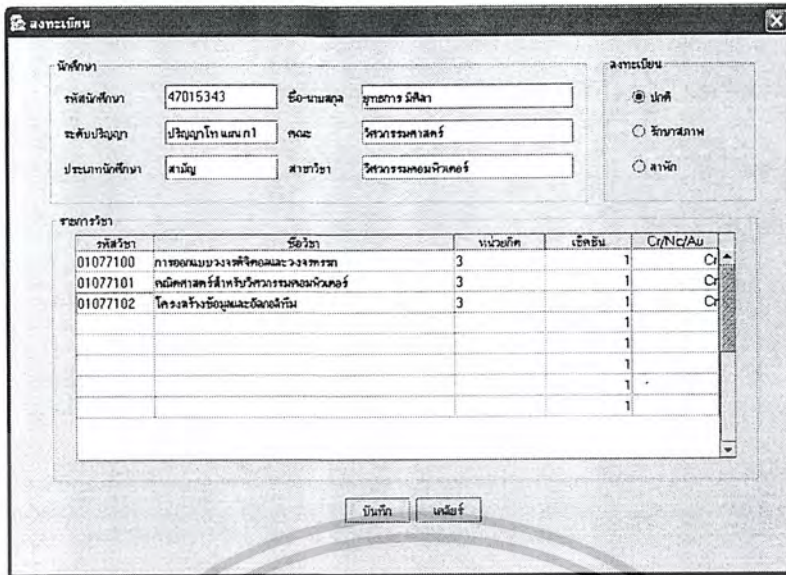


รูปที่ 7-9 แสดงข้อมูลในตาราง APPOINTMENT

#### 7.1.4 ฟังก์ชันการทำงานของลงทะเบียนปกติ

เป็นฟังก์ชันการลงทะเบียนปกติ การทำงานคือก่อนที่จะทำการจัดการข้อมูลต่างๆ ต้องทำการเลือกว่าเป็นการลงทะเบียนปกติ เมื่อทำการเลือกแล้วก็สามารถทำการลงทะเบียนได้เลยโดยการป้อนรหัสวิชาที่ตามที่ต้องการได้ดังรูปที่ 7-10

เมื่อทำงานเกี่ยวกับฟังก์ชันการลงทะเบียนปกตินี้ จะมีผลต่อฐานข้อมูลในระบบคือ ตาราง REGISTRATION จะเห็นข้อมูลการลงทะเบียนปกติที่เพิ่มเข้าไปในฐานข้อมูลคือรายละเอียดการลงทะเบียนของนักศึกษาคนนั้นๆ



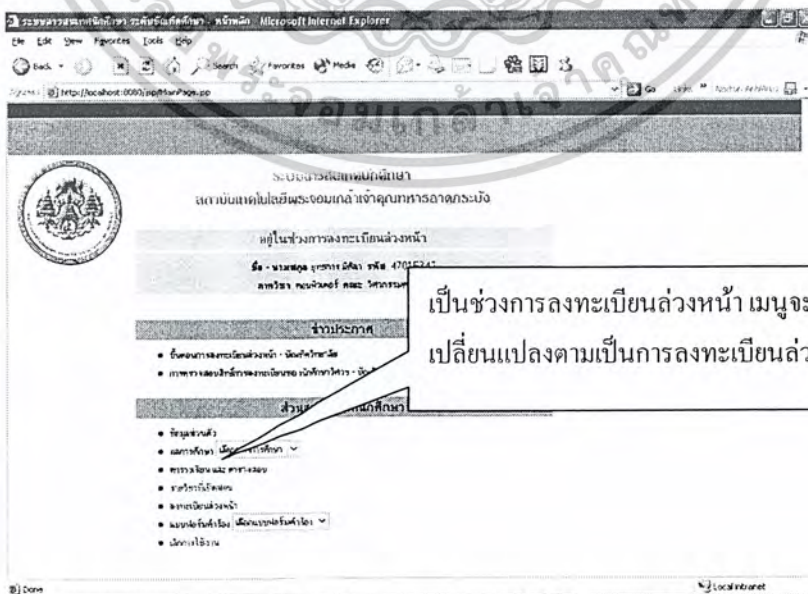
รูปที่ 7-10 การลงทะเบียนปกติ

จะเห็นว่าการทดสอบระบบในส่วนแอปพลิเคชันสำหรับเจ้าหน้าที่ เป็นไปได้อย่างถูกต้องตาม กระบวนการที่ได้ออกแบบไว้ สำหรับฟังก์ชันการทำงานในส่วนอื่นๆก็สามารถทดสอบได้ในกรณีเดียวกัน

7.2 ส่วนของเว็บแอปพลิเคชันสำหรับนักศึกษา

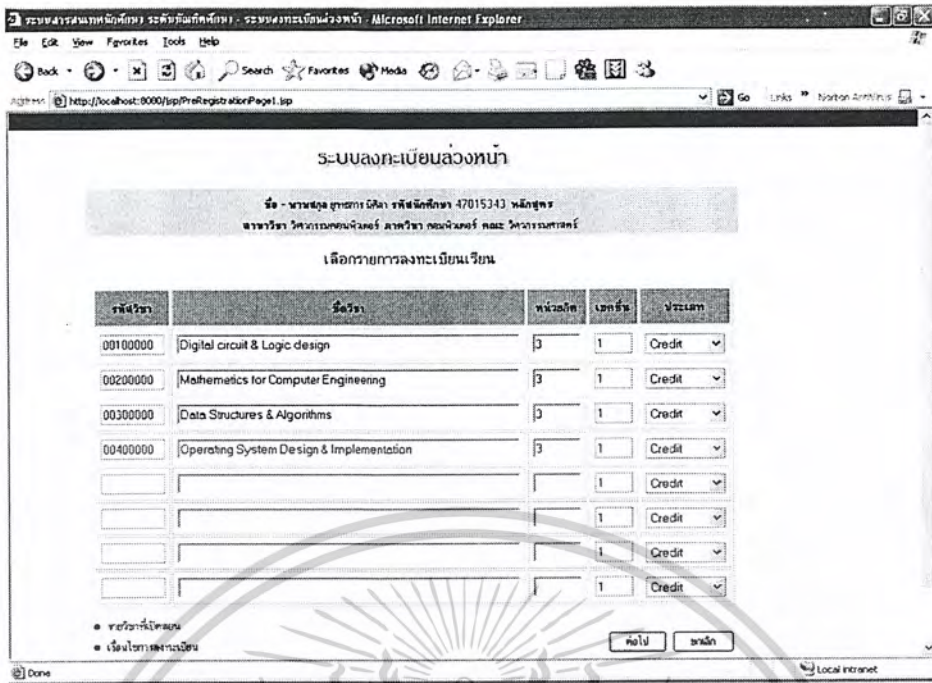
การทดสอบในส่วนของเว็บแอปพลิเคชันนั้น เพื่อให้เข้าใจในการทำงานของระบบ ในที่นี้จะ ยกตัวอย่างการลงทะเบียนล่วงหน้าของนักศึกษา

1. การลงทะเบียนล่วงหน้า นั้น เป็นหนึ่งในเมนูที่มีการเปลี่ยนแปลงตามเวลาปัจจุบัน กล่าวคือ หาก วันที่นักศึกษามีการล็อกอินเข้ามาอยู่ในช่วงของการลงทะเบียนล่วงหน้า เมนูการลงทะเบียนล่วงหน้าก็จะ สามารถทำงานได้(Active) ซึ่งจะสามารถตรวจสอบช่วงเวลาได้จาก แถบแสดงเวลา



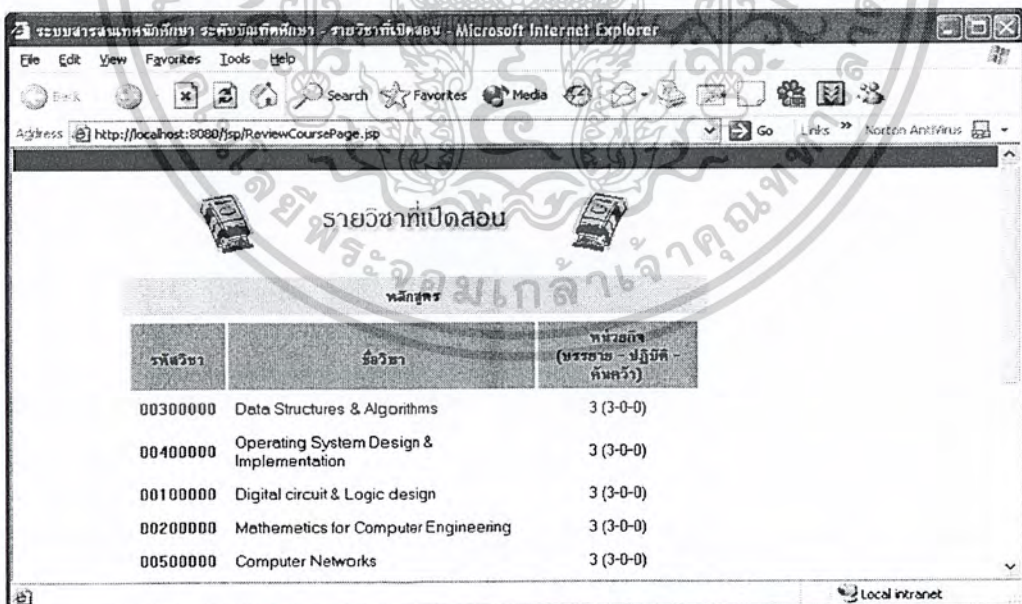
รูปที่ 7-11 การแสดงช่วงเวลาซึ่งเมนูจะเปลี่ยนแปลงตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-12 แสดงให้เลือกลงทะเบียนตามรายวิชาที่เปิดสอนตามหลักสูตร

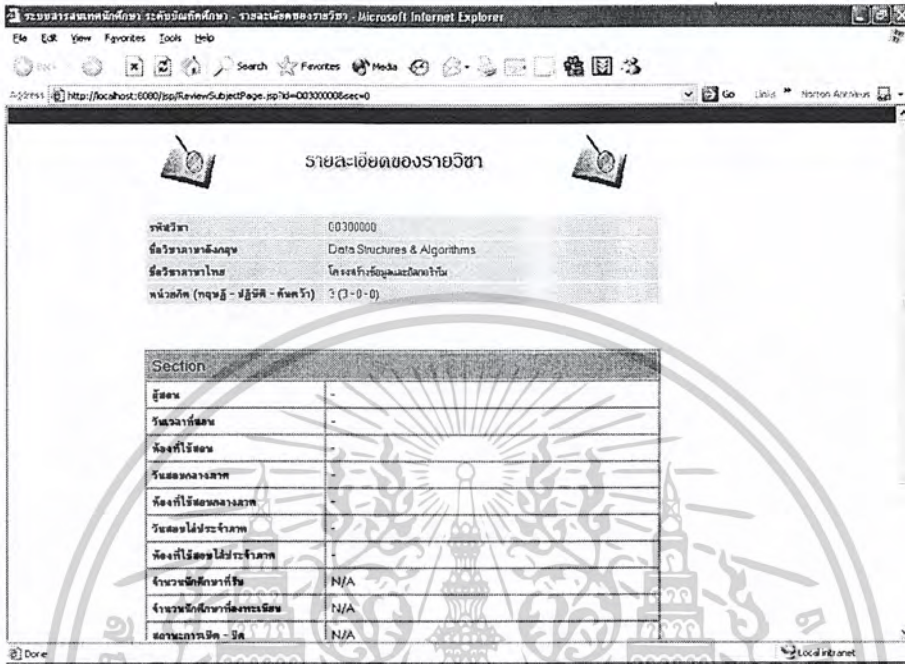
2. นักศึกษาจะทำการเลือกลงทะเบียนรายวิชาโดยป้อนรหัสวิชา ตามรายวิชาที่เปิดสอนตามหลักสูตร โดยนักศึกษาสามารถตรวจสอบรายชื่อวิชาที่เปิดสอนได้จากเมนู รายชื่อวิชาที่เปิดสอน ดังรูป



รูปที่ 7-13 แสดงรายวิชาที่เปิดสอนตามหลักสูตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยนักศึกษาสามารถเข้าไปดูรายละเอียดของแต่ละรายวิชา คำอธิบายรายวิชา และดูว่าวิชาดังกล่าวมีวิชาบังคับก่อนหรือไม่ หรือคุณสมบัติของรายวิชาว่าเปิด หรือปิด สามารถดูจำนวนนักศึกษาที่ลงทะเบียนไปแล้ว และจำนวนนักศึกษาที่รับสูงสุดได้ ในกรณีที่รายวิชามีการจำกัดจำนวนผู้ที่ลงทะเบียนรายวิชาดังกล่าวไว้



รูปที่ 7-14 แสดงรายละเอียดรายวิชาที่เปิดสอนตามหลักสูตร

3. ระบบจะทำการตรวจสอบเงื่อนไขการลงทะเบียนต่างๆ เช่น วิชาที่จะลงทะเบียนมีวิชาบังคับก่อนหรือไม่ เคยลงทะเบียนเรียนวิชาดังกล่าวมาแล้วหรือยัง และจำนวนหน่วยกิตรวมที่สามารถลงทะเบียนได้ รวมถึงรายวิชาที่มีการจำกัดจำนวนผู้ที่ลงทะเบียน ถ้าไม่ผ่านการตรวจสอบก็ไม่สามารถลงทะเบียนวิชาดังกล่าวได้ ระบบก็จะทำการแสดงรายวิชาที่ลงทะเบียนอีกครั้ง พร้อมทั้งแสดงรายละเอียดค่าใช้จ่ายทั้งหมดให้นักศึกษาทราบ

4. ขั้นตอนสุดท้าย ระบบจะทำการเก็บข้อมูลการลงทะเบียนไว้เพื่อนำไปทดสอบกับการลงทะเบียนจริง และแสดงข้อความเพื่อบอกช่วงเวลาที่นักศึกษาต้องมาลงทะเบียนจริงอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบลงทะเบียน

ชื่อ - นามสกุล สุภากร วัฒนา จฬินภักดิ์ 47015343 วิทยาลัย  
สาขาวิชา วิศวกรรมคอมพิวเตอร์ สาขาวิชา คอมพิวเตอร์ และ วิศวกรรมศาสตร์

\* วิชาดังกล่าวนั้นมี Prerequisite โปรดตรวจสอบรายวิชาที่เปิดสอนอีกครั้ง \*

รหัสวิชา	ชื่อวิชา	หน่วยกิต	เรียน	ประเภท
00200000	Mathematics for Computer Engineering	3	1	Cr

ยังไม่ไป

รูปที่ 7-15 แสดงระบบตรวจสอบวิชาที่ต้องเรียนวิชาบังคับก่อน

ระบบลงทะเบียนล่วงหน้า

ชื่อ - นามสกุล สุภากร วัฒนา จฬินภักดิ์ 47015343 วิทยาลัย  
สาขาวิชา วิศวกรรมคอมพิวเตอร์ สาขาวิชา คอมพิวเตอร์ และ วิศวกรรมศาสตร์

รายวิชาทั้งหมดที่ลงทะเบียน

รหัสวิชา	ชื่อวิชา	หน่วยกิต	เรียน	ประเภท
00100000	Digital circuit & Logic design	3	1	Credit
00300000	Data Structures & Algorithms	3	1	Credit
00500000	Computer Networks	3	1	Credit
	รวม	9	หนึ่งภาค	

ไปยังไม่ไป

รูปที่ 7-16 แสดงรายวิชาที่ลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบลงทะเบียนล่วงหน้า

ชื่อ - นามสกุล ผู้สมัคร ปีการศึกษา 47015343 หลักสูตร  
สาขาวิชา ศึกษาระดับมัธยมศึกษา สาขาวิชา คอมพิวเตอร์ และ ศึกษาระเบียบ

รายละเอียดค่าใช้จ่าย ภาคเรียนที่ 1 ปีการศึกษา 2547

รายละเอียด	จำนวนเงิน (บาท)
ค่าลงทะเบียนสาขาวิชา	2700.00
ค่าลงทะเบียนวิชาศึกษา	200.00
ค่าจัดระงับทรัพย์สิน	20.00
ค่าเอกสารของมหาวิทยาลัยใหม่	50.00
ค่าไปรษณีย์เอกสาร	400.00
ค่าบำรุงสถานที่	500.00
ค่าบำรุงเครื่องพิมพ์ (ยกเว้น)	200.00
ค่าบำรุงคอมพิวเตอร์ (ยกเว้น)	50.00
เงินค่าลงทะเบียน	20000.00
<b>รวมเงินที่ต้องชำระ</b>	<b>24120.00</b>

โปรดชำระเงินค่าลงทะเบียนก่อน

รูปที่ 7-17 แสดงค่าใช้จ่ายตามหลักสูตร

ระบบลงทะเบียนล่วงหน้า

ขณะนี้ขอแจ้งเรื่องระเบียบล่วงหน้าเรียนสมบูรณ์แล้ว  
หนังสือจากกรมการศึกษาระดับมัธยมศึกษา (10 มี.ค. 2547 - 17 มี.ค. 2547)  
คุณแจ้งเรื่องขอทราบผลการลงทะเบียนจริง  
ภายใน ( 18 มี.ค. 2547 - 25 มี.ค. 2547 )

รูปที่ 7-18 แสดงข้อความการลงทะเบียนสมบูรณ์

จะเห็นว่าการทดสอบระบบในส่วนเว็บแอปพลิเคชันสำหรับนักศึกษา เป็นไปได้ถูกต้องตามกระบวนการที่ได้ออกแบบไว้ สำหรับฟังก์ชันการทำงานในส่วนอื่นๆก็สามารถทดสอบได้ในกรณีเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 8

# บทสรุปและบทวิจารณ์

### 8.1 บทสรุป

การปรับปรุงระบบสารสนเทศนักศึกษา ระดับบัณฑิตศึกษาด้วยวิธีเชิงวัตถุนับว่าได้รับความสำเร็จอย่างยิ่งในแง่ของการวิเคราะห์และออกแบบ เพราะในระบบนี้การวิเคราะห์และออกแบบได้ทำอย่างรอบคอบที่สุด จึงทำให้ได้ระบบที่อำนวยความสะดวกและครอบคลุมประโยชน์การใช้สอยมากในระดับที่น่าพอใจ ส่วนในเรื่องของการสร้างหรือการเขียนโปรแกรมนั้นก็สามารถดำเนินการได้ครอบคลุมในส่วนระบบการลงทะเบียนได้เกือบทั้งหมด ทั้งนี้เรายังได้นำ โปรแกรมตัวอย่าง(Prototype) ไปให้เจ้าหน้าที่บัณฑิตวิทยาลัยลองทำการทดสอบระบบ ก็ได้รับความพอใจจากทางเจ้าหน้าที่มากในระดับที่น่าพอใจ

ส่วนเครื่องมือที่เราใช้ ในการทำงานนั้นคือ Case tool ต่าง ๆ ซึ่งได้แก่

- โปรแกรม Rational Rose 2002 เป็น CASE Tool ที่สามารถทำให้เราสร้างโมเดลของระบบโดยใช้ UML ได้อย่างสะดวก
- โปรแกรม JBuilder เป็นเครื่องมือที่ใช้ในการเขียนภาษาจาวาได้เป็นอย่างดีด้วยความสามารถที่มีให้มาไม่ว่าจะเป็น การเขียนโปรแกรมแบบแอปพลิเคชัน, Servlet หรือ JSP ก็ตาม
- โปรแกรมฐานข้อมูล DB2 จัดเป็น DBMS ที่มีความสามารถในด้านการสร้างและจัดการฐานข้อมูลได้ดี

ส่วนในเรื่องของการศึกษาและทำความเข้าใจระบบทะเบียนนักศึกษา ระดับบัณฑิตศึกษานั้น ได้ทำการวิเคราะห์หาจุดอ่อนต่าง ๆ ซึ่งได้กล่าวมาแล้วข้างต้น รวมทั้งยังสามารถนำไปสร้างเป็น โมเดลของระบบที่พยายามสร้างความสะดวกให้มากที่สุดอีกด้วย

ส่วนในการเขียนโปรแกรมนั้นได้มีการทำงานไปประมาณ 90 เปอร์เซ็นต์ ซึ่งจะขอชี้แจงรายละเอียดดังนี้

