

การพัฒนาระบบสารสนเทศศึกษาระดับบัณฑิตศึกษา โดยวิธีเชิงวัตถุ
OBJECT-ORIENTED SOFTWARE DEVELOPMENT OF STUDENT
INFORMATION SYSTEM FOR GRADUATE STUDENTS



เลขหมู่.....
เลขทะเบียน.....46187
วัน, เดือน, ปี 20 ส.ค. 2546

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

๒๑๑๒๘๘๕๗๐

การพัฒนาระบบสารสนเทศศึกษาระดับบัณฑิตศึกษา โดยวิธีเชิงวัตถุ
OBJECT-ORIENTED SOFTWARE DEVELOPMENT OF STUDENT
INFORMATION SYSTEM FOR GRADUATE STUDENTS



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาระบบสารสนเทศศึกษาระดับบัณฑิตศึกษา โดยวิธีเชิงวัตถุ

OBJECT-ORIENTED SOFTWARE DEVELOPMENT OF STUDENT INFORMATION SYSTEM
FOR GRADUATE STUDENTS

ผู้จัดทำ

- | | |
|------------------------|-----------------------|
| 1. นายมาศพงษ์ ฌ ลำปาง | รหัสนักศึกษา 42015313 |
| 2. นายสรวิทย์ นุพลกรัง | รหัสนักศึกษา 42015319 |
| 3. นายสมบูรณ์ เกิดทอง | รหัสนักศึกษา 42015323 |

(ดร.วิศิษฐ์ หิรัญกิตติ)

อาจารย์ที่ปรึกษา



การพัฒนาระบบสารสนเทศศึกษาระดับบัณฑิตศึกษา โดยวิธีเชิงวัตถุ

นายมาศพงษ์	ณ ลำปาง	42015313
นายศราวุฒิ	นุพลกรัง	42015319
นายสมภรณ์	เกิดทอง	42015323
ดร.วิศิษฐ์	หิรัญกิตติ	อาจารย์ที่ปรึกษา
ปีการศึกษา 2544		

บทคัดย่อ

ระบบสารสนเทศศึกษาระดับบัณฑิตศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่มีอยู่เดิมนั้นได้ประสบปัญหาในการใช้งานอยู่หลายด้าน เนื่องจากขั้นตอนการลงทะเบียนที่มีหลายขั้นตอน อีกทั้งในระบบข้อมูลทะเบียนนักศึกษาระดับบัณฑิตศึกษานั้นมีความซับซ้อนและมีเงื่อนไขต่างๆ อยู่มาก และเปลี่ยนแปลงอยู่ตลอดเวลา ระดับความถูกต้องและการตรวจสอบกระบวนการต่างๆ ยังทำโดยใช้พนักงานดำเนินการเป็นหลัก ทำให้เกิดความผิดพลาดขึ้น ดังนั้นควรมีการปรับปรุงระบบสารสนเทศนักศึกษาเดิมเพื่อแก้ไขปัญหาดังกล่าว

ปัญหานี้พบนั้ได้ศึกษา วิเคราะห์ ออกแบบและพัฒนาระบบสารสนเทศศึกษาระดับบัณฑิตศึกษาขึ้นมาใหม่ โดยใช้วิธีการใหม่ คือใช้วิธีการพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-Oriented Software Development) ในการวิเคราะห์และออกแบบได้ใช้ภาษาสร้างแบบจำลอง UML และทำการพัฒนาระบบโดยใช้ภาษาจาวา รวมทั้งยังได้ใช้ Java Servlet และ Java Server Page (JSP) ในการพัฒนาเว็บแอปพลิเคชัน ในส่วนของฐานข้อมูลนั้นเลือกใช้ฐานข้อมูลเชิงสัมพันธ์

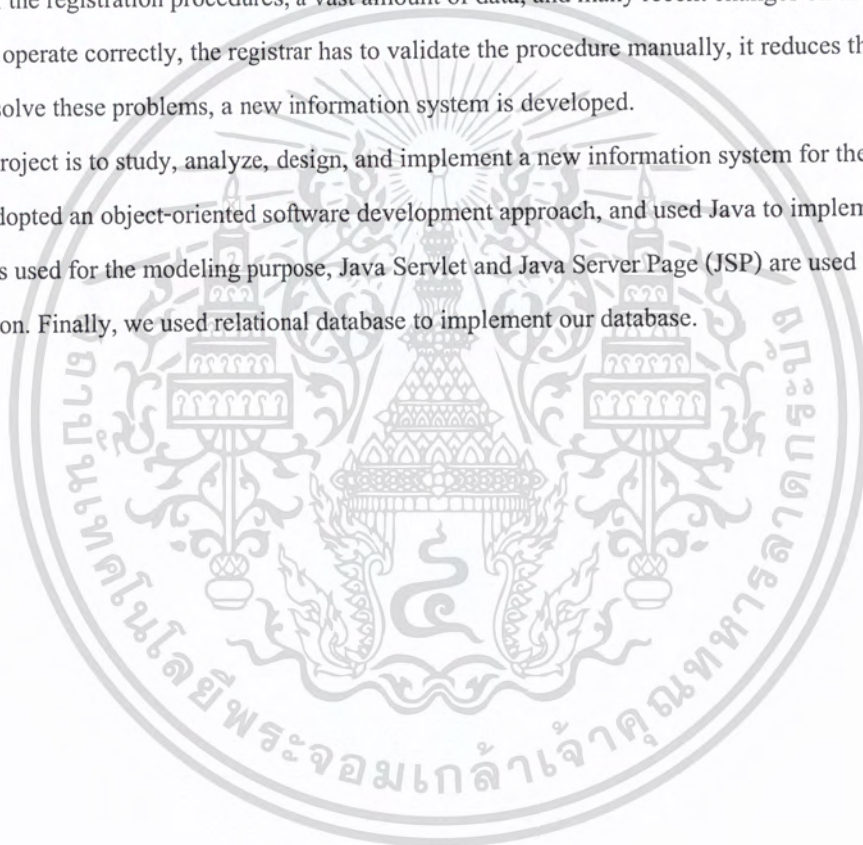
OBJECT-ORIENTED SOFTWARE DEVELOPMENT OF STUDENT INFORMATION SYSTEM FOR GRADUATE STUDENTS

Maspong	Nalampang
Sarawut	NoopholKrang
Somphoon	Kerdthong
Dr.Visit	Hirankitti Advisor

ABSTRACT

The exiting information system for graduate students at KMITL suffers many problems due to the complication of the registration procedures, a vast amount of data, and many recent changes on the regulations. To operate correctly, the registrar has to validate the procedure manually, it reduces the efficiency. To solve these problems, a new information system is developed.

This project is to study, analyze, design, and implement a new information system for the graduate students. We adopted an object-oriented software development approach, and used Java to implement the system. UML is used for the modeling purpose, Java Servlet and Java Server Page (JSP) are used to develop a web application. Finally, we used relational database to implement our database.



กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี เนื่องจากทางคณะผู้จัดทำได้รับความร่วมมือ และความอนุเคราะห์จากบุคคลต่าง ๆ หลายฝ่าย ทั้งทางด้านการทำงาน กำลังใจ ทางคณะผู้จัดทำขอขอบพระคุณท่านเหล่านั้นมา ณ โอกาสนี้

ดร.วิศิษฎ์ หิรัญกิตติ อาจารย์ที่ปรึกษาที่ได้ให้ความกรุณาช่วยเหลือ ให้คำปรึกษาที่เป็นประโยชน์ ซึ่งแรงข้อผิดพลาด และเสียสละเวลาดูแลเอาใจใส่ในการทำงานของพวกเรามาโดยตลอด

คณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ได้ให้ความรู้ทั้งทางด้านวิชาการและด้านการใช้ชีวิตในสังคมแก่พวกเรา

บริษัทไอบีเอ็มประเทศไทย ที่ได้ให้ความอนุเคราะห์ โปรแกรมและเครื่องมือต่าง ๆ ที่ใช้ในการพัฒนางานวิจัยนี้

ภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้ให้ความร่วมมือและความสะดวกสบายในทุก ๆ ด้านแก่พวกเรา

บิดา มารดาอันเป็นที่รักและเคารพที่ได้ให้ความรักและกำลังใจเสมอมา เพื่อน ๆ พี่ ๆ น้อง ๆ ชาวลาดกระบังทุกคนที่ให้พวกเราได้รับรู้ชีวิตนักศึกษา ให้มีความสุขสนุกสนานรื่นเริงกับพวกเรา

ในช่วงเวลา 3 ปีที่ผ่านมาชีวิตอีกหลายปีที่ต้องดำเนินต่อไป พวกเราขอสัญญาว่าจะพยายามใช้สิ่งต่าง ๆ ที่ได้รับจากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังแห่งนี้ให้เกิดประโยชน์แก่ตัวเราเอง และประเทศชาติสืบไป

นายมาศพงษ์ ณ ลำปาง

นายศราวุฒิ นูพลกรัง

นายสมภูรณ เกิดทอง

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VI
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 การพัฒนาระบบด้วยวิธีเชิงวัตถุ	5
2.3 ข้อดีของการพัฒนาระบบด้วยวิธีเชิงวัตถุ	5
2.4 ความสามารถของคลาส	7
2.5 Object Oriented Methodologies	8
2.6 Pattern และ Frameworks	11
2.7 กระบวนการพัฒนาซอฟต์แวร์ (Software Process) ที่ดี	12
2.8 ส่วนประกอบของ UML	14
บทที่ 3 Servlet, JSP และ Java Bean	21
3.1 เซิร์ฟเล็ต (Servlet)	21
3.2 เจเอสพี (Java Server Page : JSP)	23
3.3 จาวาบีน (JavaBean)	26
บทที่ 4 เครื่องมือที่ใช้ในการพัฒนาโปรแกรมและฐานข้อมูล	29
4.1 บทนำ	29
4.2 ลักษณะของ DB2ที่ใช้ในโครงการ	29
4.3 การเชื่อมต่อเข้ากับฐานข้อมูลโดยใช้ JDBC	29
4.4 ประโยชน์ของการใช้ VisualAge for Java	33
4.5 ข้อดีของ VisualAge for Java	34
4.6 ข้อเสียของ VisualAge for Java	35
บทที่ 5 การวิเคราะห์และออกแบบระบบทะเบียนนักศึกษาในระดับบัณฑิตศึกษา	36

5.1 ฝ่ายทะเบียนนักศึกษาของบัณฑิตวิทยาลัย	36
5.2 ปัญหาที่พบในระบบทะเบียนนักศึกษาเดิม	37
5.3 การวิเคราะห์ระบบ	37
5.4 การออกแบบระบบโดยใช้ยูเอ็มแอล	38
5.5 การออกแบบการทำงานของฟังก์ชันต่าง ๆ	41
5.6 การออกแบบการรายละเอียดของคลาสต่าง ๆ	44
5.7 การออกแบบโครงสร้างของระบบ	45
บทที่ 6 การพัฒนาและทดสอบระบบ	46
6.1 การสร้างฐานข้อมูล	46
6.2 การเขียนโปรแกรมในส่วนของแอปพลิเคชัน	48
6.3 การทดสอบโปรแกรมในส่วนของแอปพลิเคชัน	51
6.4 การเขียนโปรแกรมในส่วนของเว็บแอปพลิเคชัน	58
6.5 การทดสอบโปรแกรมในส่วนของเว็บแอปพลิเคชัน	59
บทที่ 7 บทสรุปและบทวิจารณ์	62
7.1 บทสรุป	62
7.2 บทวิจารณ์	63
7.3 ปัญหาที่พบในการดำเนินงาน	63
7.4 ข้อเสนอแนะสำหรับระบบทะเบียนนักศึกษา ของบัณฑิตวิทยาลัย	63
7.5 ข้อเสนอแนะในการนำไปพัฒนาต่อ	64
ภาคผนวก ก	65
ภาคผนวก ข	104
ภาคผนวก ค	115
บรรณานุกรม	139

สารบัญรูปภาพ

เรื่อง	หน้า
บทที่ 2 การพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุและยูเอ็มแอล	
2-1 การเชื่อมโยงของ Use Case Model กับ Model อื่น ๆ	10
2-2 Best Practices of Software Engineering	12
2-3 Deployment Iteratively Process	12
2-4 สถาปัตยกรรมของ View	15
2-5 แสดงไคอะแกรมต่างๆ ของ UML	16
2-6 ตัวอย่าง Use Case ในระบบลงทะเบียน	16
2-7 แสดงตัวอย่างที่คอลเลโบเรชั่นไคอะแกรม	18
2-8 แสดงตัวอย่างซีเคิร์นซีไคอะแกรม	18
2-9 แสดงตัวอย่างแอคทิวิตีไคอะแกรม	19
2-10 แสดงตัวอย่างคอมโพเน้นไคอะแกรม	20
2-11 แสดงตัวอย่างดีพลอยเม้นไคอะแกรม	20
บทที่ 3 Servlet, JSP และ Java Bean	
3-1 รูปอธิบายการทำงานของ JSP	25
บทที่ 4 เครื่องมือที่ใช้ในการพัฒนาโปรแกรมและฐานข้อมูล	
4-1 แสดงลักษณะการทำงานของ JDBC	30
4-2 แสดงรูปแบบ 2-Tier	31
4-3 แสดงรูปแบบ 3-Tier	31
4-4 ระดับการเชื่อมต่อของ JDBC API	31
4-5 ลำดับการเชื่อมต่อเข้าฐานข้อมูลโดยใช้ JDBC	33
4-6 แสดงข้อดีกรณีของ Parser	34
บทที่ 5 การวิเคราะห์และออกแบบระบบทะเบียนนักเรียนระดับมัธยมศึกษา	
5-1 Use Case Diagram ของระบบ	39
5-2 Activity Diagram ของ Registration Usecase	42
5-3 Sequence Diagram ของ Registration Usecase	43
5-4 ตัวอย่าง Class Diagram ของระบบ	44
5-5 Component Diagram ของระบบ	45
5-6 Deployment Diagram ของระบบ	45
บทที่ 6 การพัฒนาและทดสอบระบบ	
6-1 การเชื่อมต่อฐานข้อมูลด้วย Client Configuration Assistant	47
6-2 การใช้ Command Center เพื่อการสร้างตาราง	48

6-3 รูปส่วนอินเทอร์เฟซของ MainPanel	50
6-4 รูปส่วนอินเทอร์เฟซของ AcademicPanel	50
6-5 รูปส่วนอินเทอร์เฟซของ AcademicPanel ที่ประกอบอยู่บน MainPanel	51
6-6 แสดงหน้าจอการล็อกอิน	51
6-7 รูปแสดงข้อมูลที่อยู่ในฐานข้อมูล	52
6-8 แสดงการเลือก menu หลัก	52
6-9 แสดงข้อมูลของคณะ	53
6-10 แสดงข้อมูลในตาราง Faculty	53
6-11 แสดงข้อมูลเมื่อเลือกคณะ	54
6-12 แสดงการแก้ไขข้อมูลของคณะ	54
6-13 แสดงการเพิ่มภาควิชา	55
6-13 แสดงผลการเพิ่มภาควิชา	55
6-14 แสดงการเพิ่มสาขาวิชา	56
6-15 แสดงข้อมูลหลังจากการเพิ่มสาขาวิชา	56
6-16 แสดงการลบภาควิชา	57
6-17 แสดงข้อมูลในตาราง Department	57
6-18 การทำงานของ Websphere Test Environment	58
6-19 การทดลองใช้งานผ่านเว็บเบราว์เซอร์	58
6-20 แสดงช่วงเวลาปัจจุบัน	59
6-21 แสดงเมนูที่เปลี่ยนแปลงตามช่วงเวลา	59
6-22 แสดงรายวิชาตามหลักสูตรในภาคการศึกษาที่ลงทะเบียน	60
6-23 แสดงรายวิชาที่ลงทะเบียน	60
6-24 แสดงค่าใช้จ่ายตามหลักสูตร	61
6-25 แสดงข้อความการลงทะเบียนสมบูรณ์	61

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

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

ในการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ นั้น ได้นำเอาภาษาสัญลักษณ์เข้ามามีใช้ในการสื่อความหมายว่าระบบทำงานอย่างไร แต่ละวัตถุทำงานและเชื่อมต่อกันอย่างไร ภาษาสัญลักษณ์ดังกล่าวมีชื่อว่า ยูเอ็มแอล(Unified Modeling Language: UML) ซึ่งอยู่เราใช้ยูเอ็มแอลแสดงถึงมุมมองต่าง ๆ ของระบบออกมาในรูปของไดอะแกรม(Diagram) ซึ่งมีหลายไดอะแกรม แต่ละไดอะแกรมก็จะแสดงมุมมองแต่ละมุมของระบบออกมาอีกทั้งการพัฒนาโปรแกรมเชิงวัตถุยังมีโปรแกรมที่ใช้เป็น เครื่องมือช่วยในการพัฒนาโปรแกรม(CASE Tools) อยู่มาก จึงทำให้เพิ่มความสะดวกและมาตรฐานในการพัฒนาโปรแกรมขึ้นอย่างมาก

จากเหตุผลข้างต้นการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ นั้นจึงได้รับการนิยมนำมาใช้ในการพัฒนาโปรแกรมในระดับต่าง ๆ มากขึ้น แต่สำหรับในประเทศไทยนั้น การพัฒนาโปรแกรมคอมพิวเตอร์ในภาคอุตสาหกรรมนั้นยังมีไม่มากนัก จึงส่งผลให้วิธีการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุถูกนำไปใช้น้อยเมื่อเทียบเชิงปริมาณ บุคลากรที่มีความรู้ความเข้าใจในด้านดังกล่าวก็มีน้อยด้วยเช่นกัน ดังนั้นในอนาคตความต้องการผู้ที่มีความรู้ความเข้าใจในด้านนี้ก็จะเพิ่มมากขึ้นเรื่อย ๆ ตามความเติบโตของอุตสาหกรรมซอฟต์แวร์

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

และที่ยากยิ่งอีกอย่างหนึ่งก็คือความเข้าใจอย่างถูกต้องในการใช้สัญลักษณ์ต่าง ๆ ในยูเอ็มแอลเพราะเราอาจใช้อย่างผิด ๆ ทำให้การสื่อความหมายไม่ถูกต้อง และอาจไม่ก่อให้เกิดประโยชน์ใด ๆ เลย เพราะในการใช้ยูเอ็มแอลนั้นเพียงการเรียนรู้แต่ไม่ได้ลงมือทำอย่างจริงจังแล้วนั้น ไม่อาจทำให้เข้าใจในยูเอ็มแอลได้อย่างแน่นอน ถึงแม้จะเข้าใจก็อาจเข้าใจเพียงบางแง่มุมเท่านั้นเอง ดังนั้นการที่ได้ลงมือปฏิบัติจริงในการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ นั้น จึงก่อให้เกิดประโยชน์อย่างมากกับผู้ที่ศึกษา ทั้งก่อให้เกิดความเข้าใจและมองเห็นข้อเด่น ข้อด้อย และข้อควรระวังในการใช้วิธีนี้

อีกประการหนึ่งก็คือ ระบบจัดการข้อมูลทะเบียนนักศึกษาระดับบัณฑิตศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่มีอยู่เดิมนั้น ได้ประสบปัญหาในดำเนินงานอยู่หลายด้าน ไม่ว่าจะเป็นเรื่องของความเร็วและความสะดวกที่มีไม่มากเท่าที่ควร อีกทั้งในระบบข้อมูลทะเบียนนักศึกษาระดับบัณฑิตศึกษานั้นมีความซับซ้อน และมีเงื่อนไขอยู่มาก ระดับความถูกต้องและการตรวจสอบกระบวนการต่าง ๆ ยังทำโดยใช้พนักงานดำเนินการเป็นหลักจึงอาจมีข้อผิดพลาดและข้อสับสนเกิดขึ้นได้ง่าย

การเข้ามาวิเคราะห์และพัฒนาระบบดังกล่าวให้มีความสามารถขึ้นและเพิ่มความสะดวกให้กับผู้ใช้ระบบซึ่งมีทั้งนักศึกษา เจ้าหน้าที่ทะเบียน อาจารย์จึงนับว่ามีประโยชน์อย่างยิ่ง

1.2 วัตถุประสงค์

1.2.1 เพื่อนำวิธีการทางวิศวกรรมซอฟต์แวร์เชิงวัตถุไปพัฒนาระบบทะเบียนนักศึกษาในระดับบัณฑิตศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังให้มีความสามารถยิ่งขึ้น สร้างความสะดวกและความถูกต้องในการทำงานมากขึ้น

1.2.2 เพื่อศึกษาและทำความเข้าใจระบบทะเบียนนักศึกษาในระดับบัณฑิตศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังเดิมว่าเป็นอย่างไร และสามารถปรับปรุงให้ดีขึ้นโดยไม่ทำให้เกิดผลกระทบกับผู้ใช้ปฏิบัติงานและ ระบบอื่นที่เกี่ยวข้องมากนัก

1.2.3 เพื่อศึกษาและทำความเข้าใจวิธีการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุว่ามีกระบวนการอย่างไร ต้องวิเคราะห์และดำเนินการอย่างไรจึงจะถูกต้อง และทำให้โปรแกรมที่ออกมามีความถูกต้องและมีประโยชน์มากที่สุด

1.2.4 เพื่อศึกษาและทำความเข้าใจในการใช้ประโยชน์จากเครื่องมือช่วยในพัฒนาโปรแกรม(CASE Tools) ว่ามีตัวใดบ้าง ใช้งานในกระบวนการใดบ้าง ใช้อย่างไร มีความสามารถใดบ้าง

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้จะเป็นการดำเนินการศึกษาและทำความเข้าใจการพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ และการใช้ประโยชน์จากเครื่องมือช่วยในการพัฒนาโปรแกรม และนำความรู้ดังกล่าวเข้ามาช่วยในการพัฒนาระบบทะเบียนนักศึกษาในระดับบัณฑิตศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ให้มีความสามารถและมีความสะดวกมากขึ้น โดยให้นักศึกษาสามารถดำเนินการเกี่ยวกับระบบทะเบียนและเข้าถึงข้อมูลของตนโดยผ่าน โปรแกรมประยุกต์บนอินเทอร์เน็ต(Web Application) และเจ้าหน้าที่ทะเบียนสามารถดำเนินงานโดยผ่านโปรแกรมประยุกต์ผ่านระบบเครือข่ายภายในสถาบันได้ รวมถึงได้ทำการวิเคราะห์และปรับขั้นตอนการทำงานของงานทะเบียนนักศึกษาให้มีลำดับขั้นตอนอย่างชัดเจน ให้มีความคล่องตัวมากขึ้น และสามารถกำหนดระยะเวลาในการดำเนินการต่าง ๆ ให้ได้

แต่อย่างไรก็ตามขอบเขตที่จะดำเนินการนั้นจะครอบคลุมงานที่เกี่ยวข้องกับงานทะเบียนของนักศึกษาเป็นหลักไม่จะเป็นการลงทะเบียน การเปลี่ยนแก้ไขรายการลงทะเบียน ข้อมูลประวัตินักศึกษา เกรดนักศึกษา ตารางเรียน ตารางสอบ การดำเนินการตามคำร้องต่าง ๆ และอื่น ๆ ที่เกี่ยวข้อง

อีกประการหนึ่งก็คือในการพัฒนาระบบดังกล่าวนี้จะพยายามไม่ทำการเปลี่ยนแปลงลักษณะสำคัญของระบบเดิม เพื่อไม่ให้กระทบกับตัวระบบเดิม และระบบอื่นที่เกี่ยวข้องมากจนเกินไป ดังนั้นจึงมีงานบางอย่างที่อาจมีความยุ่งยากมากกว่าที่ควรจะเป็น แต่ถึงอย่างนั้นก็ยังเป็นการเพิ่มความสะดวกและถูกต้องให้มากขึ้นในระดับที่น่าพอใจ และเอื้อต่อการนำไปพัฒนาต่อในอนาคต

1.4 วิธีการดำเนินงาน

งานวิจัยในโครงการนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งมีแบ่งเป็น 2 ส่วนหลัก ๆ นั่นก็คือ เรื่องของกระบวนการพัฒนาโปรแกรมเชิงวัตถุ ยูเอ็มแอล(Unified Modeling Language: UML) ซึ่งกล่าวถึงในบทที่ 2 และระบบทะเบียนศึกษาทั้งในระดับบัณฑิตศึกษา และระดับต่ำกว่าบัณฑิตศึกษา โดยการเข้าไปทำการสัมภาษณ์เจ้าหน้าที่ที่ปฏิบัติงานอยู่ในระบบเดิม รวมทั้งอาจารย์และนักศึกษาในระดับบัณฑิตศึกษาอีกด้วย และยังเข้าไปศึกษาถึงรูปแบบฟอร์มต่าง ๆ ที่ใช้อยู่เดิมด้วย แล้วนำความรู้ดังกล่าวมาวิเคราะห์ และใช้ความรู้ใน 2 เรื่องแรกมาเขียนเป็นยูเอ็มแอลเพื่ออธิบายมุมมองต่าง ๆ ของระบบดังกล่าวซึ่งมีรายละเอียดอยู่ในบทที่ 5

ทฤษฎีอีกส่วนหนึ่งที่ต้องทำการศึกษาก็คือเรื่องของเครื่องมือที่ใช้ในการพัฒนาโปรแกรม ไม่ว่าจะเป็น DB2 ซึ่งเป็นโปรแกรมจัดการฐานข้อมูล และโปรแกรมที่ใช้ในการเขียนโปรแกรมนั้นคือ IBM VisualAge รวมถึงส่วนของภาษาจาวาด้วย ซึ่งจะมียูเอ็มแอลในรายละเอียดในบทที่ 3, และ 4 ตามลำดับ

หลังจากที่เราได้โมเดลของระบบแล้วสิ่งที่เราต้องทำการออกแบบต่อไปก็คือเรื่องของฐานข้อมูล ซึ่งเราก็นำข้อมูลจากคลาสต่าง ๆ ที่ได้เราออกแบบไว้แล้วนั้นมาเป็นข้อมูลในการเขียนอีอาร์ไดอะแกรม(ER-Diagram) แล้วนำมาแปลงให้เป็นโครงสร้างตารางข้อมูลต่อไป ขั้นตอนต่อจากนี้ไปก็จะเป็นการเขียนโปรแกรมทั้งในส่วนของแอปพลิเคชัน และเว็บแอปพลิเคชัน ทั้งสองส่วนนี้ใช้ภาษาจาวาในการพัฒนาซึ่งจะกล่าวถึงในบทที่ 6

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 และวิธีสุดท้าย วิธีแบบจำลองรูปภาพเชิงวัตถุทางวิศวกรรมซอฟต์แวร์ (Object Oriented Software Engineering: OOSE) จะมีจุดแข็งในคุณสมบัติของการวิเคราะห์แต่มีจุดอ่อนในจุดต่างๆอีก เมื่อเวลาผ่านไป การใช้วิธีนำเสนอรายละเอียดงานที่ต่างกัน ทำให้เกิดความสับสน ดังนั้นจึงจำเป็นต้องมีวิธีที่มาตรฐานที่สามารถเข้าใจได้ง่าย และยังคงอยู่บนพื้นฐานของวิธีเดิมที่มีอยู่ UML จึงได้ถือกำเนิดขึ้นมาจากการร่วมมือกันของทั้ง Booch ,Jacopson ,Rumbaugh รวมถึง คนอื่น โดยได้นำข้อดีของแต่ละวิธีมารวมกัน

Unified Model Language หรือ UML เป็นภาษาโมเดลที่นำเอาสัญลักษณ์มาช่วยในการเพิ่มประสิทธิภาพในส่วนของการวิเคราะห์(Analysis) และออกแบบ(Design) ซึ่งมีประโยชน์ในหลายด้าน โดยจะเป็นรูปแบบของออปเจกต์โอเรียนเท็ด ซึ่งการออกแบบจะทำให้เราสามารถวิเคราะห์ส่วนประกอบโดยรวมของระบบ และยังสามารถที่จะรองรับภาษาที่จะใช้ในการพัฒนาได้หลากหลายภาษามาก และยังหลีกเลี่ยงความซ้ำซ้อนและเฉพาะเจาะจงเกินไป ซึ่งการออกแบบโดยวิธี UML นี้สามารถที่จะแสดงออกมาในรูปแบบของไดอะแกรม (Diagram)ต่างๆ

แอปพลิเคชันขั้นสูงที่พัฒนาขึ้นนั้นต่างก็มีขนาด ความซับซ้อน และการกระจายที่มากขึ้น การพัฒนาซอฟต์แวร์ให้สามารถทำงานได้ทั้งเร็ว และถูกต้องนั้น จึงจำเป็นต้องมีการพัฒนาร่วมกันเป็นทีม ซึ่งความยากอย่างหนึ่งของทีมพัฒนาซอฟต์แวร์ก็คือการสื่อสารระหว่างนักพัฒนาในทีม เนื่องจากเมื่อซอฟต์แวร์มีขนาดใหญ่ นักพัฒนาในทีมก็มีจำนวนมากขึ้นตามขนาดของซอฟต์แวร์ และแต่ละท่านก็มีความเชี่ยวชาญในด้านต่างๆกัน นอกจากนั้นเทคโนโลยีในการพัฒนายังมีการเปลี่ยนแปลงที่รวดเร็วมาก

นอกจากนั้น ปัญหาของการพัฒนาซอฟต์แวร์ ส่วนหนึ่งเกิดจากการที่นักพัฒนาไม่สามารถจัดการกับความต้องการที่เปลี่ยนแปลงของแต่ละโมดูล รวมถึงซอฟต์แวร์ที่พัฒนาขึ้นนั้นไม่สามารถทำงานร่วมกันได้ และยังดูแลรักษาและเปลี่ยนแปลงได้ยาก นักพัฒนาในทีมไม่สามารถติดตามได้ว่าใครแก้ไขอะไร เมื่อใด ที่ไหน และทำไม

สาเหตุสำคัญที่ทำให้เกิดปัญหาในการพัฒนาซอฟต์แวร์ก็คือ ไม่มีระบบจัดการความต้องการที่ดีเพียงพอ มีการสื่อสารที่ไม่รัดกุมและกำกวม สถาปัตยกรรมของซอฟต์แวร์ไม่มั่นคง

ดังนั้น การจัดการกระบวนการในการพัฒนาจึงจำเป็นต้องการพัฒนาซอฟต์แวร์ให้ถูกต้อง และมีคุณภาพตามที่ต้องการ จึงจำเป็นต้องอย่างยิ่งที่จะต้องศึกษาการพัฒนาระบบด้วยวิธีเชิงวัตถุ, กระบวนการพัฒนาซอฟต์แวร์ที่ดี, ส่วนประกอบของยูเอ็มแอล และเครื่องมือที่ใช้วิเคราะห์และออกแบบซึ่งใช้ Rational Rose เพื่อการพัฒนา ระบบซอฟต์แวร์ที่มีคุณภาพตามความต้องการมากที่สุด

2.2 การพัฒนาระบบด้วยวิธีเชิงวัตถุ

การพัฒนาระบบด้วยวิธีเชิงวัตถุนั้นเป็นการพัฒนาที่ช่วยลดความยุ่งยากซับซ้อนของระบบใหญ่ๆ ได้ โดยผู้พัฒนาจะมองระบบและส่วนประกอบหรือระบบย่อย (Sub System) เป็นวัตถุ (Object) เนื่องจากสิ่งต่างๆ ที่มีอยู่ในโลกความเป็นจริงนั้นก็ถือได้ว่าเป็นวัตถุได้อยู่แล้ว และด้วยการมองระบบเป็นวัตถุทำให้ผู้พัฒนาระบบสามารถมองระบบได้ชัดเจนมากยิ่งขึ้น ส่งผลให้การวิเคราะห์และออกแบบระบบทำได้ง่ายและมีความถูกต้องมากขึ้น

การพัฒนาระบบด้วยวิธีเชิงวัตถุนั้น ยังมีส่วนเกี่ยวข้องกับทุกขั้นตอนของการพัฒนาระบบ ทั้งการรวบรวมความต้องการ (Requirement gathering), การวิเคราะห์ (analysis), การออกแบบ (design), การเขียนโปรแกรม (coding), และการทดสอบ (testing) ซึ่งเป็นการแก้ปัญหาของการพัฒนาระบบสมัยก่อนแบบที่เรียกว่า การพัฒนาแบบโครงสร้าง (Structural) ซึ่งเป็นการบอกเพียงว่าระบบจะมีลักษณะการทำงานอย่างไรแต่ไม่ได้บอกถึงวิธีการพัฒนาระบบหรือรูปแบบการเขียนโปรแกรมเลย จุดนี้เองทำให้หลายๆ ครั้ง ระบบที่พัฒนาขึ้นมีปัญหาเนื่องจากการตรวจสอบความถูกต้องของระบบ (Validate) ระบบในเชิงพฤติกรรม (Behavior) นั้นทำได้ยาก