- ส่วนของแอปพลิเคชัน ได้ทำการพัฒนาในส่วนของการจัดการข้อมูลหลักสูตร นักศึกษา หน่วยงานการศึกษา ระยะเวลาการลงทะเบียน การจัดการรายวิชา จัดการค่าใช้จ่ายต่าง ๆ ลงทะเบียนเรียน เพิ่ม ถอน เปลี่ยนรายวิชา ป้อนเกรดในแต่ละรายวิชา การลงทะเบียนลาพัก การลงทะเบียนรักษาสภาพและจัดการใบทรานสคริปต์
- ในส่วนของเว็บแอปพลิเคชันสำหรับนักศึกษาได้ทำการพัฒนาในส่วนของการลงทะเบียน ล่วงหน้า ลงทะเบียนปกติ ตรวจสอบรายละเอียดรายวิชา ตรวจสอบข้อมูลการลงทะเบียน ตรวจสอบเกรด และยื่นแบบฟอร์มคำร้อง
- ส่วนเว็บแอปพลิเคชันสำหรับอาจารย์ในการกรอกข้อมูลเกรดของนักศึกษา และการตรวจสอบการลงทะเบียนของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ระบบที่พัฒนาขึ้นสามารถทำการตรวจสอบเงื่อนไขได้ต่างๆ เช่น
  - ตรวจสอบรายวิชาที่นักศึกษาได้ลงทะเบียนไปแล้ว จึงทำให้ไม่เกิดปัญหาในการลงทะเบียนซ้ำซ้อนในรายวิชาเดียวกัน
  - ตรวจสอบจำนวนผู้ลงทะเบียน เนื่องจากมีบางคณะจำกัดจำนวนผู้ลงทะเบียนแต่ละวิชา โดยถ้ามีผู้ลงทะเบียนครบแล้ว ผู้ที่ลงทะเบียนทีหลังจะไม่สามารถลงทะเบียนได้
  - ตรวจสอบรายวิชาที่จำเป็นต้องเรียนมาก่อน (Prerequisite)
  - ตรวจสอบรายวิชาที่ถูกปิดไป โดยถ้ารายวิชาถูกปิดไป จะไม่สามารถทำการลงทะเบียนรายวิชาดังกล่าวได้
  - ตรวจสอบจำนวนรวมหน่วยกิต สูงสุด , ต่ำสุด ที่นักศึกษาสามารถทำการลงทะเบียนได้
  - สามารถตรวจสอบ ว่านักศึกษาถูกทำทัณฑ์บน หรือ พ้นสภาพ หรือไม่
  - ตรวจสอบเวลาเรียนและเวลาสอบว่ามีการชนกันหรือไม่ในวิชาที่นักศึกษาต้องการลงทะเบียน
  - การตรวจสอบกรณีนักศึกษาได้รับการยกเว้นต่าง ๆ เช่น ได้รับการยกเว้นค่าหน่วยกิต ยกเว้นเงินสนับสนุน ยกเว้นทั้งหมด ยกเว้นหน่วยกิต/เงินสนับสนุน
  - การตรวจสอบเงื่อนไขสถานะของนักศึกษา (นักศึกษาเก่าหรือนักศึกษาใหม่)
  - การตรวจสอบเงื่อนไขค่าใช้จ่ายการจ่ายเงินอุดหนุนสมทบ

นอกจากนี้ได้เพิ่มข้อมูลประเภทกลุ่มการเรียน คือ ภาคพิเศษ(เสาร์-อาทิตย์), ภาคพิเศษ(นอกเวลา ราชการ) และ หลักสูตรนานาชาติ ซึ่งตามระบบเดิมนั้นมีเพียง ประเภทกลุ่มการเรียน ภาคปกติและภาคสมทบ เท่านั้น

จากสิ่งที่ได้กล่าวมาจะเห็นได้ว่าการศึกษาวิจัยนี้ได้ทำครอบคลุมตามวัตถุประสงค์ทั้ง 4 ข้อที่ได้ตั้งไว้ข้างต้นแล้ว

## 8.2 บทวิจารณ์

หลังจากที่ได้ลงมือทำโครงการวิจัยนี้ สิ่งที่ผู้วิจัยได้ทำการศึกษาและค้นคว้า รวมถึงทำการวิเคราะห์ทำให้สามารถทราบได้ถึงข้อเท็จจริงต่าง ๆ ที่ส่งผลให้การทำงาน ซึ่งถ้าไม่ลงมือเข้าไปวิเคราะห์และศึกษาอย่างจริงจังอาจเป็นการยากที่จะทราบถึงข้อเท็จจริงเหล่านั้นได้ และอาจจะมีข้อจำกัดบางประการที่จะทำให้ความสามารถของระบบไม่สามารถใช้ได้ เช่นถ้าอาจารย์ทำการส่งเกรดล่าช้าก็จะทำให้ไม่สามารถทำการลงทะเบียนล่วงหน้าได้ตามเวลาที่วางไว้ ผู้วิจัยจึงจะขอกล่าวถึงข้อเท็จจริงเหล่านั้นเพื่อให้ผู้อ่านสามารถใช้ประโยชน์ในปฏิญญาพันธ์ และระบบที่พัฒนาขึ้นนี้ได้เต็มที่

### 8.3 ปัญหาที่พบในการดำเนินงาน

1. ด้วยกฎระเบียบข้อบังคับต่าง ๆ ที่มีมากมายทำให้เกิดความยุ่งยากและสับสน อีกทั้งยังมีความแตกต่างกันในแต่ละคณะซึ่งทำให้ยังเป็นการเพิ่มรายละเอียดในการวิเคราะห์เป็นอย่างยิ่ง ยกตัวอย่างเรื่องของการที่อาจารย์ที่ปรึกษาต้องทำการตรวจสอบการลงทะเบียนของนักศึกษาซึ่งทำให้นักศึกษาต้องเข้าพบอาจารย์ที่ปรึกษา ทั้ง ๆ ในความเป็นจริงอาจารย์ที่ปรึกษานั้นไม่ได้ตรวจสอบอย่างจริงจัง
2. ผู้ปฏิบัติงานไม่ทราบขั้นตอนการปฏิบัติงานอย่างชัดเจน ทำให้ต้องมีการปรับแก้เงื่อนไขข้อมูลที่ได้มาบางอย่าง นั้นหมายถึงการปรับแก้เรื่องของความต้องการของผู้ใช้ ซึ่งส่งผลกระทบต่อระบบที่ได้วิเคราะห์ไว้แล้ว ต้องทำการวิเคราะห์เปลี่ยนแปลงอีกครั้ง
3. การวิเคราะห์ระบบนั้นต้องพิจารณาทั้งระบบเดิมและระบบที่พัฒนามาแล้ว เพื่อเปรียบเทียบหาข้อมูลซึ่งมีรายละเอียดมาก ทำให้การดำเนินงานในช่วงการวิเคราะห์และออกแบบระบบเพื่อหาความต้องการที่เพิ่มเติมและที่ต้องแก้ไขนั้นต้องใช้ระยะเวลาาน นอกจากนั้นการที่ต้องไปสัมภาษณ์ทั้งเจ้าหน้าที่ทะเบียนและผู้พัฒนาระบบเดิมนั้นต้องใช้เวลาและต้องวิเคราะห์สกัดหาความต้องการสุทธิออกมาให้ได้
4. เครื่องมือที่ใช้ ในการเขียนโปรแกรมไม่ว่าจะเป็น Rational Rose , JBuilder หรือ IBM DB2 ล้วนต้องการทรัพยากรของเครื่องคอมพิวเตอร์ที่ใช้สูงทำให้การทำงานเป็นไปอย่างเชื่องช้า

### 8.4 ข้อเสนอแนะสำหรับระบบทะเบียนนักศึกษา ของบัณฑิตวิทยาลัย

1. งานในระบบทะเบียนนักศึกษาของบัณฑิตวิทยาลัย นั้นมีความซับซ้อนและมีรายละเอียดเฉพาะจึงเป็นการถูกต้องแล้วที่แยกออกมาทำงานเฉพาะจากงานทะเบียนของนักศึกษาระดับปริญญาตรี แต่สิ่งที่ควรที่จะต้องทำการปรับปรุงนั้นคือ ควรมีการกระจายงานของฝ่ายทะเบียนของบัณฑิตวิทยาลัยส่วนกลางไปยังส่วนคณะ อย่างเช่นการกรอกตารางเรียนตารางสอน การกรอกเกรด การเปิดปิดรายวิชาซึ่งจะทำให้เกิดความรวดเร็วและคล่องตัวมากกว่า ทั้งยังสามารถจัดการกับข้อผิดพลาดได้โดยตรง เพราะในปัจจุบันมีข้อผิดพลาดบางอย่างเกิดขึ้น อย่างเช่นตารางเรียนผิด ไม่สามารถหาผู้รับผิดชอบได้ เพราะ มีการส่งต่องานจากส่วนคณะเข้ามาในส่วนกลางด้วย
2. ในเรื่องของระบบสารสนเทศนั้น โดยปกติแล้วสำนักวิจัยและบริการคอมพิวเตอร์จะเป็นผู้ให้การสนับสนุนแก่บัณฑิตวิทยาลัย แต่ด้วยความแตกต่างของระบบปริญญาตรี และระดับบัณฑิตศึกษาทำให้เกิดปัญหาหลายอย่างขึ้นในส่วนของบัณฑิตวิทยาลัย ซึ่งการแก้ปัญหาจากสำนักวิจัยและบริการคอมทำได้ล่าช้าทางบัณฑิตวิทยาลัยจึงคิดที่จะสร้างระบบสารสนเทศของตัวเองขึ้นมา ซึ่งในมุมมองของผู้ทำการวิจัยนี้เห็นว่างานระบบสารสนเทศนั้นทั้งในด้านของโปรแกรม และฐานข้อมูลยังควรให้สำนักวิจัยรับผิดชอบ แต่ทางสำนักวิจัยควรจัดกำลังคน หรือใช้กำลังคนของทางบัณฑิตวิทยาลัย ขึ้นมาเฉพาะเพื่อดูแลระบบในส่วนนี้ ทั้งนี้เพื่อให้สามารถใช้ทรัพยากรของสำนักวิจัยที่มีอยู่แล้วได้อย่างเต็มที่ ส่วนบุคลากรที่จัดไว้ดูแลเฉพาะนี้ก็จะสามารถจัดการกับปัญหาต่าง ๆ หรือเพิ่มแก้ไขได้ในทันที ซึ่งปัญหาเรื่องของภาษาโปรแกรมที่ใช้ไม่ได้เป็นปัญหาอยู่แล้วเพราะถึงอย่างไรก็สามารถใช้ฐานข้อมูลร่วมกันได้
3. ควรมีการประชุมหารือเพื่อปรับแก้ข้อบังคับต่าง ๆ หาข้อดีข้อเสียของเงื่อนไขแต่ละข้ออย่างชัดเจน รวมทั้งให้แต่ละคณะมีกฎระเบียบไปในแนวทางเดียวกัน หรือถ้าไม่สามารถทำให้กฎระเบียบเหมือนกันได้ก็

เอกสารแห่งนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามเผยแพร่โดยไม่ได้รับอนุญาต หากฝ่าฝืนจะดำเนินการตามกฎหมายอย่างเต็มที่

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควรระบุถึงข้อแตกต่างนั้นอย่างชัดเจนให้เจ้าหน้าที่ผู้ปฏิบัติงานได้ทราบ นอกจากกฎระเบียบแล้วยังมีแนวทางต่าง ๆ เช่นการเขียนหลักสูตรควรเขียนให้มีความเป็นมาตรฐานมีรายละเอียดครบถ้วน อย่างเช่นทุกหลักสูตรควรมีการกำหนดแผนการศึกษา(Course Work) การกำหนดรหัสวิชาที่ไม่ควรซ้ำกัน ควรใช้รหัสวิชาที่เป็นตัวเลขทั้งหมด

4. นอกจากเรื่องของกฎระเบียบแล้ว การปฏิบัติตามก็ควรทำอย่างเคร่งครัดด้วย อย่างเช่นเรื่องกำหนดการส่งเกรดของอาจารย์ควรจะให้ส่งตรงตามกำหนดเวลา

### 8.5 ข้อเสนอแนะในการนำไปพัฒนาต่อ

1. เนื่องจากความต้องการของผู้ใช้มีการเปลี่ยนแปลง จึงควรมีการปรับปรุงระบบให้ตรงตามความต้องการของผู้ใช้
2. หาจุดบกพร่องต่างๆที่เกิดขึ้นใน โปรแกรมและทำการปรับปรุงแก้ไขเพิ่มเติม
3. นำเอาเทคนิคใหม่ เข้ามาใช้ เพื่อทำให้ระบบมีความสะดวกมากขึ้น เช่นการจัดการพิมพ์รายงานอาจใช้เทคนิคการทำ Scalable Vector Graphics (SVG) เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

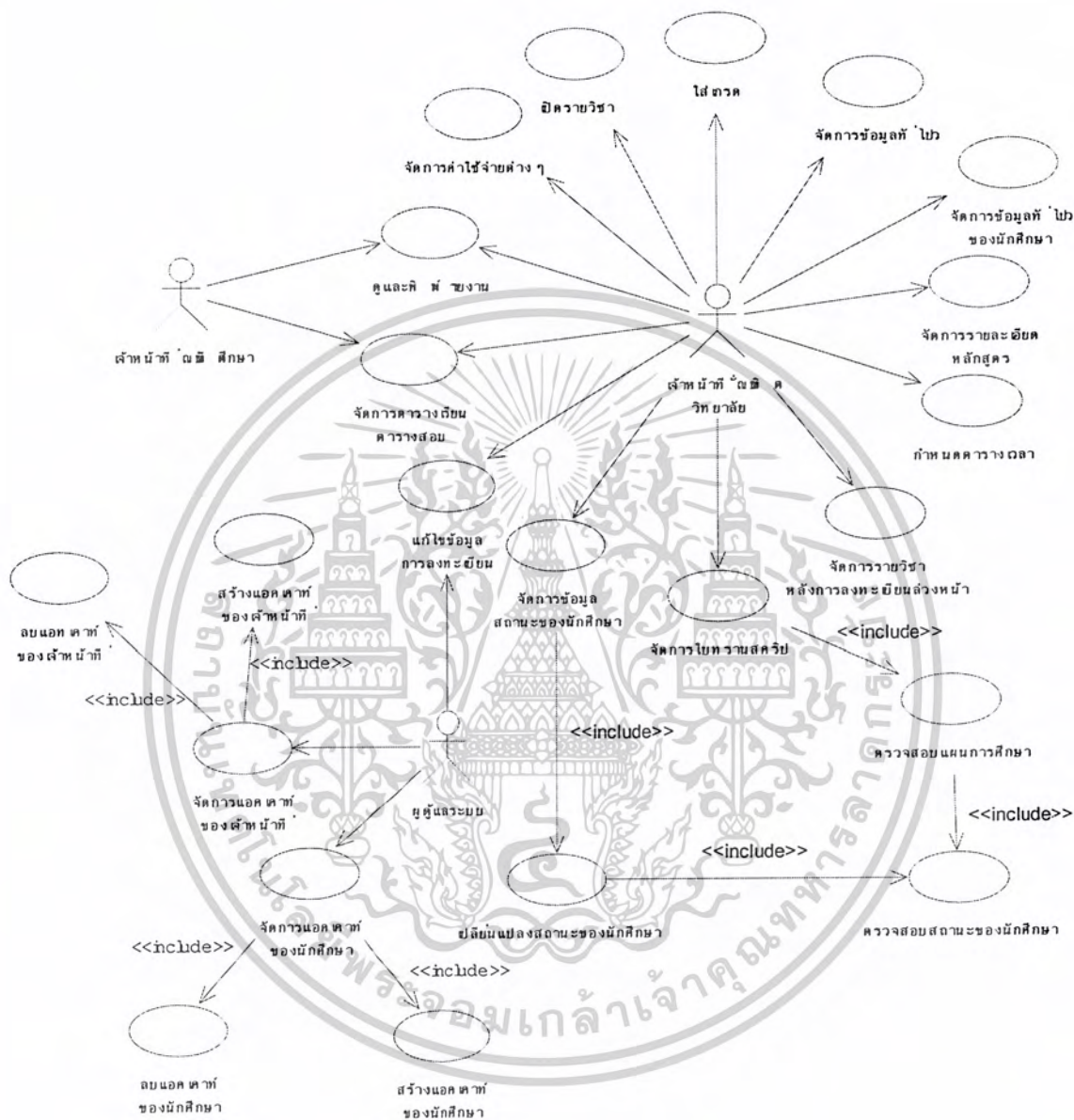


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Use Case Diagram ในส่วนของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Use Case Diagram ในส่วนของเจ้าหน้าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Activity Diagram ของ การลงทะเบียนล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### Activity Diagram ของ การลงทะเบียนปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



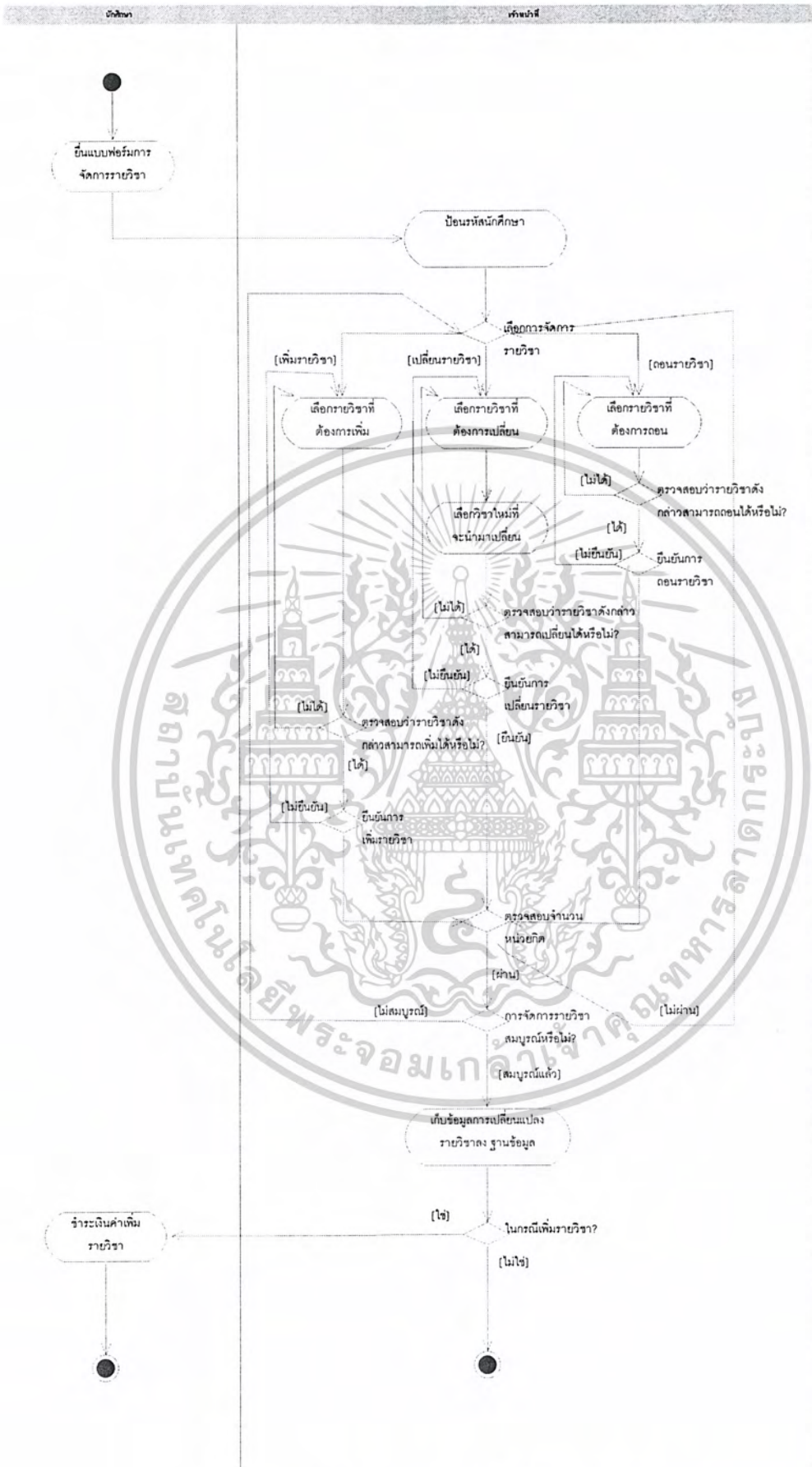
**Activity Diagram ของ การลงทะเบียนรัชสาภาพ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



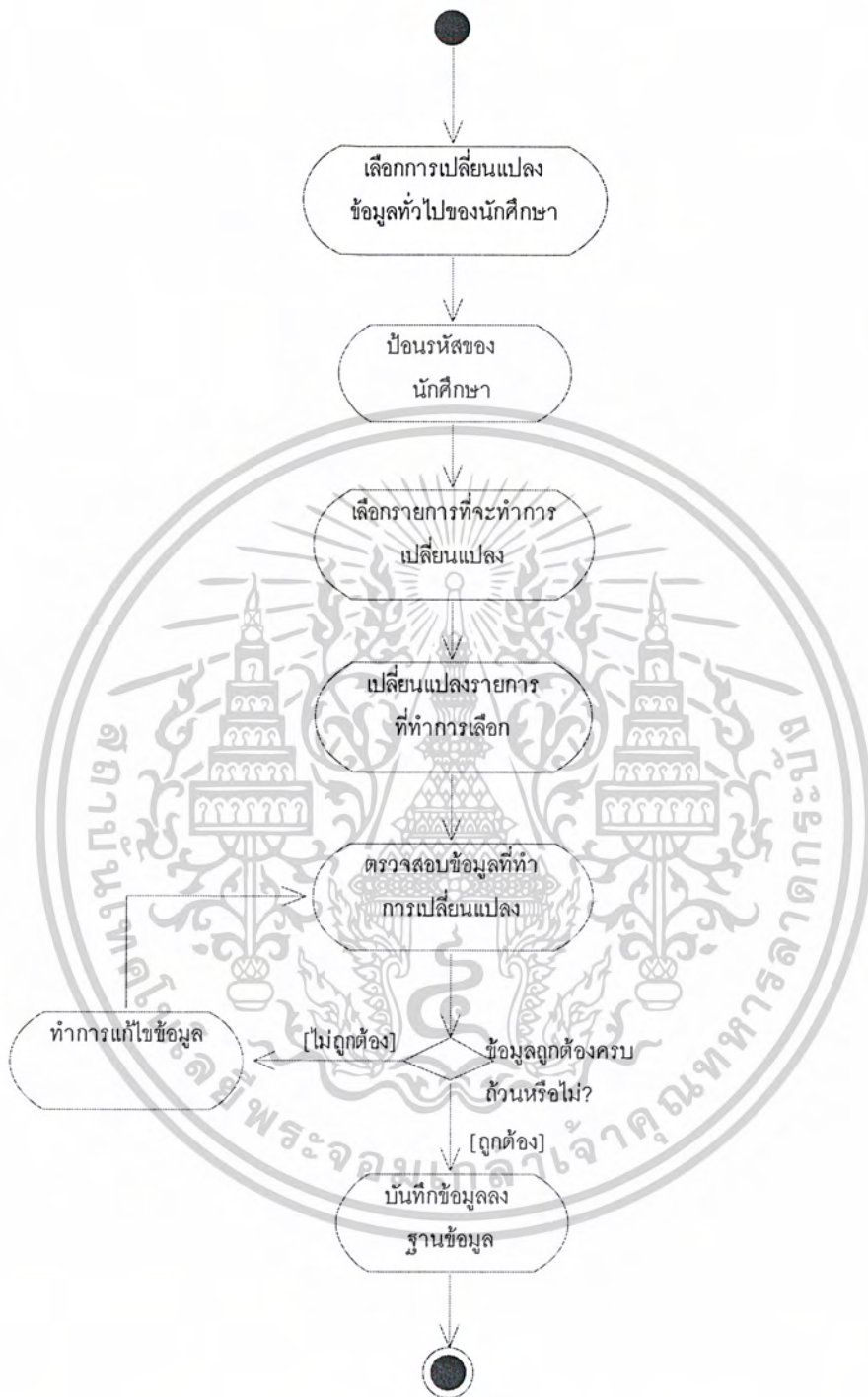
*Activity Diagram* ของ การจัดการรายวิชาหลังจากการลงทะเบียนล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