2.3 ข้อดีของการพัฒนาระบบด้วยวิธีเชิงวัตถุ

2.3.1 Higher level of abstraction วิธีการวิเคราะห์ระบบแบบโครงสร้างนั้นมีข้อจำกัดในการแยกแยะเอกลักษณ์ และการซ่อนรายละเอียด อย่างไรก็ตามการวิเคราะห์และพัฒนาระบบแบบโครงสร้างนี้ จะแยกแยะโครงสร้างและการทำงานของระบบออกจากกัน เพื่อให้เกิดประสิทธิภาพในการทำงาน ตัวอย่างเช่น เซนเซอร์ จะได้รับการโมเดลให้เป็นส่วนหนึ่งของข้อมูลการอ่านหรือการกระทำอื่นๆ กับเซนเซอร์จะถือว่าเป็นโอเปอเรชัน (Operation) แต่การโมเดลนี้ไม่ได้แสดงความสัมพันธ์ระหว่างข้อมูลและโอเปอเรชัน

ส่วนโมเดลของการพัฒนาเชิงวัตถุนั้นจะแสดงความสัมพันธ์ระหว่างข้อมูลกับโอเปอเรชันที่กระทำกับข้อมูลอย่างใกล้ชิด เนื่องจากโมเดลนี้อ้างอิงมาจากโลกจริงๆ การแยกแยะเอกลักษณ์ของการพัฒนาเชิงวัตถุนั้น ให้ความเข้าใจและความสามารถที่ดีกว่า รวมทั้งมีการตั้งชื่อสำหรับวัตถุนั้นมาจากชื่อที่มีอยู่จริงในโดเมนของปัญหา มุมมองของวัตถุจะเป็นการมองในระดับสูงและใกล้เคียงกับโดเมนของปัญหามากกว่า

2.3.2 Seamless transition among different phases of software development การพัฒนาซอฟต์แวร์นั้นต้องการวิธีการ (Methodology) ที่แตกต่างกันในแต่ละช่วงของการพัฒนา การเปลี่ยนแปลงจากช่วงการทำงานหนึ่งไปช่วงอื่นๆ ในแต่ละ Process จะมีความซับซ้อน และยังไม่มีความสอดคล้องกัน และยังซ้ำด้วย และยังทำให้ขนาดของงานใหญ่ขึ้น และมีโอกาสที่จะเกิดความผิดพลาดขึ้นด้วย แต่วิธีการของเชิงวัตถุนั้นจะใช้ภาษาเดียวกันในการติดต่อระหว่างการวิเคราะห์ การออกแบบ การเขียนโปรแกรม และการออกแบบฐานข้อมูลด้วย ทำให้ลดระดับของความซับซ้อนลง และช่วยให้การเข้าใจระบบก็ชัดเจนด้วย

2.3.3 Encouragement of good programming techniques เป็นการสนับสนุนเทคนิคการเขียนโปรแกรมที่ดี เพราะการเขียนโปรแกรมเชิงวัตถุ นั้นจะเป็นการเขียนในลักษณะของคลาส ซึ่งบอกว่าคลาสนี้ทำอะไร และทำอะไร และคุณสมบัติของระบบนั้นสามารถที่จะรวมคลาสเข้าด้วยกันเป็นระบบย่อยได้ และการเปลี่ยนแปลงคลาสหนึ่งก็จะมีผลกระทบต่อคลาสอื่นๆ ด้วย อย่างไรก็ตามภาษาต่างๆ เช่น C++, Smalltalk หรือ Java

ก็เพิ่มส่วนที่สามารถรองรับการออกแบบเชิงวัตถุด้วยเพื่อง่ายต่อการสร้าง และการนำ Code มาใช้ใหม่กับแนวคิดของคลาสและการสืบทอด (Inheritance)

2.3.4 Promotion of Reusability ระบบแบบโครงสร้างมีข้อจำกัดในการนำสิ่งที่พัฒนาไปแล้วกลับมาใช้ใหม่ แต่ในความเป็นจริงแล้ว ถ้าผู้พัฒนาระบบมีคอมโพเนนท์ อยู่แล้วก็ควรจะใช้ได้เลย แต่ระบบแบบโครงสร้างนี้ผู้ที่นำมาใช้ต้องทำการเปลี่ยนแปลงซอร์สโค้ด (Source Code) เพื่อให้เหมาะสมกับความต้องการของระบบใหม่เพื่อที่จะใช้งานได้ แต่การพัฒนาเชิงวัตถุนั้นมีคุณสมบัติที่สนับสนุนการนำกลับมาใช้ใหม่โดยตรง คือ Generalization และ Refinement โดย Generalization คือการสืบทอดคุณสมบัติโดยจะได้คุณสมบัติเดิมทุกประการและยังสามารถเพิ่มและขยายคุณสมบัติได้ด้วย โดยไม่ต้องมีการเปลี่ยนแปลงซอร์สโค้ดเก่าเลย ลักษณะการนำมาใช้ใหม่นี้เรียกว่า การเขียนโปรแกรมแบบแตกต่าง ส่วน Refinement นั้นก็คล้ายๆกับ Generalization แต่จะยอมให้นักพัฒนาใช้วัตถุที่ไม่สมบูรณ์โดยการเพิ่มส่วนที่หายไป ตามที่ตนเองต้องการได้

2.3.5 Promotion of Scalability ระบบโครงสร้างมีข้อจำกัดในการปรับเปลี่ยนขนาดของระบบ ทั้งนี้เนื่องจากระบบแบบโครงสร้างมี การแยกแยะเอกลักษณ์และการซ่อนรายละเอียดที่ไม่ดี ทำให้ระบบแบบโครงสร้างมีความซับซ้อนมากขึ้นเมื่อมีปัญหาที่เกิดขึ้นกับระบบมากขึ้น แต่ระบบที่พัฒนาด้วยวิธีเชิงวัตถุ นั้น จะแยกแยะเอกลักษณ์และการซ่อนรายละเอียดที่ดีส่งผลให้คอมโพเนนท์แต่ละส่วนแยกออกจากกันอย่างชัดเจน และการใช้เครื่องหมาย (Notation) เดียวกันตลอดการพัฒนา ทำให้สามารถเปลี่ยนจากภาวะวิเคราะห์เป็นการออกแบบได้ง่าย การปรับเปลี่ยนระบบให้มีขนาดใหญ่ขึ้นก็สามารถทำได้ง่ายตามด้วย

2.3.6 Promotion of Reliability and Safety เนื่องจากการพัฒนาเชิงวัตถุมีการแยกแยะเอกลักษณ์และการซ่อนรายละเอียดที่ดีกว่าทำให้การสื่อสารระหว่างวัตถุที่ไม่เกี่ยวข้องกันจำกัดอยู่ด้วยอินเทอร์เฟซ ที่ผู้พัฒนาได้กำหนดไว้เท่านั้น ส่งผลให้ระบบที่พัฒนาด้วยวิธีเชิงวัตถุมีความน่าเชื่อถือได้มากกว่าเพราะว่าผู้พัฒนาสามารถควบคุมการติดต่อระหว่างคอมโพเนนท์ที่ได้ และยังสามารถเพิ่มเงื่อนไขก่อนหน้าและท้าย (Pre and Post Condition) เพื่อให้ระบบสามารถทำงานได้อย่างถูกต้อง

2.3.7 Promotion of Concurrency การทำงานไปพร้อมๆกัน (Concurrency) เป็นรูปแบบการทำงานของสิ่งที่มีตัวตนในโลกแห่งความ เป็นจริง และยังเป็นคุณสมบัติสำคัญของคอมพิวเตอร์ยุคใหม่ด้วย การพัฒนาแบบโครงสร้างไม่มีสัญลักษณ์สำหรับการทำงานพร้อมกัน การจัดการงาน (Task) หรือการซิงโครไนเซชัน (Synchronization) ระหว่างงานดังนั้นโครงสร้างที่สำคัญๆของระบบอาจจะไม่สามารถพัฒนาขึ้นได้โดยการใช้รูปแบบพัฒนาแบบโครงสร้างมาตรฐานได้ แต่การพัฒนาแบบวิธีเชิงวัตถุสนับสนุนการทำงานพร้อมกัน ในตัวเอง และรายละเอียดของงาน การทำ Synchronization ระหว่างงานก็สามารถแสดงออกมาในแผนภาพ (Statecharts) ของวัตถุที่ทำงานแบบแอคทีฟ (Active Object) และการส่งแมสเสจ (Message) ระหว่างวัตถุ แผนภาพเหล่านี้เป็นเครื่องมือที่มีความเหมาะสมสำหรับการสร้างระบบที่มีเงื่อนไขทางด้านประสิทธิภาพ

2.4 ความสามารถของคลาส

2.4.1 ลำดับชั้นของคลาส Class Hierarchy

2.4.1.1 Inheritance คือการสืบทอดคุณสมบัติเป็นการสร้างคลาสใหม่ขึ้นมา โดยมีพื้นฐานมาจากคลาสเดิม แต่จะมีข้อมูล (Data) หรือเมธอด ที่พิเศษเป็นของตัวเองเพิ่มขึ้นมาจากคลาสพื้นฐานเดิม ซึ่งก่อให้เกิดการแตกสายพันธุ์ของวัตถุ ขึ้นเป็นวัตถุใหม่ที่มีคุณสมบัติของวัตถุเดิมซ่อนอยู่ไม่มากนักน้อยนั่นเอง

ปัญหาสำคัญของการสืบทอดคุณสมบัติ อยู่ตรงที่ว่า ในการสร้างคลาสขึ้นมาใหม่แต่ละครั้งนั้น คลาสเหล่านั้นมีอะไรที่เป็นพื้นฐานเหมือนกันและมีคุณสมบัติอะไรที่แตกต่างกัน นักพัฒนาจะจำแนกคลาสเหล่านั้นได้อย่างไร จึงจะเหมาะสมกับพฤติกรรมการทำงานของมัน คำตอบก็คือ จะต้องเริ่มต้นมองที่ยอดบนของลำดับชั้นของคลาสดีก่อน คลาสที่อยู่สูงที่สุดจะต้องเป็นคลาสที่มีคุณสมบัติพื้นฐาน (Simplest) ของคลาสต่างๆ ให้มากที่สุด แล้วค่อยๆ แบ่งลำดับชั้นออกไปตามคุณสมบัติอื่นๆ ที่รองลงไป โดยเมื่อคุณสมบัติอย่างหนึ่งที่กำหนดลงในคลาสพื้นฐานแล้ว ทุกๆ คลาสที่สืบทอดคุณสมบัติภายใต้คลาสนั้นจะได้รับคุณสมบัติติดตามไปด้วยอัตโนมัติ

2.4.1.2 Multiple Inheritance หรือ MI เป็นการสร้างคลาสขึ้นมาโดยการสืบทอดคุณสมบัติมาจากคลาสพื้นฐานที่มีมากกว่า 1 ตัว ทำให้ภาษาในลักษณะของการเขียนโปรแกรมเชิงวัตถุมีความยืดหยุ่นในการทำงานเพิ่มมากขึ้นอีกด้วย

2.4.2 แพ็กเกจ (Packages) เป็นการรวมคลาสที่มีความสัมพันธ์ใกล้ชิดกันเข้าไว้ในแพ็กเกจ เพื่อประโยชน์สำหรับระบบที่มีคลาสเป็นจำนวนมาก ในระบบที่มีความซับซ้อนเราอาจจะสร้างแพ็กเกจขึ้นมาก่อนแล้วจึงบรรจุคลาสและความสัมพันธ์เข้าไว้ในแพ็กเกจ

2.4.3 Encapsulation and Information Hiding เอ็นแคปซูลชัน คือ การรวมกันของโครงสร้างข้อมูล (Data Structure) กับฟังก์ชันที่เกี่ยวข้องหรือเรียกใช้ข้อมูลนั้น (เรียกว่าเมธอด หรือ Action) เกิดเป็นวัตถุตัวใหม่ที่สามารถทำการซ่อนข้อมูลจากภายนอกได้ ซึ่งจะเรียกดั้งเดิมว่า คลาส นั่นเอง

ในการพัฒนาโดยวิธีแบบโครงสร้างในสมัยก่อนนั้น ส่วนที่เป็นโครงสร้างข้อมูลและฟังก์ชันที่เรียกใช้ข้อมูลนั้นจะถูกสร้างขึ้นแยกจากกัน นั่นคือจะมองข้อมูลและฟังก์ชันเป็นโมดูลเดี่ยวๆ ไม่มีความสัมพันธ์กันแต่อย่างใด ข้อมูลชุดหนึ่งๆ อาจมีฟังก์ชันหลายๆ ตัวมาเรียกใช้งานโดยตรง ซึ่งแต่ละฟังก์ชันอาจจัดการกับโครงสร้างของข้อมูลไม่เหมือนกัน การเข้าถึงข้อมูลโดยตรงของฟังก์ชันเหล่านี้จะนำไปสู่ปัญหาต่อไป

เอ็นแคปซูลชันก้าวมาในจุดนี้ ผลจากการที่นำโครงสร้างข้อมูลและฟังก์ชันที่ใช้งานข้อมูลนั้นมารวมกันสร้างความสัมพันธ์ให้กันและกัน จะทำให้ข้อมูลมีความมั่นคงขึ้น โดยการจะเข้ามาใช้ข้อมูลจะต้องเรียกผ่านเมธอดเท่านั้น ผลก็คือ ไม่มีการเข้าถึงข้อมูลโดยตรงอีกต่อไป ซึ่งนั่นก็คือ เป็นการทำ Data Hiding นั่นเอง

ข้อดีอีกประการหนึ่งของเอ็นแคปซูลชัน ก็คือ เราสามารถแบ่งระดับของการเข้าถึงข้อมูลได้โดยตัวของเมธอดของวัตถุเอง

2.4.4 Polymorphism โพลิมอร์ฟิซึม หมายถึงการที่คลาสหนึ่งๆ สามารถเปลี่ยนไปได้หลายรูปร่างขึ้นอยู่กับสภาพแวดล้อมหรือสถานการณ์ในขณะนั้น

โดยจะเป็นการยอมให้ผู้พัฒนาใช้เมธอดชื่อเดียวกัน แต่การทำงานไม่เหมือนกัน จำแนกไปตามชนิดของข้อมูลหรือเมสเสจ(Message) ที่เมธอดจะได้รับขณะทำงาน เมื่อได้รับข้อมูลที่ต้องการแล้วก็จะไปเลือกเมธอดที่เหมาะสมที่สุดกับข้อมูลนั้นๆ ไปทำงานเองต่อไป

ความสามารถอีกอย่างหนึ่งของโพลิมอร์ฟิซึมก็คือ สามารถกำหนดการดำเนินการใหม่ให้กับตัวดำเนินการหรือเมธอดได้ เรียกว่าการทำ " Overloading " ยังผลให้เราสามารถเรียกชื่อเมธอดหรือตัวดำเนินการนั้น มาใช้งานกับวัตถุต่างๆ ในกรณีที่แตกต่างกัน ขึ้นอยู่กับว่าในขณะนั้น โปรแกรมกำลังทำงานอยู่กับวัตถุอะไร

2.4.5 ความสัมพันธ์

2.4.5.1 Generalization เป็นความสัมพันธ์แบบการให้กำเนิดคลาสย่อย (Subclass) จากคลาสแม่ (Superclass) โดยคลาสย่อยจะสืบทอดแอตทริบิวต์, โอเปอเรชันและความสัมพันธ์ทั้งหมดจากคลาสแม่ ทั้งของตัวเองและจากคลาสแม่ของคลาสแม่ขึ้นไปเรื่อยๆ คลาสย่อยอาจมีการเพิ่มแอตทริบิวต์และโอเปอเรชันในคลาสของมันเอง ความสัมพันธ์แบบนี้อาจถูกเรียกว่า is-a หรือ kind-of

2.4.5.2 Aggregation เป็นความสัมพันธ์แบบเป็นส่วนประกอบหรือที่เรียกว่า Part-of คำว่าเป็นส่วนประกอบบางครั้งอาจมีความหมายที่ต่างกัน เช่น เครื่องยนต์และล้อเป็นส่วนประกอบของรถยนต์ในขณะที่เรากำลังขับรถอยู่ แต่ถ้ารถยนต์คันนั้นอยู่ในระหว่างการซ่อมที่อู่ซ่อมรถ นั่นคือเครื่องยนต์และล้อก็ไม่ได้เป็นส่วนประกอบของรถยนต์แล้ว

2.4.5.3 Association เป็นความสัมพันธ์แบบสองทิศทาง ซึ่งใช้ในการติดต่อระหว่างคลาส ความสัมพันธ์แบบ แอสโซซิเอชันระหว่างคลาสหมายถึง การเชื่อมต่อระหว่างออบเจกต์ในคลาสทั้งสอง ที่ปลายของเส้นความสัมพันธ์จะมีการระบุบทบาทเพื่อบอกว่าคลาสนั้นถูกมองว่าเป็นอย่างไรจากคลาสอื่น

2.4.5.4 Constraints เป็นการกำหนดข้อบังคับของความสัมพันธ์ระหว่างเอนติตี้ โดยเอนติตี้จะหมายถึงออบเจกต์, คลาส, บทบาท, แอตทริบิวต์, การเชื่อมต่อ (Link) และความสัมพันธ์ ข้อบังคับนี้จะเป็นการจำกัดค่าที่เอนติตี้สามารถเป็นไปได้

2.5 Object Oriented Methodologies

วิธีการพัฒนาระบบเชิงวัตถุนี้มีมากมายหลายวิธี โดยแต่ละวิธีก็มีจุดเด่นและจุดด้อย แตกต่างกันไป โดยผู้พัฒนาควรจะใช้วิธีการที่มีประสิทธิภาพที่ดีที่สุด เพื่อระบบที่ดีที่สุด โดยจะใช้วิธีผสมนำข้อดีของแต่ละวิธีมาใช้ก็ได้ดังเช่น Unified Modeling Language (UML) ที่นำข้อดีของ Rambaugh et al., Booch และ Jacobson มาใช้ และยังจะสามารถใช้เทคนิคต่างๆ ที่เกิดขึ้นมาพัฒนาด้วยเช่น Patterns และ Frameworks เป็นต้น

2.5.1 Rambaugh et al.'s Object Modeling Technique (OMT) ถูกนำเสนอโดย Jim Rambaugh และผู้ร่วมงาน โดยจะบอกถึงวิธีการวิเคราะห์ ออกแบบ และ ลงมือทำ ของระบบโดยวิธีเชิงวัตถุ OMT จะมีรายละเอียดต่างๆ เช่น คลาส, แอตทริบิวต์, เมธอด, การสืบทอดคุณสมบัติและแอสโซซิเอชันสามารถแสดงได้อย่างง่าย โดย OMT จะประกอบไปด้วย 4 ขั้นตอนต่อไปนี้ และในแต่ละขั้นตอนสามารถกลับมาทำซ้ำอีกหลายครั้งได้

- การวิเคราะห์ (Analysis)
- ออกแบบระบบ (System design)

- ออกแบบวัตถุ (Object design)
- การสร้าง (Implementation)

OMT แบ่งแยก model ต่างๆออกเป็น 3 ส่วนดังต่อไปนี้คือ

- Object model สามารถนำเสนอได้โดยใช้โมเดลวัตถุและการอธิบายข้อมูล(Data dictionary)
- Dynamic model สามารถนำเสนอได้โดยใช้สเตทไดอะแกรมและอีเวนทโฟลไดอะแกรม
- Functional model สามารถนำเสนอได้โดยใช้ดาต้าโฟลไดอะแกรมและข้อบังคับต่างๆ

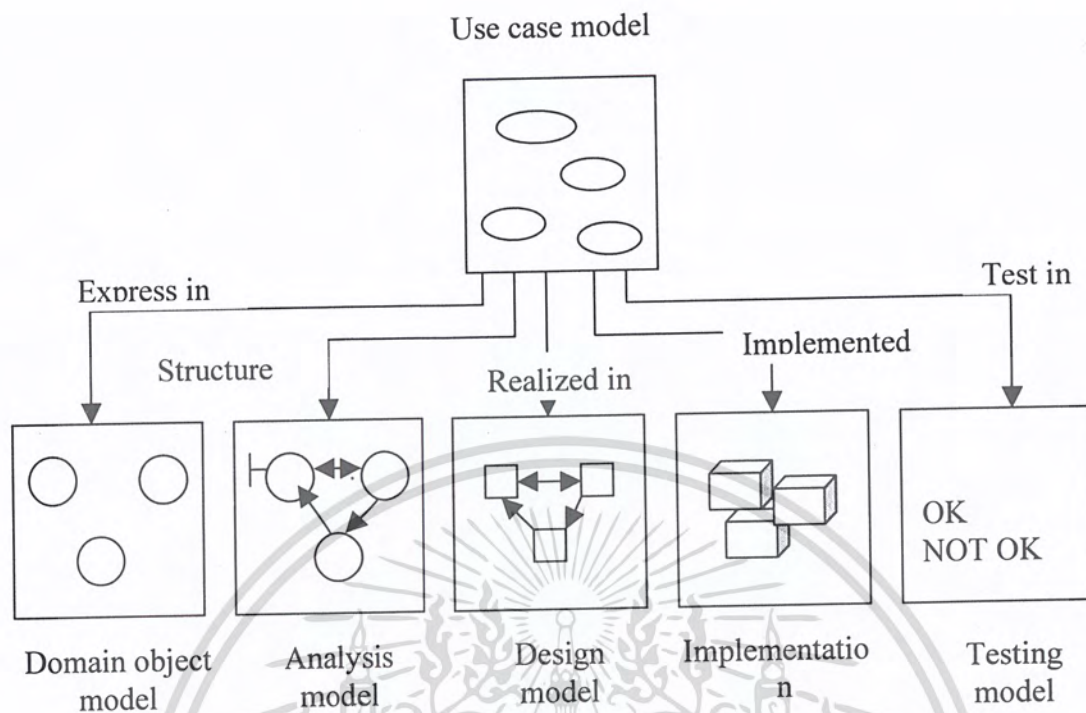
2.5.2 The Booch Methodology เป็นที่นิยมใช้กันอย่างกว้างขวางโดยวิธีเชิงวัตถุซึ่งช่วยให้ออกแบบระบบที่ใช้ วัตถุ เป็นแบบอย่างได้ โดยครอบคลุมไปถึงการวิเคราะห์และการออกแบบ วิธีการของ Booch ประกอบไปด้วย ไดอะแกรมต่างๆต่อไปนี้

- คลาสไดอะแกรม
- ออบเจกต์ไดอะแกรม
- สเตททรานซิชันไดอะแกรม
- โมดูลไดอะแกรม
- โพเชสไดอะแกรม
- อินเตอร์แอคชันไดอะแกรม

2.5.3 The Jacobson et al. Methodologies หัวใจหลักของพัฒนาระบบ วิธีนี้คือ หลักการซึ่งทำให้เกิด Objectory ขึ้นมา

2.5.3.1 Object Oriented Software Engineering: เรียกว่า Objectory ซึ่งเป็นวิธีการพัฒนาระบบด้วยวิธีเชิงวัตถุ ซึ่งสร้างโมเดลที่แตกต่างกันดังนี้

- ยูสเคสโมเดล (Use case Model) ซึ่งกำหนดสิ่งที่มากกระทำกับระบบ (Actors) และมาใช้ระบบอย่างไร
- โดเมนออบเจกต์โมเดล (Domain object Model) เป็นการแมปออบเจกต์ที่มีอยู่จริงให้เป็นไปตามโดเมนของระบบ
- โมเดลการวิเคราะห์ (Analysis object Model) เป็นการนำเสนอว่าควรสร้างระบบอย่างไร มีการส่งข้อมูลอย่างไรบ้าง
- โมเดลของการสร้าง (Implementation model) เป็นการแสดงการสร้างระบบ
- โมเดลการทดสอบ (Test model) เป็นการวางแผนการทดลองระบบที่สร้างขึ้น และทำรายงานต่างๆ



รูปที่ 2-1 การเชื่อมโยงของ Use Case Model กับ Model อื่นๆ

2.5.3.2 Object Oriented Business Engineering (OOBE) เป็นวิธีการเชิงวัตถุที่มีระดับความเสี่ยง โดยใช้ยูสเคสเป็นตัวหลักในการทำโมเดลต่างๆ ตาม Software engineering process โดยแบ่งเป็นขั้นตอนต่างๆ ดังนี้

- Analysis phase เป็นการวิเคราะห์และกำหนดว่าระบบสร้างขึ้นมาเพื่อทำอะไร ตามโมเดลของโดเมนปัญหาของระบบ (Domain object model), โมเดลของความต้องการ (Requirement model) และ โมเดลการวิเคราะห์ (analysis model)
- Design and implementation phase เป็นการกำหนดสิ่งที่จำเป็นต่างๆ ในโมเดลของการออกแบบ (Design model) ซึ่งจะรวมแพ็คเกจต่างๆ ไว้ เช่น ระบบการจัดการฐานข้อมูล (DBMS), การกระจายโปรเซสการเขียนโปรแกรมการใช้เครื่องมือ (Tools) ต่างๆ เป็นต้น
- Testing phase สุดท้ายเป็นการทดสอบระบบโดย Jacobson ได้แสดงไว้หลายวิธีในระดับต่างๆ คือการทดสอบในระดับยูนิต (unit testing), การทดสอบในระดับการทำงานร่วมกัน (Integrate testing) และ การทดสอบระบบ(system testing)

2.6 Pattern และ Frameworks

Pattern เป็นการจัดรูปแบบการแก้ปัญหาให้อยู่ในแนวทางเดียวกันซึ่งมีหลายรูปแบบ เช่น แบบ Model-View-Controller, Master Slave, Microkernel, Proxy, and Observer เป็นต้น

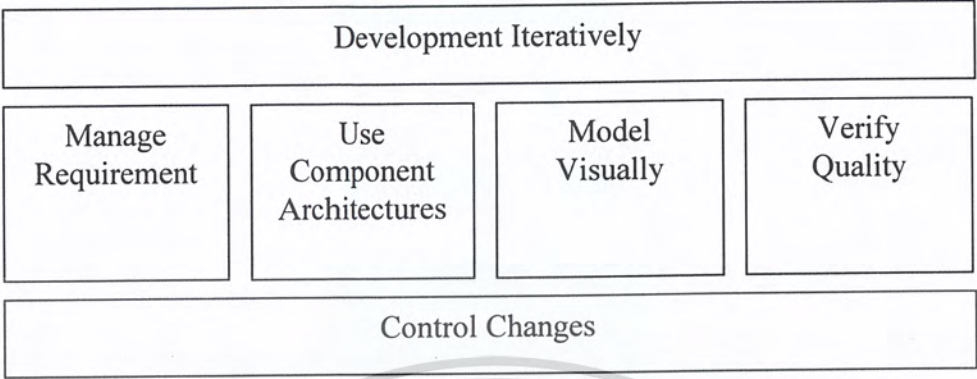
Model-View-Controller (MVC) ประกอบด้วยออบเจกต์ 3 ชนิด คือ Model ประกอบด้วยออบเจกต์หลักของแอปพลิเคชัน, View เป็นการนำเสนอข้อมูลผ่านหน้าต่างผู้ใช้ และ Controller เป็นการกำหนดวิธีการทำงานร่วมกันระหว่างส่วนติดต่อผู้ใช้กับอินพุตจากผู้ใช้

แนวคิดหลักของ MVC คือการแยกโมเดล(ออบเจกต์หลักของแอปพลิเคชัน) ซึ่งมีความสัมพันธ์ใกล้ชิดกับโดเมนธุรกิจออกจากวิวโดยใช้คอนโทรลเลอร์ดูแลการใช้งานจากผู้ใช้ และเป็นตัวกลางระหว่างออบเจกต์ส่วนติดต่อกับออบเจกต์ของโมเดล เหตุผลหลักของการแยกคือในขณะที่ออบเจกต์ของโมเดลมีการเปลี่ยนแปลงน้อยแต่วิวอาจมีการเปลี่ยนแปลงได้บ่อยตามความต้องการของผู้ใช้ และออบเจกต์โมเดลหนึ่งสามารถสัมพันธ์กับวิวได้หลายวิวระบบที่สนับสนุน MVC จำเป็นต้องมีเฟรมเวิร์กซึ่งรับบทบาทเป็น Controller คอยอำนวยความสะดวกการสื่อสารระหว่างวิวกับโมเดล

Framework เป็นกลุ่มของคลาสที่ได้ออกแบบมาเพื่อการนำกลับไปใช้ใหม่ในแอปพลิเคชันเฉพาะประเภทเท่านั้น เฟรมเวิร์กจะวางโครงสร้างของระบบโดยแบ่งเป็นออบเจกต์, คลาส, หน้าที่ได้รับผิดชอบหลัก และวิธีการทำงานร่วมกันของคลาสและออบเจกต์ รวมทั้งกระบวนการควบคุมการทำงานทั้งหมดของระบบ เฟรมเวิร์กจะกำหนดรูปแบบการออกแบบแอปพลิเคชัน ทำให้ผู้พัฒนาสามารถสร้างระบบขึ้นมาจากส่วนประกอบหลักๆที่เฟรมเวิร์กมีให้ ทำให้ผู้พัฒนาสามารถพัฒนาแอปพลิเคชันได้เร็วขึ้นและยังสามารถดูแลรักษาซอฟต์แวร์ได้ง่ายขึ้น การเปลี่ยนแปลงโครงสร้างของเฟรมเวิร์กจะมีผลกระทบต่อโครงสร้างของแอปพลิเคชันด้วย ดังนั้นในการออกแบบเฟรมเวิร์กจึงควรมีการนำแพทเทิร์น ไปประยุกต์ใช้หลายๆแบบเพื่อให้เฟรมเวิร์กมีความยืดหยุ่นและสามารถขยายฟังก์ชันเพิ่มเติมได้ ความแตกต่างระหว่างเฟรมเวิร์กและแพทเทิร์นมีดังนี้

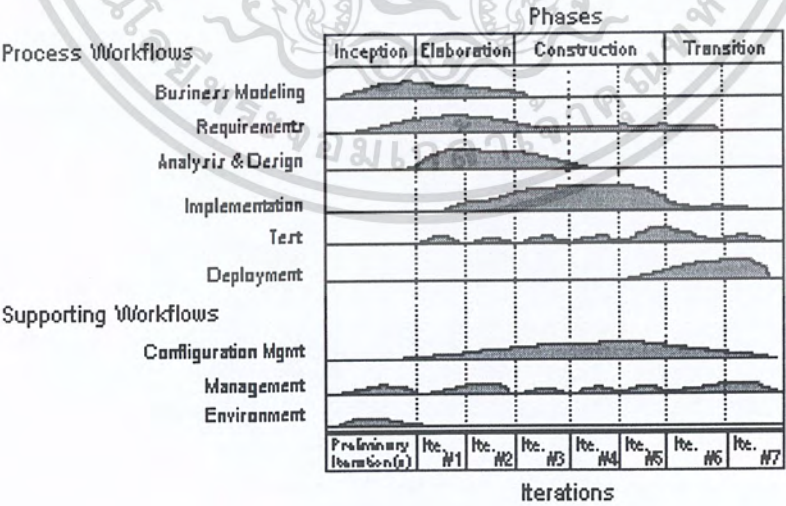
- แพทเทิร์นเป็นสิ่งที่อยู่ในความคิดมากกว่าเฟรมเวิร์ก เนื่องจากเฟรมเวิร์กจะมีโค้ดมาให้ใช้ ทำให้มีความพร้อมในการนำไปใช้หรือการนำกลับไปใช้ แต่แพทเทิร์นผู้พัฒนาจะต้องสร้างโค้ดขึ้นมาเองเมื่อต้องการใช้
- แพทเทิร์นมีส่วนประกอบเล็กกว่าเฟรมเวิร์ก โดยทั่วไปเฟรมเวิร์กจะประกอบด้วยหลายๆแพทเทิร์น
- แพทเทิร์นมีความเจาะจงน้อยกว่าเฟรมเวิร์ก โดยทั่วไปเฟรมเวิร์กนั้นได้รับการออกแบบและพัฒนาขึ้นมาสำหรับงานเฉพาะประเภทเท่านั้น แต่แพทเทิร์นสามารถนำไปใช้กับแอปพลิเคชันได้อย่างกว้างขวาง

2.7 กระบวนการพัฒนาซอฟต์แวร์(Software Process) ที่ดี



รูปที่ 2-2 Best Practices of Software Engineering

2.7.1 การพัฒนาแบบวนรอบ (Development Iteratively) จากรูปจะเห็นได้ว่า ส่วนนี้จะนำไปใช้กับส่วนของ Manage Requirement ,Use Component Architecture ,Model Visually และ Verify Quality ดังนั้นการพัฒนาแบบวนรอบนี้ ถือว่าเป็นกระบวนการที่สำคัญกระบวนการหนึ่งทีเดียว คุณลักษณะของการพัฒนานวนรอบคือ นักพัฒนาจะสามารถวิเคราะห์ และแก้ปัญหาที่มีความเสี่ยงสูง ก่อนที่จะมีการลงทุนพัฒนาระบบ การพัฒนาแบบวนรอบสามารถได้รับความต้องการอย่างต่อเนื่องจากผู้ใช้ได้ นอกจากนั้น การทดสอบและการรวมระบบจะค่อยเป็นค่อยไปอย่างต่อเนื่อง การมี milestone ทำให้นักพัฒนาทราบว่าตอนนี้การพัฒนากำลังเน้นไปที่จุดใด