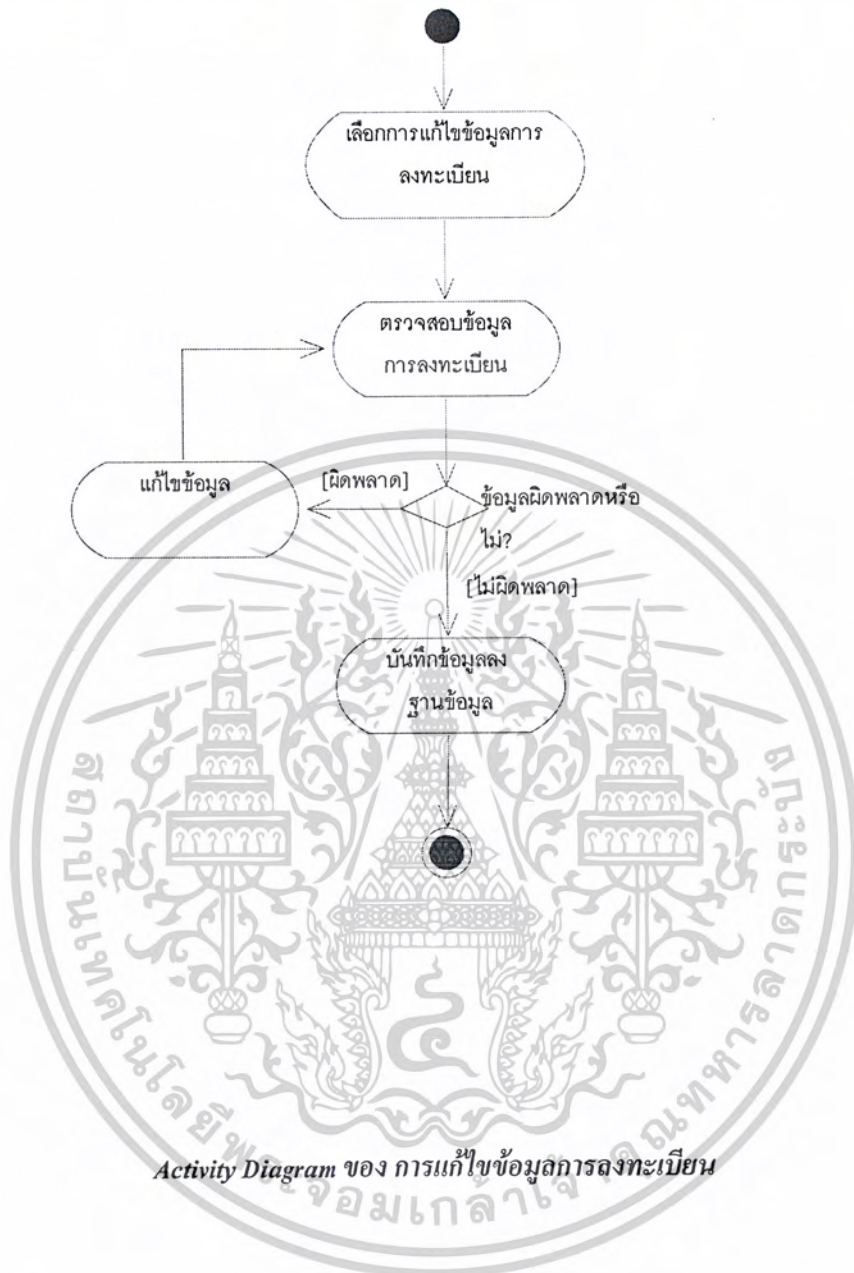
**Activity Diagram ของ การจัดการรายวิชา**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

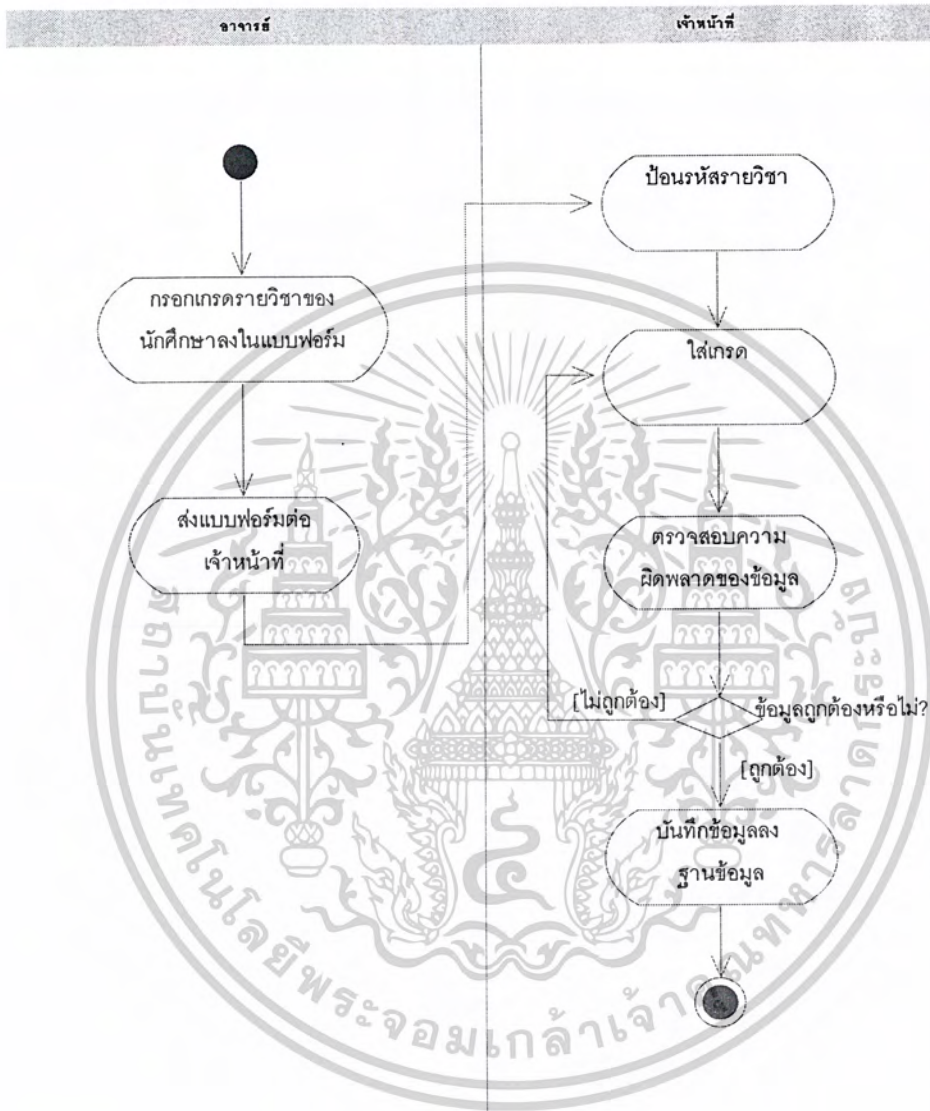


*Activity Diagram* ของ การเปลี่ยนแปลงข้อมูลทั่วไปของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

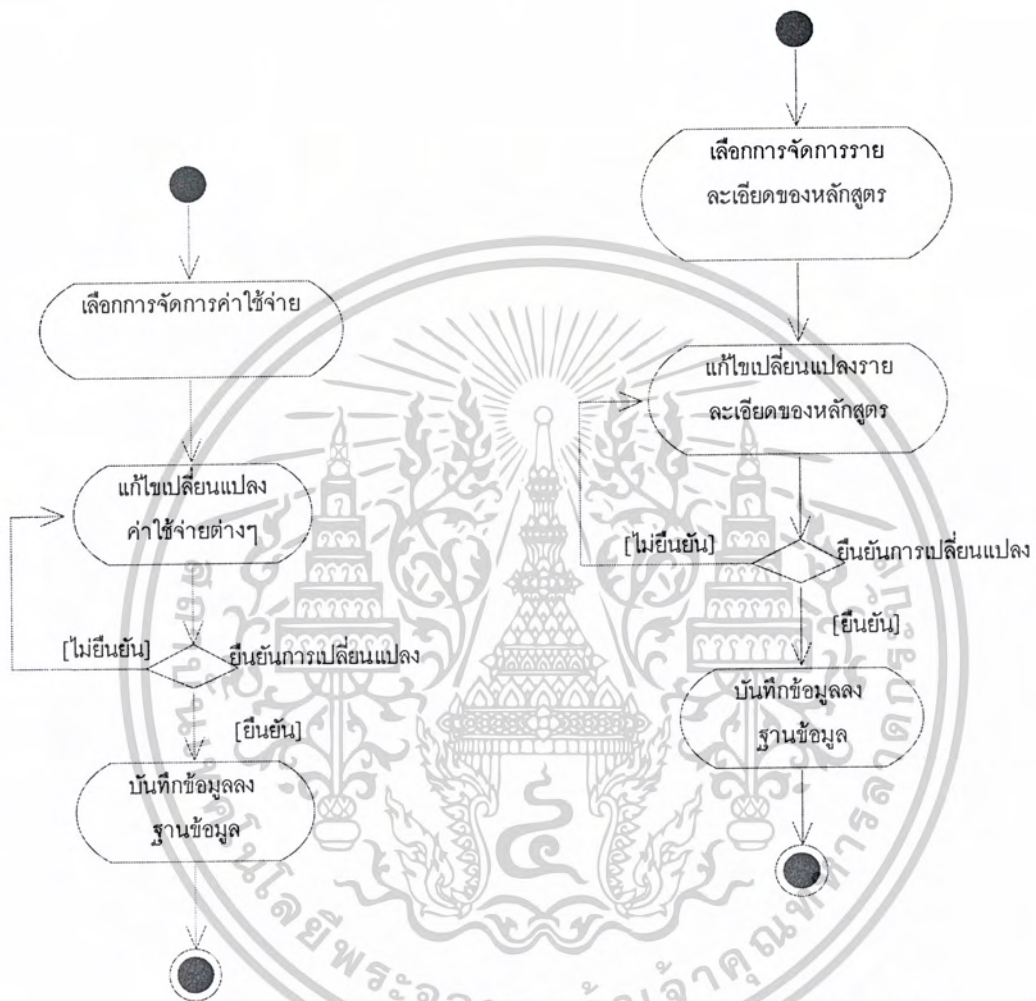


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Activity Diagram ของ การใส่เกรด

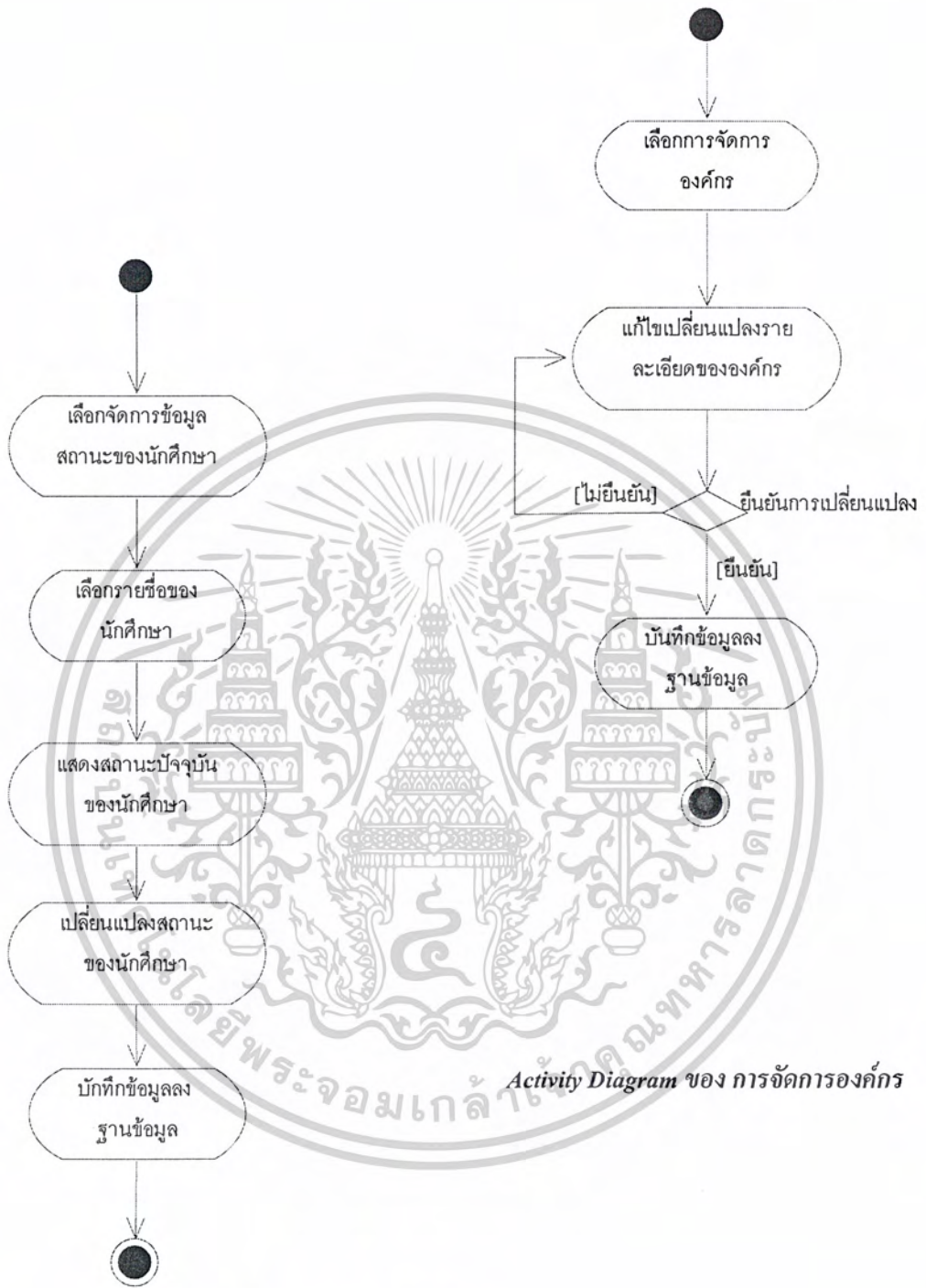
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Activity Diagram ของ การจัดการรายละเอียดของหลักสูตร

Activity Diagram ของ การจัดการค่าใช้จ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Activity Diagram ของ การจัดการองค์กร

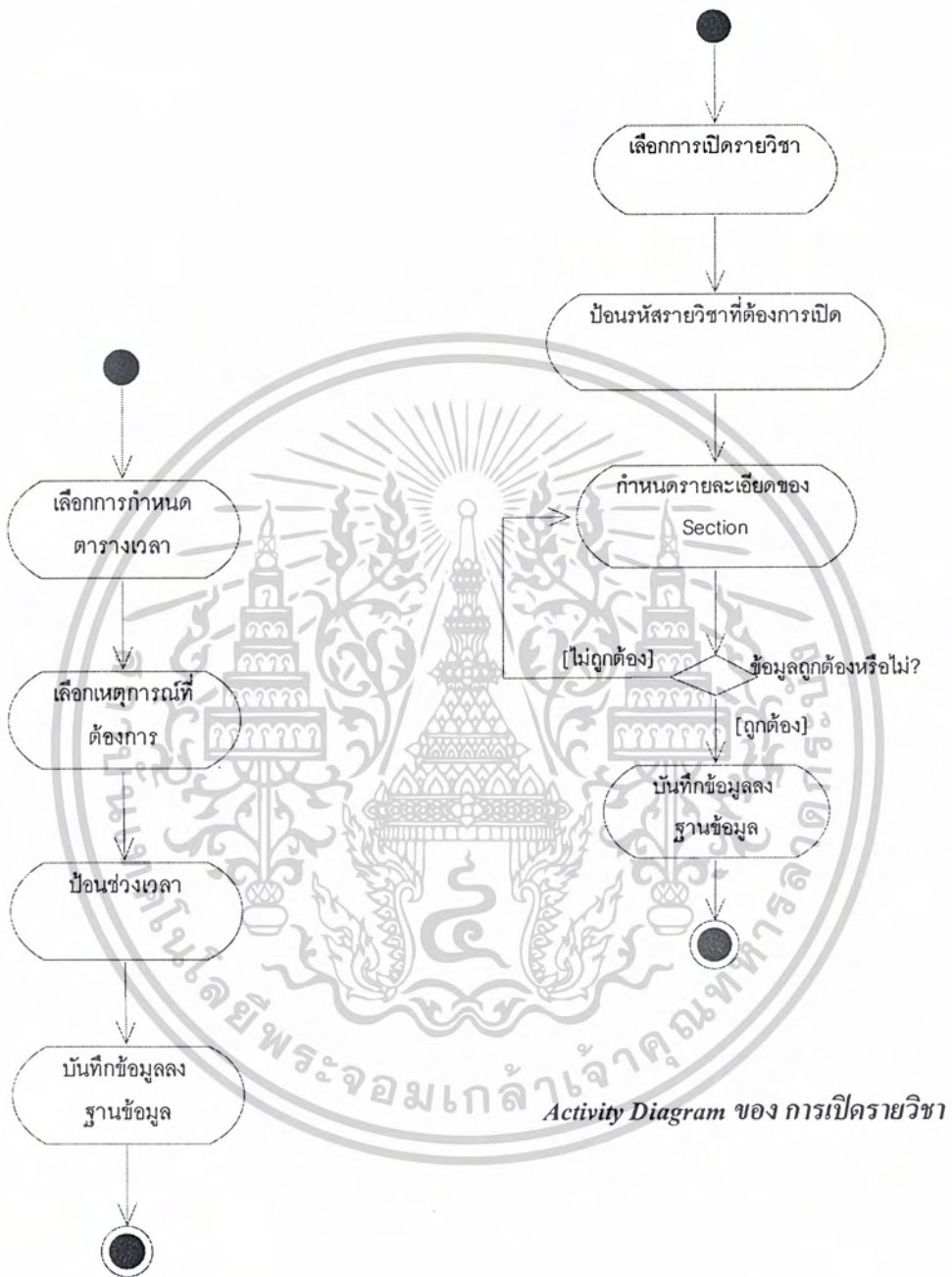
Activity Diagram ของ การจัดการข้อมูลสถานะของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



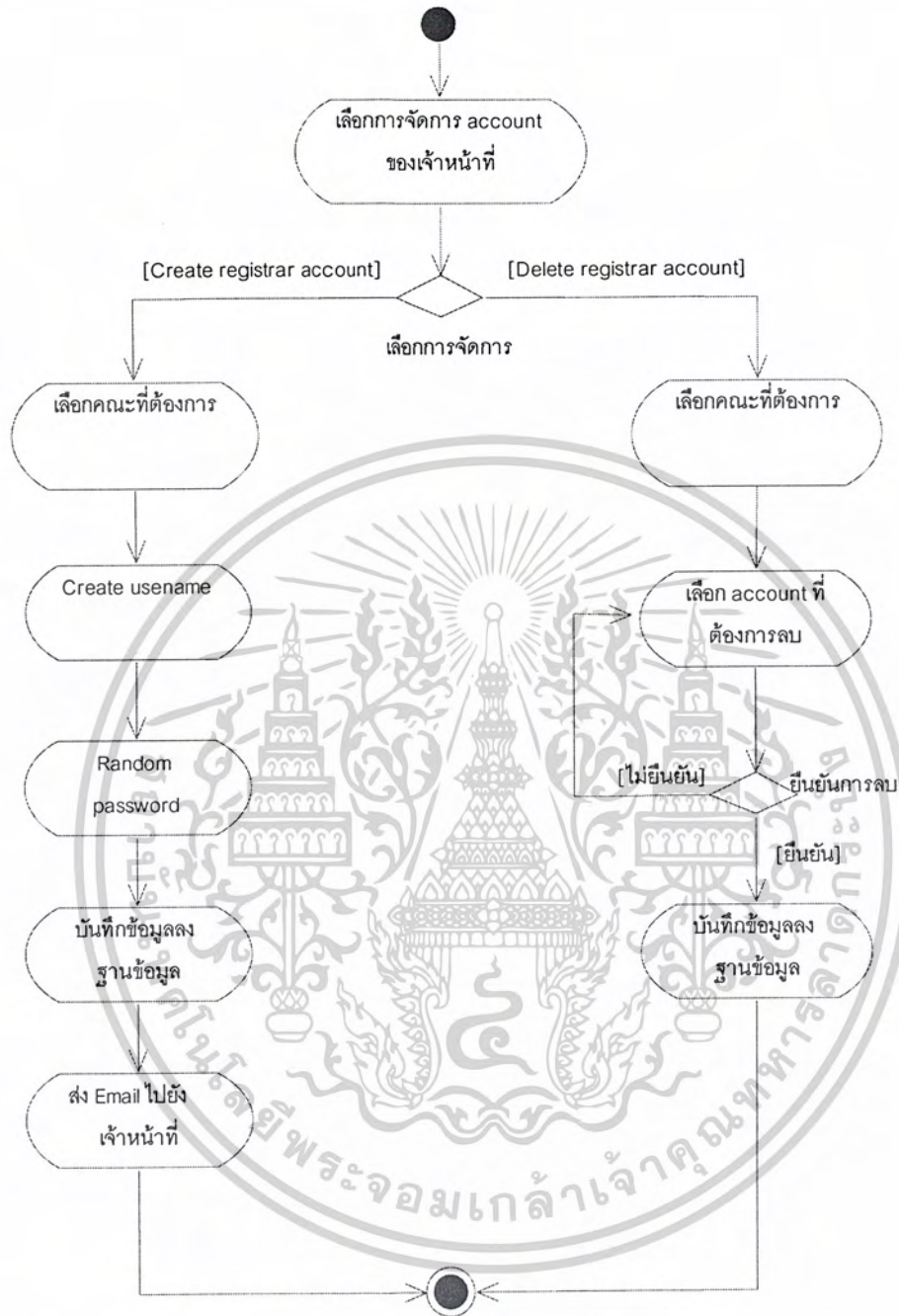
*Activity Diagram ของ การจัดการใบ transcript*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



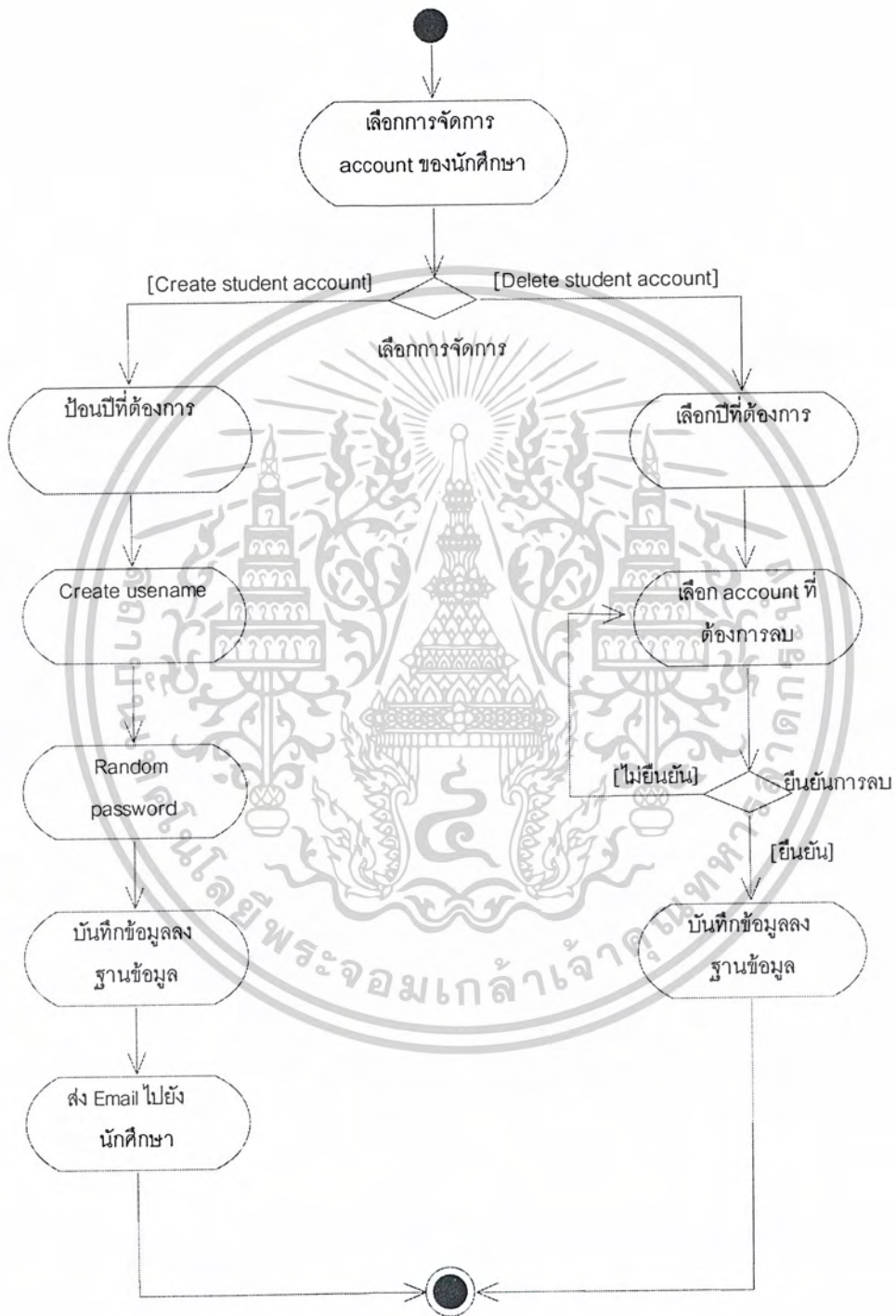
Activity Diagram ของ การกำหนดตารางเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



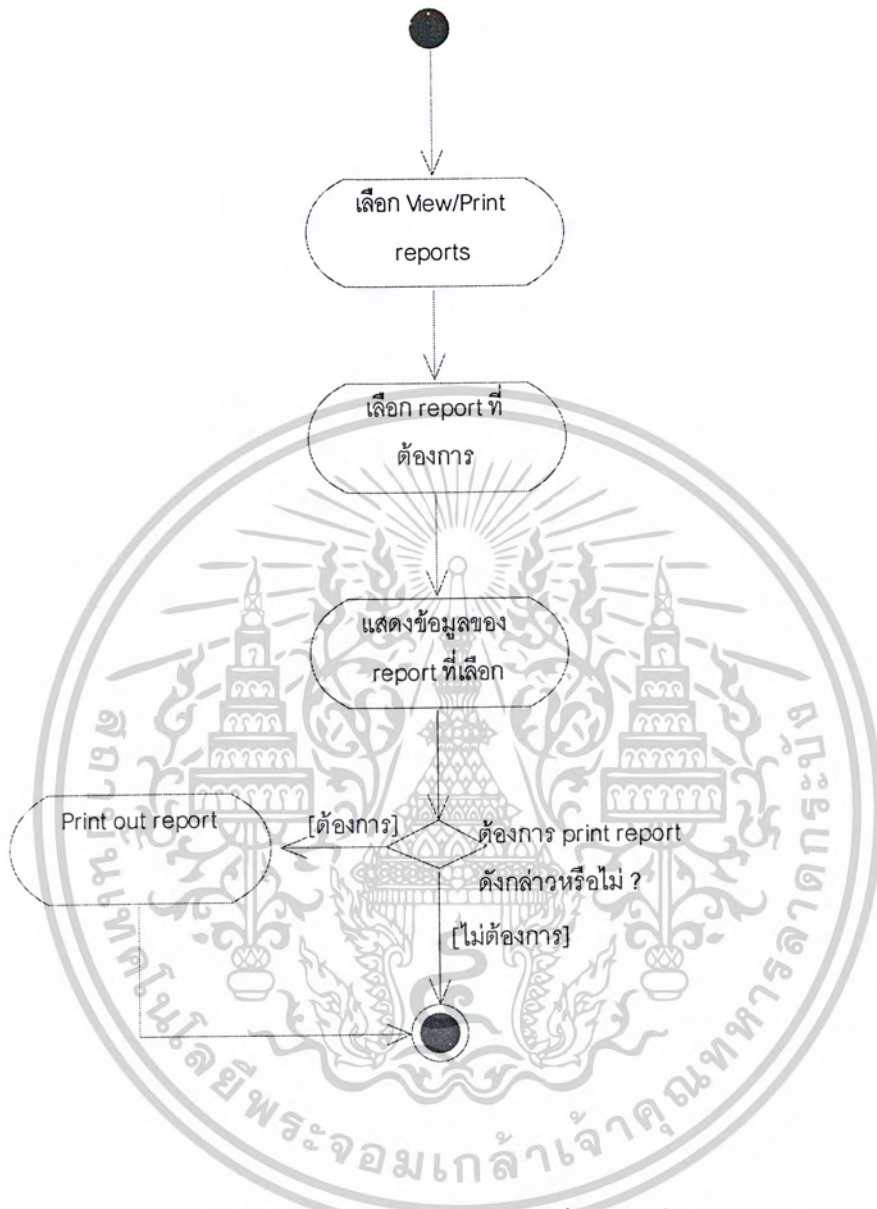
**Activity Diagram ของ การจัดการ account ของเจ้าหน้าที่**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**Activity Diagram ของ การจัดการ account ของนักศึกษา**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



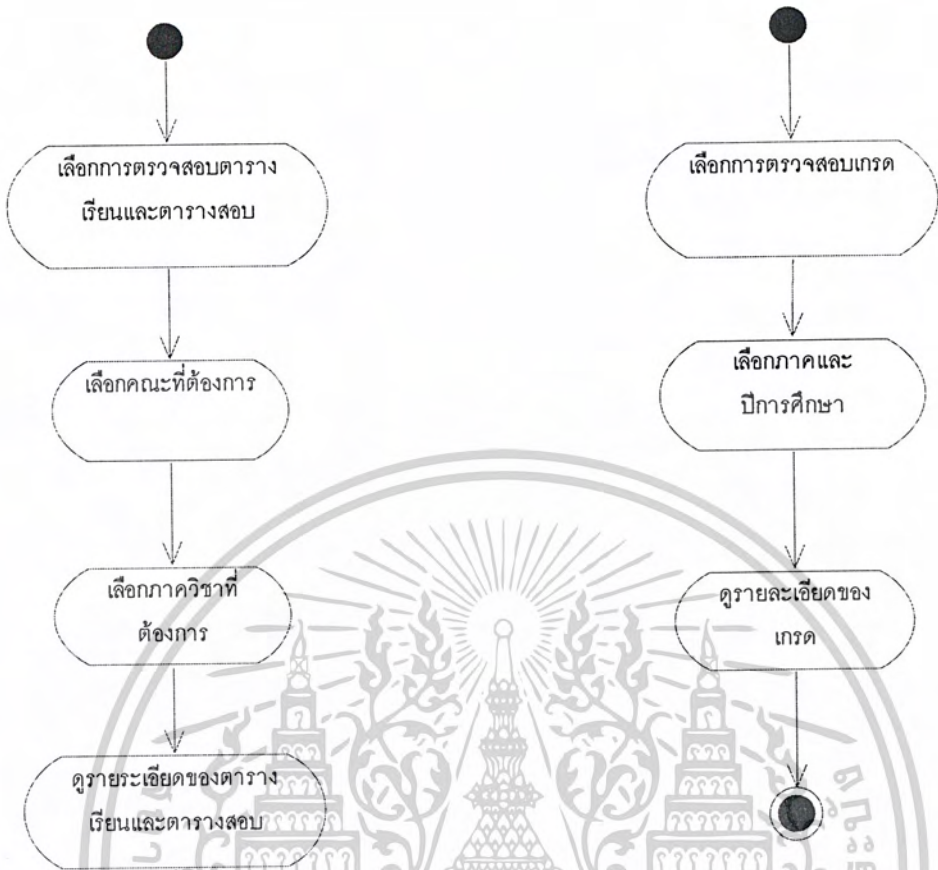
Activity Diagram ของ ดูและพิมพ์รายงานต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Activity Diagram ของ ยื่นแบบฟอร์มคำร้องต่างๆ

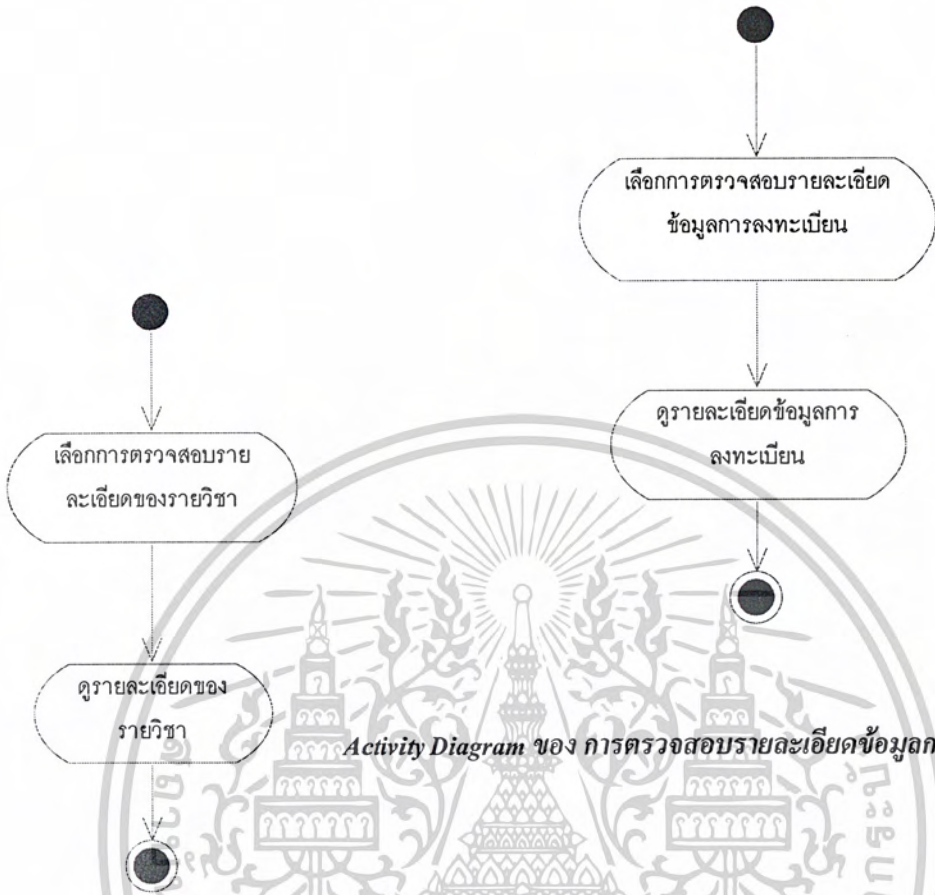
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



*Activity Diagram* ของ การตรวจสอบเกรด

*Activity Diagram* ของ การตรวจสอบตารางเรียนและตารางสอบ

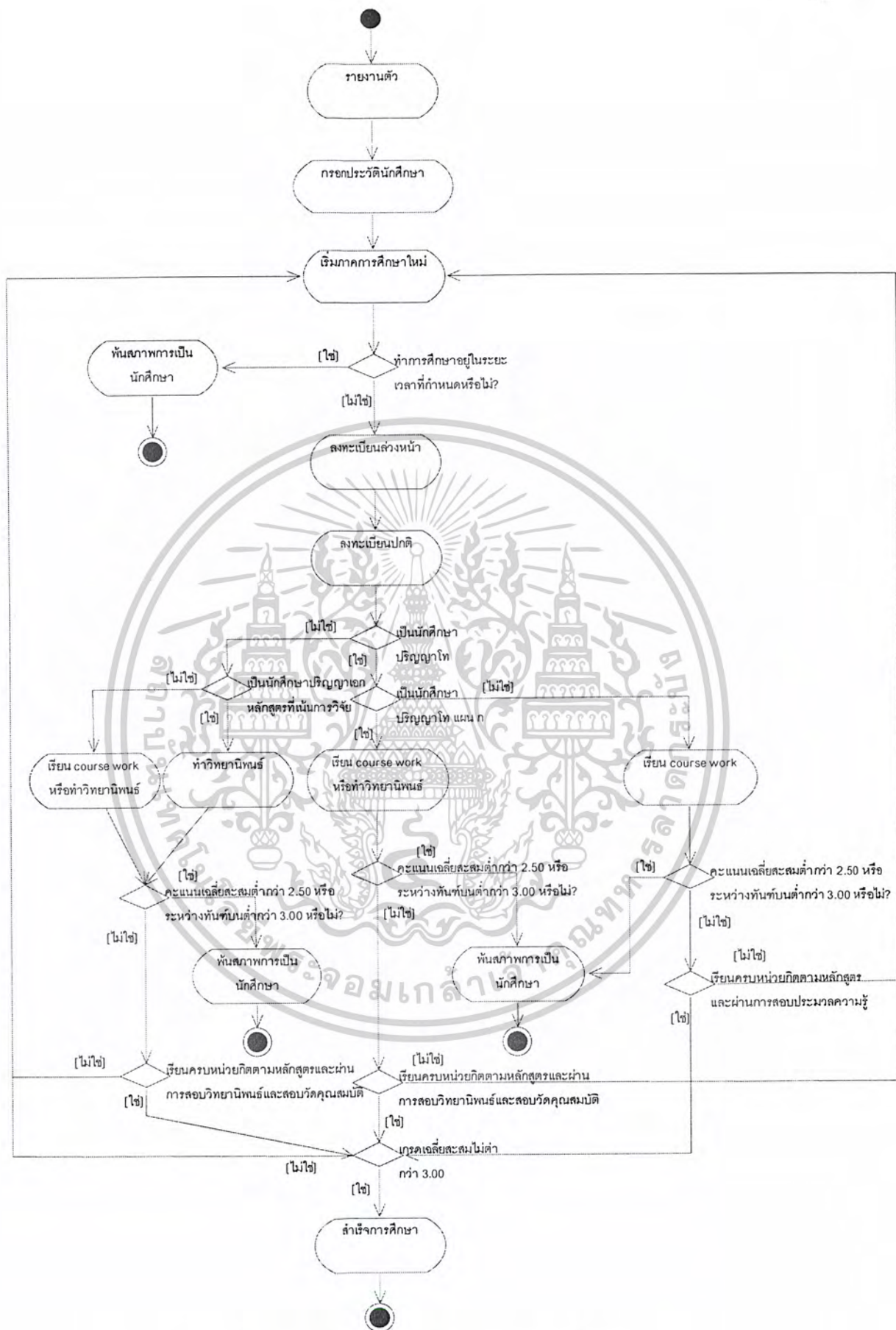
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Activity Diagram ของ การตรวจสอบรายละเอียดข้อมูลการลงทะเบียน

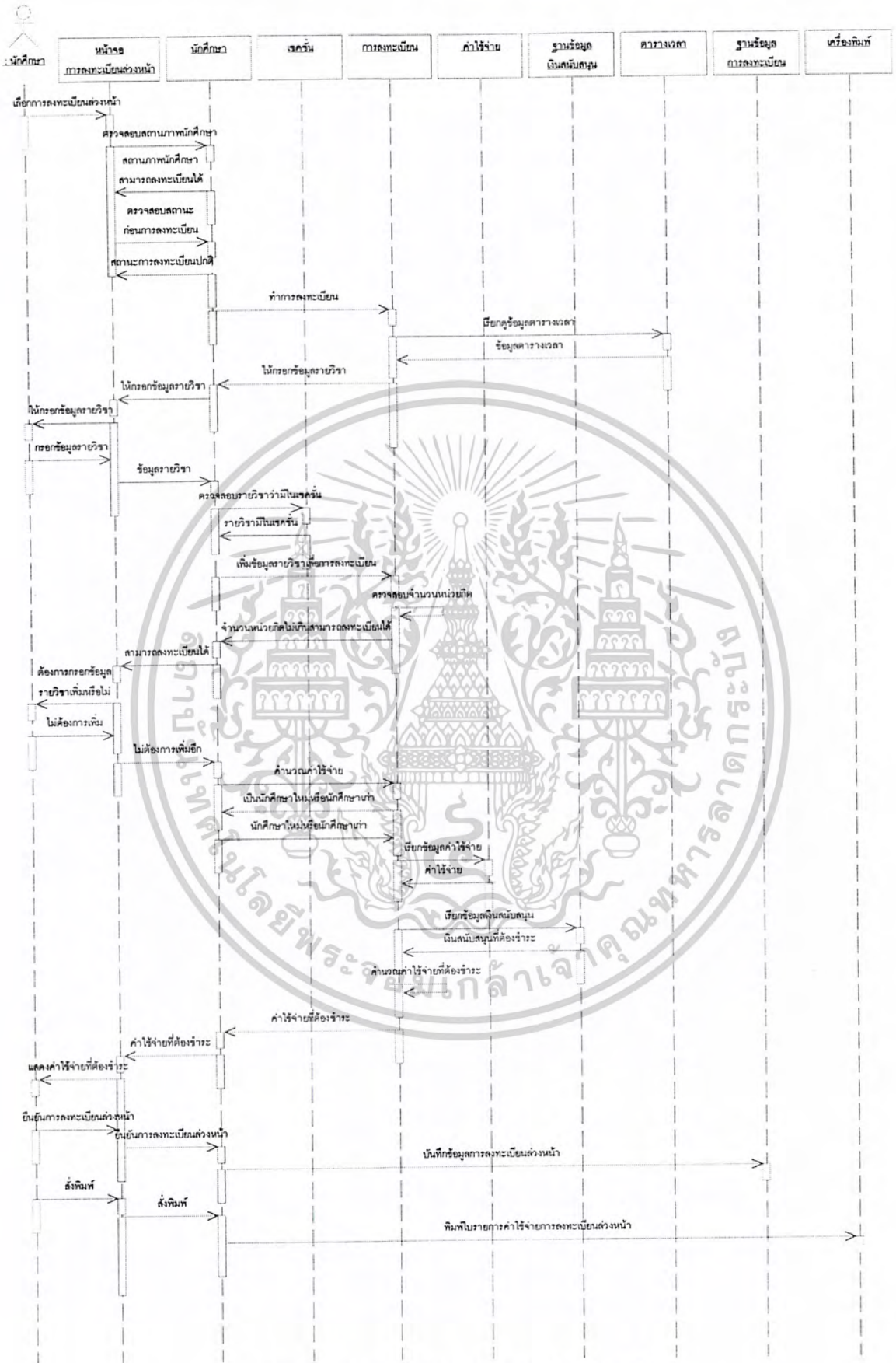
Activity Diagram ของ การตรวจสอบรายละเอียดของรายวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