รูป 2-3 Deployment Iteratively Process

จากรูป แสดงให้เห็นถึงการประยุกต์ใช้การพัฒนาแบบวนรอบ สำหรับในการพัฒนารอบแรกนั้นจะ ให้ความสำคัญกับกระบวนการ Business Modeling มากที่สุด ในรอบต่อไปก็จะให้ความสำคัญ Requirement มากที่สุด แต่ในทุกรอบการทำงานก็จะทำในกระบวนการทุกกระบวนการ แต่จะมากขึ้นแตกต่างกันไปตามรอบของการพัฒนา

2.7.2 การจัดการความต้องการของระบบ (Requirement Management) เป็นกระบวนการที่เป็นระบบ ในการค้นหา จัดโครงสร้าง และทำเอกสารความต้องการของระบบ และเป็นกระบวนการในการจัดตั้ง ดูแลข้อตกลงระหว่างลูกค้า หรือผู้ใช้ กับทีมพัฒนาของโครงการในการเปลี่ยนแปลงความต้องการของระบบ

2.7.3 การใช้สถาปัตยกรรมคอมโพเนนต์(Use Component Architectures) สถาปัตยกรรมซอฟต์แวร์ เป็นโครงสร้างหลักของระบบซอฟต์แวร์และเป็นส่วนหนึ่งที่ได้จากการออกแบบระบบซึ่งเกี่ยวข้องกับโครงสร้างและพฤติกรรมของระบบในด้านของการใช้งาน หน้าที่การทำงาน ประสิทธิภาพ และเงื่อนไขของเทคโนโลยี กระบวนการนี้ จะทำให้สถาปัตยกรรมซอฟต์แวร์มีพื้นฐานอยู่บน หลักการของ Component เป็นหลัก ซึ่งจะทำให้สถาปัตยกรรมที่ได้นั้นมีความยืดหยุ่นและทำให้สามารถดูแลรักษา เพิ่มเติมระบบได้ง่าย และยังสามารถนำกลับมาใช้ได้

2.7.4 แบบจำลองรูปภาพ (Model Visually) - เป็นการนำเอาโมเดลที่เป็นรูปภาพมาช่วย ซึ่งทำให้สามารถเก็บโครงสร้างและพฤติกรรมของสถาปัตยกรรม แสดงส่วนประกอบในระบบ หรือแสดงรายละเอียดของงาน โดยมีการดูแลรักษาความตรงกันของการออกแบบและการพัฒนา และยังช่วยลดความกำกวมในการสื่อสาร ส่วนนี้จะเป็นส่วนที่เรารู้จักถึง UML กัน ซึ่งรายละเอียดจะกล่าวต่อไป

2.7.5 การทดสอบคุณสมบัติ (Verify Quality) การพัฒนาแบบวนรอบนั้นทำให้นักพัฒนาสามารถทดสอบระบบได้ดียิ่งขึ้น ทั้งนี้แทนที่จะทำการทดสอบระบบครั้งใหญ่ครั้งเดียวก่อนที่จะส่งให้กับผู้ใช้ ด้วยการพัฒนาแบบวนรอบจะทำให้มีการทยอยทดสอบในแต่ละรอบ ส่งผลให้ระบบในรอบต่อไปมีคุณภาพมากยิ่งขึ้น

2.7.6 การควบคุมความเปลี่ยนแปลง (Control Changes) หลักการในการจัดการความเปลี่ยนแปลงก็คือจะต้องมีการแยกสถาปัตยกรรมของระบบออกเป็นระบบย่อยๆ และมีการกำหนดความรับผิดชอบของระบบย่อยให้กับทีมพัฒนาหนึ่ง มีการจัดตั้งพื้นที่ทำงานสำหรับนักพัฒนาแต่ละคน มีการกำหนดการรวมกันของแต่ละพื้นที่ทำงาน มีการกำหนดวิธีการควบคุมความเปลี่ยนแปลง มีการบันทึกความเปลี่ยนแปลงในแต่ละรุ่น มีการทดสอบในแต่ละรอบของการพัฒนาระบบ

ระบบที่ถูกดัดนั้น จะมีการจัดการความต้องการความต้องการพื้นฐาน จากนั้นจึงเปรียบเทียบ ความเปลี่ยนแปลงที่เกิดขึ้นโดยใช้สถาปัตยกรรมคอมโพเนนต์ นักพัฒนาสามารถที่จะตรวจสอบรุ่นของ คอมโพเนนต์ที่ใช้โมเดลด้วยรูปภาพ ทำให้สามารถตรวจสอบความเปลี่ยนแปลงได้ง่าย ผลที่ได้จากการตรวจสอบจะต้องสามารถติดตามได้ในแต่ละรุ่นของระบบ จากนั้นจะพัฒนายวนรอบต่อไป

จากรายละเอียดที่ได้กล่าวข้างต้น การสื่อสารระหว่างนักพัฒนา นับเป็นปัญหาหนึ่งในการพัฒนาระบบซอฟต์แวร์ใหญ่ๆ ดังนั้นในกระบวนการการพัฒนาซอฟต์แวร์ที่ดีนั้นจึงนำเอา แบบจำลองรูปภาพ(Visual Modeling) มาใช้ในการแสดงรายละเอียดของงาน แทนที่จะใช้ข้อความในการอธิบายเหมือนระบบในอดีต ซึ่งการใช้ข้อความนั้นอาจจะทำให้ความหมายนั้นกำกวม ต่างกับรูปภาพที่แทนคอมโพเนนต์ ในระบบจริงๆ ที่สามารถสื่อความหมายได้ชัดเจน

2.7.7 แบบจำลองรูปภาพ (Visual Modeling Technology: VMT) VMT คือแนวทางหนึ่งของการพัฒนาระบบ โดยการออกแบบระบบด้วยโมเดลตามแนวความคิดของโลกแห่งความเป็นจริง โมเดลที่ถูกออกแบบมานั้น จะมีประโยชน์ช่วยในการเข้าใจถึงปัญหาของระบบ, การติดต่อกันของผู้ที่จัดทำโครงการงาน (เช่น ลูกค้า, ผู้ดูแลระบบ, นักวิเคราะห์ และนักออกแบบ เป็นต้น) , การเตรียมการทำเอกสาร ,การออกแบบโปรแกรมและดาตาเบส

โมเดลเป็นนามธรรมที่แสดงถึงลักษณะที่สำคัญของปัญหาหรือโครงสร้างของระบบ โดยการลดทอนรายละเอียดที่ไม่สำคัญออก ดังนั้นจึงเป็นการทำให้ปัญหานั้นง่ายต่อการเข้าใจ เราจึงควรสร้างโมเดลสำหรับระบบที่ซับซ้อน เพราะความสามารถของคนที่จะเข้าใจระบบที่ซับซ้อนนั้นมีจำกัด การสร้างโมเดลจะทำให้นักออกแบบสามารถเห็นภาพกว้างของโครงการได้ โมเดลที่ดีควรจะใช้สัญลักษณ์ที่ถูกต้องและมีการตรวจสอบโมเดลให้ตรงความต้องการของระบบ

Visual Modeling Technique จะมีส่วนเกี่ยวข้องกับทุกขั้นตอนของการพัฒนาโปรแกรมเชิงวัตถุทั้งการรวบรวมความต้องการ(requirement gathering), การวิเคราะห์ (analysis), การออกแบบ (design), การเขียนโปรแกรม (coding) และการทดสอบ (testing)

2.8 ส่วนประกอบของ UML

UML ประกอบด้วยหลายส่วนดังนี้

View : แสดงมุมมองต่างๆ ของระบบที่ออกแบบขึ้นมา โดยการอธิบายรายละเอียดในแต่ละส่วนใน View จะใช้ ไดอะแกรม ที่มีอยู่ใน UML อธิบาย

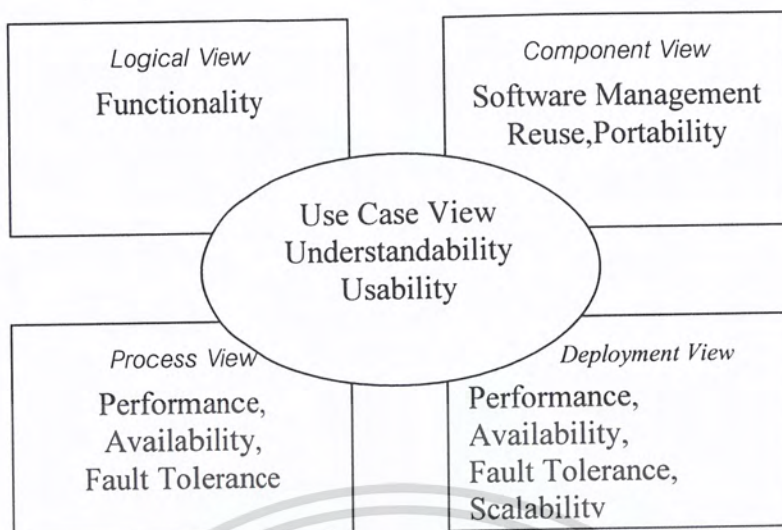
Diagram : เป็นไดอะแกรมที่ใช้ในการอธิบายส่วนต่างๆของ View

Model element : เป็นสัญลักษณ์ที่ใช้ในไดอะแกรมเพื่อแสดงหรือเป็นตัวแทนของสิ่งต่างๆ เช่น คลาส (class) ,ออปเจกต์,เมสเสจ(message) และ ความสัมพันธ์(relationship) เป็นต้น

General mechanism : เป็นส่วนที่แสดงคอมเม้นเพิ่มเติม (extra comment) , ข้อมูลอื่นๆ ที่จำเป็นหรือความหมายของโมเดลเอลิเมนต์ (model element)

2.8.1 View ในการออกแบบระบบที่มีขนาดใหญ่และมีความซับซ้อนมากขนาดนั้นจะทำให้ผู้ออกแบบระบบไม่สามารถที่จะออกแบบระบบได้ครบถ้วน ดังนั้นจึงต้องมีการมองระบบเป็นมุมมองต่างๆ เพื่อทำให้ง่ายในการออกแบบ เช่น มุมมองด้านฟังก์ชันนอล , นอนฟังก์ชันนอล(Nonfunctional), มุมมองขององค์กร เป็นต้น ดังนั้นระบบจึงมี View ที่ต่างๆ กันซึ่งแต่ละ view จะแสดงมุมมองเฉพาะของระบบซึ่งอธิบายรวมกันเป็นระบบที่สมบูรณ์ ซึ่งจะประกอบด้วย View ต่างๆ ดังนี้

2.8.1.1 Use-case View อธิบายการทำงานต่างๆ ของระบบที่ถูกมองจากภายนอกหรือผู้ใช้ระบบ Use-case view ซึ่งอธิบายโดยยูสเคสไดอะแกรม(use-case diagram) เป็นมุมมองสำหรับลูกค้า ,ผู้ออกแบบ,ผู้พัฒนาระบบและผู้ทดสอบระบบ



รูปที่ 2-4 สถาปัตยกรรมของ View

2.8.1.2 Logical view อธิบายการทำงานต่างๆ ที่ถูกออกแบบไว้ภายในระบบ ว่าระบบจะมีบริการอะไรให้กับผู้ใช้อย่างไร โดยจะแสดงโครงสร้างแบบสถิต (Static) เช่น คลาส, วัตถุ, ความสัมพันธ์และการทำงานร่วมกันแบบไดนามิก (dynamic collaboration) ซึ่งจะเกิดขึ้นเมื่อวัตถุส่งข้อความระหว่างกันในการทำงาน

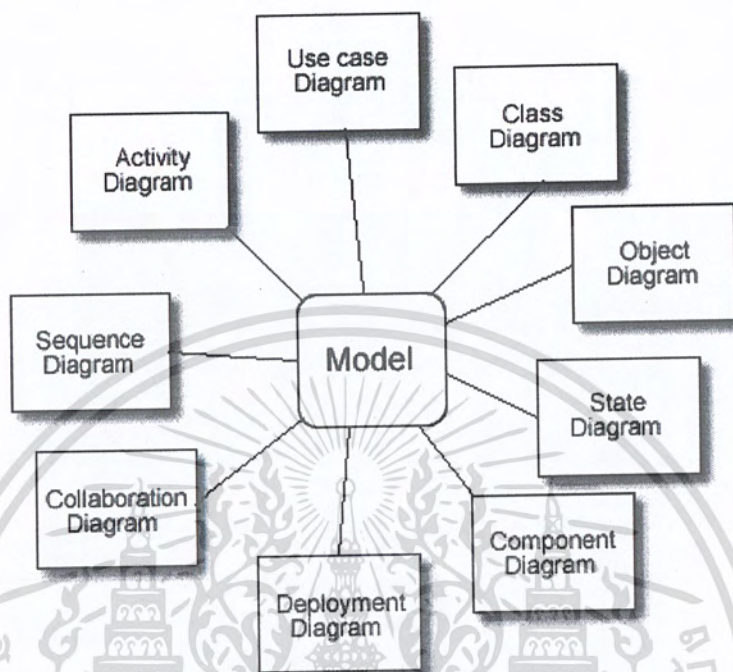
โครงสร้างแบบสถิตจะอธิบายโดยใช้ คลาสไดอะแกรม (Class diagram) และวัตถุไดอะแกรม (object diagram) ส่วนการทำงานร่วมกันแบบไดนามิกจะอธิบายโดยใช้ สเตตไดอะแกรม (state diagram) และแอคทิวิตีไดอะแกรม (activity diagram)

2.8.1.3 Component View อธิบายการสร้างและสร้างความขึ้นต่อกันของโมดูล (Module) ที่ใช้ในการพัฒนาระบบ โดยใช้คอมโพเนนต์ไดอะแกรม (Component diagram) ในการอธิบาย

2.8.1.4 Deployment View อธิบายการจัดวางระบบให้เหมาะสมในด้านกายภาพ (Physical) แสดงด้วยคอมพิวเตอร์และโหนด (nodes) ต่างๆ เพื่อให้ระบบมีเสถียรภาพมากขึ้น โดยใช้ดีพลอยเมนต์ไดอะแกรม (deployment diagram) ในการอธิบาย

2.8.1.5 Process View แสดงการทำงานร่วมกันและการติดต่อกันของส่วนต่างๆ ในระบบ

2.8.2 Diagram เป็นไดอะแกรมซึ่งแสดงสัญลักษณ์ที่ถูกจัดเรียงเพื่ออธิบายระบบในมุมมองต่างๆ ซึ่งในระบบหนึ่งๆ จะประกอบด้วยหลายๆ ไดอะแกรม และในแต่ละไดอะแกรมยังสามารถมองในหลายๆ มุมมองได้ UML มีไดอะแกรมที่ต่างกันอยู่ดังนี้



รูปที่ 2-5 แสดงไดอะแกรมต่างๆ ของ UML

2.8.2.1 Use Case Diagram ยูสเคสไดอะแกรมจะแสดงความสัมพันธ์ระหว่างผู้ใช้งาน (User) กับระบบ โดยใช้แอ็กเตอร์ (actors) แทนผู้ใช้งานและแอ็กเตอร์จะต้องติดต่อกับระบบโดยผ่าน ยูสเคส(Use case) ต่างๆ ซึ่งยูสเคสใน Use case diagram ก็คือการทำงานต่างๆ ของระบบที่ผู้ใช้งานต้องการ



รูปที่ 2-6 ตัวอย่าง Use Case ในระบบลงทะเบียน

แอ็กเตอร์ไม่ใช่ส่วนประกอบของระบบ แต่จะเป็นส่วนที่ใช้ติดต่อกับระบบ ซึ่งอาจจะเป็นเพียงการป้อนข้อมูลเข้าไปในระบบหรือเป็นเพียงการรับข้อมูลจากระบบ หรืออาจจะเป็นทั้งสองอย่างก็ได้ ในระบบหนึ่งๆ จะต้องมียแอ็กเตอร์ซึ่งจะต้องทำการกำหนดคุณลักษณะของแอ็กเตอร์ (Identify actor) ให้ได้ก่อนดังนี้

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

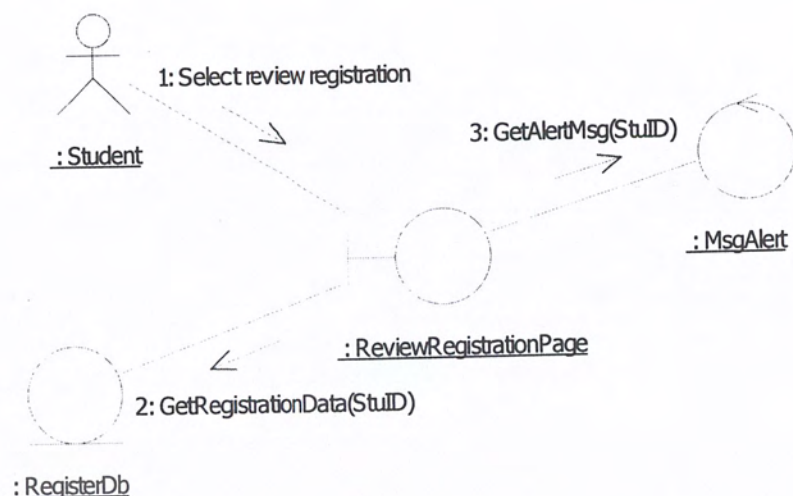
2.8.2.2 Class Diagram คลาสไดอะแกรมเป็นไดอะแกรมที่มีความสำคัญมากที่สุดในการวิเคราะห์และออกแบบโดยใช้วิธีเชิงวัตถุ เนื่องจากคลาสไดอะแกรมจะแสดงโครงสร้างของวัตถุและคลาสที่มีในระบบรวมทั้งแสดงความสัมพันธ์ด้วย คลาสไดอะแกรมจะเป็นโครงสร้างหลักของระบบและยังใช้ในการแยกย่อยรายละเอียด (Decomposite) ด้วยวิธีเชิงวัตถุ

สัญลักษณ์ความสัมพันธ์ต่างๆ ของคลาสไดอะแกรม มีดังนี้

- | | |
|------|---|
| 1 | หมายถึงจะมีออบเจ็กต์ในคลาสไดอะแกรมได้หนึ่งออบเจ็กต์เท่านั้น |
| 0..1 | หมายถึงจะมีออบเจ็กต์ในคลาสไดอะแกรมได้แค่หนึ่งอาจจะไม่มีก็ได้ |
| M..N | หมายถึงจะมีออบเจ็กต์ในคลาสไดอะแกรมได้ตั้งแต่ M ถึง N (เมื่อ M และ N เป็นจำนวนเต็มบวก) |
| * | หมายถึงจะมีออบเจ็กต์ในคลาสไดอะแกรมได้ตั้งแต่ศูนย์ขึ้นไป |
| 0..* | หมายถึงจะมีออบเจ็กต์ในคลาสไดอะแกรมได้ตั้งแต่ศูนย์ขึ้นไป |
| 1..* | หมายถึงจะมีออบเจ็กต์ในคลาสไดอะแกรมได้ตั้งแต่หนึ่งขึ้นไป |

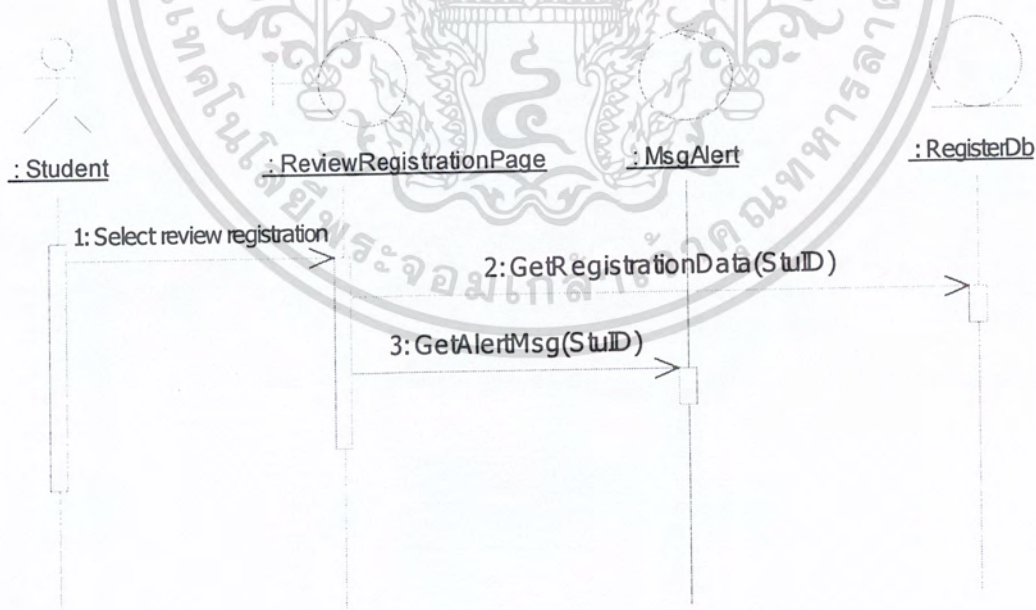
2.8.2.3 Object Diagram ออบเจ็กต์ไดอะแกรมจะได้มาจากคลาสไดอะแกรม ซึ่งจะเป็นการแสดงความเชื่อมต่อระหว่างอินชแทนซ์ (Instances) ต่างๆ ที่อยู่ภายในคลาสไดอะแกรมอย่างละเอียด เพื่อให้เป็นตัวอย่างสำหรับคลาสไดอะแกรมที่มีความซับซ้อนโดยจะนำมาแทนเป็น ออบเจ็กต์จริงๆ และแสดงความสัมพันธ์ต่างๆ ให้เห็น โดยสัญลักษณ์ของอินชแทนซ์ในออบเจ็กต์ไดอะแกรม จะใช้การขีดเส้นใต้เป็นสัญลักษณ์ ภายในกรอบสี่เหลี่ยม

2.8.2.4 Collaboration Diagram คีอแลโบลันไดอะแกรมใช้อธิบายการทำงานร่วมกันของออบเจ็กต์ที่มีการส่งแมสเสจระหว่างกัน เพื่อบอกถึงลำดับขั้นตอนการทำงานของระบบ โดยใช้สัญลักษณ์ลูกศรเป็นตัวกำหนดทิศทางซึ่ง คีอแลโบลันไดอะแกรมจะเป็นส่วนอธิบายการทำงานของออบเจ็กต์ไดอะแกรม



รูปที่ 2-7 แสดงตัวอย่างคอลเลบอเรชันไดอะแกรม

2.8.2.5 Sequence Diagram ซีเควินซ์ไดอะแกรมแสดงการติดต่อโต้ตอบระหว่างออบเจกต์ ที่เน้นไปที่ลำดับก่อนหลังและเวลาที่มีความสำคัญในการเปลี่ยนแปลง จะมีการส่งเมสเสจ เพื่อให้ออบเจกต์ ต่างๆ ที่อยู่ในระบบทำงานโดยใช้สัญลักษณ์ที่เรียกว่า ออบเจกต์เมสเสจ ซีเควินซ์ชาต (Object message sequence chart) ในการแสดงดังรูป



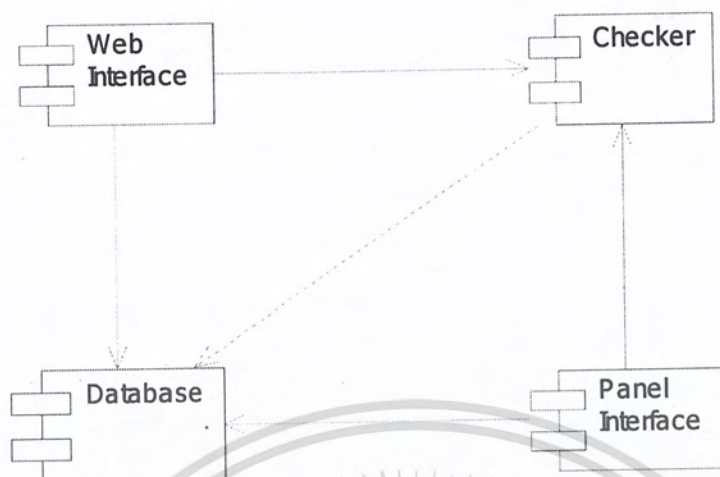
รูปที่ 2-8 แสดงตัวอย่าง ซีเควินซ์ไดอะแกรม

2.8.2.6 State Diagram สเตทไดอะแกรมใช้อธิบายคลาสต่างๆ ในระบบโดยจะแสดงทุกๆ สถานะที่เป็นไปได้และเหตุการณ์ทำให้ออบเจกต์เหล่านั้นเกิดการเปลี่ยนแปลง โดยเหตุการณ์ที่ทำให้เกิดการเปลี่ยนแปลงอาจเกิดจากออบเจกต์อื่นส่งมาเสมอเสมอ การเปลี่ยนแปลงสถานะเรียกว่า ทรานซิชัน (Transitions)

2.8.2.7 Activity diagram แอคติวิตีไดอะแกรมแสดงลำดับการไหลของกิจกรรม(activity) ต่างๆ โดยจะอธิบายกิจกรรมต่างๆ ในลักษณะของการกระทำในไดอะแกรม จะแสดงเป็นสถานะการกระทำ (Action State) ซึ่งสถานะจะเปลี่ยนไปเมื่อเกิดการกระทำอย่างใดอย่างหนึ่งเกิดขึ้นตามเงื่อนไขหรือการตัดสินใจที่กำหนดไว้เพื่อควบคุมการไหลของกิจกรรมรวมถึงสามารถมีเสมอเสมอที่รับ – ส่งระหว่างแต่ละกิจกรรม

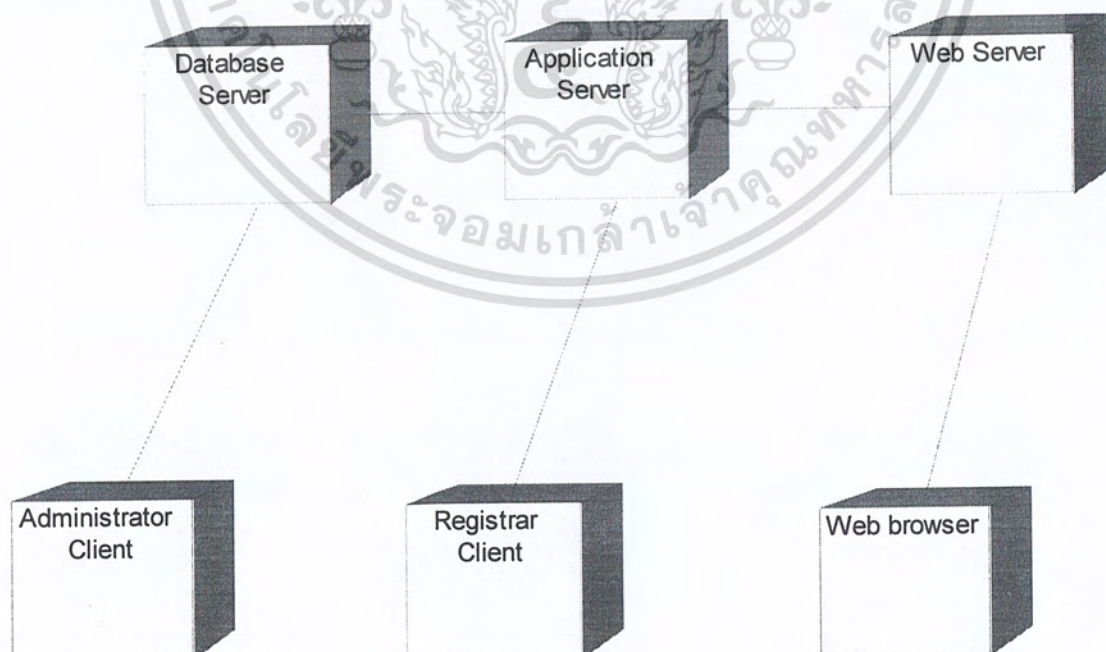


2.8.2.8 Component diagram คอมโพเนนต์ไดอะแกรมแสดงโครงสร้างทางกายภาพของโค้ด ในรูปคือคอมโพเนนต์ของโค้ด (Code) อาจเป็นส่วนประกอบของซอร์สโค้ด (source code) คอมโพเนนต์จะประกอบด้วยข้อมูลเกี่ยวกับ ลอจิกคลาส (logical class) ของคอมโพเนนต์ในไดอะแกรม มีการแสดงความสัมพันธ์หรือความพึ่งพากันของคอมโพเนนต์ให้ช่วยในการวิเคราะห์ว่าเมื่อเกิดการเปลี่ยนแปลงของ คอมโพเนนต์หนึ่งจะมีผลต่อคอมโพเนนต์อื่นๆ อย่างไร ซึ่งช่วยในการโปรแกรม



รูปที่ 2-10 แสดงตัวอย่างคอมโพเน้นไดอะแกรม

2.8.2.9 Deployment diagram คือพลอยเม้นไดอะแกรมแสดงสถาปัตยกรรมทางกายภาพของส่วนประกอบต่างๆ ของฮาร์ดแวร์(hardware) ซึ่งถูกเรียกว่า โหนด(Node) ในระบบซึ่งสามารถแสดงเป็นโหนดและการเชื่อมต่อระหว่างชนิดของการเชื่อมต่อ นอกจากนี้ภายในโหนด ยังสามารถมีคอมโพเน้น หรือออบเจ็กต์ที่สามารถปฏิบัติกับโหนดเพื่อแสดงว่าโปรแกรมส่วนใดถูกปฏิบัติบนโหนดใดและความสัมพันธ์ระหว่างคอมโพเน้นที่อยู่บนโหนด ซึ่งโหนดจะถูกแทนด้วยสัญลักษณ์ของสี่เหลี่ยมลูกบาศก์ดังรูป



รูปที่ 2-11 แสดงตัวอย่างดีพลอยเม้นไดอะแกรม

บทที่ 3

Servlet, JSP และ Java Bean

3.1 เซิร์ฟเล็ต (Servlet)

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

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

3.1.1 ข้อแตกต่างของ Servlet เมื่อเทียบกับ CGI Servlet เป็นเทคโนโลยี ที่มีรูปแบบการทำงานเหมือนกับ เทคโนโลยี CGI ที่มีการใช้งานอยู่ แต่เนื่องจาก Servlet ถูกพัฒนามาจากเทคโนโลยีจาวาอีกที ดังนั้น ต่อไปนี้คือข้อเปรียบเทียบซึ่งแสดงให้เห็น ถึงความแตกต่างระหว่าง Servlet และ CGI

3.1.1.1 ประสิทธิภาพ การทำงานของ CGI จะมีการสร้างโปรเซส (Process) ใหม่ทุกครั้ง ที่มีการเรียกใช้งานจากผู้ใช้ ดังนั้นหากมีการเรียกใช้งาน จำนวนหลายครั้งในช่วงเวลาเดียวกัน ก็จะมีจำนวนโปรเซสเกิดขึ้นเท่ากับจำนวนครั้งที่เรียกใช้งาน โปรเซสจะมีขนาดและรูปแบบการทำงาน ที่เหมือนกัน ทำให้กินเวลาในการทำงานตามจำนวนโปรเซส แต่สำหรับ Servlet แล้ว ระบบเวอร์ชวลแมชีน (Java Virtual Machine) จะถูกดึงขึ้นมาทำงานก่อน เพื่อทำให้เกิดโปรเซสตามลักษณะการเรียกใช้ของผู้ใช้เช่นกัน แต่การทำงานของ Servlet จะเป็นรูปแบบเธรด(Thread) แทนที่จะเรียกว่าโปรเซส ซึ่งแต่ละเธรดขนาดการทำงานเล็กกว่าเดิม และกินระยะเวลาการทำงานน้อยกว่า จึงทำให้รูปแบบของ Servlet มีประสิทธิภาพกว่านั่นเอง

3.1.1.2 ความสะดวก หากคุณสามารถเขียนโปรแกรม ด้วยภาษา C, C++ หรือ Java อยู่แล้ว จะเกิดความสะดวกและรวดเร็วในการสร้าง Servlet มากกว่า แทนที่จะเริ่มศึกษาภาษาคำสั่งใหม่ๆ และ Servlet ก็มีความสามารถเพียงพอสำหรับการใช้งานบนเว็บอยู่แล้ว เช่น การรับข้อมูลจากฟอร์ม, การอ่านและตั้งค่า HTTP Header, การทำงานกับ Cookie, การทำงานแบบ Session และอื่นๆอีกมาก

3.1.1.3 ความสามารถ Servlet มีข้อดีที่เห็นได้ชัดคือ สามารถติดต่อกับเว็บเซิร์ฟเวอร์ได้โดยตรง ในขณะที่ CGI ปกติทำไม่ได้ (ยกเว้นรุ่นใหม่ๆ) และ Servlet ยังสามารถแชร์ข้อมูลการทำงานร่วมกับโปรแกรมอื่นๆได้ด้วย

3.1.1.4 การโยกย้าย เนื่องจาก Servlet ถูกสร้างมาจาก Java ซึ่งเป็นเทคโนโลยี ที่สามารถถูกใช้งาน ณ. ที่ใดก็ได้ โดยไม่จำกัดรูปแบบ ดังนั้น หากพัฒนา Servlet เพื่อใช้งานแล้ว สามารถที่จะโยกย้ายผลผลิตไปยัง ระบบอื่นๆ ได้ โดยไม่ต้องปรับปรุงหรือเปลี่ยนแปลง

3.1.1.5 ไม่แพง ในปัจจุบันเว็บเซิร์ฟเวอร์ที่สนับสนุนการทำงาน Servlet มีอยู่มากมาย ทั้งที่เป็นแบบฟรี(ดาวน์โหลด) และซื้อ-ขายกัน นอกจากนี้ ในอนาคตแผนการผลิตเว็บเซิร์ฟเวอร์อื่นๆ ยังมีการสนับสนุนการใช้งาน Servlet อีกด้วย

3.1.2 โครงสร้างของ Servlet การรับส่งข้อมูลโดยใช้เว็บ ผ่านทางแท็กฟอร์ม <FORM ...> มีรูปแบบการส่งอยู่ 2 ลักษณะคือ GET และ POST โดยที่การส่งแบบ GET เป็นรูปแบบการส่งข้อมูล ผ่านทาง URL (ช่องพิมพ์ชื่อเว็บหรือเว็บแอดเดรส) ซึ่งผู้ใช้อาจจะพิมพ์ได้เองโดยตรง เช่น `http://www.javacentrix.com?user=1245&pwd=4444` หรือฟอร์มที่แท็ก กำหนดแอททริบิวต์ METHOD เป็น GET เช่น <FORM METHOD="GET" ... เป็นต้น (ตัวอย่างที่เป็นตัวหนาถือการส่งข้อมูลผ่านฟอร์มในแบบ GET)

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

3.1.3 การใช้ Servlet ส่งข้อมูลรูปแบบ HTML เน้นอนว่า การสร้าง 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 ชื่อไดเรกทอรี และชื่อไฟล์เว็บเพจที่ต้องการดู เช่น หากข้อมูล URL คือ `http://www.javacentrix.com/tutorials/api/index.htm` นั้นหมายถึงต้องการอ่านเว็บเพจที่ชื่อ `index.htm` จากไดเรกทอรี `tutorials/api` จากเว็บเซิร์ฟเวอร์ที่ชื่อ `www.javacentrix.com`

สำหรับ URL นั้นนอกจาก จะใช้ชี้ตำแหน่งไฟล์ที่ต้องการเรียกดูแล้ว ยังสามารถใช้สำหรับส่งข้อมูลให้กับระบบเซิร์ฟเวอร์ เพื่อใช้ในการทำงาน หรือดำเนินการอย่างใดอย่างหนึ่ง ลักษณะ URL ประเภทนี้หลังจากชื่อไฟล์แล้ว 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 เจเอสพี (Java Server Page : JSP)

Java Server Page หรือที่เรียกกันง่าย ๆ ว่า JSP เป็นเทคโนโลยีที่เป็นผลรวม ของการนำเอาหลักการ สร้างเว็บแบบสแตติกด้วย HTML มารวมกับการสร้างเว็บแบบไดนามิก ก่อให้เกิดการแสดงผลเว็บ ที่มีการ เปลี่ยนแปลงข้อมูลได้ตามการใช้งานของผู้ใช้ 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 สังเกตจากตัวหนาที่ อยู่ใน แท็กพิเศษ <% ... %> ซึ่งเป็นคำสั่งในรูปแบบ JSP ทำให้ส่วนดังกล่าวสร้างข้อมูลในแบบไดนามิก ไว้ใน เว็บเพจตามคำสั่ง ไฟล์ดังกล่าวจะบันทึกไว้ในนามสกุล .jsp จาก HTML เดิม

รูปแบบดังกล่าวไม่สามารถทำได้สำเร็จตามเทคโนโลยี JSP หากไฟล์ตัวอย่างด้านบนไม่ ได้ถูกเก็บไว้ในเว็บเซิร์ฟเวอร์ที่สนับสนุนการทำงานของ JSP ข้อความที่เป็นไดนามิกนั้น ถูกสำรวจเพื่อประมวลผล และใส่ ลงในตำแหน่งดังกล่าว ด้วยความสามารถของ เซิร์ฟเวอร์ที่สนับสนุน JSP ก่อนส่งกลับไปให้ผู้ใช้งาน โดยผู้ใช้งาน จะได้รับข้อความ Hello Java Server Page เท่านั้นโดยไม่เห็นคำสั่ง JSP เลย

3.2.1 ข้อแตกต่างของ JSP เมื่อเทียบกับเทคโนโลยีอื่นๆ JSP ไม่ได้เป็นเทคโนโลยีเดียว ในปัจจุบันที่ สามารถทำให้เว็บ แสดงข้อมูลในแบบไดนามิกได้ มีเทคโนโลยีอื่นๆที่ทำงานในลักษณะนี้หลายแบบ เนื้อหา ส่วนนี้ เป็นการแนะนำให้ผู้อ่านได้เห็นข้อแตกต่างบางประการ เมื่อเทียบกับ JSP

- Active Server Page(ASP) ASP เป็นเทคโนโลยีที่เหมือนกับ JSP แต่เป็นเทคโนโลยีที่พัฒนามา จาก บริษัทไมโครซอฟต์ ผู้ผลิตระบบปฏิบัติการวินโดวส์ที่รู้จักกัน โดยทั่วไป ลักษณะ JSP มีรูปแบบที่แตกต่างกันเด่นชัด 2 ประการคือ ประการแรก JSP สามารถสร้างได้จาก ภาษาจาวา ต่างจากภาษา VB หรือภาษาไคๆ ซึ่งเป็นข้อกำหนดเฉพาะของ ไมโครซอฟต์ ประการที่สอง JSP ซึ่งเป็นผลพวงมาจากประการแรกคือ สามารถโยกย้ายการทำงาน ไปใช้งานระบบปฏิบัติการใดก็ได้ ที่มีการใช้งานกันอยู่ในปัจจุบัน โดยไม่ต้องแก้ไขหรือเปลี่ยนแปลงใดๆ ในขณะที่ ASP ก็โยกย้ายได้เช่นเดียวกัน แต่ต้องอยู่ในระบบปฏิบัติการที่ ไมโครซอฟต์กำหนดขึ้นเท่านั้น

- Servlet JSP สามารถทำงานได้เช่นเดียวกับที่ Servlet ทำได้ แต่มีจุดเด่นมากกว่าที่ JSP มีความสะดวกในการสร้าง และเปลี่ยนแปลง มากกว่า เพราะทำโดยตรงที่ไฟล์ HTML มากกว่าการที่ต้องลงมือเขียนคำสั่งภาษาจาวา โดยตรงเหมือนกันสร้าง Servlet ซึ่งต้องมีการนำไปคอมไพล์ก่อนการนำไปใช้งาน
- Server-Side Includes(SSI) SSI คือรูปแบบการสร้างไดนามิกเว็บเช่นเดียวกัน แต่รูปแบบการทำงานคือการนำเอาข้อมูลที่มีอยู่แล้ว บนเซิร์ฟเวอร์นำมาประกอบใส่ ในเว็บเท่านั้น ต่างจาก JSP ซึ่งสามารถนำข้อมูลมาได้ แต่มีรูปแบบในการทำงาน มีรูปแบบในการประมวลผล หรือการเรียกโปรแกรมภายนอก เช่น Servlet มาช่วยในการทำให้ข้อมูล มีรูปแบบต่างๆ เช่น ดึงข้อมูลมาจากระบบฐานข้อมูล (Database) นอกจากข้อมูลที่มีอยู่บนเซิร์ฟเวอร์เช่นที่ใช้ใน SSI เท่านั้น
- JavaScript เป็นการทำเนื้อหาไดนามิกให้กับเว็บ แต่รูปแบบการทำงานเกิดขึ้นของ JavaScript เกิดจากการประมวลผลและดึงข้อมูล ที่มีอยู่บนเครื่องผู้ใช้ หรือไคลเอนต์เท่านั้น มารวบรวมเนื้อหาในเว็บ ต่างกับ JSP ที่เป็นการทำไดนามิก แต่ข้อมูลถูกสร้าง และดึงมาจากระบบเซิร์ฟเวอร์ ที่มีหลากหลายมากกว่า และทำให้เนื้อหาข้อมูลเป็นเนื้อเดียวกัน เมื่อผู้ใช้เรียกดู แต่ JavaScript เนื้อข้อมูลเป็นของเครื่องผู้ใช้อง ซึ่งผู้ใช้อย่างคน(ต่างเครื่อง) ก็จะให้ข้อมูลที่คล้ายกัน ก็ไม่เหมือนกันทั้งหมด
- Static HTML หรือสแตติก HTML เน้นอนข้อนี้ คงเห็นได้ชัดเจนที่ว่า สแตติก HTML ให้เนื้อหาข้อมูลที่คงที่ตลอดเวลา ไม่ว่าจะถูกเรียกดูในเวลาใดๆ (ยกเว้นมีคนมาแก้ไขคำสั่ง HTML) ในขณะที่ JSP ทำให้เนื้อหาข้อมูลมีการเปลี่ยนแปลงตามการใช้งาน หรือเครื่องผู้ใช้ โดยไม่ต้องเปลี่ยนแปลงแก้ไขคำสั่ง JSP

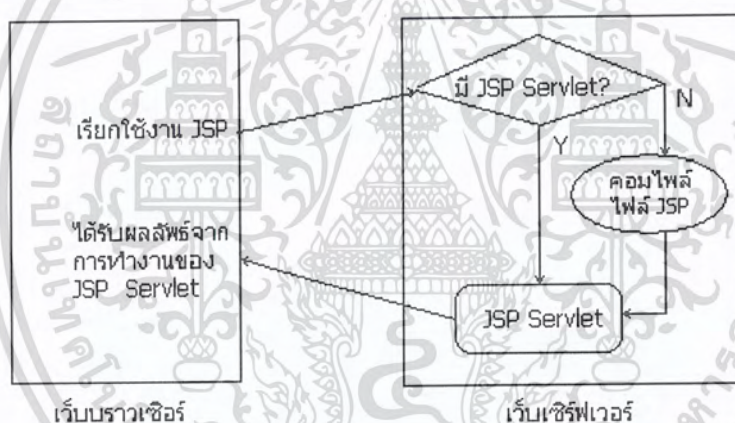
	CGI/Perl	Mod_Perl	ASP	JSP
เว็บเซิร์ฟเวอร์	Any Web server	Apache Web Server	Microsoft IIS or Personal Web Server	Any Web server, including Apache, Netscape, IIS today
โยกย้ายการทำงาน	No	No	No	Yes
ลักษณะนำกลับมาใช้	No	No	No	Yes
สคริปต์หรือภาษา	C, Perl	Perl	VBScript, JScript	Java
ปกป้องการใช้งาน หน่วยความจำ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

ตารางที่ 3-1 เปรียบเทียบเทคโนโลยี JSP กับเทคโนโลยีอื่น

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

3.2.3 JSP Servlet คือ Servlet ที่เกิดขึ้นจากชุดคำสั่ง JSP ที่ถูกคอมไพล์ไปเป็น JSP Servlet โดยขบวนการนี้เกิดโดยอัตโนมัติภายในเว็บเซิร์ฟเวอร์ที่สนับสนุน การทำงานของ JSP และ JSP Servlet นี้เองที่เป็นส่วน ที่เกิดการประมวลผลบนเซิร์ฟเวอร์ ดังนั้นไฟล์นามสกุล JSP ที่ถูกเรียกใช้งาน จะต้องถูกคอมไพล์ไปเป็น JSP Servlet ก่อนในครั้งแรกของการเรียกใช้งาน หรือเมื่อมีการเปลี่ยนแปลงชุดคำสั่ง

3.2.4 การทำงานของ JSP



รูปที่ 3-1 รูปอธิบายการทำงานของ JSP

เมื่อเว็บเบราว์เซอร์ มีการร้องขอโดยผู้ใช้เพื่อเรียกข้อมูล JSP ที่อยู่บนเซิร์ฟเวอร์ ซึ่งก็หมายถึงไฟล์ที่มีนามสกุล JSP เว็บเซิร์ฟเวอร์จะทำการตรวจสอบว่า ไฟล์นั้นมีการคอมไพล์ไปเป็น JSP Servlet แล้วหรือยัง ถ้ายังไฟล์ JSP จะถูกคอมไพล์ โดยตัวเว็บเซิร์ฟเวอร์ ไปเป็น JSP Servlet แต่ถ้าไฟล์ถูกคอมไพล์แล้ว (ซึ่งเกิดขึ้นจากการใช้งานในครั้งแรก) เว็บเซิร์ฟเวอร์จะนำ JSP Servlet มาใช้งาน โดยการเรียกเพื่อประมวลผล ได้ผลลัพธ์อะไรก็จะเป็นข้อมูล ซึ่งเว็บเซิร์ฟเวอร์ จะส่งกลับไปให้ผู้เรียกใช้งาน และการประมวลผลของ JSP Servlet นี้เอง ที่ก่อให้เกิด ข้อมูลเว็บในแบบไดนามิก

3.3 จาวาบีน (JavaBean)

ในการเขียน JSP สำหรับระบบใหญ่ ๆ นักพัฒนาจะไม่นิยมใส่ java code ลงไปในไฟล์ JSP มากนัก สิ่งที่อยู่ในไฟล์มักจะเป็นเพียง HTML code และค่าของ variables ต่าง ๆ ที่ได้มาจาก JavaBeans เท่านั้น. ข้อดีของการใช้ JavaBean คือการง่ายต่อการเปลี่ยนแปลงลักษณะของ look-and-feel ของไฟล์ JSP ที่เป็นเช่นนั้น เพราะ ส่วนที่เป็น data (variables) กับส่วนที่เป็น presentation (HTML) จะอยู่ด้วยกันอย่างหลวม ๆ ดังนั้นเราจึงสามารถเปลี่ยนตัว presentation เมื่อไหร่ก็ได้ โดยตัว JavaBean ที่ใช้เก็บค่าของ variables ต่าง ๆ ยังคงเหมือนเดิม

สำหรับคนที่ไม่คุ้นกับ JavaBean ให้นึกถึง JavaBean เป็นลักษณะของกล่อง โดยกล่องนี้มีส่วนที่ใช้ติดต่อกับโลกภายนอกอยู่สองส่วนหลัก ๆ คือ

- ส่วนที่ใช้สำหรับรับข้อมูลต่าง ๆ โดยข้อมูลเหล่านี้จะถูกใช้เป็นตัวแปรในการควบคุมหรือเปลี่ยนแปลงคุณสมบัติและการทำงานของกล่อง (Setter)
- ส่วนที่ใช้สำหรับอ่านคุณสมบัติของกล่อง (Getter) ซึ่งโดยส่วนมาก(แต่ไม่ทั้งหมด) ก็คือค่าของตัวแปรต่าง ๆ ที่บรรจุอยู่ในกล่อง

ในการใช้งานกล่อง (JavaBean) ดังกล่าวร่วมกับ JSP เราสามารถที่จะสร้างกล่องขึ้นมาโดยใช้ `<jsp:useBean .../>`, ส่งค่าเข้าไปในกล่องโดยใช้ `<jsp:setProperty .../>` และอ่านค่าต่าง ๆ ที่อยู่ในกล่อง โดยใช้ `<jsp:getProperty .../>` tag แต่ก่อนที่จะเรารู้ถึงการใช้ tag เหล่านี้ เรามาพูดถึง JavaBean ในรายละเอียดอีกสักเล็กน้อยเสียก่อน

3.3.1 Bean Conventions สำหรับคนที่เคยเขียน application ที่เกี่ยวกับ JavaServlet หรือ RMI จะพบว่าแต่ละคลาสที่เราเขียนขึ้น อาจจะต้องมีการ extend คลาสใดคลาสหนึ่งก่อน เช่น HttpServlet (ในกรณีของการเขียน JavaServlet) หรืออาจต้องทำการ implement interface ต่าง ๆ เช่น Serializable (ในกรณีของการเขียน RMI) แต่สำหรับการเขียนคลาสให้กลายเป็น JavaBean แล้ว เราไม่จำเป็นต้องทำสิ่งต่าง ๆ ที่กล่าวมาข้างต้นเลย. JavaBean จริง ๆ แล้วก็คือ จาวาคลาสธรรมดา ๆ แต่ถูกเขียนขึ้นโดยใช้ naming และ design conventions ที่อยู่ใน JavaBean Specification เท่านั้น. Conventions หลัก ๆ ที่เราจะพูดถึงสำหรับการเขียน JavaBean เพื่อใช้ใน JSP มีดังต่อไปนี้คือ

3.3.2 The Bean Constructor ในการเขียน Bean ให้สามารถใช้ได้กับ `<jsp:useBean .../>` สิ่งหนึ่งที่ Bean ของเราจะต้องมีคือ ตัว constructor ที่ไม่ต้องการ argument สำหรับการ initialize ของ Bean * เราสามารถนำคลาสเหล่านี้ไปใช้กับ `<jsp:useBean .../>` ได้ อย่างไรก็ตามคลาสนี้ยังไม่มีประโยชน์อะไรสำหรับไฟล์ JSP จริง ๆ ประโยชน์ที่เราจะได้จาก Bean ก็คือการส่งค่าต่าง ๆ เพื่อทำการควบคุมและเปลี่ยนแปลงคุณสมบัติของ Bean รวมไปถึงการอ่านค่าคุณสมบัติต่าง ๆ ของ Bean ซึ่งสิ่งเหล่านี้จะเกิดขึ้นได้โดยอาศัยสิ่งหนึ่งที่เรียกว่า Bean's Properties

3.2.3 Bean's Properties Bean's Properties คือตัวที่บอกว่า เราสามารถทำอะไรกับ JavaBean นั้นได้บ้าง [โดยส่วนมาก(แต่ไม่ทั้งหมด) ก็คือการเปลี่ยนและอ่านค่าต่าง ๆ ของ instance variables ที่บรรจุอยู่ข้างใน Bean นั้น ๆ] ซึ่งผลที่ได้ก็คือ การควบคุมและเปลี่ยนแปลงคุณสมบัติต่าง ๆ ของ Bean อย่างไรก็ตามบางที properties ที่เรากำหนดขึ้น อาจจะไม่ได้อะไรที่เกี่ยวข้องกับ instance variables ที่อยู่ข้างใน Bean นั้นเลย

JSP Container จะสามารถทราบได้ว่า JavaBean ของเรามี properties อะไรบ้าง ได้จากวิธีการที่เรียกว่า introspection หลักการก็คือ JSP Container จะทำการวิเคราะห์ฟังก์ชันที่เป็น public ของ Bean นั้น ซึ่งจะต้องเขียนขึ้นอย่างถูกต้องตามหลักของ JavaBean API โดยจะถือว่า property ใด ๆ ก็ตามจะเกิดขึ้นได้ก็ต่อเมื่อ Bean คลาสนั้นมี

- public ฟังก์ชันที่ใช้สำหรับควบคุมหรือเปลี่ยนแปลงค่าของ property นั้น (Setter) หรือ
- public ฟังก์ชันที่ใช้สำหรับอ่านค่าของ property นั้น (Getter) หรือ
- มีทั้งสองฟังก์ชันข้างต้น

ในการกำหนด property ขึ้นมาอันหนึ่ง เราจะต้องสร้าง public ฟังก์ชันที่ขึ้นต้นด้วยคำว่า set (Setter) ซึ่งใช้สำหรับควบคุมหรือเปลี่ยนแปลงค่าของ property นั้น และหรือ public ฟังก์ชันที่ขึ้นต้นด้วยคำว่า get (Getter) ซึ่งใช้สำหรับอ่านค่าของ property นั้น โดยรูปแบบโดยทั่วไปก็คือ

```
public void setPropertyName(PropertyType value);
```

```
public PropertyType getPropertyName();
```

ในการเขียน JavaBean เพื่อกำหนด properties ต่าง ๆ เราสามารถกำหนด properties ต่าง ๆ ขึ้นมา โดยไม่เกี่ยวข้องกับ instance variable ใด ๆ เลยก็ได้

3.2.4 Property Name Conventions โดยทั่วไป การเขียนชื่อของ properties ต่าง ๆ จะเป็นในลักษณะของลูกผสมคือ มีทั้งตัวใหญ่และตัวเล็ก ซึ่งจะขึ้นต้นด้วยตัวเล็กในคำแรกและจะตามด้วยตัวใหญ่ในคำถัด ๆ มา ยกตัวอย่างเช่น firstName, lastName เป็นต้น สำหรับตัว setter และ getter ของ properties ดังกล่าวก็จะเป็น setFirstName, setLastName ตามลำดับ

เราจะสังเกตเห็นว่า ชื่อของ properties และตัว setter, getter ของ properties เหล่านั้นจะต่างกันที่ตัวแรก หลักจากคำว่า set หรือ get เท่านั้น ในการตีความ setter, getter เพื่อหา properties ต่าง ๆ ของ JavaBean โดย JSP Container, JSP Container จะทำการแปลง case ต่าง ๆ ให้แมชกับชื่อของ properties ที่อยู่ใน `<jsp:getProperties .../>`, `<jsp:setProperty .../>` โดยอัตโนมัติ

3.2.5 Loading a Bean - `<jsp:useBean>` ก่อนที่เราจะสามารถใช้ JavaBean ใน JSP ได้ สิ่งแรกที่เราต้องทำคือ การโหลด instance ของ Bean เราสามารถโหลด instance ของ Bean เข้ามาในระบบได้โดยใช้ `<jsp:useBean .../>` โดยเราจะต้อง

- กำหนดชื่อที่จะใช้ในการอ้างถึง instance ของ Bean นี้ (id)
- กำหนดคลาสของ Bean ที่จะโหลด (class)
- ทำการเซต scope (หรือช่วงชีวิต) ของ Bean ว่าจะยาวนานแค่ไหน (scope)

ดังรูปแบบข้างล่าง

```
<jsp:useBean id="myBeanInstance" class="com.myPackage.BeanClass"
scope="page|request|session|application">
```

...body...

</jsp:useBean>

3.2.6 Initializing a Bean - <jsp:setProperty> ใน Bean Conventions เรากล่าวถึง Bean Constructor ที่ไม่ต้องการ arguments ซึ่งถูกใช้สำหรับ <jsp:useBean> อย่างไรก็ตามบางครั้ง Bean ที่เราใช้อาจต้องการค่าเริ่มต้นในการ initialize ซึ่งเราสามารถทำการเซตค่าต่าง ๆ ให้ Bean ได้โดยใช้ <jsp:setProperty .../>

ค่าที่เราจะเซตให้ Bean บางครั้งอาจมาจากค่าที่เรากำหนดขึ้น หรืออาจจะเป็นค่าที่ได้จาก request ของ client ซึ่งมาจาก element ต่าง ๆ ใน <FORM> tag ของ html ดังนั้นการเซตค่า Bean จึงถูกแบ่งออกเป็นสองแบบคือ

3.2.6.1 การเซตค่าให้ Bean โดยใช้ค่าที่เรากำหนดขึ้น เช่น

```
<jsp:useBean id="myBeanInstance" class="com.myPackage.BeanClass" scope="request">
<jsp:setProperty name="myBeanInstance" property="myProperty"
value="definedValue"/>
```

</jsp:useBean>

name คือ ชื่อของ Bean instance ที่เราต้องการอ้างถึง

property คือ ชื่อของ property ที่เราต้องการเซตค่า

value คือ ค่าที่เราจะใส่เข้าไปให้ property นั้น

3.2.6.2 การเซตค่าให้ Bean โดยใช้ค่าที่ได้จาก element ต่าง ๆ ของ <FORM> tag เช่น

```
<jsp:useBean id="myBeanInstance" class="com.myPackage.BeanClass" scope="request">
<jsp:setProperty name="myBeanInstance" property="myProperty"
param="myFormElementName"/>
```

</jsp:useBean>

param คือ ชื่อของ element ที่อยู่ใน <FORM> tag ที่ส่ง request มาที่ไฟล์ JSP ที่มี

<jsp:useBean> นี้

หลังจากที่เราทำการโหลด และเซตค่าของ property ต่าง ๆ ใน Bean แล้ว เราสามารถดูผลที่เกิดขึ้นได้โดยใช้ <jsp:getProperty name="myBeanInstance" property="myProperty"/>

ค่าที่ได้จาก <jsp:getProperty> อาจจะเป็นค่าของ instance variables ของ Bean ที่เปลี่ยนไปหลังจากที่เราใช้ <jsp:setProperty> หรือบางครั้ง อาจจะเป็นลิสต์ของตารางรายชื่อ ในกรณีที่ว่า property ที่เราเซตผ่าน <jsp:setProperty> เป็นตัว trigger ให้ Bean ส่ง query ไปที่ดาตาเบส เป็นต้น

บทที่ 4

เครื่องมือที่ใช้ในการพัฒนาโปรแกรมและฐานข้อมูล

4.1 บทนำ

ในบทนี้จะกล่าวถึงเครื่องมือที่ใช้ในการพัฒนาโปรแกรมและฐานข้อมูลซึ่งมี 2 โปรแกรมด้วยกัน นั่นก็คือ IBM DB2 ที่ใช้ในการพัฒนาฐานข้อมูลและ IBM VisualAge for Java ในการพัฒนาโปรแกรม

IBM DB2 เป็น ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management) ระบบหนึ่ง ซึ่งได้รับการอนุเคราะห์จากบริษัทไอบีเอ็ม 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

VisualAge for Java เป็นเครื่องมือที่ได้รวบรวม visual environment ซึ่งสามารถรองรับกระบวนการพัฒนาโปรแกรมด้วยภาษาจาวาได้อย่างสมบูรณ์และซึ่งกล่าวได้ว่า VisualAge for Java นี้ไม่ใช่แค่เพียงคอมไพเลอร์ของภาษาจาวาเท่านั้น แต่ได้รวบรวม environment ไว้หลายอย่าง ซึ่งเรียกว่า Integrated Development Environment (IDE) ตัวอย่างที่นำมาใช้งานเช่น

- Web Sphere Test Environment
- JSP/Servlet Development Environment

4.2 ลักษณะของ DB2ที่ใช้ในโครงการ

DB2 Universal Database (UDB) เป็น RDBMS ที่สนับสนุนความสามารถในการจัดการ RDBMS ผ่านเว็บอย่างเต็มรูปแบบ โดยโครงการนี้ใช้ version 7.1 ซึ่งมีความสามารถใหม่ๆ เช่น

- สนับสนุน XML โดย DB2 ยอมให้ XML เป็นชนิดข้อมูลในคอลัมน์ได้
- สามารถทำงานร่วมกับโปรแกรมต่างๆ เช่น The DB2 Software Developer's Kit (SDK), XML, IBM VisualAge for Java, Net.Data, WebSphere Application Server ฯลฯ

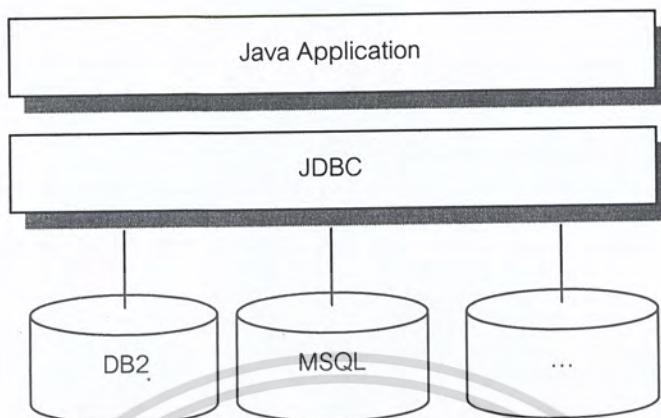
DB2 Client Tool for Window ใช้ในการติดต่อระหว่าง Client / Server เพื่อทดสอบโครงการในส่วนการติดต่อระหว่างนักศึกษากับระบบสารสนเทศ

4.3 การเชื่อมต่อเข้ากับฐานข้อมูลโดยใช้ 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 4.4

JDBC ถูกสร้างขึ้นมาในระดับของการเชื่อมต่อกับฐานข้อมูลในรูปแบบที่คล้ายกับ ODBC (ของทางบริษัทไมโครซอฟท์) หลักการของทั้งสองมาตรฐานตั้งอยู่บนฐานเดียวกันคือ X/Open SQL Call-Level

Interface ของระบบ X-windows และ JDBC driver ต้องเข้ากับระดับมาตรฐานในการเข้าถึง SQL (ANSI SQL Entry Level Standards) และต้องผ่าน Conformance test ซึ่ง JavaSoft เป็นผู้กำหนดขึ้น



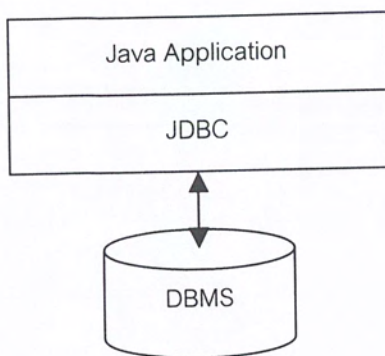
รูปที่ 4-1 แสดงลักษณะการทำงานของ JDBC

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

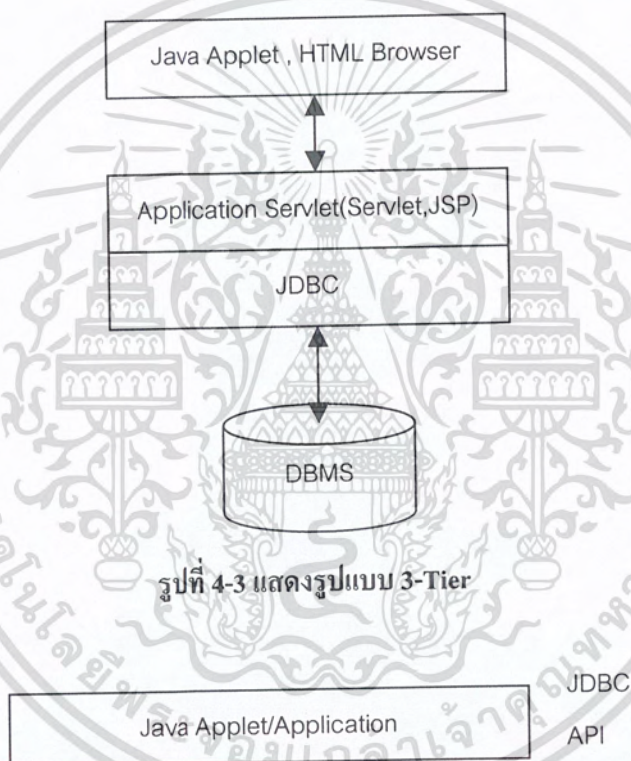
4.3.1.1 รูปแบบ 2-Tier จาวาแอปพลิเคชันหรือจาวาแอปพลิเคชันจะทำการติดต่อกับฐานข้อมูลโดยตรงจึงมีความจำเป็นที่โปรแกรมจาวาต้องการ JDBC Driver พิเศษที่สามารถสื่อสารกับระบบจัดการฐานข้อมูลชนิดนั้นได้ รูปแบบการเชื่อมต่อแสดงรูปที่ 4-5

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

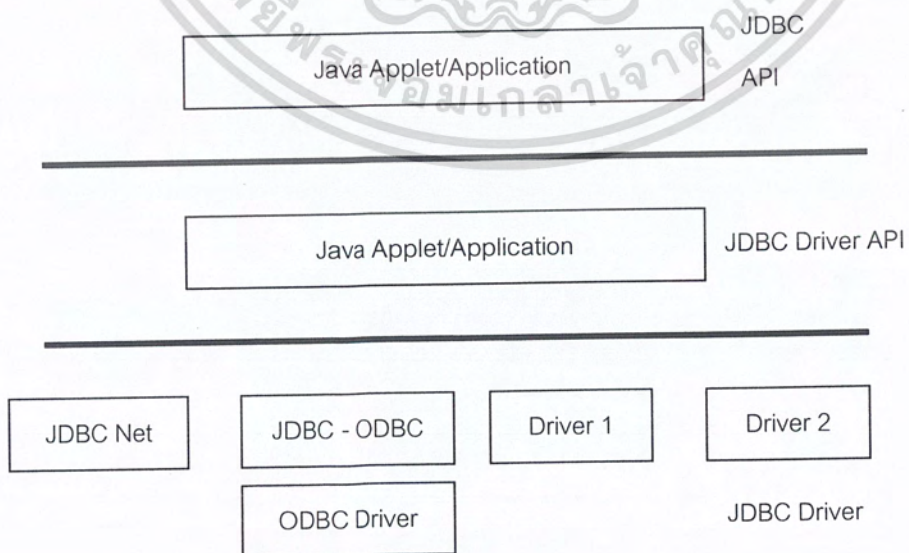
4.3.1.2 รูปแบบ 3-Tier คำสั่งต่างๆจากผู้ใช้จะถูกส่งไปให้กับ Tier กลางหรือตัวกลาง หลังจากนั้น Tier กลางจะแปลงคำสั่งเหล่านั้นให้เป็นภาษา SQL เพื่อส่งให้กับระบบฐานข้อมูลเพื่อทำการประมวลผลและทำการส่งผลลัพธ์คืนให้กับ Tier กลางและส่งให้ผู้ใช้ในที่สุด หลักการทำงานเช่นนี้มักจะพบในเครือข่ายอินเทอร์เน็ตซึ่งประกอบด้วยเครื่องคอมพิวเตอร์หลากหลายชนิดและเว็บเซิร์ฟเวอร์ (Web Server) ซึ่งทำหน้าที่เป็น Tier กลางในการจัดการให้ระบบคอมพิวเตอร์ทั้งไคลเอนต์และเซิร์ฟเวอร์ฐานข้อมูลสามารถติดต่อกันได้ การปรับเปลี่ยนระบบไคลเอนต์และเซิร์ฟเวอร์ หรือแม้กระทั่งการเปลี่ยนตัวฐานข้อมูลตัวใหม่จะไม่มีผลกระทบต่อซึ่งกันและกันเกิดขึ้น แสดงดังรูป 4.6