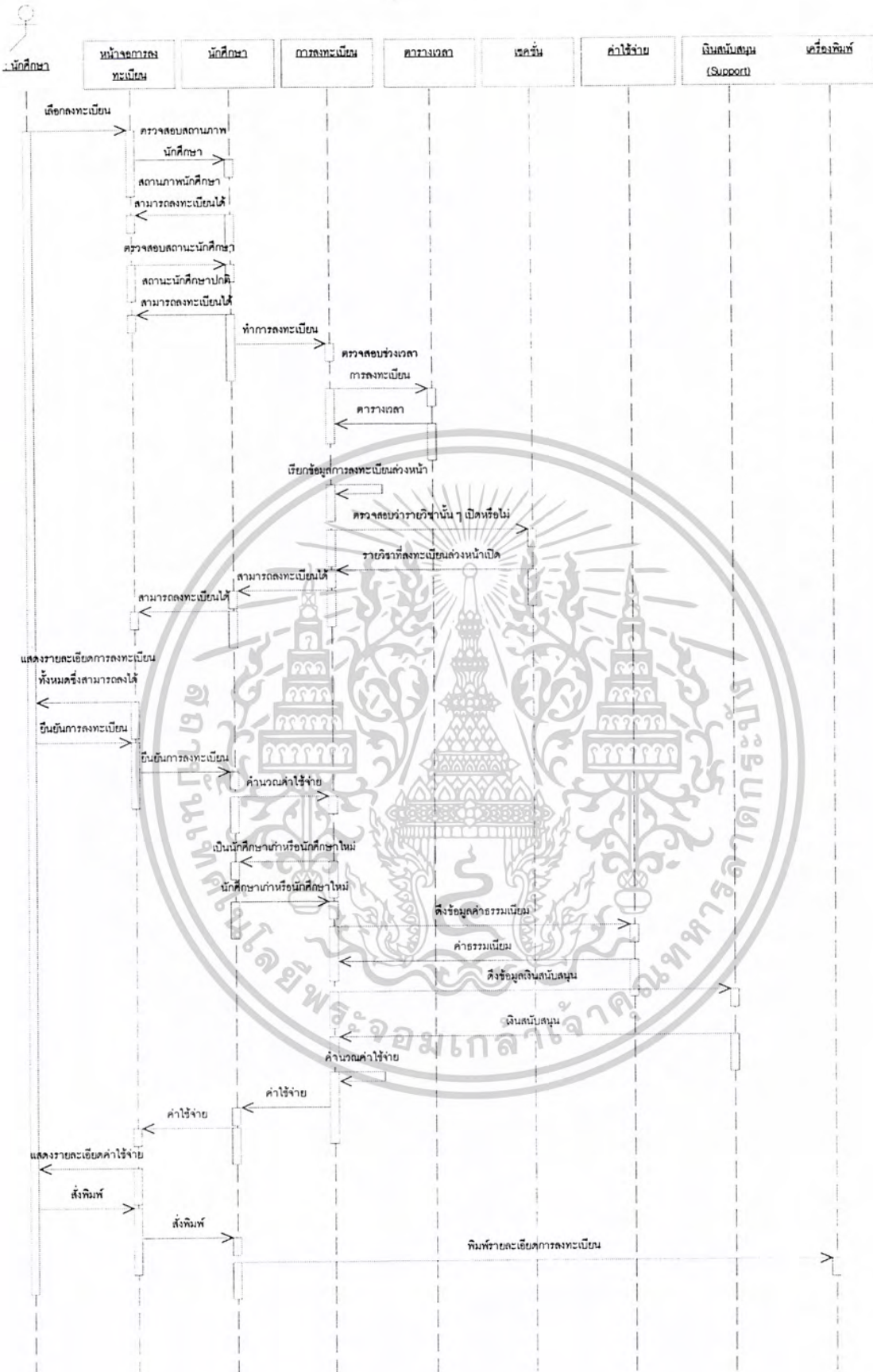
Activity Diagram ของ life cycle ของ การลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



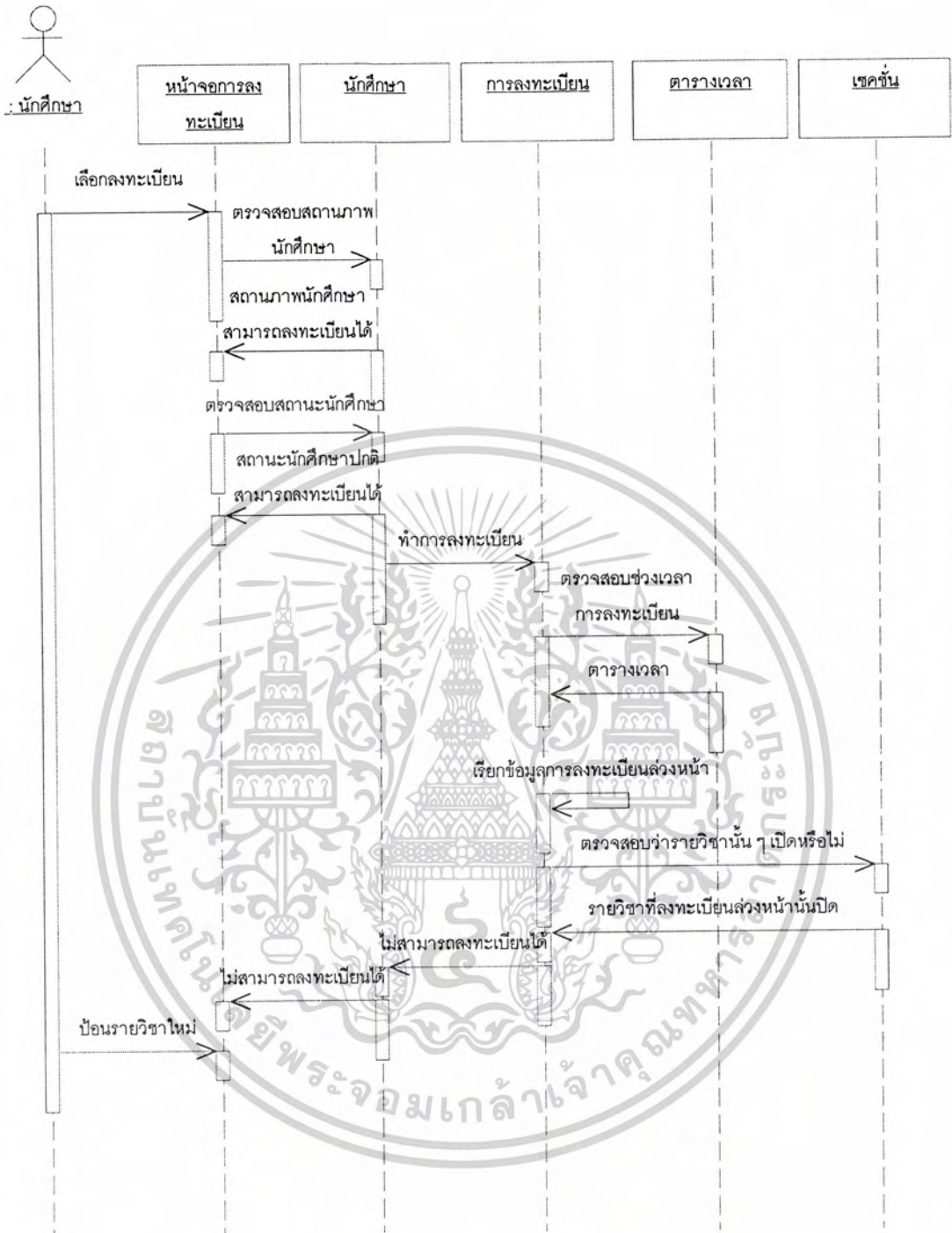
**Sequence Diagram ของ การลงทะเบียนล่วงหน้า**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



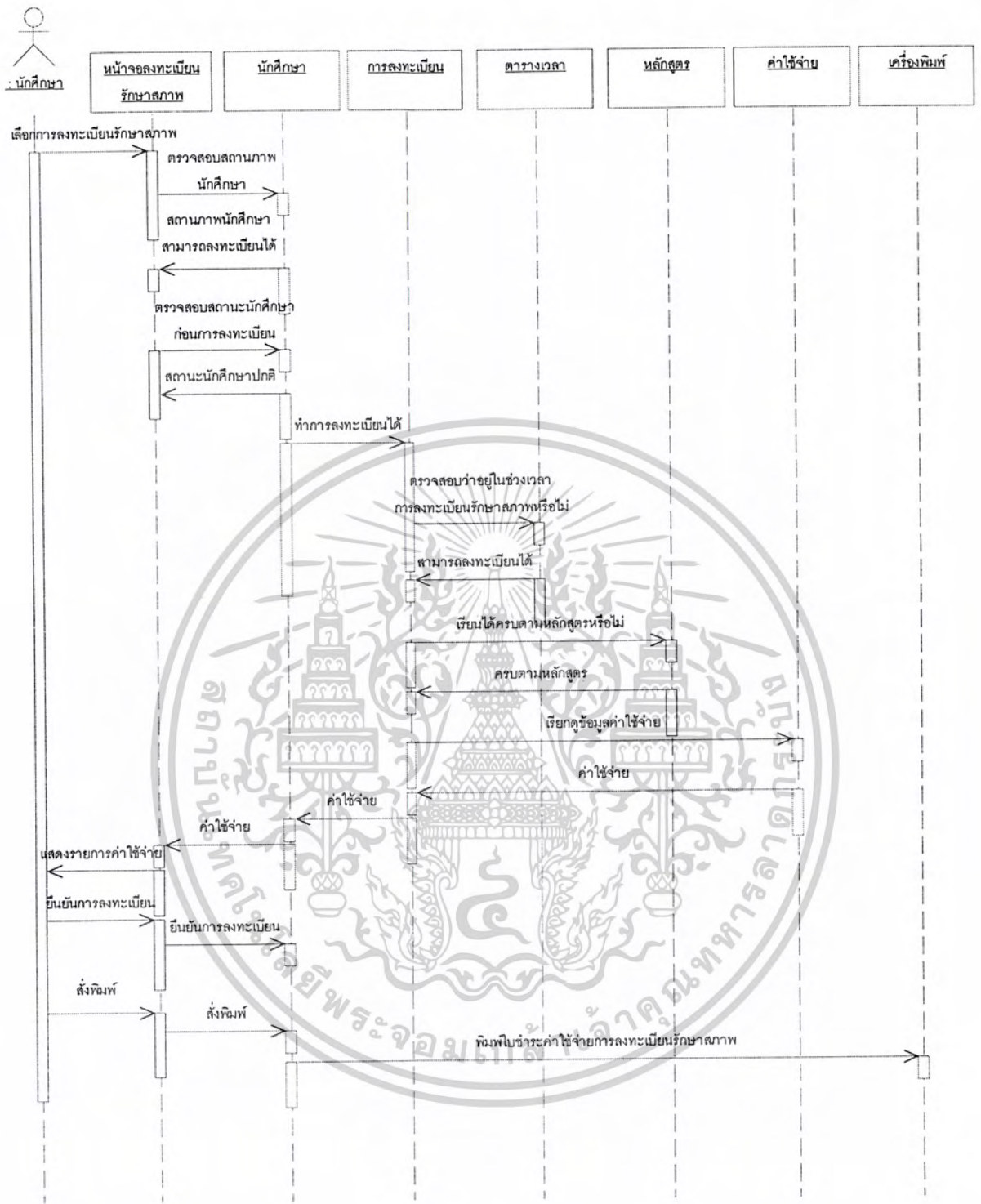
**Sequence Diagram ของ การ ลงทะเบียนปกติในกรณีที่ไม่มืวิชาถูกปิด**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



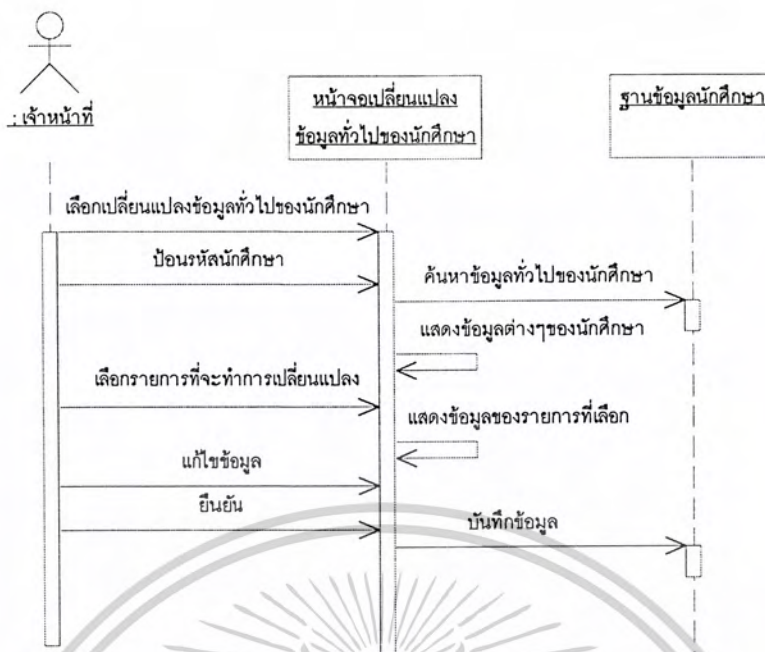
**Sequence Diagram ของ การ ลงทะเบียนปกติในกรณีที่มิวิชาถูกปิด**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

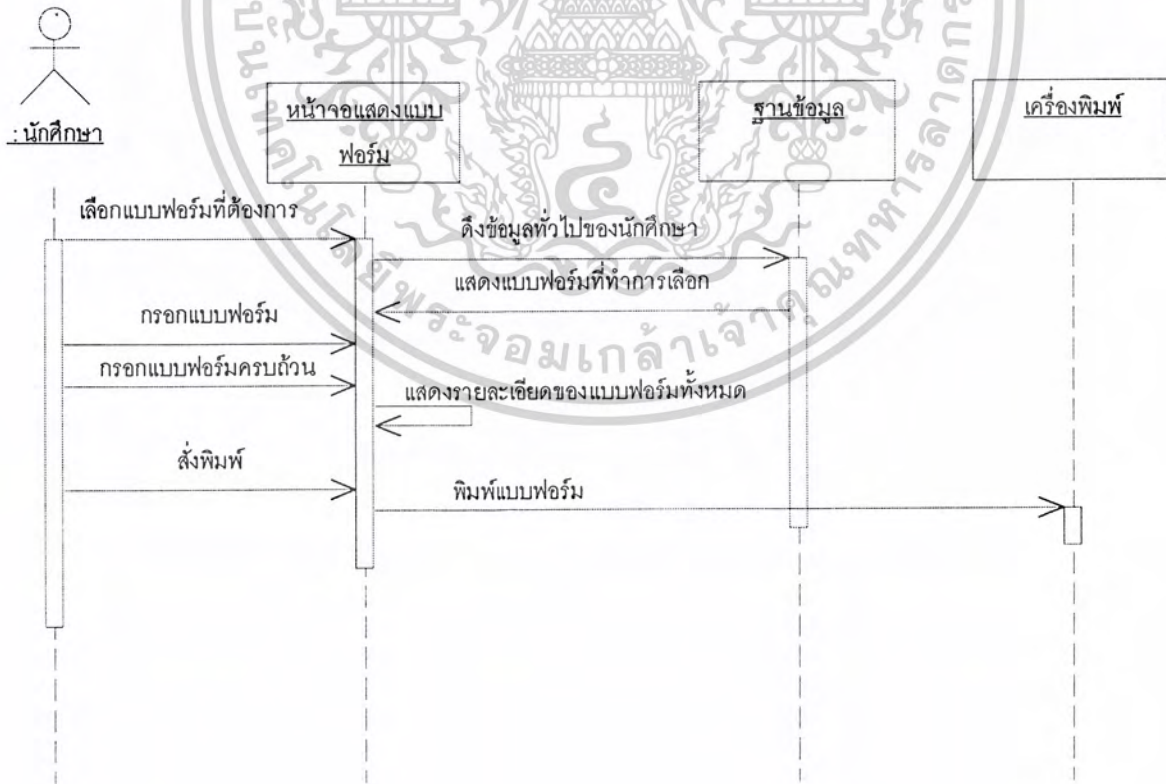


Sequence Diagram ของ การ ลงทะเบียนรักษาสภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

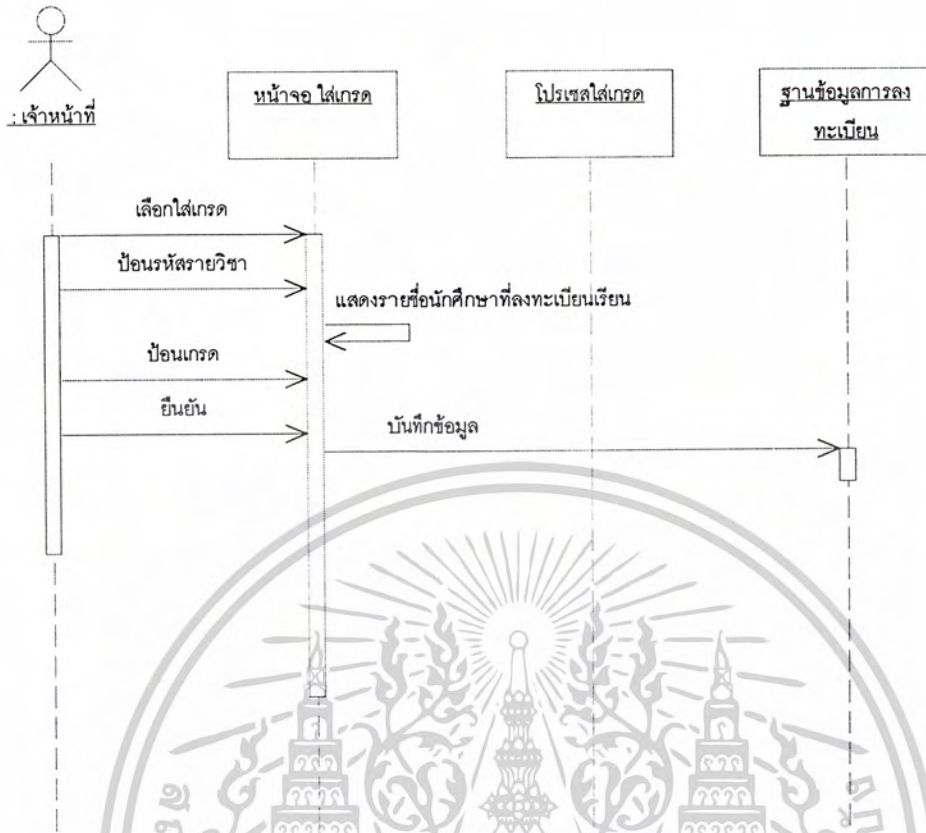


Sequence Diagram ของ การ เปลี่ยนแปลงข้อมูลทั่วไปของนักศึกษา

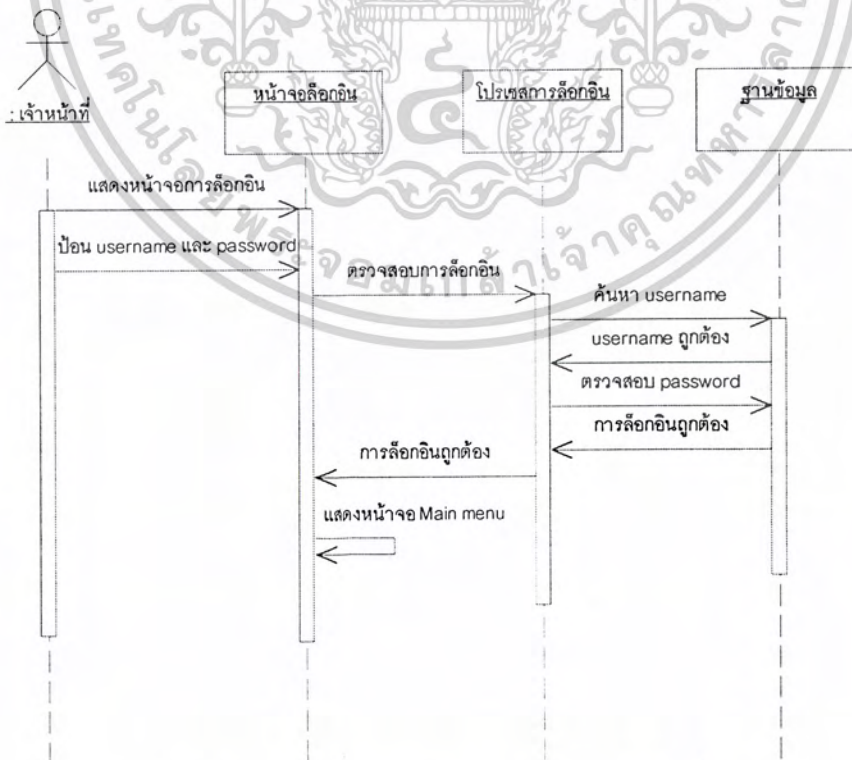


Sequence Diagram ของ การยื่นแบบฟอร์มคำร้องต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