รูปที่ 4-2 แสดงรูปแบบ 2-Tier



รูปที่ 4-3 แสดงรูปแบบ 3-Tier



รูปที่ 4-4 ระดับการเชื่อมต่อของ JDBC API

4.3.2 โครงสร้างของ JDBC โครงสร้างการเชื่อมต่อภายใน JDBC ประกอบด้วย 3 ระดับ คือ JDBC API, JDBC Driver API และ JDBC Driver ดังรูปที่ 4.7 ระดับบนสุด เป็น JDBC API เป็นระดับฟังก์ชัน API ที่อำนวยความสะดวกให้แก่โปรแกรมประยุกต์ ระดับกลาง JDBC Driver (มี Driver ที่ต่างกันอยู่ 4 ชนิด) ที่เหมาะสม

4.3.3 รูปแบบของ JDBC Driver หรือส่วนที่อยู่เบื้องหลังการทำงานของฟังก์ชัน API ต่างๆ ของ JDBC ถูกแบ่งออกเป็นทั้งหมด 4 ชนิดคือ

4.3.3.1 JDBC-ODBC Bridge

4.3.3.2 Native-API(Party-Java) Driver

4.3.3.3 Net-Protocol(All-Java) Driver

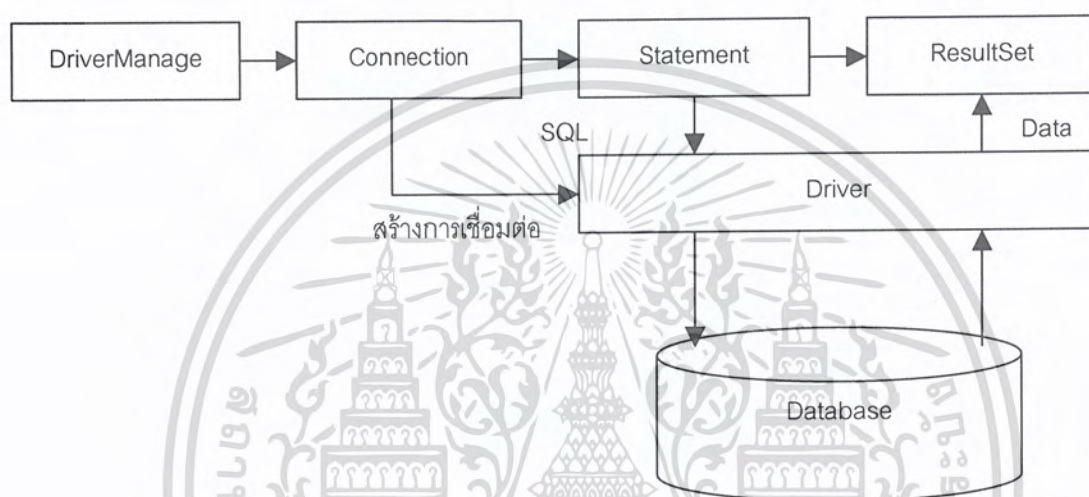
4.3.3.4 Nativer-Protocol(All-Java)

ในที่นี้จะขอกล่าวถึงแค่ JDBC/ODBC Bridge ถูกพัฒนาโดย JavaSoft และ InterSolv ทำหน้าที่เป็นตัวกลางในการเข้าถึงฐานข้อมูลได้โดยผ่านทางการทำงานของ ODBC โดยนำข้อดีของ ODBC-enabled data source ที่มีใช้อยู่โดยทั่วไปอย่างมากมา ผังโคลเ็นต์ที่เป็นจาวาแอฟเพล็ตหรือจาวาแอฟพลิชั่นจะถูกเขียนโดยใช้ JDBC API Bridge จะทำการแปลงการเรียกใช้ JDBC ไปยัง ODBC Driver ที่เหมาะสมสำหรับฐานข้อมูล ข้อดีของ Bridge ทำให้แอฟพลิชั่นสามารถติดต่อกับฐานข้อมูลได้ง่ายจากผู้ผลิตที่มากโดยสามารถเลือกได้จาก ODBC Driver ที่เหมาะสม

4.3.4 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);`
- สร้างออบเจกต์ Statement
`Statement statement = dbconn.createStatement();`
- ประมวลผลคำสั่ง SQL โดยใช้คำสั่งออบเจกต์ Statement
`statement.executeQuery("SELECT * FROM STUDENT");`
- รับผลจากการประมวลผลคำสั่งด้วยออบเจกต์ Resultset
`Resultset result = statement.executeQuery(query);`

- ปิดออบเจกต์ ResultSet
result.close()
- ปิดออบเจกต์ Statement
statement.close();
- ปิดออบเจกต์ Connection
dbcon.close();



รูปที่ 4-5 ลำดับการเชื่อมต่อเข้าฐานข้อมูลโดยใช้ JDBC

4.4 ประโยชน์ของการใช้ VisualAge for Java

4.4.1 การพัฒนาแอปพลิเคชันได้อย่างรวดเร็ว การเขียนโปรแกรมทั้ง applet และ application ซึ่งสามารถออกแบบได้ในลักษณะของ visual programming ซึ่งสามารถใช้ Visual Composition Editor (VCE) ของ VisualAge for Java ในการทำงานด้านต่าง ๆ ดังนี้คือ

- ออกแบบในส่วนที่ติดต่อกับผู้ใช้ (user interface)
- กำหนดเหตุการณ์ที่เกิดขึ้นของส่วนประกอบต่างๆในส่วนที่ติดต่อกับผู้ใช้ได้เช่นการกดปุ่ม การเลือกรายการต่างๆ เป็นต้น
- กำหนดความสัมพันธ์กันระหว่างส่วนที่ติดต่อกับผู้ใช้กับโปรแกรมอื่นๆได้

4.4.2 เพิ่มความสามารถในการคอมไพล์ VisualAge for Java จะทำการคอมไพล์ซอร์สโค้ดเป็นไบต์โค้ดให้อย่างอัตโนมัติเมื่อซอร์สโค้ดถูก import เข้าไปใน workspace (จากไฟล์จาวา) และเมื่อมีการแก้ไข ลบ เคลื่อนย้าย หรือเปลี่ยนชื่อส่วนประกอบต่างๆของโปรแกรมและจะมีการคอมไพล์อีกครั้งอย่างอัตโนมัติและจะแสดงผลกระทบของการกระทำดังกล่าวด้วยว่ามีข้อผิดพลาดที่ใดบ้าง และเมื่อมีการแสดงข้อผิดพลาดเกิดขึ้นก็จะมีแจ้งเตือนและเราสามารถที่จะแก้ไขปัญหานั้นได้เลยหรือจะแก้ภายหลังก็ได้

4.4.3 VisualAge for Java มีระบบการจัดการโค้ดที่ดีเพื่อให้ง่ายในการแก้ไขที่มีหลายๆ edition เมื่อต้องการที่จะนำโปรแกรมที่พัฒนาแล้วสุดไปใช้งานก็สามารถตั้งค่า version ของโปรแกรมที่เราพัฒนาได้และเมื่อมีข้อผิดพลาดก็กลับมาแก้ไขได้เพื่อแสดงถึงรอบของการพัฒนาด้วย

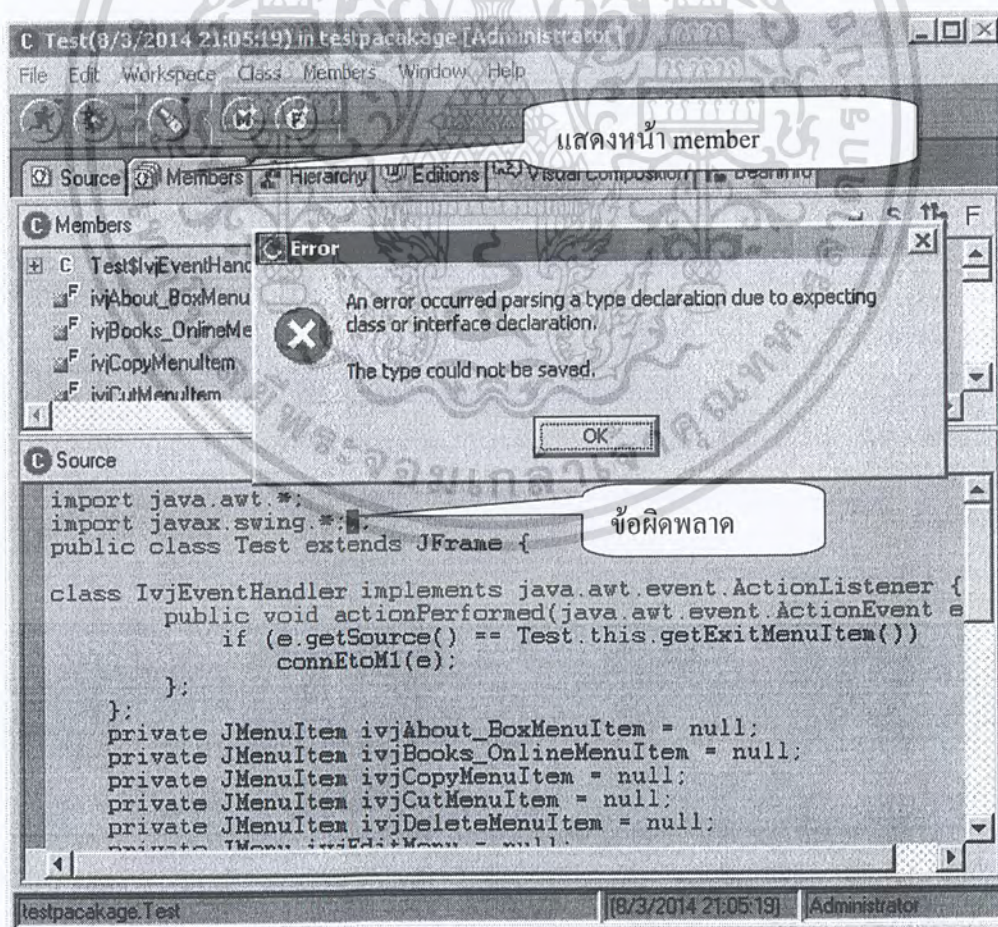
4.4.4 การพัฒนาโปรแกรมที่มีจุดคือ

- สามารถสร้าง แก้ไข และใช้ JavaBean ได้
- แสดงโค้ดเป็นชั้นลำดับชั้น เช่น project, package, class หรือ method ได้
- สามารถใช้ visual debugger ช่วยในขณะรันโปรแกรมได้

4.5 ข้อดีของ VisualAge for Java

4.5.1 สามารถออกแบบส่วนติดต่อกับผู้ใช้ (user interface) สามารถทำได้ง่ายมากโดยใช้ Visual Composite Editor ทำให้การออกแบบเป็นไปได้อย่างรวดเร็ว

4.5.2 ในกรณีที่ผู้เขียนโปรแกรมต้องการบันทึกโค้ด แต่โค้ดนั้นมีข้อผิดพลาดในหน้าของ member ในลักษณะของ type checking (Parser) แต่ VisualAge for Java จะไม่ยอมให้มีการบันทึกแต่อย่างใด โดยจะแจ้งข้อความให้ทราบเช่นดังรูป



รูปที่ 4-6 แสดงข้อดีกรณีของ Parser

4.6 ข้อเสียของ VisualAge for Java

4.6.1 ปัญหาเรื่องการแสดงผลภาษาไทย (Thai Font) จะมีฟอนต์ให้เลือกเพียงไม่กี่รายการเท่านั้น ทำให้การแสดงผลเป็นภาษาไทย จึงขาดความสวยงามเท่าที่ควร

4.6.2 ในโค้ดโปรแกรมการแสดงผลเป็นภาษาไทยจะเป็นลักษณะของยูนิโค้ดซึ่งถ้าอ่านโค้ดดูแล้วจะอ่านไม่รู้เรื่องทำให้การแก้ปัญหาทำได้ลำบากขึ้น

4.6.3 ใช้ทรัพยากรจำนวนมาก ทำให้ทำงานได้ช้า ในขณะที่ทำการรันโปรแกรม



บทที่ 5

การวิเคราะห์และออกแบบระบบทะเบียนนักศึกษาในระดับบัณฑิตศึกษา

ในบทนี้จะได้กล่าวถึงขั้นตอนการวิเคราะห์และออกแบบระบบทะเบียนนักศึกษาในระดับบัณฑิตศึกษา โดยเริ่มด้วยการอธิบายถึงหน้าที่ความรับผิดชอบ การดูแลงานด้านทะเบียนนักศึกษาของฝ่ายทะเบียนนักศึกษา บัณฑิตวิทยาลัย และวิเคราะห์ถึงข้อเสีย และจุดบกพร่องต่าง ๆ ของระบบทะเบียนนักศึกษาที่ใช้อยู่ในปัจจุบัน หลังจากนั้นก็จะกล่าวถึงการตอนการวิเคราะห์และการออกแบบ โดยการนำเอายูเอ็มแอลเข้ามาอธิบายลักษณะของระบบ ดังนี้

5.1 ฝ่ายทะเบียนนักศึกษาของบัณฑิตวิทยาลัย

ฝ่ายทะเบียนนักศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เป็นหน่วยงานที่รับผิดชอบงานในเรื่องของทะเบียนนักศึกษา ในระดับบัณฑิตศึกษาทุกชั้นตอนนับตั้งแต่ การรับนักศึกษาใหม่ การขึ้นทะเบียนนักศึกษาใหม่ การลงทะเบียน การแก้ไขข้อมูลการลงทะเบียน การเปิดปิดรายวิชา การกรอกข้อมูลเกรด การเพิ่มแก้ไขข้อมูลหลักสูตร การดำเนินการตามคำร้องต่าง ๆ ของนักศึกษา จนถึงเรื่องของการจบการศึกษา

การทำงานของฝ่ายทะเบียนในปัจจุบันนั้น มีระบบสารสนเทศของสำนักวิจัยและบริการคอมพิวเตอร์รองรับการทำงานอยู่ ซึ่งนับว่าเป็นระบบที่ได้อำนวยความสะดวกให้มากพอสมควร แต่ก็ยังคงพบปัญหาในการทำงานอยู่เช่นกัน ในที่นี้ขอยกตัวอย่างหนึ่งในงานที่ฝ่ายทะเบียนต้องรับผิดชอบ นั่นก็คือการลงทะเบียนเรียนของนักศึกษา แต่เดิมนั้นใช้รหัสแท่งเข้าช่วย โดยนักศึกษาจะต้องคัดแถบรหัสแท่งของรายวิชาที่ต้องการจะลงทะเบียน มาแนบลงบนเอกสารการลงทะเบียนแล้วนำเอกสารดังกล่าวไปให้อาจารย์ที่ปรึกษาทำการตรวจสอบและเห็นชอบ หลังจากที่อาจารย์ที่ปรึกษาได้ลงนามเห็นชอบแล้วถึงจะนำเอกสารดังกล่าวมาให้เจ้าหน้าที่ทะเบียน เพื่อเอาข้อมูลลงสู่ระบบ แล้วออกจะออกเอกสารการชำระเงินเพื่อให้นักศึกษานำเงินไปชำระที่ธนาคารที่ได้กำหนดไว้ ก็จะสิ้นสุดขั้นตอนการลงทะเบียน ต่อมาในปีการศึกษา 2544 ได้ทำการปรับระบบให้นักศึกษาสามารถทำการลงทะเบียนผ่านระบบอินเทอร์เน็ตได้ โดยมีการยกเลิกการตรวจสอบของอาจารย์ที่ปรึกษาไป หลังจากที่น่าระบบดังกล่าวเข้ามาใช้ก็ยังคงพบปัญหาหลายอย่างอันเนื่องมาจากเกิดความสับสนในขั้นตอนการปฏิบัติ รวมถึงระบบที่นำมาใช้นี้ไม่ได้มีการตรวจสอบข้อกำหนดในการลงทะเบียนอย่างครบถ้วน อย่างเช่นการตรวจสอบรายวิชาที่จำเป็นต้องเรียนมาก่อน(Prerequisite)เป็นต้น จากตัวอย่างการลงทะเบียนเรียนของนักศึกษานับเป็นงานหนึ่งในหลายงานที่ฝ่ายทะเบียนต้องรับผิดชอบ ซึ่งคงไม่สามารถกล่าวถึงรายละเอียดของงานทุกงานได้ในที่นี้

ในการดำเนินงานของฝ่ายทะเบียน และการตรวจสอบเงื่อนไขที่ระบบไม่ได้รองรับไว้นั้น จะต้องใช้บุคคลเป็นผู้ดำเนินการเป็นส่วนใหญ่ และยังมีขั้นตอนการปฏิบัติงานบางอย่างที่ยังไม่ชัดเจน รวมถึงกฎข้อบังคับต่าง ๆ ที่อาจล้าสมัยอยู่บ้าง ทำให้เกิดความสับสนในการปฏิบัติ ซึ่งเมื่อเกิดปัญหานั้นก็จะมีการแก้ปัญหาเป็นกรณี ๆ ไปซึ่งอาจก่อให้เกิดปัญหานั้นซ้ำอีกได้ในภายหลัง

5.2 ปัญหาที่พบในระบบทะเบียนนักศึกษาเดิม

ก่อนที่จะดำเนินการพัฒนาโครงการนี้ ผู้พัฒนาโครงการได้เข้าไปศึกษาทำความเข้าใจถึงการทำงานเดิมของระบบทะเบียนนักศึกษาระดับบัณฑิตศึกษา ซึ่งทำให้ได้ทราบถึงจุดอ่อนของระบบเดิมอยู่หลายประการ เช่น

5.2.1 ในขั้นตอนการลงทะเบียนของนักศึกษาระดับบัณฑิตศึกษานั้นมีรายละเอียดหลายขั้นตอน ตั้งแต่การเข้าดูรายวิชาที่เปิดสอนร่วมกับการปรึกษาอาจารย์ที่ปรึกษาหลังจากที่ได้ข้อตกลงในการลงทะเบียน แล้วนักศึกษาต้องทำการตัดกระดาษรหัสแท่ง รายวิชาที่ต้องการพร้อมมาดำเนินการลงทะเบียนที่ฝ่ายทะเบียนซึ่งนับว่าเป็นขั้นตอนที่ซับซ้อน

5.2.2 รายวิชาที่เปิดสอนในระดับบัณฑิตศึกษานั้น เปิดเพื่อให้นักศึกษาสามารถเลือกลงทะเบียน ดังนั้นอาจมีบางรายวิชาที่นักศึกษาให้ความสนใจลงทะเบียนน้อย ทำให้รายวิชานั้นต้องปิดลงไปสร้างความเดือดร้อนให้กับนักศึกษาที่ลงทะเบียนวิชานั้น ๆ อยู่

5.2.3 ระบบเดิมยังไม่ได้มีการตรวจสอบเงื่อนไขการลงทะเบียนอย่างจริงจัง อย่างเช่นการตรวจสอบรายวิชาที่จำเป็นต้องเรียนมาก่อน(Prerequisite) จำนวนหน่วยกิตรวม ทำให้เกิดการลงทะเบียนที่ผิดเงื่อนไขได้ง่าย

5.2.4 ประเภทนักศึกษาที่มีหลายประเภทเช่นนักศึกษาสามัญ นักศึกษาทดลองเรียน ซึ่งมีระเบียบปฏิบัติบางอย่างที่แตกต่างกัน

5.2.5 นักศึกษาต้องไปติดต่อกับฝ่ายทะเบียนบ่อยครั้ง ไม่ว่าจะเป็นการยื่นคำร้องต่าง ๆ การเพิ่มแก้ไขการลงทะเบียน การตรวจสอบข้อมูลเกี่ยวกับการลงทะเบียนและเกรด

นั่นคือตัวอย่างของปัญหาที่เกิดขึ้นในระบบเดิม ปัญหาเหล่านี้ก่อให้เกิดความสับสนในการทำงาน จึงเป็นการเหมาะสมที่จะพัฒนาระบบให้มีประสิทธิภาพมากขึ้น

5.3 การวิเคราะห์ระบบ

ในการวิเคราะห์ระบบนั้นนอกจากการเข้าไปศึกษาการทำงานของระบบเดิมแล้ว การหาถึงความต้องการของผู้ใช้ระบบนั้นนับว่าเป็นสิ่งสำคัญอีกประการหนึ่งในการพัฒนาระบบใด ๆ ขึ้นมา ในที่นี้ผู้พัฒนาโครงการได้ทำการสกัดความต้องการของผู้ใช้ระบบอย่างละเอียด โดยขั้นตอนในการสกัดความต้องการของผู้ใช้ดำเนินการหลายวิธีด้วยกันเช่น ทำการสัมภาษณ์กับผู้ใช้ระบบโดยตรง ไม่ว่าจะเป็นเจ้าหน้าที่ทะเบียน อาจารย์และนักศึกษา การเข้าไปศึกษาเอกสาร แบบฟอร์มต่าง ๆ แล้วนำข้อมูลที่ได้มาทำความเข้าใจและวิเคราะห์ให้ได้ทั้งกระบวนการทำงานเดิมและแนวทางที่ผู้ใช้ต้องการ ซึ่งทำให้ได้ความต้องการหลากหลายจากผู้ใช้งานกำหนดฟังก์ชันการทำงานให้กับระบบ นอกจากนั้นผู้พัฒนาโครงการยังได้พยายามเลือกใช้เทคโนโลยีที่เหมาะสม และไม่ต้องลงทุนเพิ่มเติมมากนักอีกด้วย

หลังจากที่ได้ทำการวิเคราะห์ระบบอย่างละเอียด เราได้ทั้งปัญหาของระบบและความต้องการของผู้ใช้ ซึ่งพอที่จะสรุปเป็นแนวทางในการออกแบบและพัฒนาระบบต่อไป ดังนี้

5.3.1 ให้นักศึกษาสามารถทำกิจกรรมและเข้าถึงข้อมูลต่าง ๆ ที่เกี่ยวกับงานทะเบียนได้ โดยที่ไม่ต้องมาติดต่อฝ่ายทะเบียน หรือมาติดต่อให้น้อยที่สุด เพื่อเพิ่มความสะดวกให้นักศึกษาเอง รวมถึงลดภาระงานของเจ้าหน้าที่ นั่นก็คือให้นักศึกษาสามารถทำกิจกรรมเหล่านั้นผ่านทางอินเทอร์เน็ต

5.3.2 ลดภาระการตรวจสอบเงื่อนไขต่าง ๆ ด้วยการสร้างให้โปรแกรมสามารถตรวจสอบเงื่อนไขเหล่านั้นให้ครอบคลุมและถูกต้องมากที่สุด ซึ่งเราต้องทำการศึกษาเงื่อนไขเหล่านั้นให้ชัดเจนและออกแบบเรื่องของโปรแกรมและฐานข้อมูลให้สามารถสนับสนุนการตรวจสอบดังกล่าว

5.3.3 การเพิ่ม แก้ไขข้อมูลต่าง ๆ นั้น เจ้าหน้าที่จะต้องสามารถทำได้อย่างครบถ้วนและสะดวก เพราะข้อมูลที่เกี่ยวข้องกับงานทะเบียนนั้นค่อนข้างซับซ้อน ยกตัวอย่างเช่นข้อมูลของหลักสูตรซึ่งมีข้อมูลย่อย ๆ ประกอบอยู่หลายอย่าง ระบบจึงควรเก็บข้อมูลให้ครอบคลุมให้ได้มากที่สุด

5.3.4 ระบบควรมีความเป็นอัตโนมัติในการแสดงข้อมูลต่าง ๆ ให้กับผู้ใช้ยกตัวอย่างเช่นในการตรวจสอบช่วงเวลาการลงทะเบียน ระบบควรสามารถตรวจสอบได้เอง ไม่ต้องให้เจ้าหน้าที่ทำการตั้งค่าเอง

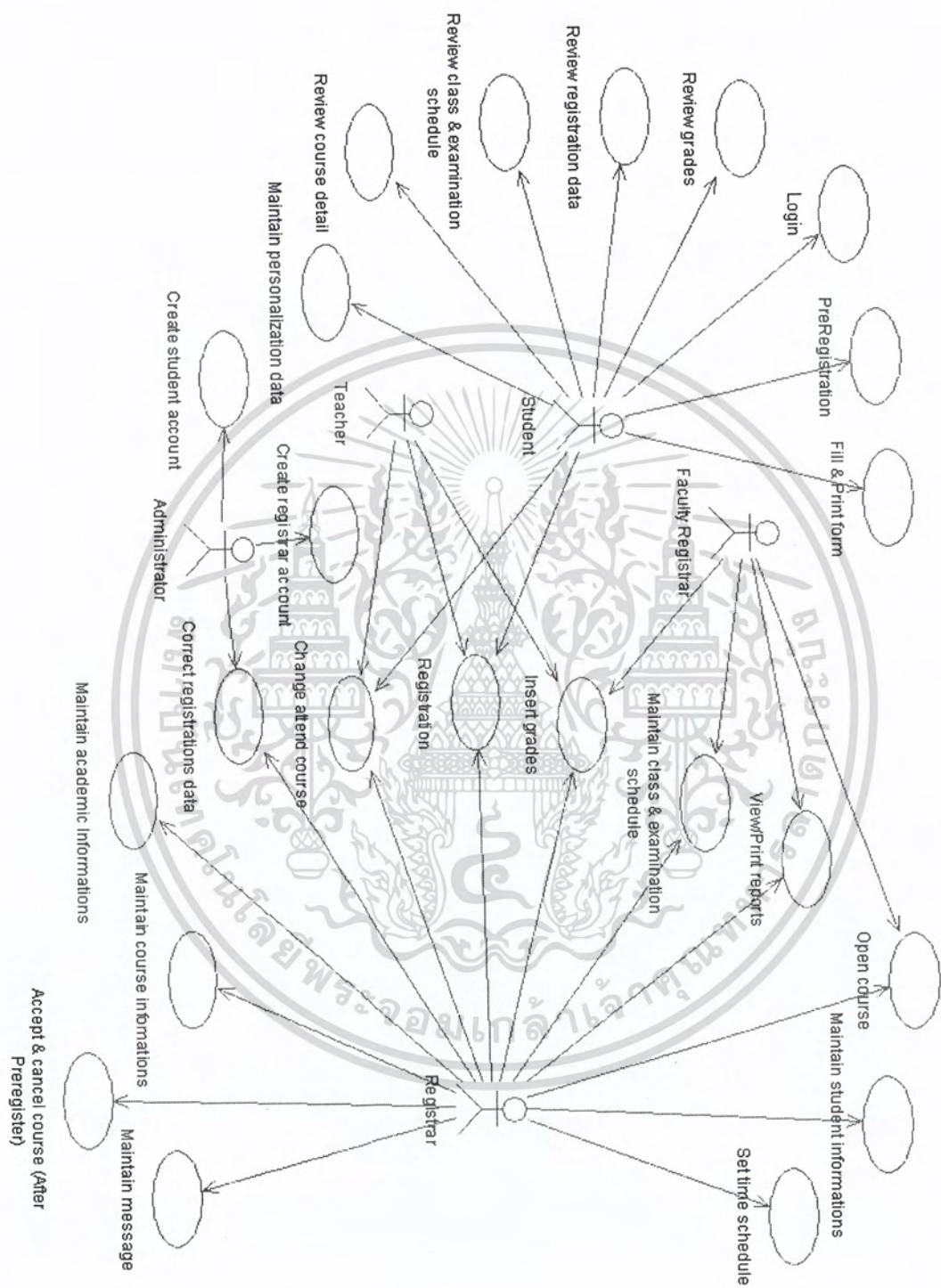
5.4 การออกแบบระบบโดยใช้ยูเอ็มแอล

หลังจากที่เราได้หาความต้องการของผู้ใช้และศึกษาระบบเดิมพอสมควรแล้วเราต้องนำข้อมูลที่ได้มาทำการวิเคราะห์และออกแบบโดยใช้ยูเอ็มแอล แต่ถึงอย่างไรก็ตามในการออกแบบระบบนี้ เราใช้กระบวนการทางวิศวกรรมซอฟต์แวร์แบบวนรอบ (Iteration Software Process) ซึ่งเป็นกระบวนการที่ทำงานเป็นวงรอบ ในแต่ละรอบก็จะมีการปรับปรุงในแต่ละขั้นตอนเสมอ ดังนั้นขั้นตอนที่จะอธิบายต่อไปนี้จะมีการปรับปรุงอยู่ตลอดเวลา และในส่วนนี้จะกล่าวถึงไดอะแกรมต่าง ๆ จึงควรศึกษาในบทที่ 2 ประกอบด้วย

การหาว่ามีใครบ้างที่ต้องใช้ระบบนี้ซึ่งเราเรียกว่าแอคเตอร์ (Actor) และหาว่าแต่ละคนทำอะไรกับระบบบ้างหรือใช้ฟังก์ชันใดของระบบบ้าง เมื่อได้แอคเตอร์และฟังก์ชันการทำงานแล้วนำมาเขียนเป็นยูสเคสไดอะแกรม (Use Case Diagram) ซึ่งเราถือว่าแต่ละฟังก์ชันการทำงานของระบบคือหนึ่งยูสเคส (Use case)

จากรูปที่ 5-1 นั่นคือยูสเคสไดอะแกรมของระบบซึ่งจะเห็นได้ว่าประกอบด้วย 5 แอคเตอร์ซึ่งมีรายละเอียดของแต่ละแอคเตอร์ดังนี้

- Student ก็คือนักศึกษาผู้ใช้ระบบ ซึ่งได้แก่นักศึกษาในระดับบัณฑิตศึกษา ซึ่งสามารถที่จะใช้งานระบบผ่านทางอินเทอร์เน็ต
- Registrar ก็คือเจ้าหน้าที่ทะเบียนของบัณฑิตวิทยาลัย ซึ่งจะเป็นผู้ที่ดูแล เพิ่ม แก้ไขข้อมูลของระบบ การทำงานกับระบบนั้นจะผ่านทางโปรแกรมที่จะพัฒนาขึ้น
- Faculty Registrar ก็คือเจ้าหน้าที่งานบัณฑิตศึกษาในส่วนคณะต่าง ๆ ก็จะสามารถในการเพิ่ม แก้ไข ข้อมูลบางอย่างของระบบ เพื่อเป็นการแบ่งภาระงานของเจ้าหน้าที่ทะเบียนของบัณฑิตวิทยาลัย รวมถึงการตัดตอนกระบวนการทำงาน ทำให้มีโอกาสเกิดข้อผิดพลาดน้อยลง
- Teacher ก็คืออาจารย์ซึ่ง อาจจะไม่ได้ใช้งานระบบโดยตรง แต่ก็มีส่วนเกี่ยวข้องกับระบบในงานบางงานเช่นการกรอกเกรดของนักศึกษา การลงทะเบียนเป็นต้น
- Administrator คือผู้ดูแลระบบซึ่งจะเป็นผู้ที่จัดการเกี่ยวกับระบบโดยตรงกับฐานข้อมูลทำให้สามารถจัดการกับข้อมูลได้ทั้งหมด รวมถึงเป็นผู้ที่ดูแลเกี่ยวกับความปลอดภัยของระบบด้วย



รูปที่ 5-1 Usecase ของระบบ

จากคำอธิบายข้างต้น ฟังก์ชันการทำงานต่าง ๆ ของระบบที่แอดมินสามารถใช้ได้นั้นก็จะประกอบไปด้วย 3 ส่วนนั้นคือ

- ส่วนที่เป็นแอปพลิเคชันซึ่งเป็นโปรแกรมที่พัฒนาด้วยภาษาจาวาเพื่อให้เจ้าหน้าที่ทะเบียนทั้งส่วนกลางและส่วนคณะได้ใช้
- ส่วนที่เป็นเว็บแอปพลิเคชัน ซึ่งเป็นส่วนที่ให้นักศึกษาใช้ในการติดต่อกับระบบ
- ส่วนที่เป็นเอกสาร หรือโปรแกรมพื้นฐานที่มีอยู่แล้ว ซึ่งไม่ต้องทำการพัฒนาอย่างเช่นระบบงานในส่วนของผู้ดูแลระบบที่ใช้โปรแกรมจัดการฐานข้อมูล เป็นต้น

นอกจากในส่วนของแอดมินแล้ว จะขออธิบายถึงฟังก์ชันการทำงานต่าง ๆ ของระบบนั้นก็คือยูสเคสแต่ละยูสเคสนั้นเอง มีรายละเอียดดังนี้

- Login เป็นส่วนที่ให้นักศึกษาเข้าสู่ระบบโดยการกรอก Username และรหัสผ่าน นอกจากจะเป็นการเข้าสู่ระบบแล้ว ยูสเคสนี้จะทำการประเมินข้อมูลต่าง ๆ ของนักศึกษาคนนั้นแล้วแสดงออกมาด้วย
- Maintain personalization data เป็นส่วนที่ให้นักศึกษาสามารถแก้ไขข้อมูลส่วนตัวบางส่วนอย่างเช่นที่อยู่ปัจจุบัน อีเมล เป็นต้น
- Review class & examination schedule เป็นส่วนที่ให้นักศึกษาจะเข้าไปดูตารางเรียนและตารางสอบ
- Review course detail เป็นส่วนที่ให้นักศึกษาดูรายละเอียดของวิชาที่เปิดสอน และข้อมูลต่าง ๆ ของรายวิชานั้น ๆ
- PreRegistration เป็นส่วนที่ให้นักศึกษาทำการลงทะเบียนล่วงหน้า
- Registration เป็นส่วนที่ให้นักศึกษาลงทะเบียนจริง
- Change attend course เป็นส่วนที่ให้นักศึกษาเพิ่มเปลี่ยนถอน รายวิชา
- Review registration data เป็นส่วนที่ให้นักศึกษาดูผลการลงทะเบียน
- Review grades เป็นส่วนที่ให้นักศึกษาดูผลการศึกษา
- Fill & Print form เป็นส่วนที่ให้นักศึกษาสามารถกรอกคำร้องและพิมพ์ออกมา เพื่อนำไปดำเนินการต่อไป ซึ่งนักศึกษาไม่จำเป็นต้องกรอกข้อมูลทั้งหมด เพราะข้อมูลบางอย่างจะถูกอ่านขึ้นมาจากฐานข้อมูลให้
- Insert grades เป็นส่วนที่ให้อาจารย์ และเจ้าหน้าที่กรอกเกรดเข้าสู่ระบบ
- Maintain student information เป็นส่วนที่ให้เจ้าหน้าที่เพิ่มแก้ไขข้อมูลนักศึกษา
- Maintain course information เป็นส่วนที่ให้เจ้าหน้าที่เพิ่มแก้ไขข้อมูลหลักสูตร
- Maintain academic information เป็นส่วนที่ให้เจ้าหน้าที่เพิ่มแก้ไขข้อมูลหน่วยงานการศึกษา
- Maintain class & examination schedule เป็นส่วนที่ให้เจ้าหน้าที่เพิ่มแก้ไขข้อมูลตารางเรียนและตารางสอบ
- Maintain message เป็นส่วนที่ให้เจ้าหน้าที่เพิ่มแก้ไขข้อมูลข่าวสาร

- Open course เป็นส่วนที่ให้เจ้าหน้าที่เพิ่มแก้ไขข้อมูลการเปิดปิดรายวิชา
- Set time schedule เป็นส่วนที่ให้เจ้าหน้าที่เพิ่มแก้ไขข้อมูลกำหนดระยะเวลาต่าง ๆ เช่นระยะเวลาการลงทะเบียน
- Accept & cancel course (After Reregister) เป็นส่วนที่ให้เจ้าหน้าที่สั่งให้ระบบประเมินว่ามีรายวิชาใดที่มีผู้ลงทะเบียนน้อยจากการลงทะเบียนล่วงหน้า และให้สิทธิเจ้าหน้าที่ในการปิดรายวิชานั้น
- View/Print report เป็นส่วนที่ให้เจ้าหน้าที่ดู และพิมพ์รายงานต่าง ๆ จากระบบ
- Correct registrations data เป็นส่วนที่ให้เจ้าหน้าที่และผู้ดูแลระบบแก้ไขข้อผิดพลาดต่าง ๆ ที่อาจเกิดขึ้นได้
- Create student account ให้ผู้ดูแลระบบสร้างรหัสผ่านให้กับนักศึกษา
- Create registrar account ให้ผู้ดูแลระบบสร้างรหัสผ่านให้กับเจ้าหน้าที่ทะเบียน

5.5 การออกแบบการทำงานของฟังก์ชันต่าง ๆ

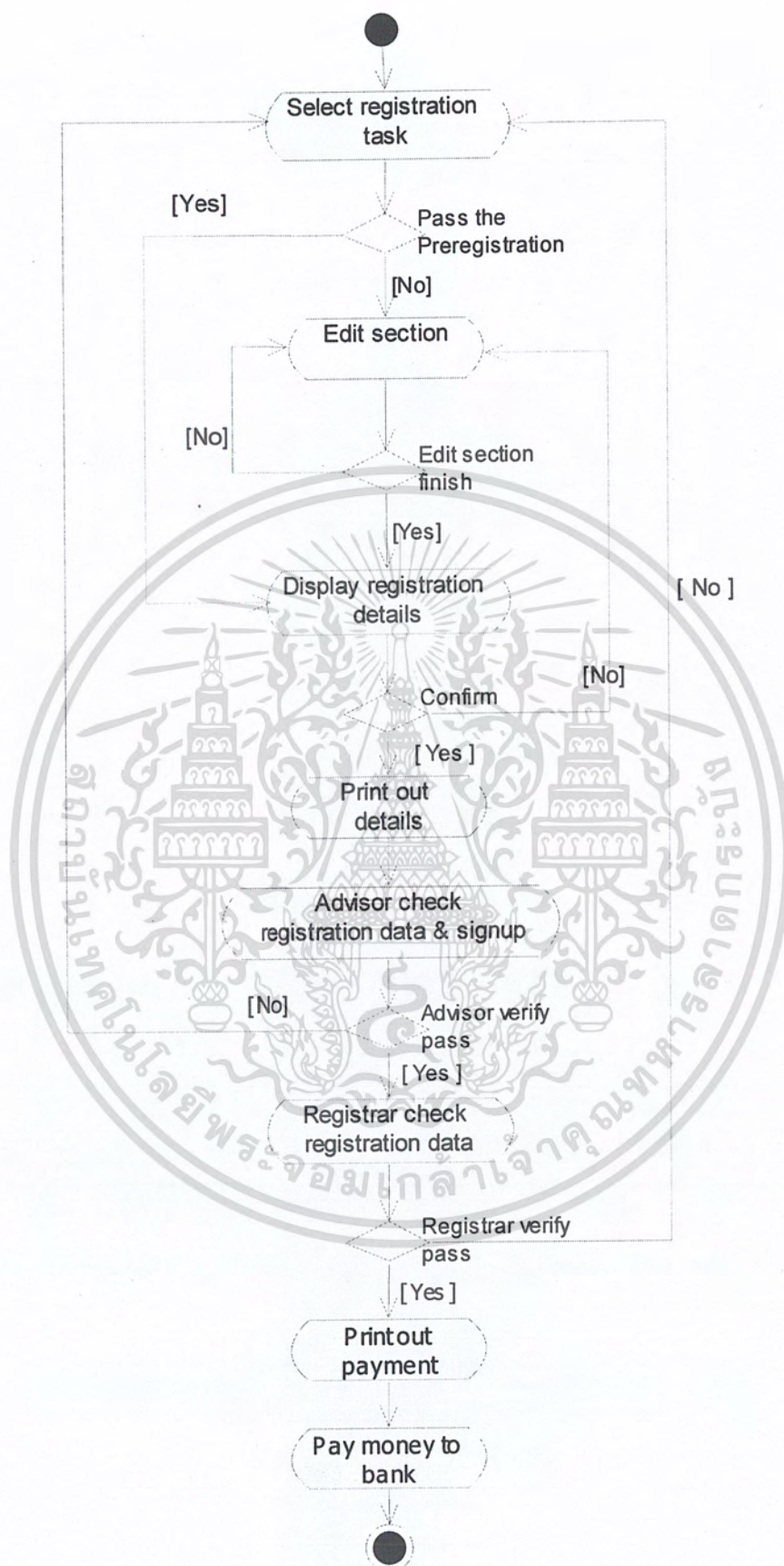
จะเห็นได้ว่าหลังจากขั้นตอนที่ผ่านมาเราจะได้ฟังก์ชันการทำงานต่าง ๆ ของระบบทั้งหมด ซึ่งเราสามารถมาทำการแยกออกแบบทีละฟังก์ชัน ซึ่งทำให้เราสามารถออกแบบได้ง่ายขึ้น สิ่งแรกที่เราจะต้องทำก็คือการอธิบายว่าในแต่ละฟังก์ชันมีกิจกรรมใดเกิดขึ้นบ้าง ซึ่งเราใช้แอกทิวิตีไดอะแกรมในการอธิบายสิ่งดังกล่าว

จากรูปที่ 5-2 จะเป็นแอกทิวิตีไดอะแกรมของ Registration Usecase ซึ่งก็คือขั้นตอนการลงทะเบียนนั่นเอง จากไดอะแกรมจะเห็นว่าเริ่มต้นเมื่อนักศึกษาเลือกที่จะทำการลงทะเบียนระบบจะตรวจสอบว่านักศึกษาค้นนี้ได้มีการลงทะเบียนล่วงหน้าหรือยัง หรือมีรายวิชาใดที่เขาลงทะเบียนล่วงหน้าไว้แล้วรายวิชานั้นถูกปิดไปหรือไม่ ซึ่งถ้าไม่มีปัญหาอะไร ระบบก็จะแสดงรายละเอียดค่าใช้จ่าย และการลงทะเบียนเพื่อให้นักศึกษาทำการยืนยันเมื่อเสร็จขั้นตอนนี้นักศึกษาจะต้องทำการพิมพ์เอกสารนี้ไปให้อาจารย์ที่ปรึกษาตรวจสอบและลงลายมือชื่อรับรอง แล้วจึงไปทำการติดต่อเจ้าหน้าที่ทะเบียนเพื่อบันทึกข้อมูลลงสู่ระบบ แล้วออกไป payment เพื่อให้นักศึกษาไปชำระเงินที่ธนาคารต่อไป ซึ่งแอกทิวิตีไดอะแกรมนี้เป็นหนึ่งในตัวอย่างของยูสเคสต่าง ๆ ซึ่งต้องทำเช่นนี้เหมือนกันจนครบ

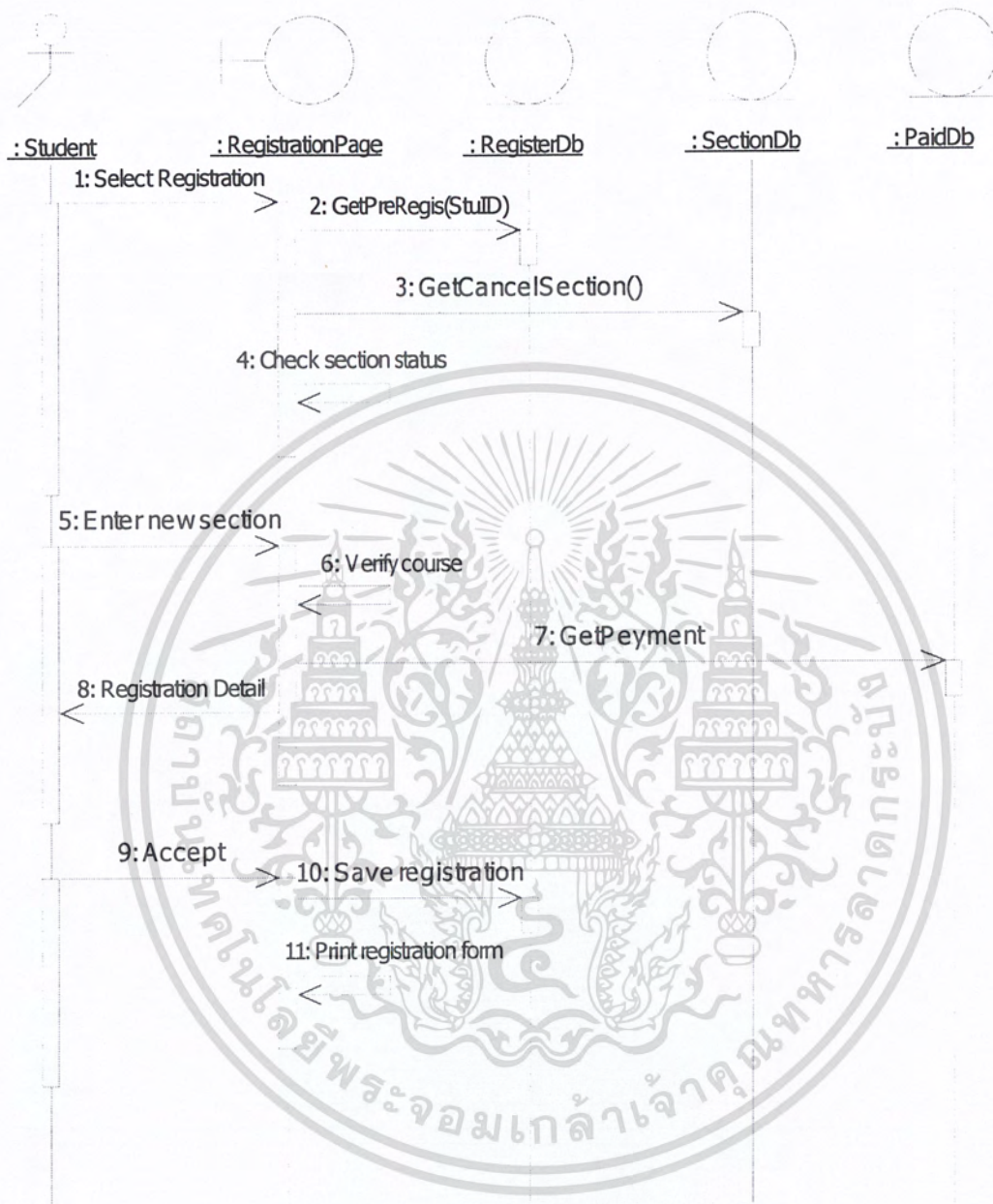
สิ่งที่ต้องทำต่อไปก็คือการอธิบายว่าในแต่ละฟังก์ชันการทำงานนั้น มี Class ใดบ้าง และมีการทำงานสื่อสารกันอย่างไร ซึ่งเราใช้ซีเควนซ์ไดอะแกรม(Sequence Diagram) เป็นตัวอธิบายสิ่งดังกล่าว ซึ่งแต่ละยูสเคสอาจมีมากกว่าหนึ่งซีเควนซ์ไดอะแกรม หรืออาจไม่มีก็ได้ หลังจากที่ได้ซีเควนซ์ไดอะแกรมเราสามารถนำไปสร้างเป็น Collaboration Diagram ซึ่ง Rational Rose สามารถสร้างให้เราได้อย่างอัตโนมัติ

จากรูป 5-3 ก็เป็นตัวอย่างของซีเควนซ์ไดอะแกรมของ Registration Usecase ซึ่งจะเห็นได้ว่าเป็นการส่งข้อมูลกันระหว่างคลาสต่าง ๆ ทำให้เราสามารถเข้าใจและง่ายในการนำไปเขียนเป็นโปรแกรมต่อไป นอกจากนี้ซีเควนซ์ไดอะแกรมแล้วยังมี Collaboration Diagram ที่สามารถอธิบายข้อมูลดังกล่าวได้เช่นกัน

ในการการออกแบบระบบในส่วนของการเขียนไดอะแกรมต่าง ๆ เหล่านี้คำนึงถึงการนำไปเขียนเป็นโปรแกรมได้ นั่นหมายความว่าแต่ละคลาสที่ได้จากการเขียนไดอะแกรมก็จะนำไปเขียนเป็นคลาสจริง ๆ ดังนั้นการสกัดเพื่อหาคลาสเหล่านี้จึงต้องทำอย่างรอบคอบและคำนึงถึงขนาดและหน้าที่ของแต่ละคลาสด้วย



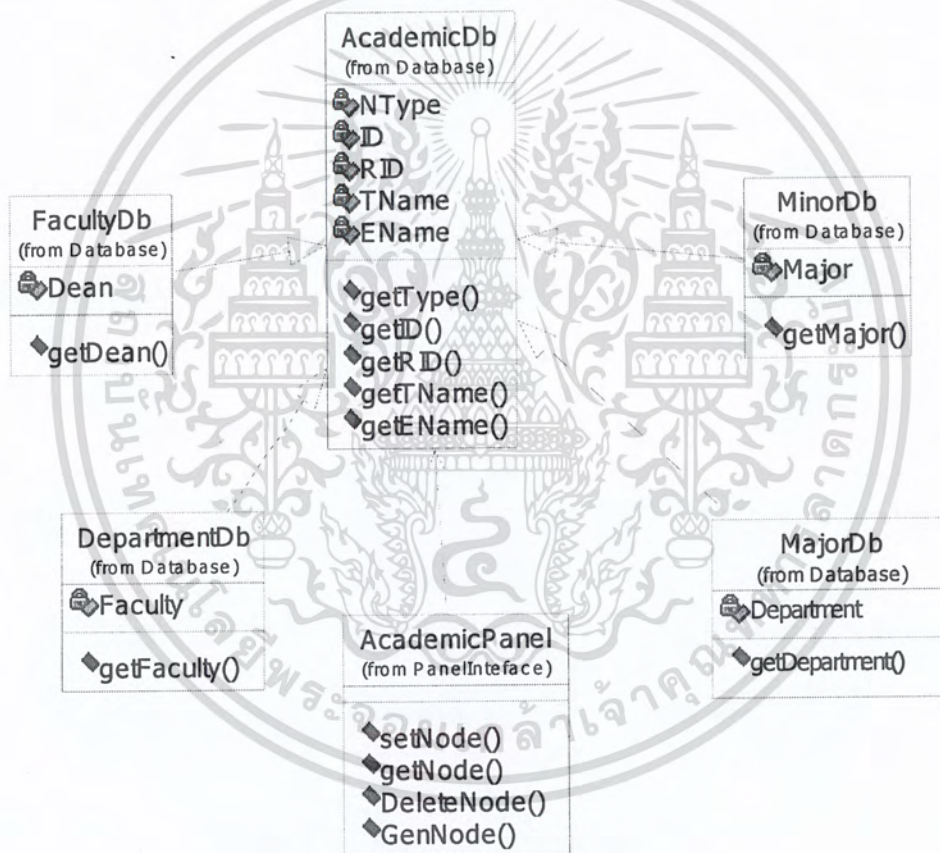
รูปที่ 5-2 Activity Diagram ของ Registration Usecase



รูปที่ 5-3 Sequence Diagram ของ Registration Usecase

5.6 การออกแบบการรายละเอียดของคลาสต่าง ๆ

มาถึงขั้นตอนนี้เราจะได้คลาสที่ใช้ในระบบอย่างครบถ้วน เราก็จะนำคลาสเหล่านี้มากำหนดความสัมพันธ์ระหว่างกัน รวมถึงอาจจัดคลาสให้เป็นกลุ่มแล้วเก็บไว้ในแพ็คเกจ(Package) แล้วกำหนดความสัมพันธ์ระหว่างแพ็คเกจด้วยก็ได้ แล้วปรับแต่งสีแฉกโค๊ดโปรแกรมเพื่อให้การส่งข้อมูลระหว่างคลาสอยู่ในรูปของการเรียกใช้เมธอด(Method) ให้ได้มากที่สุด กำหนดคุณสมบัติของคลาสแต่ละคลาส กำหนดแอตทริบิวต์(Attribute) และเมธอดของคลาสให้ครบถ้วน แล้วนำไปเขียนเป็นคลาสโค๊ดโปรแกรม เขียนความสัมพันธ์ในรูปของการสืบทอดคุณสมบัติของคลาส ดูรายละเอียดที่คลาสของระบบว่ามีคลาสใดที่แยกกันได้ รวมกันได้ หรือกำจัดออกได้ เพื่อลดความซ้ำซ้อนของคลาส



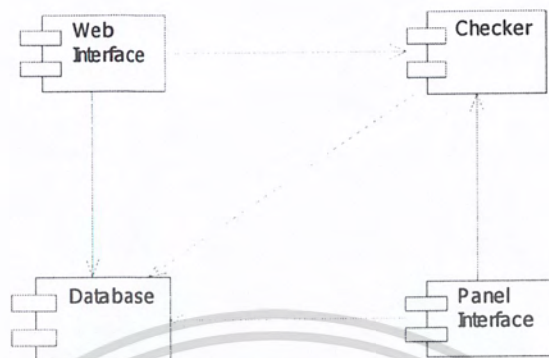
รูปที่ 5-4 ตัวอย่าง Class Diagram ของระบบ

จากรูป 5-4 เป็นตัวอย่างของหนึ่งในคลาสโค๊ดโปรแกรมของระบบซึ่งจะอธิบายว่าคลาสแต่ละคลาสมีรายละเอียดอย่างไรและสัมพันธ์กับคลาสอื่น ๆ อย่างไร

ส่วนที่ใช้อธิบายคลาสอีกอย่างหนึ่งคือสแตตโค๊ดโปรแกรม(State Diagram) ซึ่งจะใช้อธิบายคลาสที่มีการทำงานภายในเป็นแบบสแตตโค๊ด

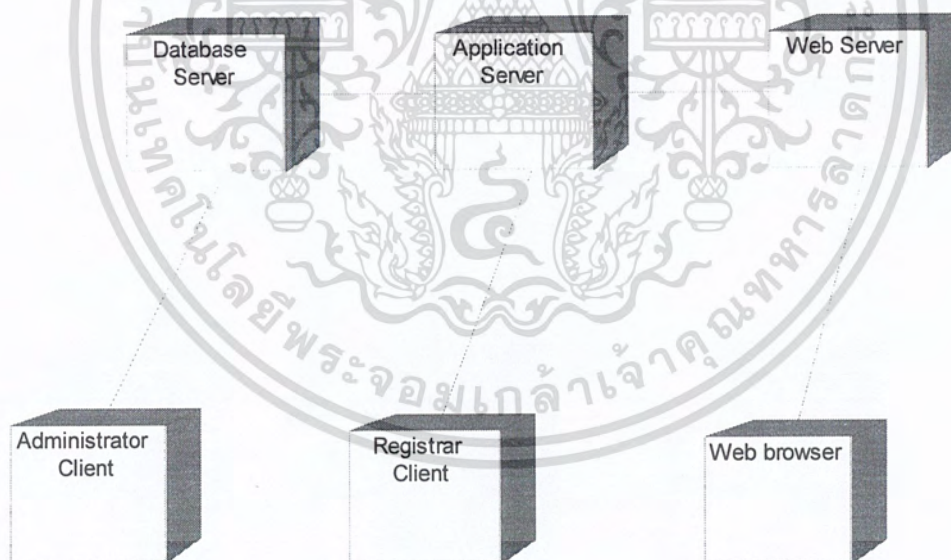
5.7 การออกแบบโครงสร้างของระบบ

อย่างแรกคือเขียนคอมโพเนนต์ไคอะแกรม(Component Diagram) ซึ่งจะต้องมองว่าโปรแกรมของเราจะแบ่งส่วนของโปรแกรมอย่างไร



รูปที่ 5-5 Component Diagram ของระบบ

เขียนดีพอยเมนต์ไคอะแกรม(Deployment Diagram) ซึ่งจะต้องดูว่าระบบของเราประกอบด้วยโครงสร้างของระบบทางกายภาพอย่างไร



รูปที่ 5-6 Deployment Diagram ของระบบ

ที่ได้กล่าวมาทั้งหมดนี้คือขั้นตอนการวิเคราะห์และออกแบบระบบ ซึ่งได้ยกเอาไคอะแกรมบางไคอะแกรมมาเพื่ออธิบายขั้นตอนการทำงาน ส่วนไคอะแกรมทั้งหมดในโครงการนี้จะอยู่ในส่วนของภาคผนวก ก ท้ายเล่ม

บทที่ 6

การพัฒนาและทดสอบระบบ

ในส่วนของบทนี้จะได้กล่าวถึงขั้นตอนการพัฒนาและทดสอบระบบ ซึ่งเราจะเริ่มตั้งแต่การสร้างในส่วน ofฐานข้อมูลขึ้นมาก่อน แล้วจึงทำการเขียนโปรแกรมซึ่งก็แบ่งออกเป็น 2 ส่วนก็คือส่วนที่เป็นแอปพลิเคชัน ให้สำหรับเจ้าหน้าที่ทะเบียน และส่วนที่เป็นเว็บแอปพลิเคชันสำหรับนักศึกษา หลังจากที่เรเขียนโปรแกรมเสร็จก็จะถึงขั้นตอนการทดสอบระบบ ซึ่งมีรายละเอียดดังนี้

6.1 การสร้างฐานข้อมูล

ในการสร้างฐานข้อมูลนั้นเราอ้างอิงจากการนำเอาโครงสร้างของคลาสต่าง ๆ ในระบบมาสร้างเป็นโครงสร้างตารางฐานข้อมูล และสร้างเป็นตัวฐานข้อมูลใน DB2 ซึ่งมีขั้นตอนดังนี้

6.1.1 นำเอาโครงสร้างของคลาสมาเขียนเป็นอีอาร์ไดอะแกรม(ER Diagram) ซึ่งจะสามารถอธิบายถึงลักษณะความสัมพันธ์ระหว่างแต่ละเอนติตี้(Entity) ได้อย่างชัดเจน

6.1.2 เราก็จะนำเอาอีอาร์ไดอะแกรมมาแปลงเป็นโครงสร้างตารางฐานข้อมูล(Database Schema)โดยใช้ทฤษฎีทางฐานข้อมูลซึ่งไม่ขอกล่าวในรายละเอียด

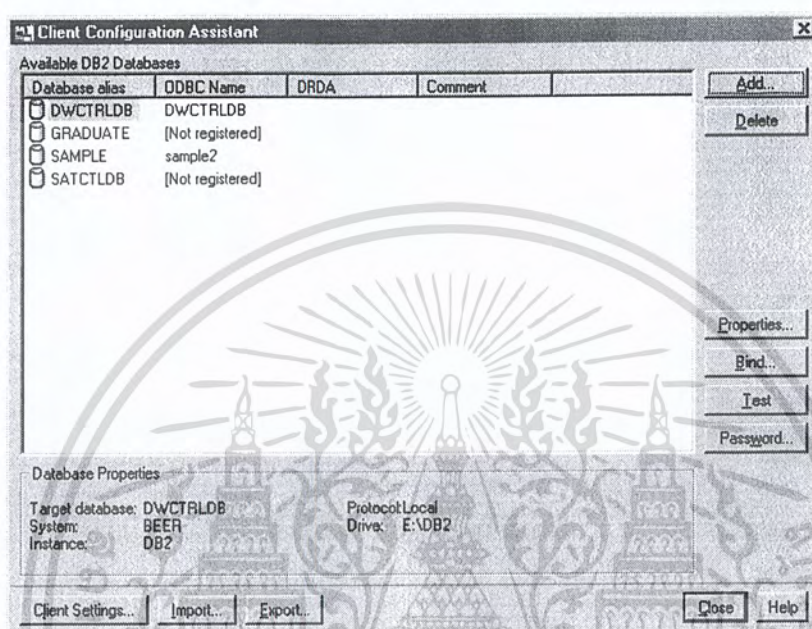
6.1.3 นำโครงสร้างตารางฐานข้อมูลที่ได้ไปทำการนอร์มัลไลซ์(Normalization) ซึ่งจากทฤษฎีแล้วโครงสร้างตารางฐานข้อมูลที่ได้จากการแปลงมาจากอีอาร์ไดอะแกรมนั้นจะยังไม่อยู่ในนอร์มัลฟอร์มแรก(First Normal Form) เราจึงต้องทำการนอร์มัลไลซ์จนโครงสร้างตารางฐานข้อมูลอยู่ในนอร์มัลฟอร์มที่ห้า(Fifth Normal Form) ซึ่งสามารถดูรายละเอียดของตารางที่ได้ในภาคผนวก ข

Subject		
SubID	VARCHAR(8)	รหัสวิชา(PK)
EName	VARCHAR(85)	ชื่อวิชาภาษาอังกฤษ
TName	VARCHAR(85)	ชื่อวิชาภาษาไทย
SType	VARCHAR(30)	กลุ่มวิชา
Credit	DECIMAL(2)	หน่วยกิตรวม
Lecture	DECIMAL(2)	เวลาบรรยาย
Practice	DECIMAL(2)	เวลาปฏิบัติ
Research	DECIMAL(2)	เวลาค้นคว้า
OutlineE	LONG VARCHAR	คำอธิบายรายวิชาภาษาอังกฤษ
OutlineT	LONG VARCHAR	คำอธิบายรายวิชาภาษาไทย
CoID	SMALLINT	รหัสหลักสูตร(FK) (PK)

ตารางที่ 6-1 ตัวอย่างตารางที่อยู่ในนอร์มัลฟอร์มที่ห้า(Fifth Normal Form)

6.1.4 กำหนดรายละเอียดของชนิดข้อมูล(Data Type) ของแต่ละแอตทริบิวต์โดยอ้างอิงตามชนิดข้อมูลของ DB2

6.1.5 ทำการสร้างฐานข้อมูลบน DB2 โดยใช้โปรแกรม Client Configuration Assistant ซึ่งเป็นโปรแกรมที่ให้มากับ DB2 ในการติดต่อกับเซิร์ฟเวอร์ฐานข้อมูล(Database Server) และสร้างฐานข้อมูลขึ้นโดยใช้โปรแกรม Control Center

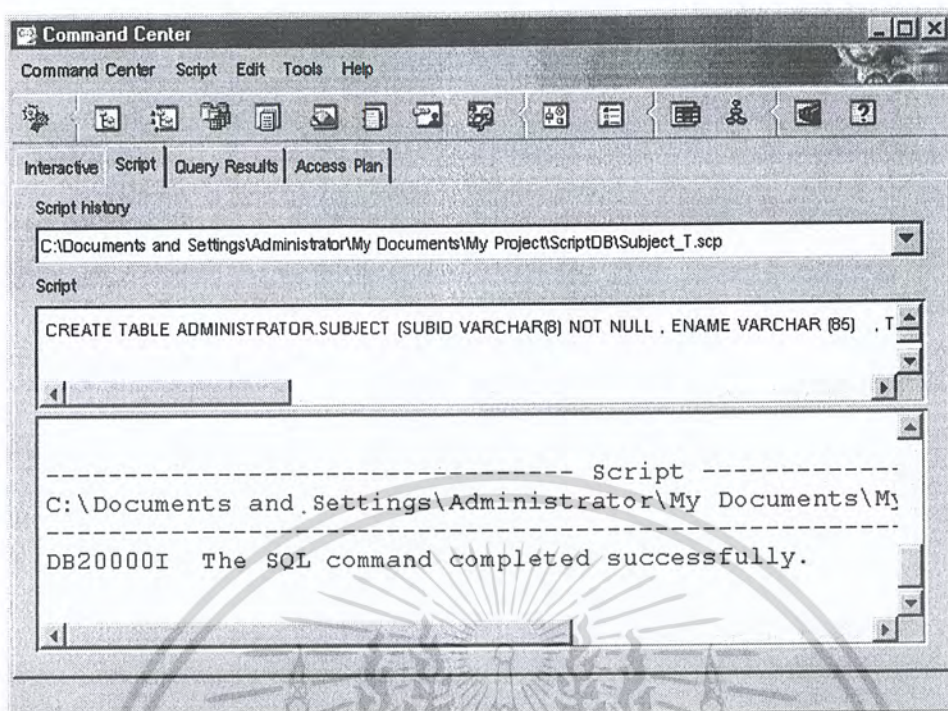


รูปที่ 6-1 การเชื่อมต่อฐานข้อมูลด้วย Client Configuration Assistant

6.1.6 ทำการสร้างตารางแต่ละตารางโดยเขียนเป็นสคริปต์(Script) ก่อนแล้วจึงนำไปสร้างตารางโดยใช้โปรแกรม Command Center เป็นโปรแกรมในการสร้างตารางแต่ละตารางขึ้น ซึ่งได้ยกตัวอย่างสคริปต์ของตารางข้อมูลของคณะมาในที่นี้ พร้อมทั้งรูปที่แสดงการใช้ Command Center เพื่อการสร้างตารางด้วย

```
CREATE TABLE ADMINISTRATOR.SUBJECT (SUBID VARCHAR(8) NOT NULL ,
ENAME VARCHAR (85) , TNAME VARCHAR (85) , STYPE VARCHAR (30) ,
CREDIT SMALLINT , LECTURE SMALLINT , PRACTICE SMALLINT , OUTLINEE
LONG VARCHAR , OUTLINET LONG VARCHAR , COID SMALLINT NOT NULL
, PRIMARY KEY (SUBID, COID) , FOREIGN KEY (COID) REFERENCES
ADMINISTRATOR.COURSE (COID) ON DELETE NO ACTION ON UPDATE NO
ACTION ) DATA CAPTURE NONE
```