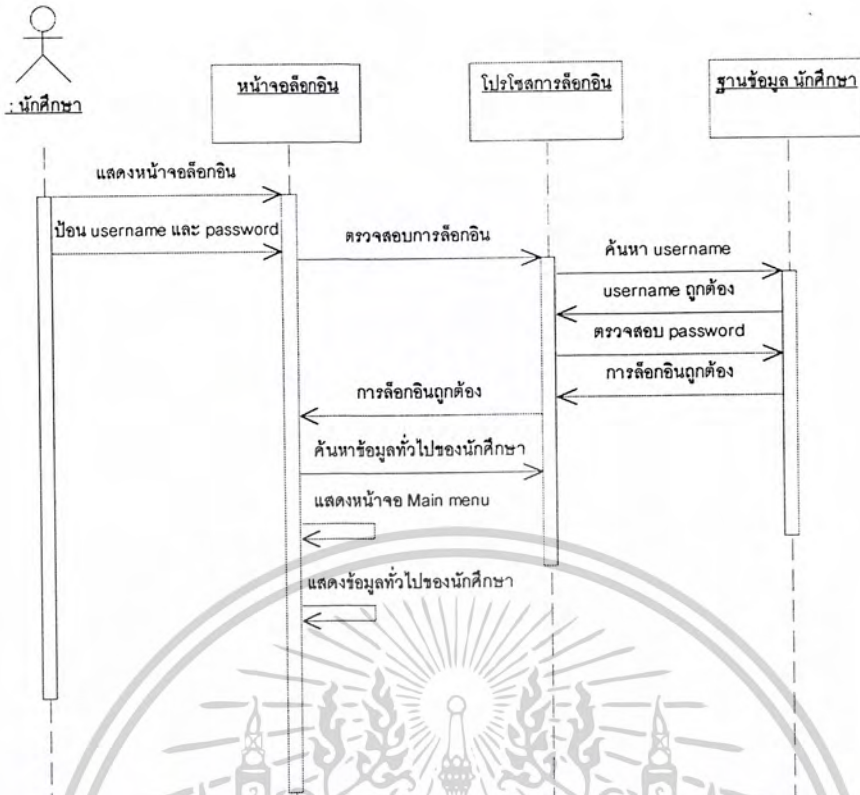


Sequence Diagram ของ การ ใ้กรวด

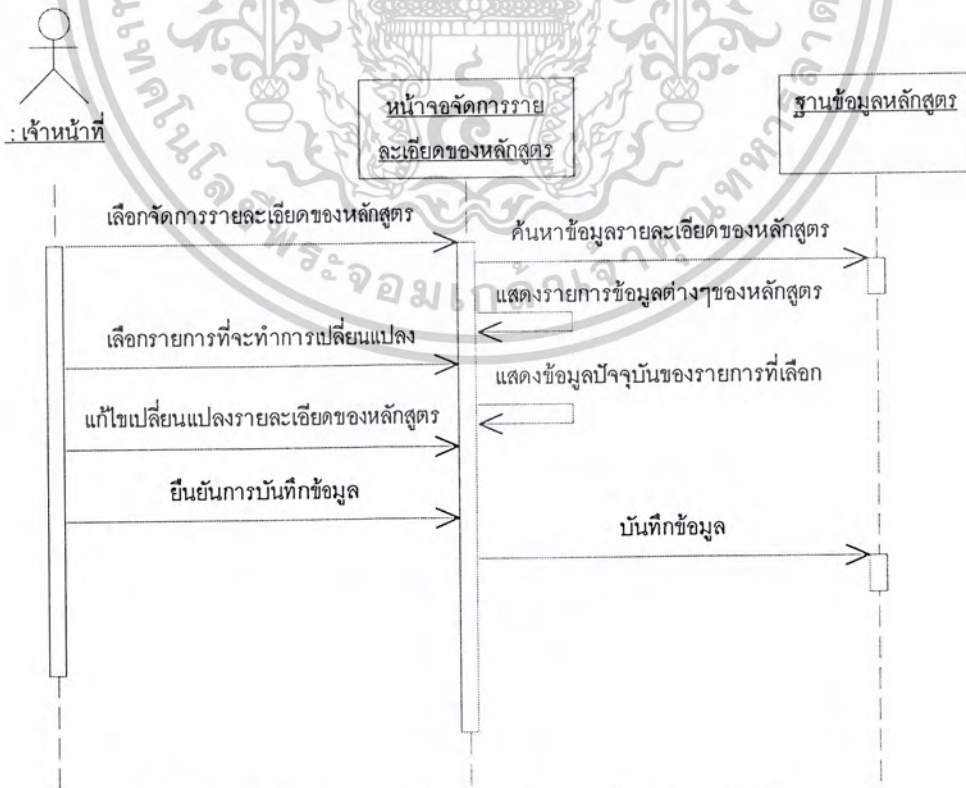


Sequence Diagram ของ การ ล็อกอินของเจ้าหน้าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

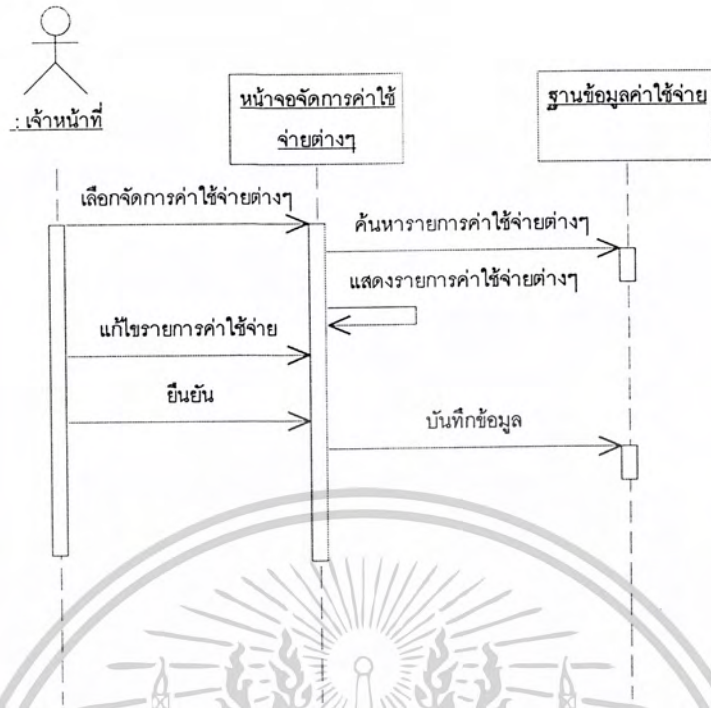


Sequence Diagram ของ การ ล็อกอินของนักศึกษา



Sequence Diagram ของ การ จัดการรายละเอียดของหลักสูตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

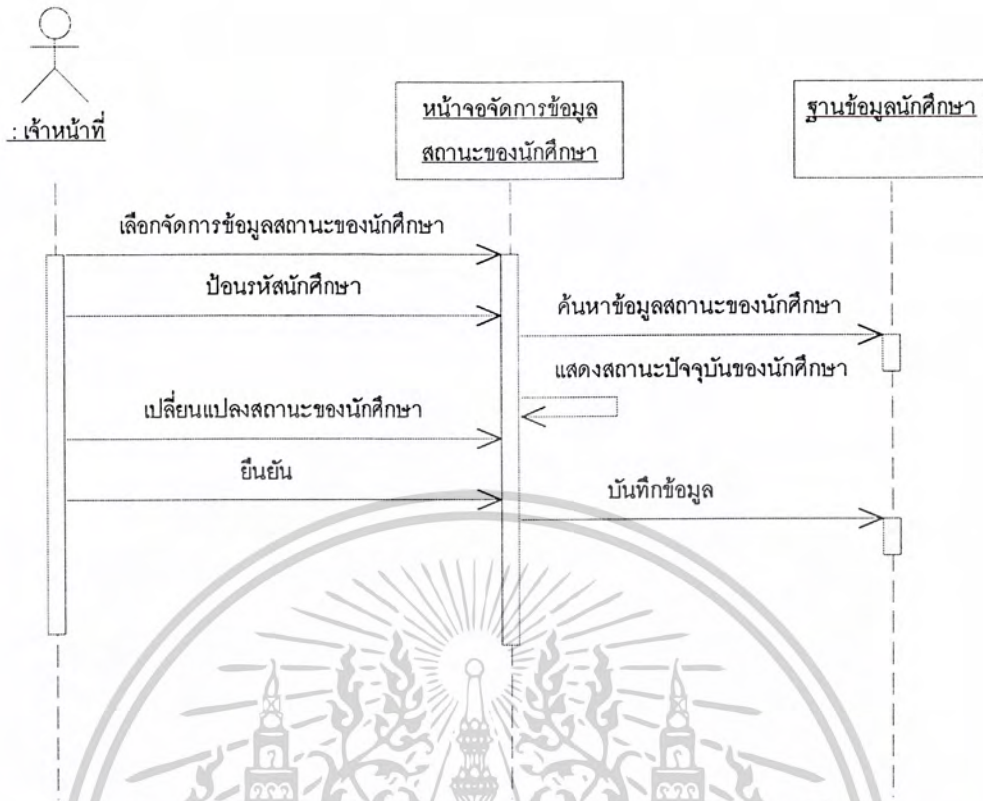


Sequence Diagram ของ การ จัดการค่าใช้จ่ายต่างๆ

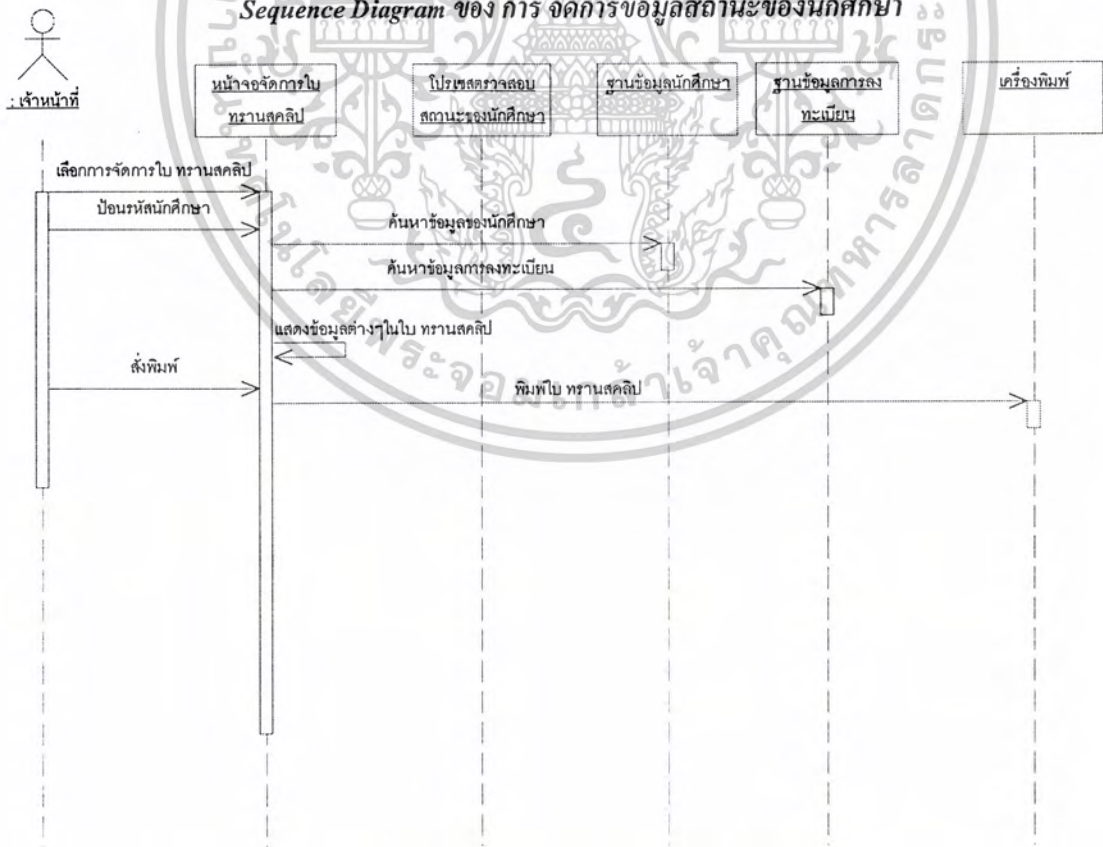


Sequence Diagram ของ การ จัดการข้อมูลองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

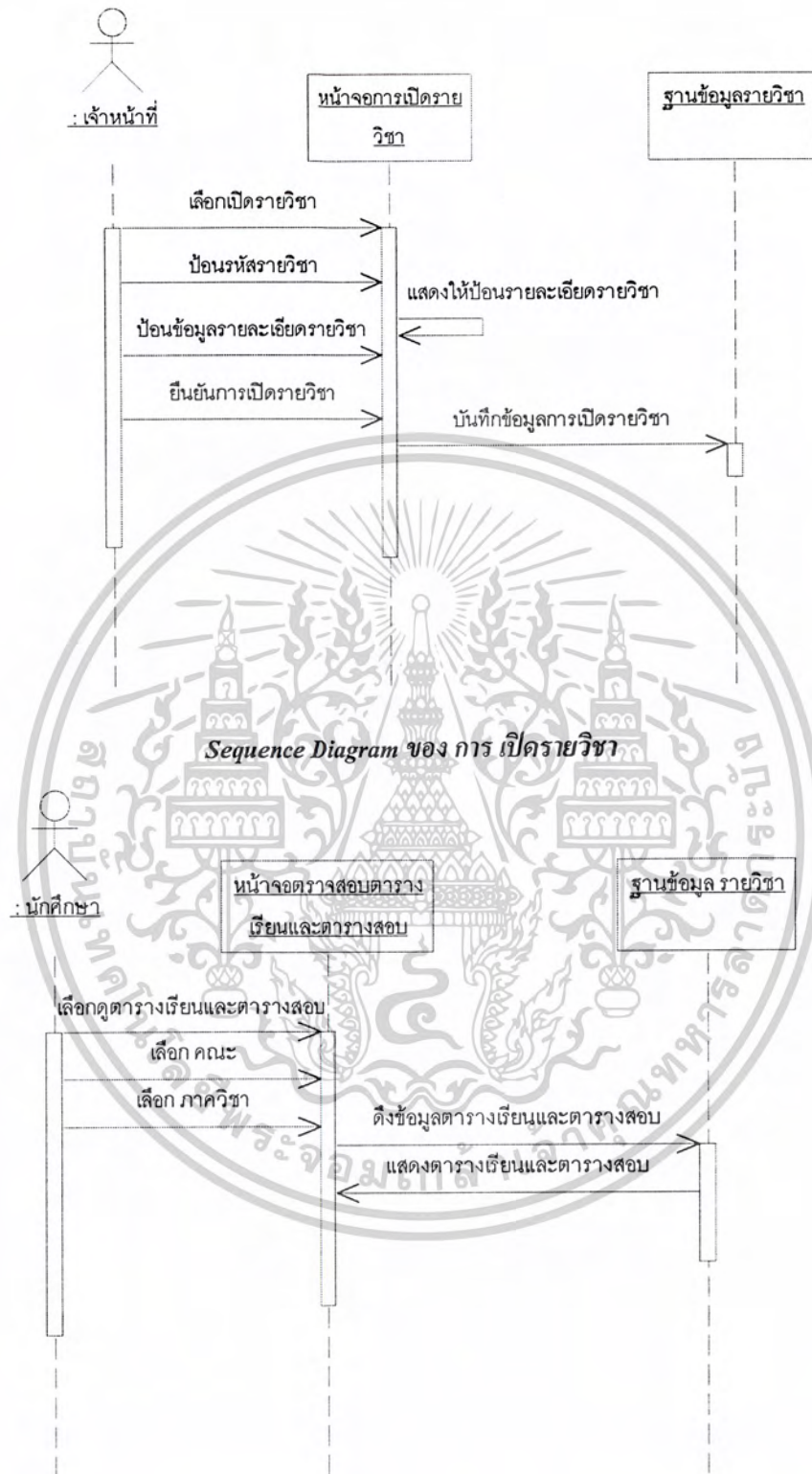


**Sequence Diagram ของ การ จัดการข้อมูลสถานะของนักศึกษา**



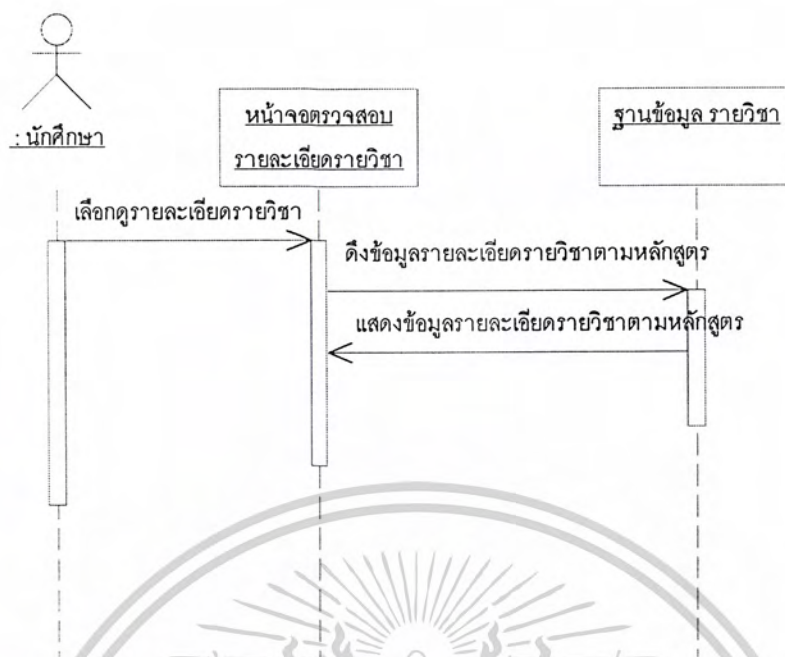
**Sequence Diagram ของ การ จัดการใบ ทรานสคริป**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