ตัวอย่างสคริปต์ในการสร้างตารางของรายวิชา



รูปที่ 6-2 การใช้ Command Center เพื่อการสร้างตาราง

6.1.6 ทำการกรอกข้อมูลบางอย่างเพื่อเป็นตัวอย่างในการทดลองเขียนโปรแกรมโดยใช้โปรแกรม Command Center

6.1.7 ในการติดต่อกับฐานข้อมูลต่าง ๆ เราจะใช้โปรแกรม Command Center ซึ่งสามารถทำงานด้วยภาษาเอสคิวแอล(SQL)ได้ และโปรแกรม Control Center ซึ่งจะเป็นตัวจัดการในการบริหารฐานข้อมูล

หลังจากทั้ง 7 ขั้นตอนนี้เราจะมีฐานข้อมูลอยู่บนเครื่องเซิร์ฟเวอร์ที่เราสามารถติดต่อเข้ามาใช้งานได้จากเครื่องคอมพิวเตอร์อื่นได้โดยผ่านทางโปรโตคอลที่ซีพีทีบีไอพี(TCP/IP Protocol)

6.2 การเขียนโปรแกรมในส่วนของแอปพลิเคชัน

แอปพลิเคชันที่สร้างขึ้นนี้จากข้างต้นที่เคยกล่าวแล้วว่าใช้ภาษาจาวา ซึ่งเป็นภาษาที่มีข้อดีหลายประการและรองรับการพัฒนาโปรแกรมแบบ Object Oriented ได้เป็นอย่างดี รวมถึงใช้โปรแกรม IBM Java VisualAge 4.0 เป็นเครื่องมือในการพัฒนา ในการเขียนโปรแกรมนั้นค่อนข้างมีรายละเอียดมากมาย ผู้พัฒนาจะพยายามทำให้เนื้อหาง่ายที่สุดในการทำความเข้าใจโดยการแบ่งเป็นข้อ ๆ ตามลำดับขั้นตอนดังนี้ 6.2.1 ออกแบบยูสเซอร์อินเทอร์เฟซ(User Interface) โดยแบ่งโปรแกรมออกเป็นส่วน ๆ ตามยูสเคส ในแต่ละยูสเคสก็นำมาแบ่งเป็นส่วนย่อย ๆ อีกตามซีเควนซ์ไดอะแกรม โดยให้แต่ละส่วนเป็นพาเนล(Panel)

6.2.2 หลังที่ทำการแบ่งเป็นพาเนลได้แล้วนั้น แต่ละพาเนลก็หมายถึงหนึ่งคลาสนั่นเอง เราจึงสามารถแยกแต่ละพาเนลไปเขียนโปรแกรม อย่างอิสระ แล้วจึงค่อยนำมารวมกันในภายหลัง

6.2.3 ในการเขียนโปรแกรมติดต่อกับฐานข้อมูลนั้นเราใช้ JDBC เป็นตัวเชื่อมต่อให้โดยมีการระบุถึงเซิร์ฟเวอร์ฐานข้อมูลด้วยหมายเลขไอพี(IP Address)

6.2.4 หลังจากขั้นตอนนี้เราสามารถแยกแต่ละส่วนมาเขียนโปรแกรมได้ ซึ่งแยกไปตามคลาสต่าง ๆ ซึ่งถ้าพิจารณาในส่วนของยูเอ็มแอลแล้วจะเห็นว่าจะมีคลาสอยู่ 3 ชนิดด้วยกัน ซึ่งจะไม่กล่าวถึงรายละเอียดส่วนนี้อีก คลาสแต่ละชนิดบางชนิดอาจเขียนขึ้นมาใหม่ บางชนิดการสืบทอดมาจากคลาสพื้นฐานที่ภาษาจาวามีให้เช่น คลาสที่เป็นอินเทอร์เฟซนั้นจะสืบทอดมาจาก javax.swing.JPanel เป็นหลัก และคลาสในส่วนของการเก็บข้อมูลสืบทอดมาจาก javax.swing.tree.DefaultMutableTreeNode ซึ่งทำให้มีความสามารถในการเป็นโนด (Node) ในทรีที่เราใช้ javax.swing.Jtree ด้วย

try {

```

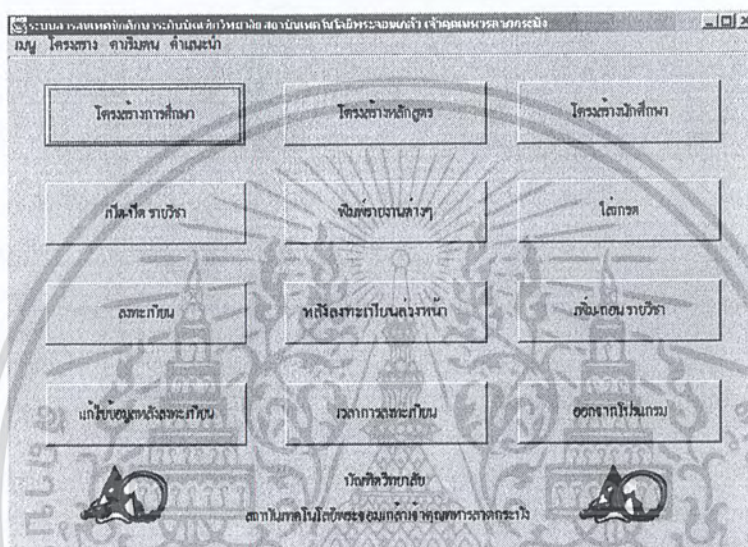
    Connection dbconn;
    String msgout = "";
    String fname = null, lname = null;
    try {
        String userid = "Administrator";
        String passwd = "maspong";
        String url = "jdbc:db2://192.168.0.13/Graduate";
        Class.forName("COM.ibm.db2.jdbc.net.DB2Driver").newInstance();
        dbconn = DriverManager.getConnection(url, userid, passwd);
        statement = dbconn.createStatement();
        treeModel = getMyTreeModel();
    } catch (ClassNotFoundException cnfex) {
        cnfex.printStackTrace();
        msgout = "Connection unsuccessful\n" + cnfex.toString();
    } catch (SQLException sqlex) {
        sqlex.printStackTrace();
        msgout = "Connection unsuccessful\n" + sqlex.toString();
        out.println(sqlex);
    } catch (Exception excp) {
        excp.printStackTrace();
        msgout = excp.toString();
    }
}

```

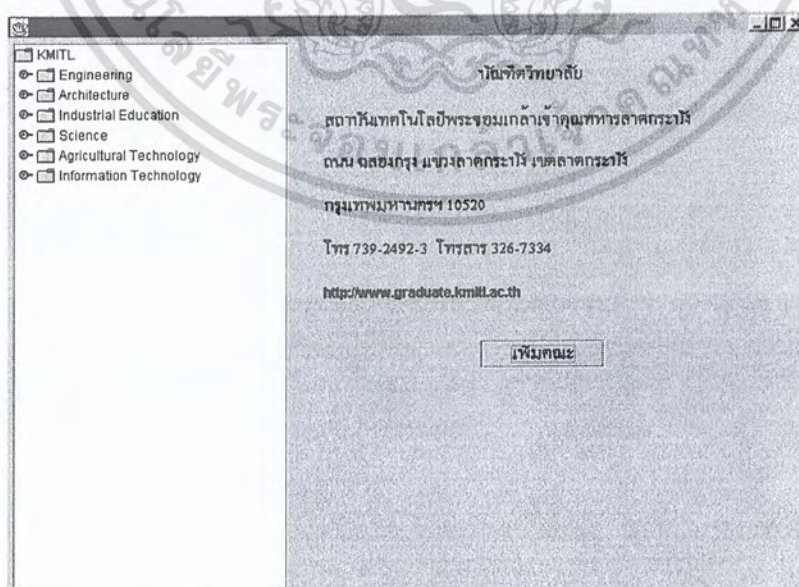
ตัวอย่างโค้ดการติดต่อฐานข้อมูลด้วย JDBC

6.2.5 สิ่งหนึ่งที่ต้องคำนึงในการเขียนโปรแกรมก็คือข้อกำหนดต่าง ๆ ที่ได้ทำการศึกษาจากระบบจริง ซึ่งเราต้องเขียนโปรแกรมให้มีการตรวจสอบตรงตามข้อกำหนดเหล่านั้นด้วย ยกตัวอย่างเช่นนักศึกษาในระดับบัณฑิตศึกษานั้นสามารถเข้ามานักศึกษาใหม่ได้ทั้งในภาคเรียนที่ 1 และภาคเรียนที่ 2 ดังนั้นการจัดแบ่งกลุ่มนักศึกษาจึงต้องแบ่งตามภาคและปีที่เข้า ไม่สามารถแบ่งตามรหัสนักศึกษาแต่เพียงอย่างเดียวได้เหมือนในระดับปริญญาตรี

6.2.6 เมื่อเราเขียนโปรแกรมแต่ละส่วนเสร็จก็จะนำมาประกอบในส่วนของพาแนลหลักซึ่งอยู่ในส่วนของ javax.swing.JFrame อีกที



รูปที่ 6-3 รูปส่วนอินเทอร์เฟซของ MainPanel



รูปที่ 6-4 รูปส่วนอินเทอร์เฟซของ AcademicPanel

ระบบสารสนเทศศึกษาระดับบัณฑิตศึกษา มหาวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เมนู โครงสร้าง คาเรียนต ค้นหา

KMITL

- Engineering
 - Electronic Engineering
 - Electrical Engineering
 - Computer Engineering
 - Control Engineer
- Architecture
 - Plan
 - Industrial Art
- Industrial Education
- Science
- Agricultural Technology
- Information Technology

รหัสคณะ 01

ชื่อคณะภาษาไทย วิศวกรรมศาสตร์

ชื่อคณะภาษาอังกฤษ Engineering

ค่านำหน้าชื่อคณะภาษาอังกฤษ Assistant Professor Doctor

ชื่อคณะภาษาอังกฤษ Tawin

นามสกุลคณะภาษาอังกฤษ Puengma

ค่านำหน้าชื่อคณะภาษาไทย ผศ. ดร.

ชื่อคณะภาษาไทย กวิด

นามสกุลคณะภาษาไทย พิงมา

เพิ่มภาควิชา แก้ไขคณะ ตัดคณะ

รูปที่ 6-5 รูปส่วนอินเทอร์เฟซของ AcademicPanel ที่ประกอบอยู่บน MainPanel

จากขั้นตอนข้างต้นอาจไม่สามารถอธิบายลงในรายละเอียดทั้งหมดได้ซึ่งสามารถดูในรายละเอียดจากซอสโค้ด(Source Code)

6.3 การทดสอบโปรแกรมในส่วนของแอปพลิเคชัน

การทดสอบนี้จะทดสอบในกรณีของโครงสร้างของหน่วยงานการศึกษา ดังขั้นตอนต่อไปนี้

1. ล็อกอินเข้าสู่ระบบ ดังรูป โดย UserName และ Password ได้จากตาราง Registrar ในฐานข้อมูลของ DB2

ระบบสารสนเทศศึกษาระดับบัณฑิตศึกษา มหาวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เมนู โครงสร้าง คาเรียนต ค้นหา

ล็อกอินเข้าสู่ระบบสารสนเทศศึกษาระดับบัณฑิตศึกษา มหาวิทยาลัยสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

กรุณาใส่ Username และ Password

Username isupanee

Password *****

ตกลง ยกเลิก

© 2012 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ตึก E-12 ถนนวิภาวดีรังสิต กรุงเทพมหานคร 10160

รูปที่ 6-6 แสดงหน้าจอการล็อกอิน

Command Center

Command Center Query Results Edit Tools Help

Interactive Script Query Results Access Plan

Perform required changes and then click the commit update button.

REGID	FACID	ENAME	TNAME	POSITION	USERNAME	PASSWORD	PERMISS...
0001	01	-	-	-	supanee	12345678	1

☐ Automatically commit updates

รูปที่ 6-7 รูปแสดงข้อมูลที่อยู่ในฐานข้อมูล

2. เลือกเมนูข้อมูลหน่วยงานการศึกษา

ระบบสารสนเทศศึกษาระดับบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เมนู แก้ไขข้อมูล ค้นหา ค้นหา

ข้อมูลหน่วยงานการศึกษา ข้อมูลหลักสูตร ข้อมูลนักศึกษา

เปิด-ปิด รายวิชา รายวิชา รายวิชา

ลงทะเบียน ตั้งค่าเวลา รายวิชาหลังลงทะเบียน

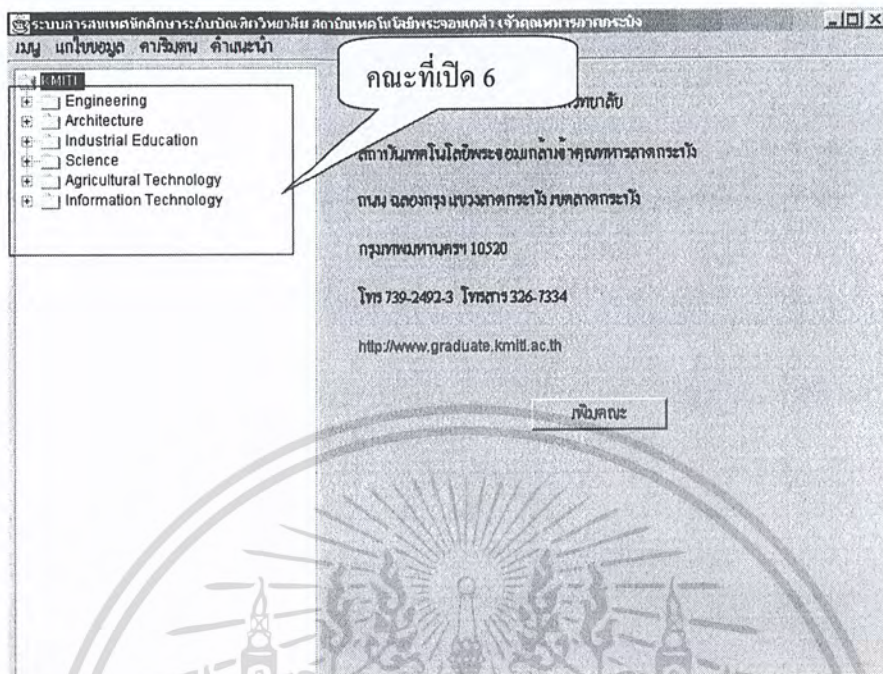
เพิ่มผลจ.รายวิชา คัดกรอง ออกจากโปรแกรม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

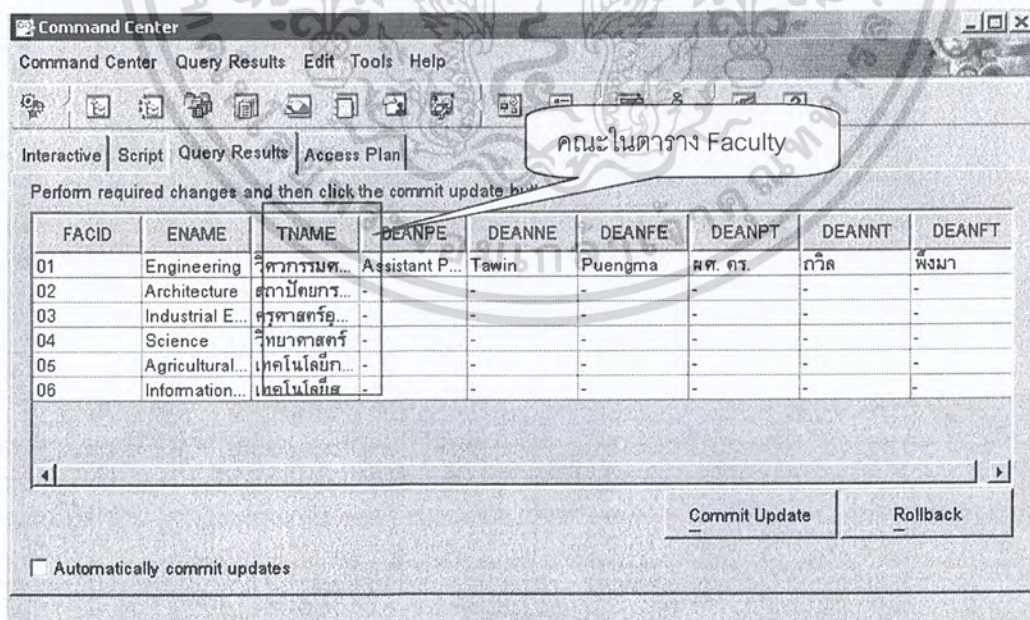
รูปที่ 6-8 แสดงการเลือก menu หลัก

3. จะได้หน้าต่างของโครงสร้างของข้อมูลหน่วยงานการศึกษา ลักษณะของทรีโมเดลดังรูป



รูปที่ 6-9 แสดงข้อมูลของคณะ

3. ลองเทียบกับข้อมูลในตาราง faculty ใน Command Center ของ db2 จะพบว่า มี 5 คณะเท่ากันดังรูป



รูปที่ 6-10 แสดงข้อมูลในตาราง Faculty

4. ลองดูรายละเอียดของคณะ Engineering และเทียบกับในฐานข้อมูลได้

ระบบสารสนเทศอิเล็กทรอนิกส์กับบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เมนู แก้ไขข้อมูล ค้นหา ค้นหา

KMITL

- Engineering
 - Electronic Engineering
 - Electrical Engineering
 - Computer Engineering
 - Control Engineer
 - Mechanical Engineering
- Architecture
- Industrial Education
- Science
- Agricultural Technology
- Information Technology

รหัสคณะ: 01

ชื่อคณะภาษาไทย: วิศวกรรมศาสตร์

ชื่อคณะภาษาอังกฤษ: Engineering

ค่านำหน้าชื่อคณะภาษาอังกฤษ: Assistant Professor Doctor

ชื่อคณะภาษาอังกฤษ: Tawin

นามสกุลคณะภาษาอังกฤษ: Puengma

ค่านำหน้าชื่อคณะภาษาไทย: ผศ. ดร.

ชื่อคณะภาษาไทย: ทวี

นามสกุลคณะภาษาไทย: พึ่งมา

เพิ่มภาควิชา แก้ไขคณะ ตัดคณะ

รูปที่ 6-11 แสดงข้อมูลเมื่อเลือกคณะ

5. สามารถแก้ไขข้อมูลของคณะ Engineering ได้ โดยคลิกที่ปุ่มแก้ไขคณะ แล้วกดปุ่มตกลงดังรูปที่ และจะได้ดังรูปที่ แล้วลองเทียบกับในฐานข้อมูล จะพบว่ามีการอัปเดตเรียบร้อยแล้ว

ระบบสารสนเทศอิเล็กทรอนิกส์กับบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เมนู แก้ไขข้อมูล ค้นหา ค้นหา

KMITL

- Engineering
 - Electronic Engineering
 - Electrical Engineering
 - Computer Engineering
 - Control Engineer
 - Mechanical Engineering
- Architecture
- Industrial Education
- Science
- Agricultural Technology
- Information Technology

รหัสคณะ: 01

ชื่อคณะภาษาไทย: วิศวกรรมศาสตร์

ชื่อคณะภาษาอังกฤษ: Engineering Faculty

ค่านำหน้าชื่อคณะภาษาอังกฤษ: Assistant Professor Doctor

ชื่อคณะภาษาอังกฤษ: Tawin

นามสกุลคณะภาษาอังกฤษ: Puengma

ค่านำหน้าชื่อคณะภาษาไทย: ผศ. ดร.

ชื่อคณะภาษาไทย: ทวี

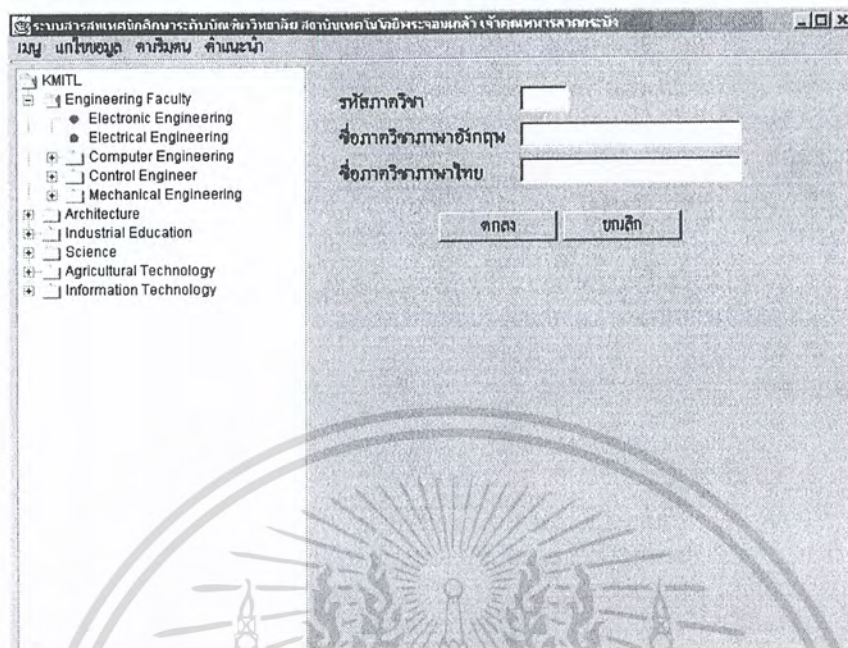
นามสกุลคณะภาษาไทย: พึ่งมา

ตกลง ยกเลิก

ส่วนที่แก้ไข

รูปที่ 6-12 แสดงการแก้ไขข้อมูลของคณะ

6. ทำการทดสอบการเพิ่มภาควิชาในคณะ Engineering โดยกดปุ่มเพิ่มภาควิชา จะได้ช่องใส่ข้อมูลดังรูปที่



ระบบสารสนเทศศึกษาระดับบัณฑิตศึกษา มหาวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เมนู แก้ไขข้อมูล ค้นหา ค้นหา

KMITL

- Engineering Faculty
 - Electronic Engineering
 - Electrical Engineering
 - Computer Engineering
 - Control Engineer
 - Mechanical Engineering
- Architecture
- Industrial Education
- Science
- Agricultural Technology
- Information Technology

รหัสภาควิชา

ชื่อภาควิชาภาษาอังกฤษ

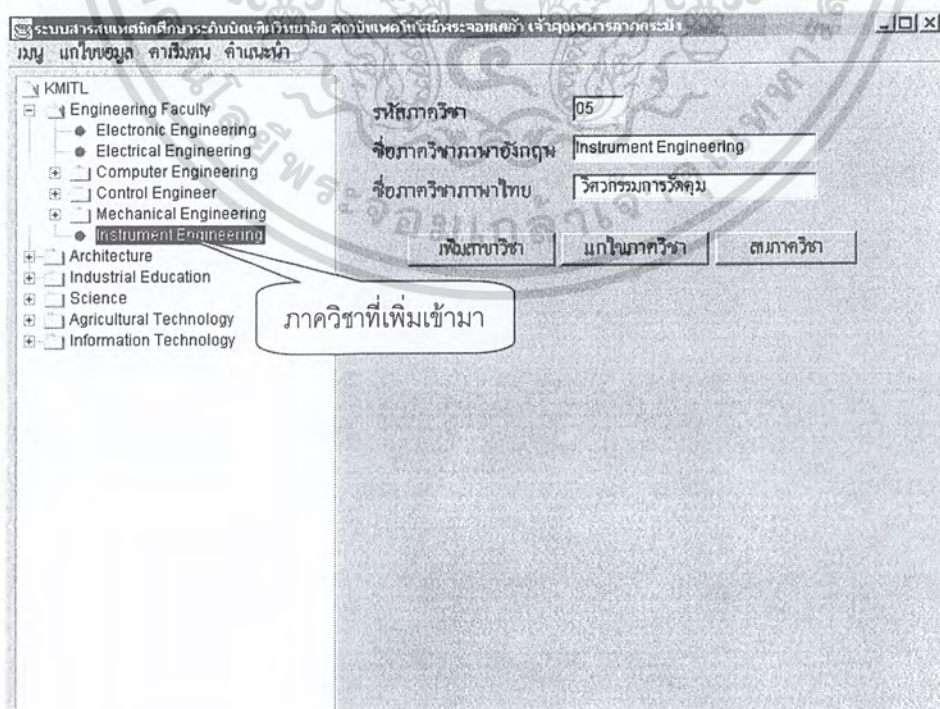
ชื่อภาควิชาภาษาไทย

ตกลง ยกเลิก

รูปที่ 6-13 แสดงการเพิ่มภาควิชา

7. ทำการใส่ข้อมูลลงไป แล้วกดปุ่มตกลง

8. ทดสอบดูกับในฐานข้อมูลของตาราง Department จะได้ดังรูป



ระบบสารสนเทศศึกษาระดับบัณฑิตศึกษา มหาวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เมนู แก้ไขข้อมูล ค้นหา ค้นหา

KMITL

- Engineering Faculty
 - Electronic Engineering
 - Electrical Engineering
 - Computer Engineering
 - Control Engineer
 - Mechanical Engineering
 - Instrument Engineering
- Architecture
- Industrial Education
- Science
- Agricultural Technology
- Information Technology

รหัสภาควิชา 05

ชื่อภาควิชาภาษาอังกฤษ Instrument Engineering

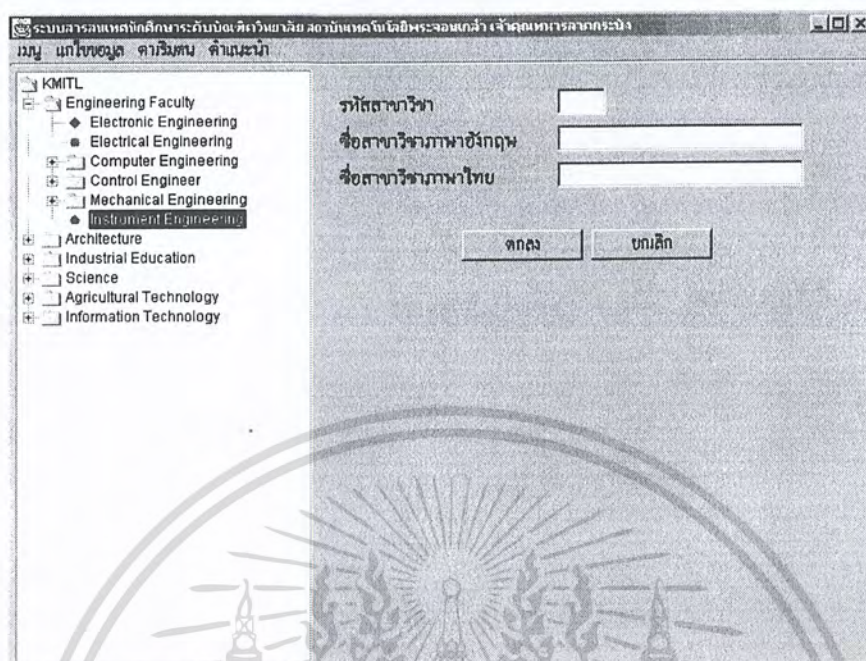
ชื่อภาควิชาภาษาไทย วิศวกรรมวิศกุ่ม

เพิ่มสาขาวิชา แก้ไขภาควิชา ลบภาควิชา

ภาควิชาที่เพิ่มเข้ามา

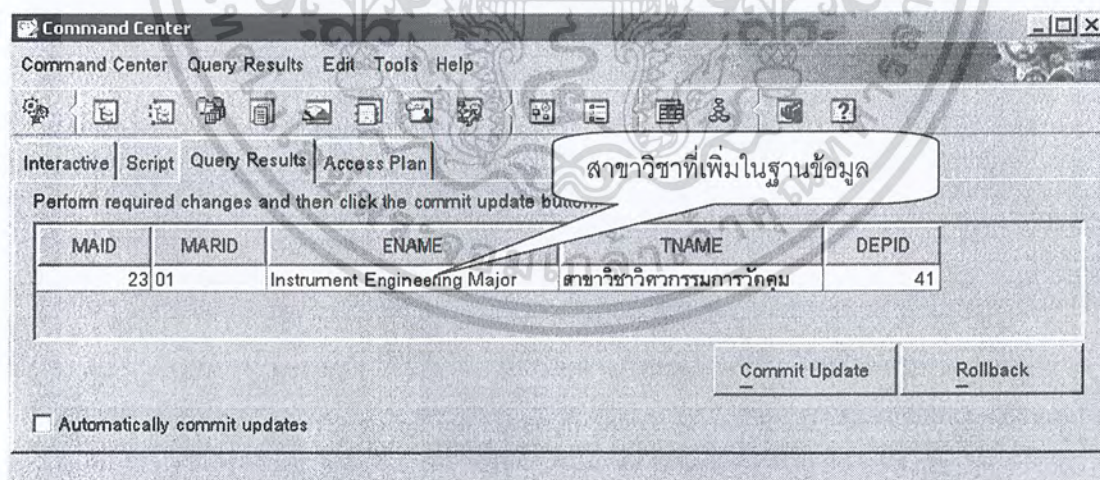
รูปที่ 6-13 แสดงผลการเพิ่มภาควิชา

9. ทดสอบการเพิ่มสาขาวิชาในภาควิชาวิศวกรรมการวัดคุม (Instrument Engineering) ดังรูป



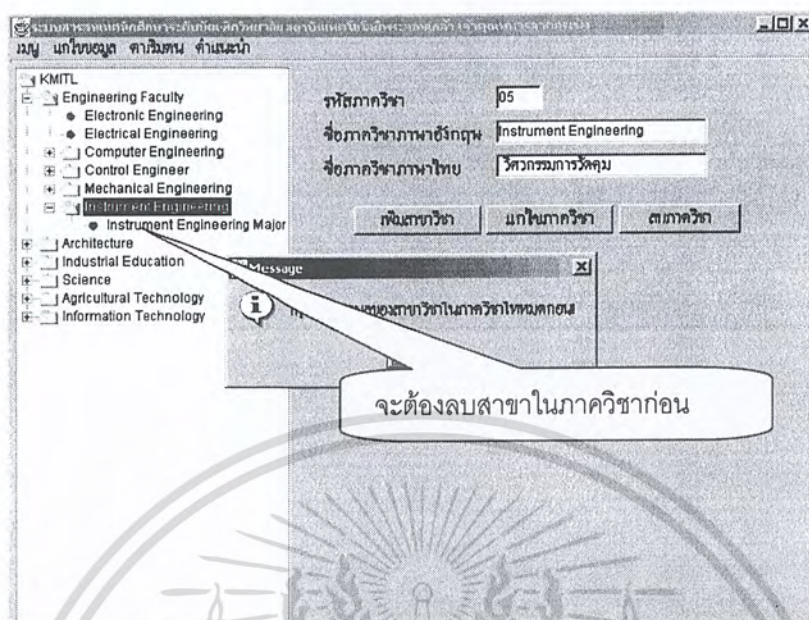
รูปที่ 6-14 แสดงการเพิ่มสาขาวิชา

9. ใส่ข้อมูลสาขาวิชาที่ต้องการลงไป แล้วคลิกปุ่มตกลง แล้วลองเทียบกับตาราง Major ในฐานข้อมูลดู



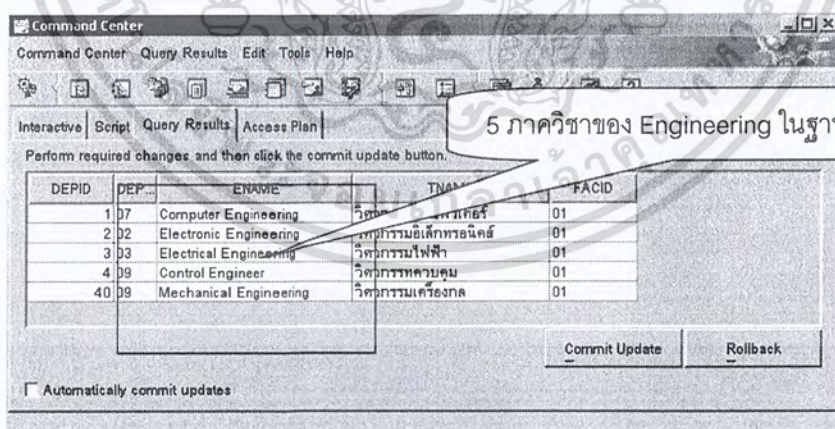
รูปที่ 6-15 แสดงข้อมูลหลังจากการเพิ่มสาขาวิชา

10. ลองทดสอบการลบภาควิชา Instrument Engineering โดยการคลิกปุ่มลบภาควิชา จำได้ดังรูป



รูปที่ 6-16 แสดงการลบภาควิชา

11. จากรูปจะพบว่าไม่สามารถลบได้ จะต้องลบข้อมูลสาขาวิชา ของภาควิชา Instrument Engineering ก่อน
12. หลังจากนั้นจึงจะลบภาควิชา Instrument Engineering ได้โดยคลิกปุ่มลบภาควิชา