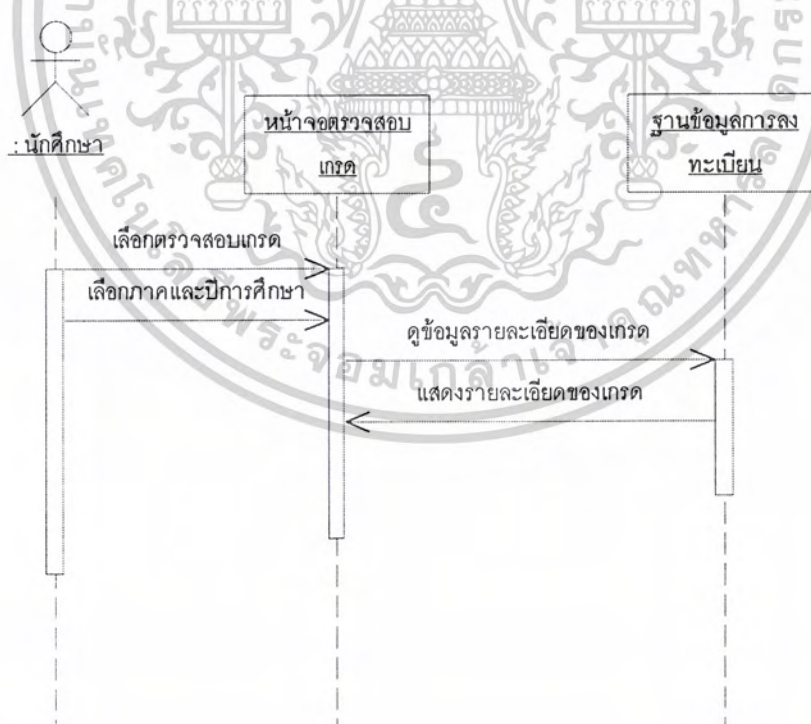


**Sequence Diagram ของ การ ตรวจสอบตารางเรียนและตารางสอบ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

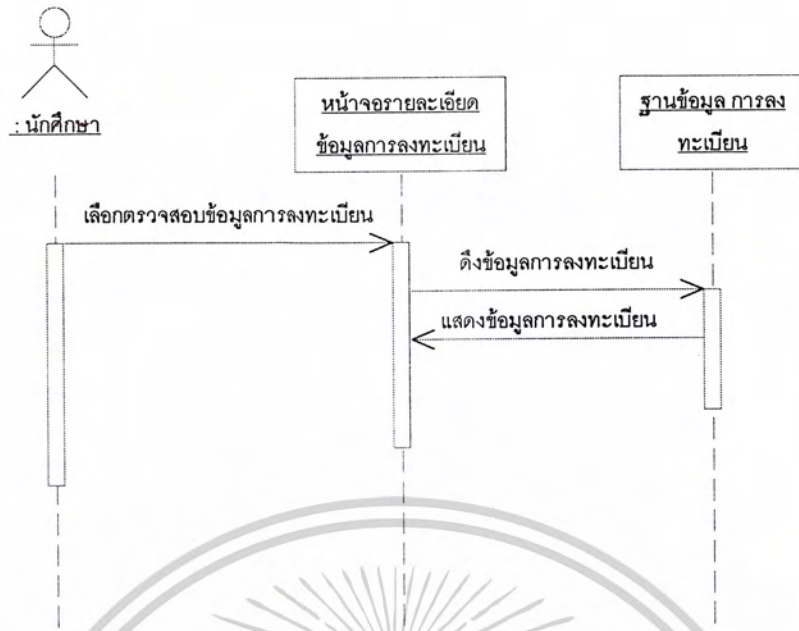


*Sequence Diagram* ของ การ ตรวจสอบรายละเอียดรายวิชา

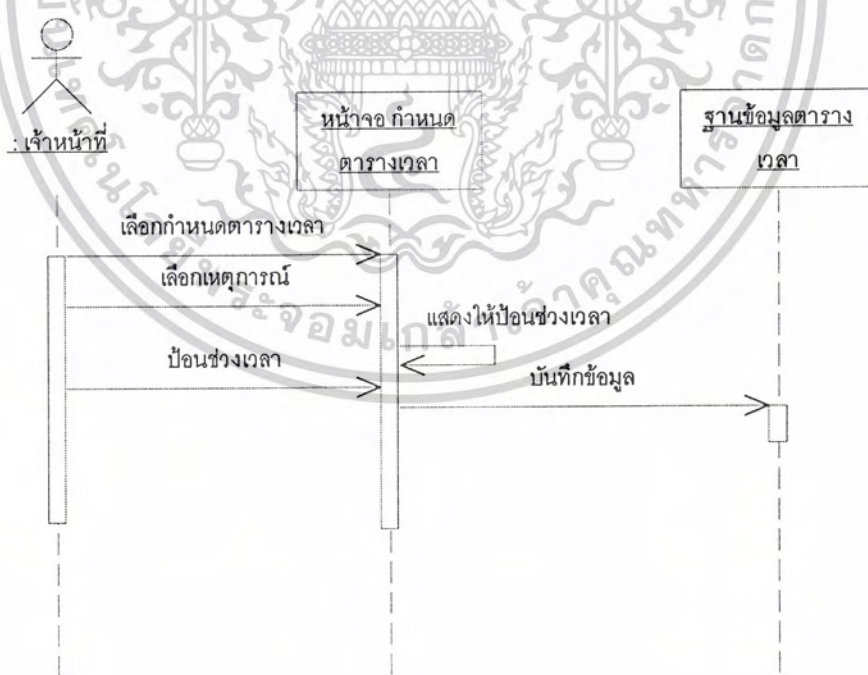


*Sequence Diagram* ของ การ ตรวจสอบเกรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

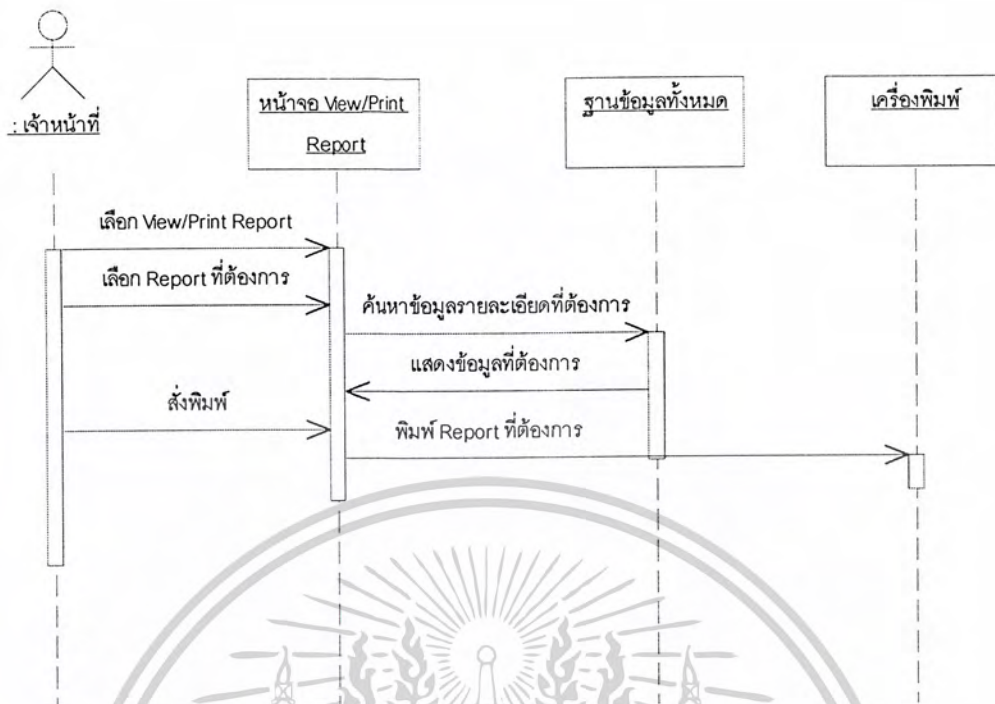


**Sequence Diagram ของ การ ตรวจสอบข้อมูลการลงทะเบียน**



**Sequence Diagram ของ การ กำหนดตารางเวลา**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

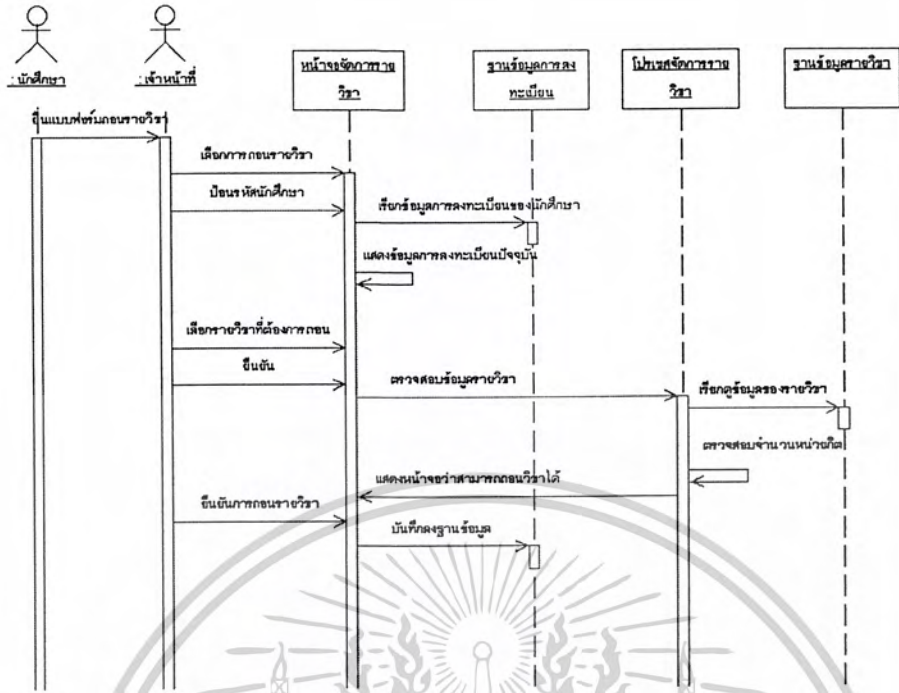


Sequence Diagram ของ การ ดูและพิมพ์รายงาน

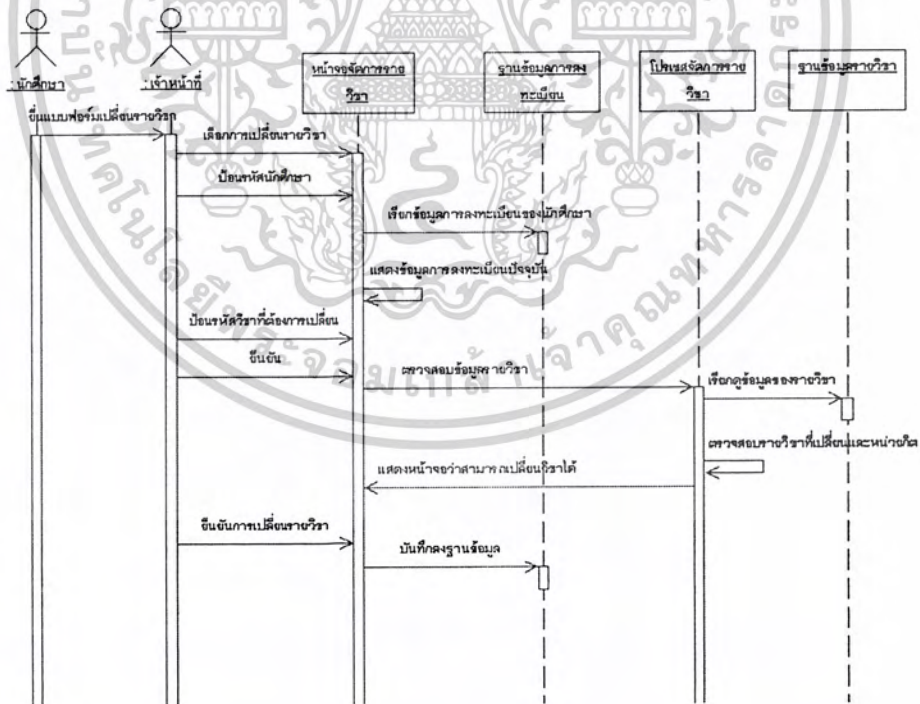


Sequence Diagram ของ การ จัดการรายวิชาหลังจากการลงทะเบียนล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

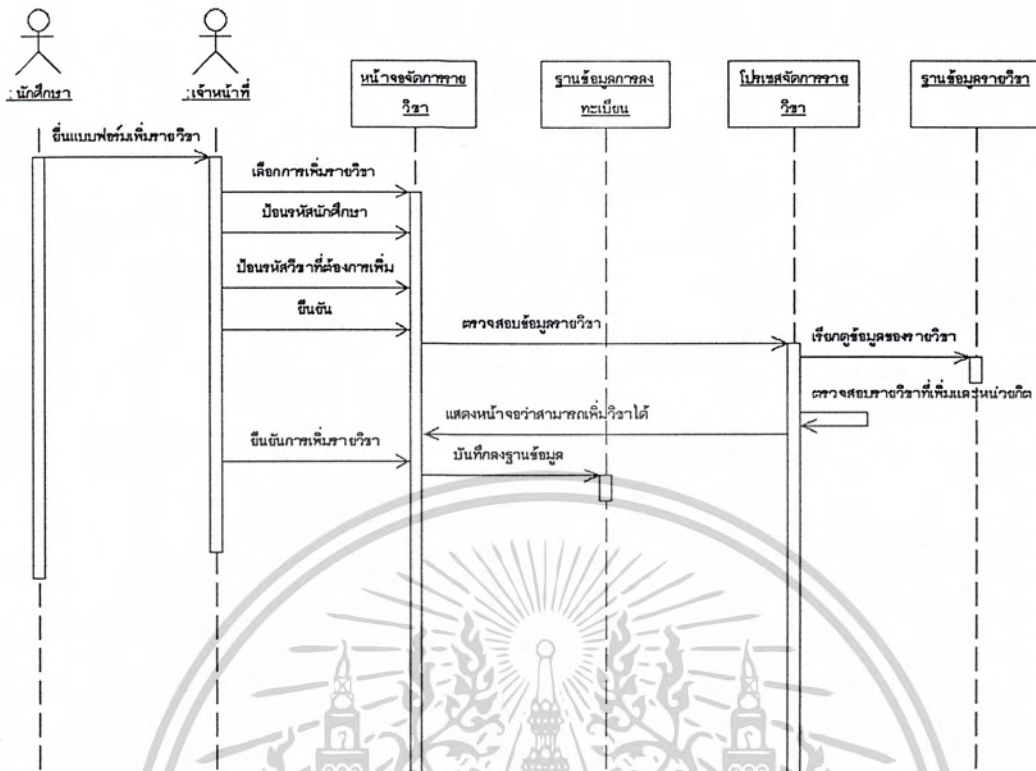


Sequence Diagram ของ การ ออกรายวิชา



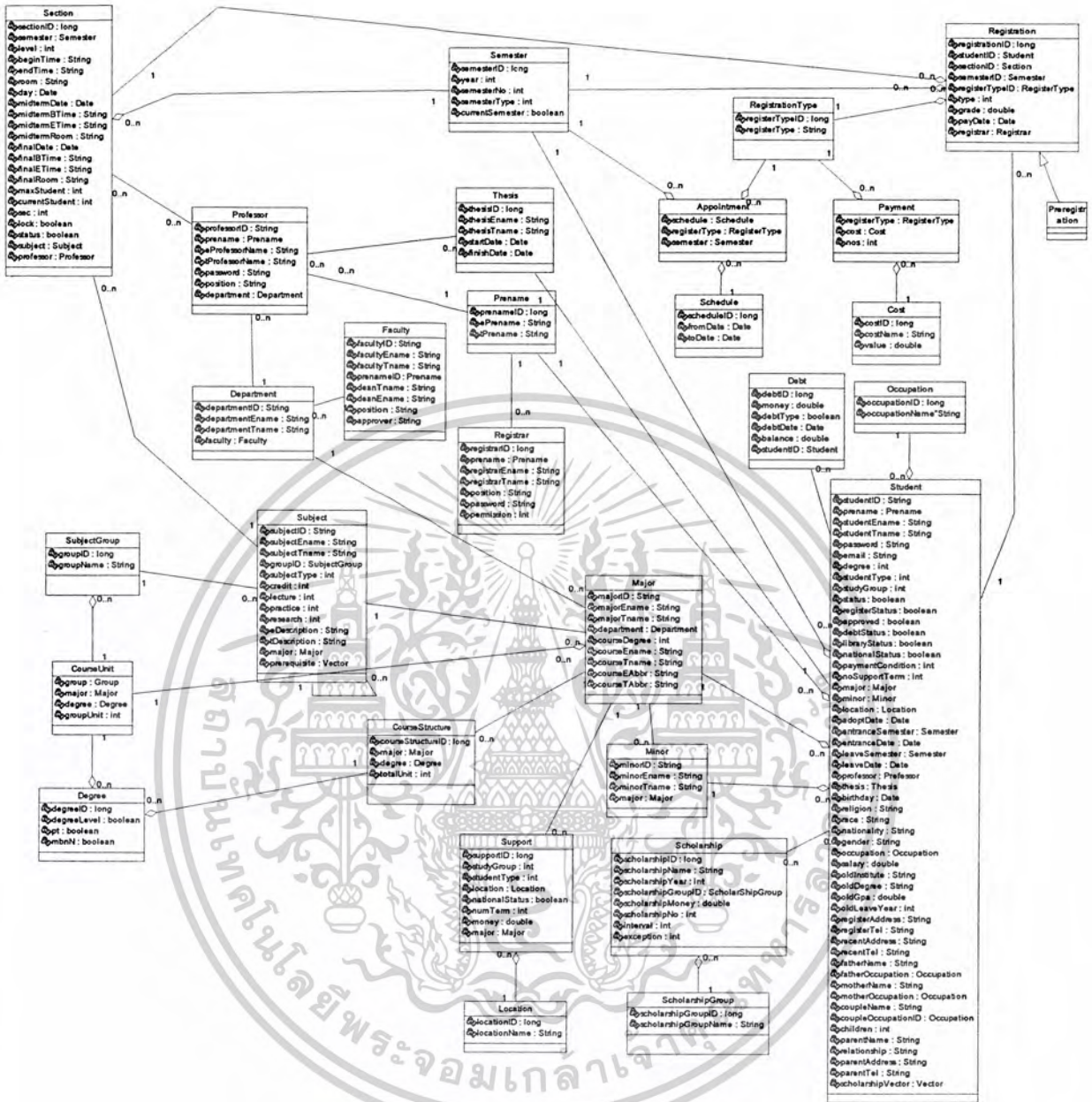
Sequence Diagram ของ การ เปลี่ยนรายวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



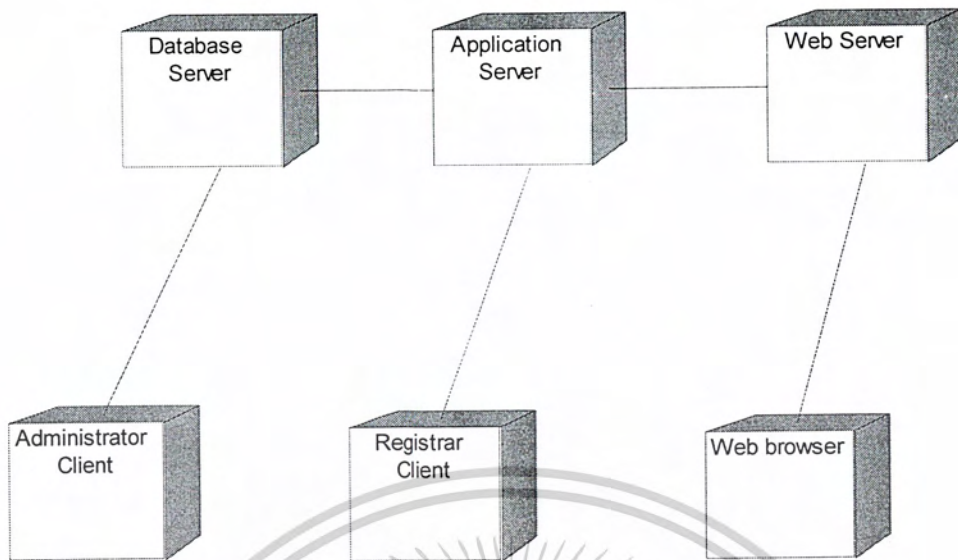
Sequence Diagram ของ การ เพิ่มรายวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



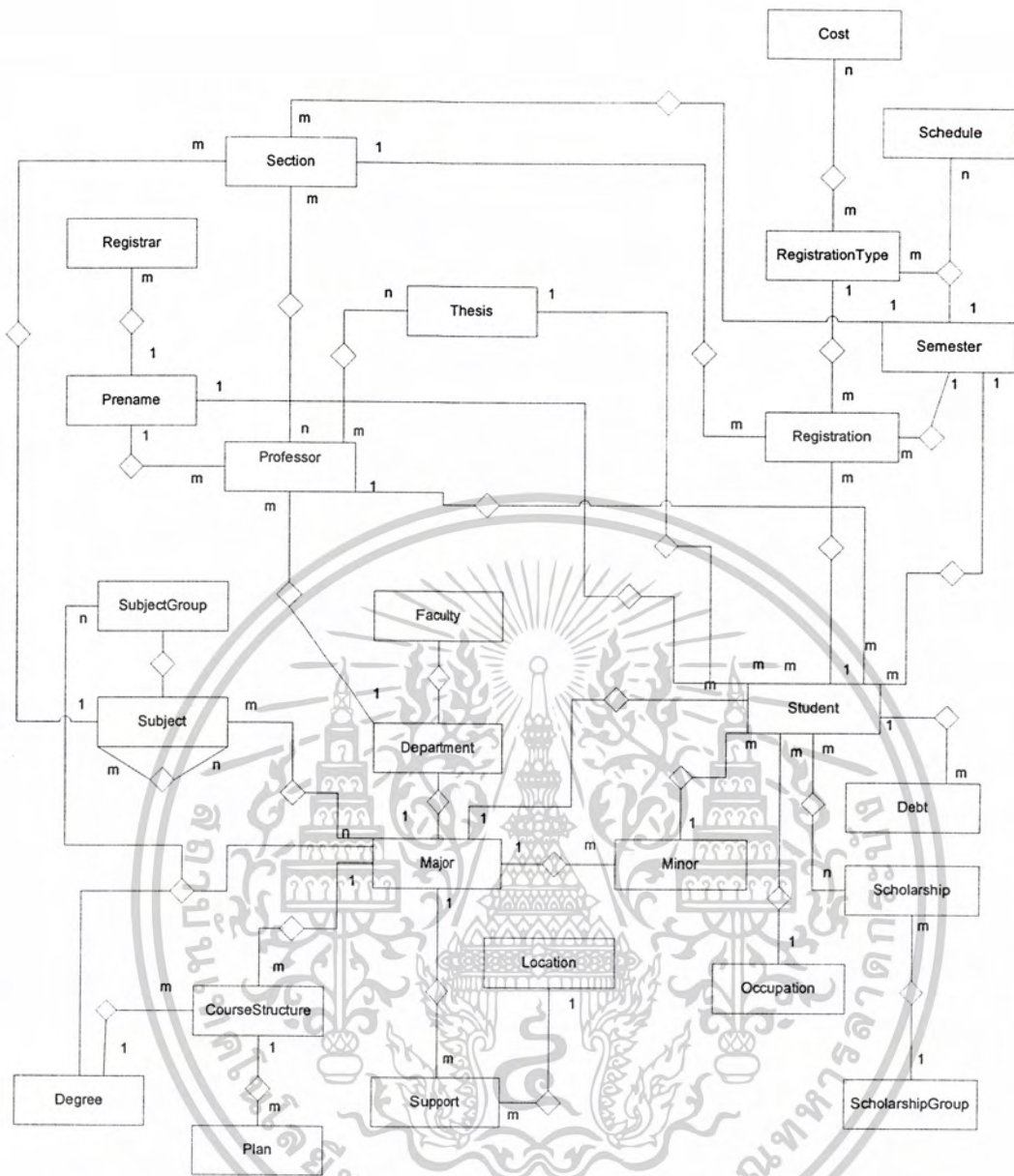
class Diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



*Deployment Diagram*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



E-R Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### โครงสร้างตารางฐานข้อมูล(Table Schema)

APPOINTMENT		
REGISTERTYPEID	INTEGER	รหัสประเภทการลงทะเบียน (PK)
SEMESTERID	INTEGER	รหัสภาคการศึกษา (PK)
FROMDATE	DATE	วันที่เริ่มกำหนด
TODATE	DATE	ถึงวันที่กำหนด

COST		
COSTID	INTEGER	รหัสค่าธรรมเนียมการศึกษา (PK)
COSTNAME	VARCHAR(50)	ชื่อค่าธรรมเนียมการศึกษา
VAL	DOUBLE	จำนวนเงินค่าธรรมเนียมการศึกษา

COURSESTRUCTURE		
COURSESTRUCTUREID	INTEGER	รหัสโครงสร้างหลักสูตร (PK)
MAJORID	VARCHAR(10)	รหัสสาขาวิชา
DEGREEID	INTEGER	รหัสระดับการศึกษา
TOTALUNIT	INTEGER	จำนวนหน่วยกิตรวมตลอดหลักสูตร

COURSEUNIT		
COURSEUNITID	INTEGER	รหัสกลุ่มวิชา (PK)
MAJORID	VARCHAR(10)	รหัสสาขาวิชา (PK)
DEGREEID	INTEGER	รหัสระดับการศึกษา (PK)
GROUPID	INTEGER	หน่วยกิตรวมของแต่ละหมวดวิชา

DEBT		
DEBTID	INTEGER	รหัสหนี้ (PK)
DEBTMONEY	DOUBLE	จำนวนเงินที่ค้างชำระหรือจำนวนเงินที่ชำระ
DEBTTYPE	SMALLINT	ประเภทจำนวนเงิน(ค้างชำระหรือที่ชำระ)
BALANCE	DOUBLE	จำนวนเงินคงเหลือที่ต้องชำระ
DEBTDATE	DATE	วันที่ชำระเงิน
STUDENTID	VARCHAR(10)	รหัสนักศึกษา (FK)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEGREE		
DEGREEID	INTEGER	รหัสระดับการศึกษา (PK)
DEGREELEVEL	SMALLINT	ระดับการศึกษา
PT	SMALLINT	แผน/แบบการศึกษา
MBN	SMALLINT	จบการศึกษาระดับ/แผนก.(1,2)

DEPARTMENT		
DEPARTMENTID	VARCHAR(10)	รหัสภาควิชา (PK)
DEPARTMENTENAME	VARCHAR(50)	ชื่อภาควิชาภาษาอังกฤษ
DEPARTMENTTNAME	VARCAHR(50)	ชื่อภาควิชาภาษาไทย
FACULTYID	VARCHAR(10)	รหัสคณะ (FK)

FACULTY		
FACULTYID	VARCHAR(10)	รหัสคณะ (PK)
FACULTYENAME	VARCHAR(50)	ชื่อคณะภาษาอังกฤษ
FACULTYTNAME	VARCAHR(50)	ชื่อคณะภาษาไทย
PRENAMEID	INTEGER	รหัสสำนักหน้าชื่อ
DEANTNAME	VARCHAR(50)	ชื่อคณบดีภาษาไทย
DEANENAME	VARCHAR(50)	ชื่อคณบดีภาษาอังกฤษ
POSITION	VARCHAR(50)	ตำแหน่ง
APPROVER	VARCHAR(50)	ผู้อนุมัติทรานสคริปต์

LOCATION		
LOCATIONID	INTEGER	รหัสศูนย์การเรียน (PK)
LOCATIONNAME	VARCHAR(50)	ชื่อศูนย์การเรียน

MAJOR		
MAJORID	VARCHAR(10)	รหัสสาขาวิชา (PK)
MAJORENAME	VARCHAR(50)	ชื่อสาขาวิชาภาษาอังกฤษ
MAJORTNAME	VARCHAR(50)	ชื่อสาขาวิชาภาษาไทย
DEPARTMENTID	VARCHAR(10)	รหัสภาควิชา
MASTERENAME	VARCHAR(50)	ชื่อหลักสูตรภาษาอังกฤษ
MASTERTNAME	VARCHAR(50)	ชื่อหลักสูตรภาษาไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAJOR		
MASTEREABBR	VARCHAR(50)	ชื่อย่อหลักสูตรภาษาอังกฤษ
MASTERTABBR	VARCHAR(50)	ชื่อย่อหลักสูตรภาษาไทย
PHDENAME	VARCHAR(50)	ชื่อหลักสูตรภาษาอังกฤษ
PHDTNAME	VARCHAR(50)	ชื่อหลักสูตรภาษาไทย
PHDEABBR	VARCHAR(50)	ชื่อย่อหลักสูตรภาษาอังกฤษ
PHDTABBR	VARCHAR(50)	ชื่อย่อหลักสูตรภาษาไทย

MINOR		
MINORID	VARCHAR(10)	รหัสแขนงวิชา (PK)
MINORENAME	VARCHAR(50)	ชื่อแขนงวิชาภาษาอังกฤษ
MINORTNAME	VARCHAR(50)	ชื่อแขนงวิชาภาษาไทย
MAJORID	VARCHAR(10)	รหัสสาขาวิชา (FK)

OCCUPATION		
OCCUPATIONID	INTEGER	รหัสอาชีพ (PK)
OCCUPATIONNAME	VARCAHR(50)	ชื่ออาชีพ

PAYMENT		
PAYMENTID	INTEGER	รหัสประเภทการลงทะเบียน (PK)
COSTID	INTEGER	รหัสค่าธรรมเนียมการศึกษา (PK)
NOS	SMALLINT	นักศึกษาเก่าหรือนักศึกษาใหม่หรือทั้งสองประเภท

PRENAME		
PRENAMEID	INTEGER	รหัสค่านำหน้าชื่อ (PK)
EPRENAME	VARCAHR(50)	ค่านำหน้าชื่อภาษาอังกฤษ
PRENAME	VARCAHR(50)	ค่านำหน้าชื่อภาษาไทย

PREREQUISITE		
PREREQUISITEID	VARCAHR(10)	รหัสวิชาที่ต้องเรียนมาก่อน (PK)
SUBJECTID	VARCAHR(10)	รหัสวิชา (PK)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROFESSOR		
PROFESSORID	VARCAHR(10)	รหัสอาจารย์ (PK)
SUBJECTID	VARCAHR(10)	รหัสวิชา
PRENAMEID	INTEGER	รหัสค่านำหน้าชื่อ
EPROFESSORID	VARCHAR(50)	ชื่ออาจารย์ภาษาอังกฤษ
PASSWORD	VARCHAR(10)	รหัสผ่านอาจารย์
POSITION	VARCHAR(50)	ตำแหน่งอาจารย์
DEPARTMENTID	VARCHAR(10)	รหัสภาควิชา (FK)

REGISTERTYPE		
REGISTERTYPEID	INTEGER	รหัสประเภทการลงทะเบียน (PK)
REGISTERTYPE	VARCAHR(50)	ประเภทการลงทะเบียน

REGISTRAR		
REGISTRARID	VARCHAR(10)	รหัสพนักงาน (PK)
PRENAMEID	INTEGER	รหัสค่านำหน้าชื่อ (FK)
REGISTRARENAME	VARCHAR(50)	ชื่อพนักงานภาษาอังกฤษ
REGISTRARTNAME	VARCHAR(50)	ชื่อพนักงานภาษาไทย
POSITION	VARCHAR(50)	ตำแหน่งพนักงาน
PASSWORD	VARCHAR(10)	รหัสผ่านพนักงาน
PERMISSION	INTEGER	สิทธิ์ในการใช้งานระบบ

REGISTRATION		
REGISTRATIONID	INTEGER	รหัสการลงทะเบียน (PK)
STUDENTID	VARCAHR(10)	รหัสนักศึกษา (FK)
SECTIONID	INTEGER	รหัสเซกชั่น (FK)
SEMESTERID	INTEGER	รหัสภาคการศึกษา (FK)
REGISTERTYPEID	INTEGER	รหัสประเภทการลงทะเบียน (FK)
TYPE	VARCHAR(5)	ประเภทการลงทะเบียน (Ad, Cr, Nc)
GRADE	VARCHAR(2)	เกรด
PAYDATE	DATE	วันที่ชำระเงิน
REGISTRARID	VARCHAR(10)	รหัสพนักงาน (FK)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCHOLAR		
SCHOLARSHIPID	INTEGER	ทุนการศึกษาที่ได้รับ (PK)
STUDENTID	VARCHAR(10)	รหัสนักศึกษาที่ได้รับทุน (PK)

SCHOLARSHIP		
SCHOLARSHIPID	INTEGER	ทุนการศึกษาที่ได้รับ (PK)
SCHOLARSHIPNAME	VARCHAR(50)	ชื่อทุนการศึกษา
SCHOLARSHIPYEAR	INTEGER	ปีการศึกษาที่ให้ทุนการศึกษา
SCHOLARSHIPGROUPID	INTEGER	รหัสประเภททุนการศึกษา (FK)
SCHOLARSHIPMONERY	DOUBLE	ทุนละ
SCHOLARSHIPNO	INTEGER	จำนวนทุนการศึกษาที่ให้
INTERVAL	INTEGER	ระยะเวลาการให้ทุน
EXCEPTION	INTEGER	ยกเว้นค่าหน่วยกิต, เงินสนับสนุน
SCHOLARSHIPGROUPID	INTEGER	รหัสประเภททุนการศึกษา (PK)
SCHOLARSHIPNAME	VARCHAR(50)	ชื่อประเภททุนการศึกษา

SECTION		
SECTIONID	INTEGER	รหัสสหกรณ์ (PK)
SEMESTERID	INTEGER	รหัสภาคการศึกษา (FK)
DEGREELEVEL	INTEGER	ชั้นปีที่เรียน
BEGINTIME	VARCHAR(10)	เวลาเริ่มเรียน
ENDTIME	VARCHAR(10)	เวลาเลิกเรียน
ROOM	VARCHAR(10)	ห้องเรียน
DAY	INTEGER	วันที่เรียน
MIDTERMDATE	DATE	วันสอบกลางภาค
MIDTERMBTIME	VARCHAR(10)	เวลาเริ่มสอบกลางภาค
MIDTERMETIME	VARCHAR(10)	เวลาเลิกสอบกลางภาค
MIDTERMROOM	VARCHAR(10)	ห้องสอบกลางภาค
FINALDATE	DATE	วันสอบปลายภาค
FIANLBTIME	VARCHAR(10)	เวลาเริ่มสอบปลายภาค
FIANLBTIME	VARCHAR(10)	เวลาเลิกสอบปลายภาค
FINALROOM	VARCHAR(10)	ห้องสอบปลายภาค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SECTION		
MAXSTUDENT	INTEGER	จำนวนนักศึกษาที่รับสูงสุด
CURRENTSTUDENT	INTEGER	จำนวนนักศึกษาที่ลงทะเบียน
SEC	INTEGER	ลำดับของเซกชั่น
LOCK	SMALLINT	การล็อกจำนวนนักศึกษาที่ลงทะเบียน
STATUS	SMALLINT	สถานะของเซกชั่น (Active/Cancel)
SUBJECTID	VARCHAR(10)	รหัสวิชา (FK)

SEMESTER		
SEMESTERID	INTEGER	รหัสภาคการศึกษา (PK)
YEAR	INTEGER	ปีการศึกษา
SEMESTERNO	INTEGER	ภาคการศึกษาที่
SEMESTERTYPE	INTEGER	ประเภทภาคการศึกษา
CURRENTSEMESTER	SMALLINT	ภาคการศึกษาปัจจุบัน
STUDENTID	VARCHAR(10)	รหัสนักศึกษา (PK)
PRENAMEID	INTEGER	รหัสคำนำหน้าชื่อ (FK)
STUDENTENAME	VARCHAR(50)	ชื่อนักศึกษาภาษาอังกฤษ
STUDENTTNAME	VARCHAR(50)	ชื่อนักศึกษาภาษาไทย

STUDENT		
PASSWORD	VARCHAR(10)	รหัสผ่านนักศึกษา
EMAIL	VARCHAR(50)	อีเมลแอดเดรส
DEGREE	INTEGER	ระดับการศึกษา
STUDENTTYPE	INTEGER	ประเภทนักศึกษา
STUDENTGROUP	INTEGER	ประเภทกลุ่มการเรียน
STATUS	INTEGER	สถานภาพนักศึกษา (กำลังศึกษา, จบการศึกษา, พ้นสภาพ, ลาออก, ปรับสภาพ)
REGISTERSTATUS	INTEGER	สถานะก่อนการลงทะเบียน
APPROVED	SMALLINT	ได้รับการอนุมัติการลงทะเบียนกรณีไม่สามารถลงทะเบียนได้ตามกำหนด
DEBTSTATUS	SMALLINT	สถานะ การผ่อนผันการชำระ
LIBRARYSTATUS	SMALLINT	สถานะการค้างหนังสือห้องสมุด
NATIONALSTATUS	SMALLINT	สถานภาพนักศึกษา (ไทยหรือต่างประเทศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STUDENT		
PAYMENTCONDITION	INTEGER	เงื่อนไขการชำระเงิน
NOSUPPORTTERM	INTEGER	จำนวนเทอมที่ต้องจ่ายเงินสนับสนุน
MAJORID	VARCHAR(10)	รหัสสาขาวิชา (FK)
MINORID	VARCHAR(10)	รหัสแขนงวิชา (FK)
LOCATIONID	INTEGER	รหัสศูนย์การเรียน (FK)
ADOPTDATE	DATE	วันเดือนปีปรับสภาพ
ENTRANCESEMESTERID	INTEGER	ปีเข้าการศึกษา (FK)
ENTRANCEDATE	DATE	วันเดือนปีที่เข้าการศึกษา
LEAVESEMESTERID	INTEGER	ปีจบการศึกษา, พ้นสภาพ, ลาออก (FK)
LEAVEDATE	DATE	วันเดือนปีที่จบการศึกษา, พ้นสภาพ, ลาออก
PROFESSORID	VARCHAR(10)	รหัสอาจารย์ที่ปรึกษา (FK)
THESISID	INTEGER	รหัสวิทยานิพนธ์ (FK)
BIRTHDAY	DATE	วันเกิด
RELIGION	VARCHAR(50)	ศาสนา
RACE	VARCHAR(50)	เชื้อชาติ
NATIONALITY	VARCHAR(50)	สัญชาติ
GENDER	VARCHAR(10)	เพศ
OCCUPATIONID	INTEGER	รหัสอาชีพ (FK)
SALARY	DOUBLE	เงินเดือน
OLDINSTITUE	VARCHAR(50)	สถาบันการศึกษาเดิม
OLDDEGREE	VARCHAR(50)	วุฒิการศึกษาเดิม
OLDGPA	DOUBLE	เกรดเฉลี่ยสะสมเดิม
OLDLEAVEYEAR	INTEGER	ปีการศึกษาที่จบ
REGISTERADDRESS	VARCAHR(100)	ที่อยู่ตามทะเบียนบ้าน
REGISTERTEL	VARCAHR(50)	เบอร์โทรบ้านตามทะเบียนบ้าน
RECENTADDRESS	VARCHAR(100)	ที่อยู่ปัจจุบัน
RECENTTEL	VARCHAR(50)	เบอร์โทรศัพท์ปัจจุบัน
FATHERNAME	VARCHAR(10)	ชื่อบิดา
FATHEROCCUPATIONID	INTEGER	รหัสอาชีพบิดา (FK)
MOHERNAME	VARCHAR(50)	ชื่อมารดา
MOTHEROCCUPATIONID	INTEGER	รหัสอาชีพมารดา (FK)
COUPLENAME	VARCHAR(50)	ชื่อคู่สมรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STUDENT		
COUPLEOCCUPATIONID	INTEGER	รหัสอาชีพคู่สมรส (FK)
CHILDREN	INTEGER	จำนวนบุตร
PARENTNAME	VARCHAR(50)	ชื่อผู้ปกครอง
RELATIONSHIP	VARCHAR(50)	ความสัมพันธ์
PARENTADDRESS	VARCHAR(100)	ที่อยู่ผู้ปกครอง
PARENTTEL	VARCHAR(50)	เบอร์โทรศัพท์ผู้ปกครอง

SUBJECT		
SUBJECTID	VARCHAR(10)	รหัสวิชา (PK)
SUBJECTENAME	VARCHAR(50)	ชื่อวิชาภาษาอังกฤษ
SUBJECTTNAME	VARCHAR(50)	ชื่อวิชาภาษาไทย
GROUPID	INTEGER	รหัสกลุ่มวิชา (FK)
SUBJECTTYPE	INTEGER	ประเภทรายวิชา
CREDIT	INTEGER	หน่วยกิตรวม
LECTURE	INTEGER	บรรยาย
PRACTICE	INTEGER	ปฏิบัติ
EDESCRIPTION	VARCHAR(50)	คำอธิบายรายวิชาภาษาอังกฤษ
TDESCRIPTION	VARCHAR(50)	คำอธิบายรายวิชาภาษาไทย
MAJORID	VARCHAR(10)	รหัสสาขาวิชา (FK)

SUBJECTGROUP		
GROUPID	INTEGER	รหัสกลุ่มวิชา (PK)
GROUPNAME	VARCHAR(50)	ชื่อกลุ่มวิชา

SUPPORT		
SUPPORTID	INTEGER	รหัสเงินสมทบการศึกษา (PK)
STUDYGROUP	INTEGER	ประเภทกลุ่มการเรียนรู้ (ภาคปกติ, สมทบ, พิเศษ (เสาร์, อาทิตย์), พิเศษนอกเวลาราชการ, หลักสูตรนานาชาติ)
STUDENTTYPE	INTEGER	ประเภทนักศึกษา (สามัญ, ทดลองเรียน)
LOCATIONID	INTEGER	รหัสศูนย์การเรียนรู้ (FK)
NATIONALSTATUS	SMALLINT	นักศึกษาไทยหรือต่างประเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUPPORT		
NUMTERM	INTEGER	จำนวนเทอมการศึกษาที่ต้องชำระ
SUPPORTMONEY	DOUBLE	จำนวนเงินที่ต้องชำระ
MAJORID	VARCHAR(10)	รหัสสาขาวิชา (FK)

TEACH		
SECTIONID	INTEGER	รหัสเซกชัน (PK)
PROFESSORID	VARCHAR(10)	รหัสอาจารย์ (PK)

THESIS		
THESISID	INTEGER	รหัสวิทยานิพนธ์
THESISENAME	VARCHAR(50)	ชื่อวิทยานิพนธ์(ภาษาอังกฤษ)
THESISTNAME	VARCHAR(50)	ชื่อวิทยานิพนธ์(ภาษาไทย)
STARTDATE	DATE	วันที่เริ่มทำวิทยานิพนธ์
FINISHDATE	DATE	วันที่ทำวิทยานิพนธ์เสร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- [1] Martin Fowler with Kendall Scott. *UML Distilled*, Second Edition: A brief guide to the standard Object Modeling Language, Addison - Wesley Longman, 2000
- [2] Ali Bahrami, Boeing Applied Research & Technology. *Object Oriented Systems Development*, International Edition 1999, Irwin McGraw-Hill, 1976
- [3] Ivar Jacobson, Grady Booch, James Rumbaugh. *The Unified Software Development Process* Rational Software Corporation, Addison -Wesley Longman, 1998
- [4] Terry Quatrani. *Visual Modeling with Rational Rose 2000 and UML*, Addison-Wesley, Pearson Education Asia Pte Ltd., 1998
- [5] อาจารย์ช่อภินันท์ อุนากุล. *Object-Oriented Analysis and Design*, แผนกคํารวาคณะวิศวกรรมศาสตร๑สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 1999
- [6] Marty Hall. *Core Servlet and Java Server Page*, Prentice Hall PTR Upper Saddle River, 2000
- [7] Dale R. Nilsson, Peter M. Jakab, Bill Sarantakos, Russell A. Stinehour. *Enterprise Development with VisualAge for Java, Version 3*, Wiley Computer Publishing, John Wiley & Sons. Inc., 2000
- [8] Osamu Takagiwa, Frederik Haesbrouck, Veronique Quilblier, Sarah Poge. *Programming with VisualAge for Java, Version 3.5*, Third Edition, Prentice Hall PTR, Upper Saddle River, New Jersey 07458, April 2001
- [9] Eric J. Naiburg, Robert A. Maksimchuk. *UML For Database Design*, Addison -Wesley, 2001
- [10] Akmal B. Chaudhri, Roberto Zicari. *Succeeding with Object Databases*, Wiley Computer Publishing, John Wiley & Sons. Inc., 2001
- [11] Wendy Boggs, Michael Boggs. *Mastering UML with Rational Rose 2002*, SYBEX Inc., 2002