รูปที่ 6-17 แสดงข้อมูลในตาราง Department

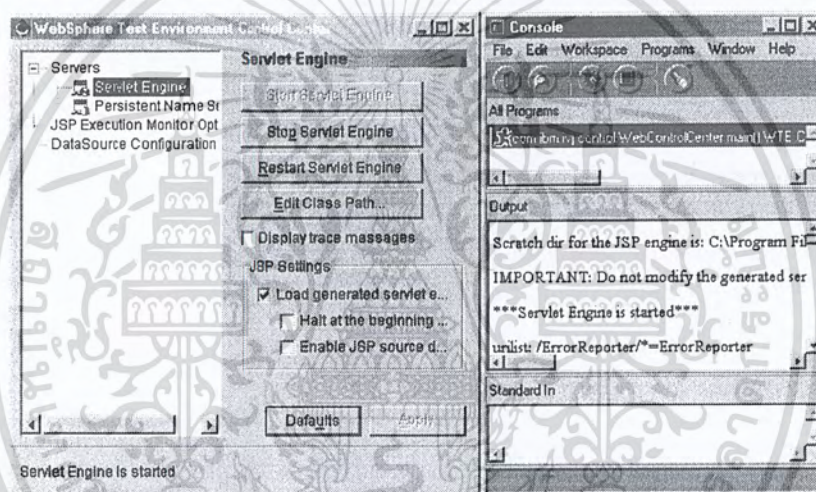
จะเห็นว่าการทดสอบ Application Program ในส่วนของข้อมูลหน่วยงานการศึกษา เป็นไปได้อย่างถูกต้องตามกระบวนการที่ได้ออกแบบไว้ ในส่วนอื่นๆก็สามารถทดสอบได้ในกรณีเดียวกัน ซึ่งยกตัวอย่างไว้เพียงส่วนของข้อมูลการศึกษาเท่านั้น

6.4 การเขียนโปรแกรมในส่วนของเว็บแอปพลิเคชัน

การเขียนโปรแกรมในส่วนนี้จะแตกต่างจากการเขียนโปรแกรมในส่วนแอปพลิเคชันเพราะว่า การแสดงผลในส่วนนี้จะอยู่บนเว็บเบราว์เซอร์(Web Brower) ดังนั้นการเขียนโปรแกรมแบบนี้จึงอยู่บนสิ่งแวดล้อมอินเทอร์เน็ตโดยใช้เทคโนโลยีของ Servlet, JSP และ JavaBean ซึ่งได้กล่าวถึงทฤษฎีทั้งสองเรื่องนี้แล้วในบทที่ 3 แล้ว

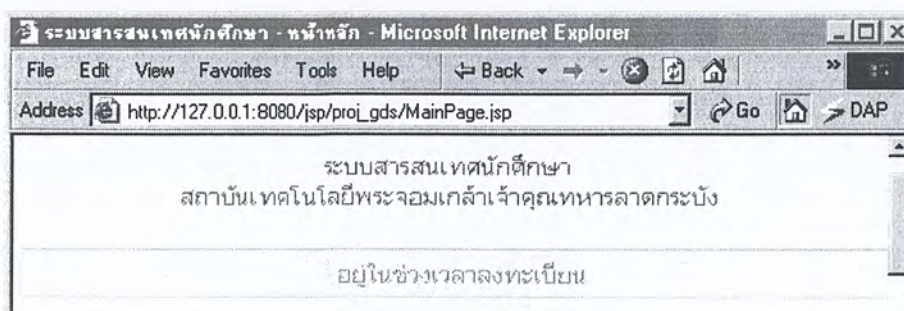
6.4.1 เพื่อให้ง่ายและเป็นการสะดวกในการเขียนโปรแกรมในส่วนนี้เราต้องจำลองสภาวะแวดล้อมให้สามารถทำงานเหมือนกับอินเทอร์เน็ตนั่นก็คือ จำลองเว็บเซิร์ฟเวอร์ขึ้นมา โดยใช้ความสามารถของ Java VisualAge ซึ่งเราต้องไปทำการตั้งค่าของ Websphere Test Environment ให้ทำงานได้

6.4.2 เราสามารถเขียนโปรแกรมของเราทั้งในส่วนคลาสที่เป็น Servlet, JSP และ JavaBean ไว้ในไฟล์เดอร์(Folder) ที่ Websphere Test Environment สร้างไว้ให้ เราก็จะสามารถทดลองโปรแกรมที่เราเขียนได้โดยผ่านทางเว็บเบราว์เซอร์บนเครื่องเราเอง



รูปที่ 6-18 การทำงานของ Websphere Test Environment

6.4.3 ขั้นตอนต่อไปก็คือการออกแบบยูสเซอร์อินเทอร์เฟซในแต่ละส่วน แล้วก็สามารถนำไปเขียนโปรแกรม โดยแยกจากกันเพื่อทดลองทีละส่วนได้



รูปที่ 6-19 การทดลองใช้งานผ่านเว็บเบราว์เซอร์

6.4.4 ในขั้นตอนนี้เราสามารถเขียนคลาสต่าง ๆ ขึ้นมาใช้งานได้ตามปกติ การติดต่อกับฐานข้อมูลก็ทำคล้ายกับการติดต่อในส่วนแอปพลิเคชัน แต่คลาสที่เราสร้างขึ้นเพื่อนำไปใช้ควรสร้างให้เป็น JavaBean เพื่อจะมีความสามารถสูงในการนำไปใช้ใน Servlet และ JSP

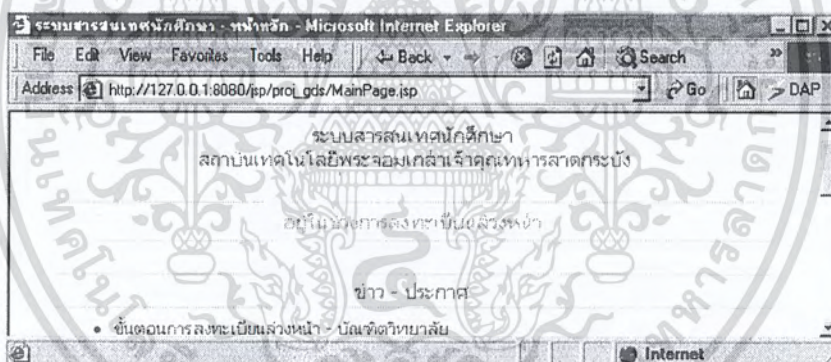
6.4.5 นอกจากใช้ IBM Java VisualAge แล้วในการเขียน JSP ซึ่งเป็นการทำงานแบบ Server Side Included อย่างหนึ่งจำเป็นต้องสร้างโครงร่างของหน้า html ขึ้นมาก่อนซึ่งในที่นี้เราใช้ Macromedia Dreamweaver 4 เข้ามาช่วยเสริมการทำงานในส่วนนี้ด้วย

6.4.5 นำส่วนต่าง ๆ มาประกอบกันให้สามารถทำงานเชื่อมต่อกันได้ด้วยการกำหนดลิงก์(Link) ให้เชื่อมโยงถึงกัน

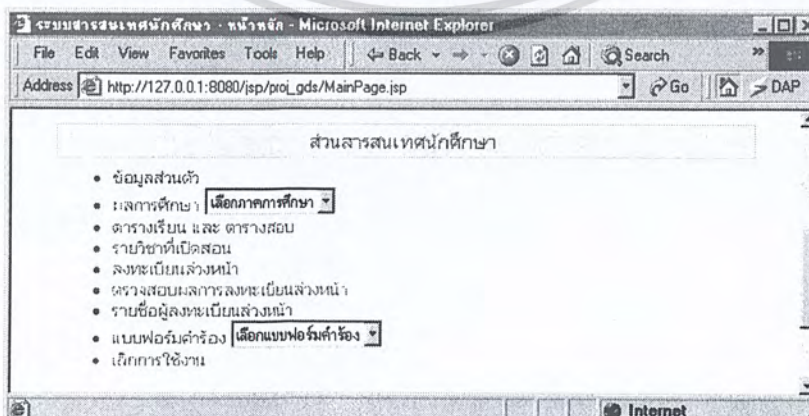
6.5 การทดสอบโปรแกรมในส่วนของเว็บแอปพลิเคชัน

การทดสอบในส่วนของเว็บแอปพลิเคชันนั้น เพื่อให้เข้าใจในการทำงานของระบบ ในที่นี้จะยกตัวอย่างการลงทะเบียนล่วงหน้าของนักศึกษา

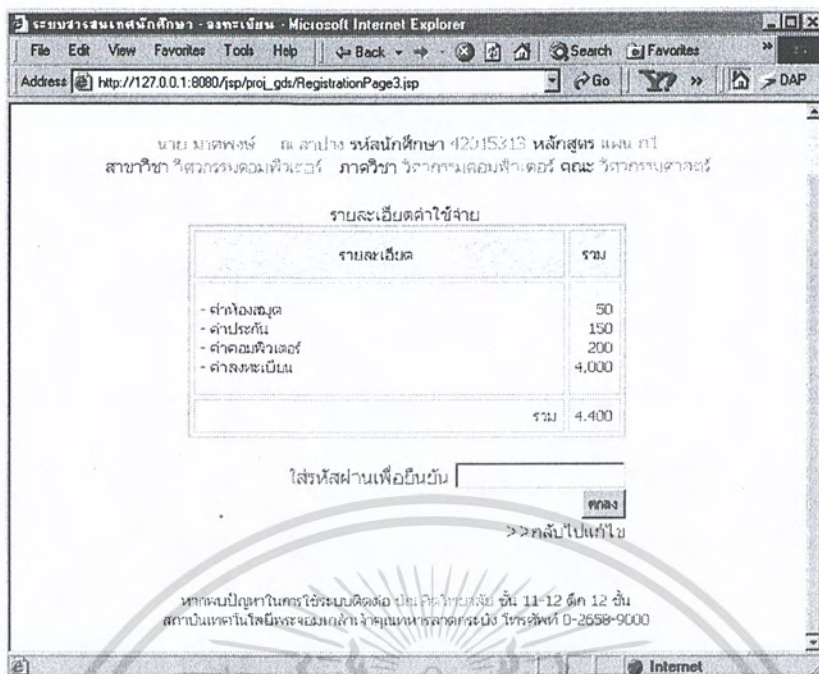
6.5.1 การลงทะเบียนล่วงหน้าเป็น 1 ในเมนูที่มีการเปลี่ยนแปลงตามเวลาปัจจุบัน กล่าวคือ หากวันที่นักศึกษามีการล็อกอินเข้ามาอยู่ในช่วงของการลงทะเบียนล่วงหน้า เมนูการลงทะเบียนล่วงหน้าก็จะสามารถทำงานได้(Active) ซึ่งจะสามารถตรวจสอบช่วงเวลาได้จาก แถบแสดงเวลา



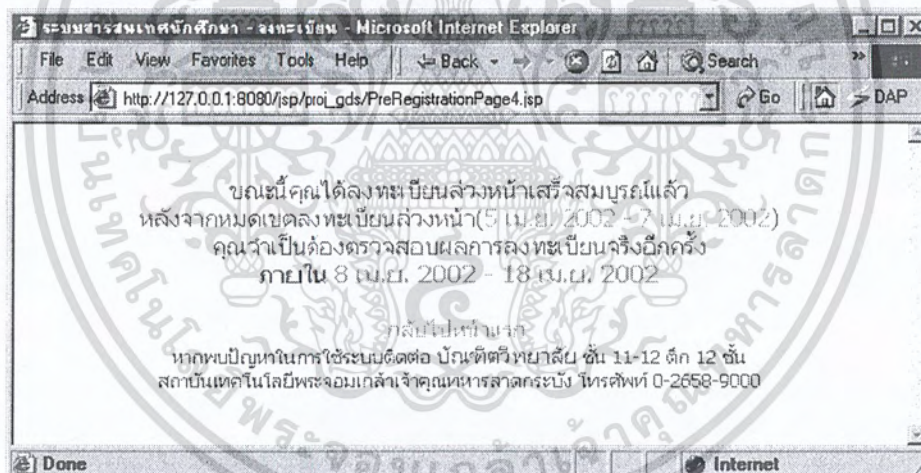
รูปที่ 6-20 แสดงช่วงเวลาปัจจุบัน



รูปที่ 6-21 แสดงเมนูที่เปลี่ยนแปลงตามช่วงเวลา



รูปที่ 6-24 แสดงค่าใช้จ่ายตามหลักสูตร



รูปที่ 6-25 แสดงข้อความการลงทะเบียนสมบูรณ์

บทที่ 7

บทสรุปและบทวิจารณ์

7.1 บทสรุป

การพัฒนากระบวนสารสนเทศนักศึกษา ระดับบัณฑิตศึกษาด้วยวิธีเชิงวัตถุนับว่ามีได้รับความสำเร็จอย่างยิ่งในแง่ของการวิเคราะห์และออกแบบ เพราะในระบบนี้การวิเคราะห์และออกแบบได้ทำอย่างรอบคอบที่สุด จึงทำให้ได้ระบบที่อำนวยความสะดวกและครอบคลุมประโยชน์การใช้สอยมากในระดับที่น่าพอใจ แต่ในเรื่องของการสร้างหรือการเขียน โปรแกรมนั้นยังสามารถดำเนินการได้ครบทุกส่วน อันเนื่องมาจากในโครงการนี้จะเน้นไปที่การวิเคราะห์และออกแบบเป็นหลัก โดยส่วนที่เหลือก็จะมีการพัฒนาต่อไป

ส่วนเครื่องมือที่เราใช้ในการทำงานนั้นคือ Case tool ต่าง ๆ ซึ่งได้แก่

- โปรแกรม Rational Rose 2000 เป็น CASE Tool ที่สามารถทำให้เราสร้างโมเดลของระบบโดยใช้ UML ได้อย่างสะดวก
- โปรแกรม IBM VisualAge for Java 4.0 เป็นเครื่องมือที่ใช้ในการเขียนภาษาจาวาได้เป็นอย่างดี ด้วยความสามารถที่มีให้มาไม่ว่าจะเป็น การเขียนโปรแกรมแบบแอปพลิเคชัน หรือใช้ Servlet ,JSP ก็ตาม
- โปรแกรมฐานข้อมูล DB2 จัดเป็น DBMS ที่มีความสามารถในการสร้างและจัดการฐานข้อมูลได้ดี แต่ในที่นี้ยังไม่ได้มีการทดสอบเรื่องของประสิทธิภาพการทำงาน

ส่วนในเรื่องของการศึกษาและทำความเข้าใจระบบทะเบียนนักศึกษา ระดับบัณฑิตศึกษานั้น ได้ทำการวิเคราะห์หาจุดอ่อนต่าง ๆ ซึ่งได้กล่าวมาแล้วข้างต้น รวมทั้งยังสามารถนำไปสร้างเป็นโมเดลของระบบที่พยายามสร้างความสะดวกให้มากที่สุดอีกด้วย

ส่วนในการเขียนโปรแกรมนั้นได้มีการทำงานไปประมาณ 60 เปอร์เซ็นต์ ซึ่งจะขอชี้แจงรายละเอียดดังนี้

- ส่วนของแอปพลิเคชันได้พัฒนาในส่วนของการเพิ่ม แก้ไขข้อมูลหลักสูตร นักศึกษา หน่วยงานการศึกษา ระยะเวลาการลงทะเบียน
- ในส่วนของเว็บแอปพลิเคชันได้พัฒนารอบทุกฟังก์ชันแต่อาจมีบางฟังก์ชันที่ต้องพัฒนาเพิ่มเติมเล็กน้อย

จากสิ่งที่ได้กล่าวมาจะเห็นได้ว่าการศึกษาวิจัยนี้ได้ทำครอบคลุมตามวัตถุประสงค์ทั้ง 4 ข้อที่ได้ตั้งไว้ข้างต้นแล้ว

7.2 บทวิจารณ์

หลังจากที่ได้ลงมือทำโครงการวิจัยนี้ สิ่งที่คุณวิจัยได้ทำการศึกษาและค้นคว้า รวมถึงทำการวิเคราะห์ทำให้สามารถทราบได้ถึงข้อเท็จจริงต่าง ๆ ที่ส่งผลให้การทำงาน ซึ่งถ้าไม่ลงมือเข้าไปวิเคราะห์และศึกษาอย่างจริงจังอาจเป็นการยากที่จะทราบถึงข้อเท็จจริงเหล่านั้นได้ ผู้วิจัยจึงจะขอกล่าวถึงข้อเท็จจริงเหล่านั้นเพื่อให้ผู้อ่านสามารถใช้ประโยชน์ในปริญญานิพนธ์ และระบบที่พัฒนาขึ้นนี้ได้อย่างเต็มที่

7.3 ปัญหาที่พบในการดำเนินงาน

7.3.1 ด้วยกฎระเบียบข้อบังคับต่าง ๆ ที่มีมากมายทำให้เกิดความยุ่งยากและสับสนอีกทั้งยังมีความแตกต่างกันในแต่ละคณะซึ่งทำให้ยังเป็นการเพิ่มรายละเอียดในการวิเคราะห์เป็นอย่างยิ่ง ยกตัวอย่างเรื่องของการที่อาจารย์ที่ปรึกษาต้องทำการตรวจสอบการลงทะเบียนของนักศึกษาซึ่งทำให้นักศึกษาต้องเข้าพบอาจารย์ที่ปรึกษา ทั้ง ๆ ในความเป็นจริงอาจารย์ที่ปรึกษานั้นไม่ได้ตรวจสอบอย่างจริงจัง

7.3.2 ผู้ปฏิบัติงานไม่ทราบขั้นตอนการปฏิบัติงานอย่างชัดเจน ทำให้ต้องมีการปรับแก้ไขเงื่อนไขข้อมูลที่ได้มาบางอย่าง นั่นหมายถึงการปรับแก้เรื่องของความต้องการของผู้ใช้ ซึ่งส่งผลต่อระบบที่ได้วิเคราะห์ไว้แล้วต้องทำการวิเคราะห์เปลี่ยนแปลงอีกครั้ง

7.3.3 เครื่องมือที่ใช้ ในการเขียนโปรแกรมไม่ว่าจะเป็น Rational Rose หรือ IBM VisualAge for Java ล้วนต้องการทรัพยากรของเครื่องคอมพิวเตอร์ที่ใช้สูงทำให้การทำงานเป็นไปอย่างเชื่องช้า และเครื่องเกิดหยุดทำงานได้ง่าย

7.3.4 เครื่องมือที่ใช้ในการเขียน โปรแกรมที่เราใช้ นั่นก็คือ IBM VisualAge for Java นั้น ถึงแม้ว่าจะสามารถทำการตรวจสอบการทำงานของโปรแกรม(Debug)ได้ แต่ก็อาจจะไม่สมบูรณ์มากนักจึงยากในการหาข้อผิดพลาดที่เกิดขึ้น

7.3.5 เซิร์ฟเวอร์ที่จะใช้งานจริงไม่มีเสถียรภาพเพียงพอ เนื่องด้วยปัญหาทางด้านเครือข่าย และตัวเซิร์ฟเวอร์เองจึงทำให้การทำงานไม่สามารถทดลองได้จริง และต้องเสียเวลากับเรื่องดังกล่าวไปมาก

7.4 ข้อเสนอแนะสำหรับระบบทะเบียนนักศึกษา ของบัณฑิตวิทยาลัย

7.4.1 งานในระบบทะเบียนนักศึกษาของบัณฑิตวิทยาลัย นั้นมีความซับซ้อนและมีรายละเอียดเฉพาะจึงเป็นการถูกต้องแล้วที่แยกออกมาทำงานเฉพาะจากงานทะเบียนของนักศึกษาระดับปริญญาตรี แต่สิ่งที่ควรที่จะต้องทำการปรับปรุงนั้นคือ ควรมีการกระจายงานของฝ่ายทะเบียนของบัณฑิตวิทยาลัยส่วนกลางไปยังส่วนคณะ อย่างเช่นการกรอกรายเรียนตารางสอน การกรอกรายวิชาซึ่งจะทำให้เกิดความรวดเร็ว และคล่องตัวมากกว่า ทั้งยังสามารถจัดการกับข้อผิดพลาดได้โดยตรง เพราะในปัจจุบันมีข้อผิดพลาดบางอย่างเกิดขึ้น อย่างเช่นตารางเรียนผิด ไม่สามารถหาผู้รับผิดชอบได้ เพราะ มีการส่งต่องานจากส่วนคณะเข้ามาในส่วนกลางด้วย

7.4.2 ในเรื่องของระบบสารสนเทศนั้น โดยปกติแล้วสำนักวิจัยและบริการคอมพิวเตอร์จะเป็นผู้ให้การสนับสนุนแกบัณฑิตวิทยาลัย แต่ด้วยความแตกต่างของระบบปริญญาตรี และระดับบัณฑิตศึกษาทำให้เกิดปัญหาหลายอย่างขึ้นในส่วนของบัณฑิตวิทยาลัย ซึ่งการแก้ปัญหาจากสำนักวิจัยและบริการคอมพิวเตอร์ทำได้ล่าช้า

ทางบัณฑิตวิทยาลัยจึงคิดที่จะสร้างระบบสารสนเทศของตัวเองขึ้นมา ซึ่งในมุมมองของผู้ทำการวิจัยนี้เห็นว่างานระบบสารสนเทศนั้นทั้งในด้านของโปรแกรม และฐานข้อมูลยังควรให้สำนักวิจัยรับผิดชอบ แต่ทางสำนักวิจัยควรจัดกำลังคน หรือใช้กำลังคนของทางบัณฑิตวิทยาลัย ขึ้นมาเฉพาะเพื่อดูแลระบบในส่วนนี้ ทั้งนี้เพื่อให้สามารถใช้ทรัพยากรของสำนักวิจัยที่มีอยู่แล้วได้อย่างเต็มที่ ส่วนบุคลากรที่จัดไว้ดูแลเฉพาะนี้ก็จะสามารถจัดการกับปัญหาต่าง ๆ หรือเพิ่มแก้ไขได้ในทันที ซึ่งปัญหาเรื่องของภาษาโปรแกรมที่ใช้ไม่ได้เป็นปัญหาอยู่แล้ว เพราะถึงอย่างไรก็สามารถใช้ฐานข้อมูลร่วมกันได้

7.4.3 ควรมีการประชุมหาหรือเพื่อปรับแก้ข้อบกพร่องต่าง ๆ หาข้อดีข้อเสียของเงื่อนไขแต่ละข้ออย่างชัดเจน รวมทั้งให้แต่ละคณะมีกฎระเบียบไปในแนวทางเดียวกัน หรือถ้าไม่สามารถทำให้กฎระเบียบเหมือนกันได้ ก็ควรระบุถึงข้อแตกต่างนั้นอย่างชัดเจนให้เจ้าหน้าที่ผู้ปฏิบัติงานได้ทราบ นอกจากกฎระเบียบแล้วยังมีแนวทางต่าง ๆ เช่น การเขียนหลักสูตรควรเขียนให้มีความเป็นมาตรฐานมีรายละเอียดครบถ้วน อย่างเช่นทุกหลักสูตรควรให้มีการกำหนดแผนการศึกษา(Course Work) การกำหนดรหัสวิชาที่ไม่ควรซ้ำกัน ควรใช้รหัสวิชาที่เป็นตัวเลขทั้งหมด

7.4.4 นอกจากเรื่องของกฎระเบียบแล้ว การปฏิบัติตามก็ควรทำอย่างเคร่งครัดด้วย อย่างเช่นอาจารย์ที่ทำการตรวจสอบเรื่องของการลงทะเบียนของนักศึกษาที่ควรตรวจสอบอย่างจริงจัง การตรวจสอบในเรื่องของวิชาที่ต้องลงก่อนล่วงหน้าเป็นต้น

7.5 ข้อเสนอแนะในการนำไปพัฒนาต่อ

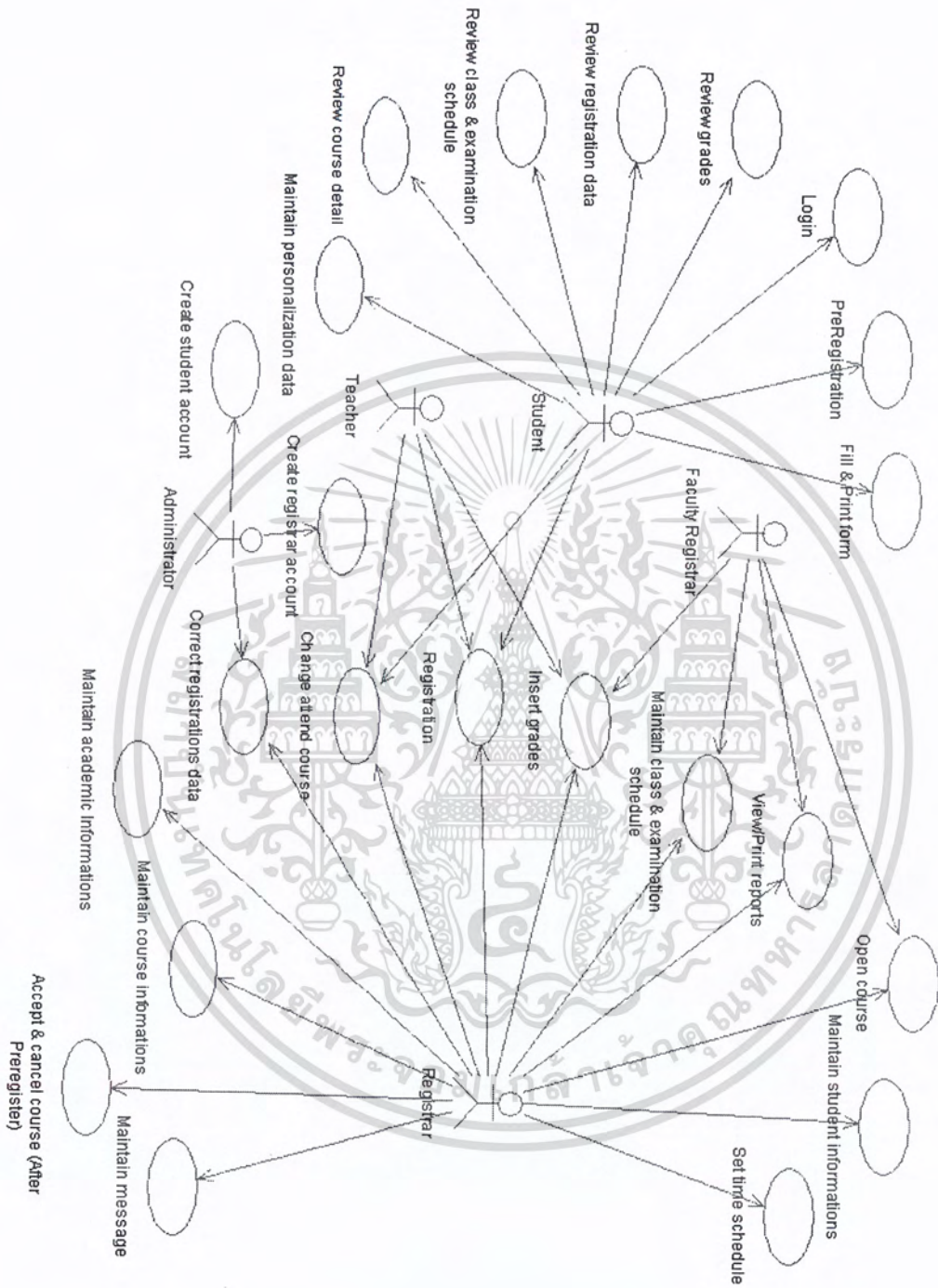
7.5.1 แก้ปัญหาเรื่องการแสดงผลภาษาไทย ที่ยังแสดงผลได้ไม่ดีนัก

7.5.2 จัดทำระบบเพิ่มเติม ในส่วนที่ได้ออกแบบไว้แล้ว แต่ยังไม่ได้มีการสร้างจริง เนื่องจากด้วยข้อจำกัดเรื่องระยะเวลาทำให้ผู้วิจัยไม่สามารถทำการสร้างโปรแกรมได้ครบทุกส่วน แต่ได้ทำการวิเคราะห์และออกแบบไว้ทั้งหมดแล้ว ซึ่งผู้ที่ทำการพัฒนาต่อสามารถนำไปสร้างต่อได้

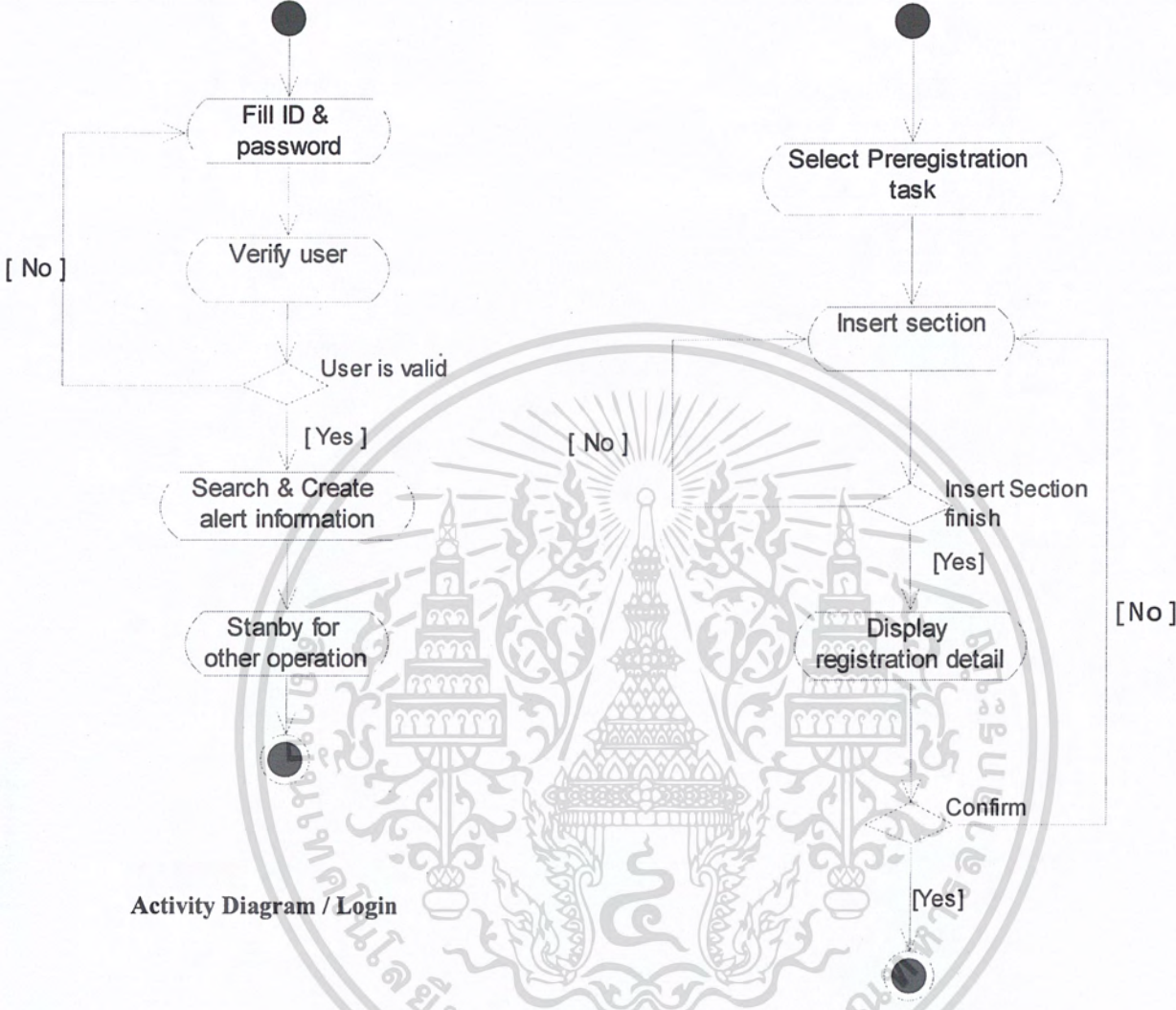
7.5.3 จัดระบบแนวทางปฏิบัติงานของฝ่ายทะเบียนอย่างชัดเจน เนื่องจากการปฏิบัติงานบางอย่างของฝ่ายทะเบียนอาจยังไม่มีมาตรฐานชัดเจน จึงควร เข้าไปทำการวิเคราะห์ปรึกษาหารือให้ได้แนวทางการปฏิบัติงานที่เป็นขั้นตอน

7.5.4 นำเอาเทคนิคใหม่ เข้ามาใช้ เพื่อทำให้ระบบมีความสะดวกมากขึ้น อย่างเช่นในเรื่องของการกรอกเกรดอาจจะทำให้อาจารย์แต่ละท่านสามารถกรอกเกรดผ่านระบบอินเทอร์เน็ตก็ได้เพื่อความสะดวก อาจพัฒนาในเรื่องการส่งข้อความ และการสื่อสารระหว่างอาจารย์และนักศึกษาให้สามารถสื่อสารข้อความกันได้ผ่านทางระบบนี้ได้เลย



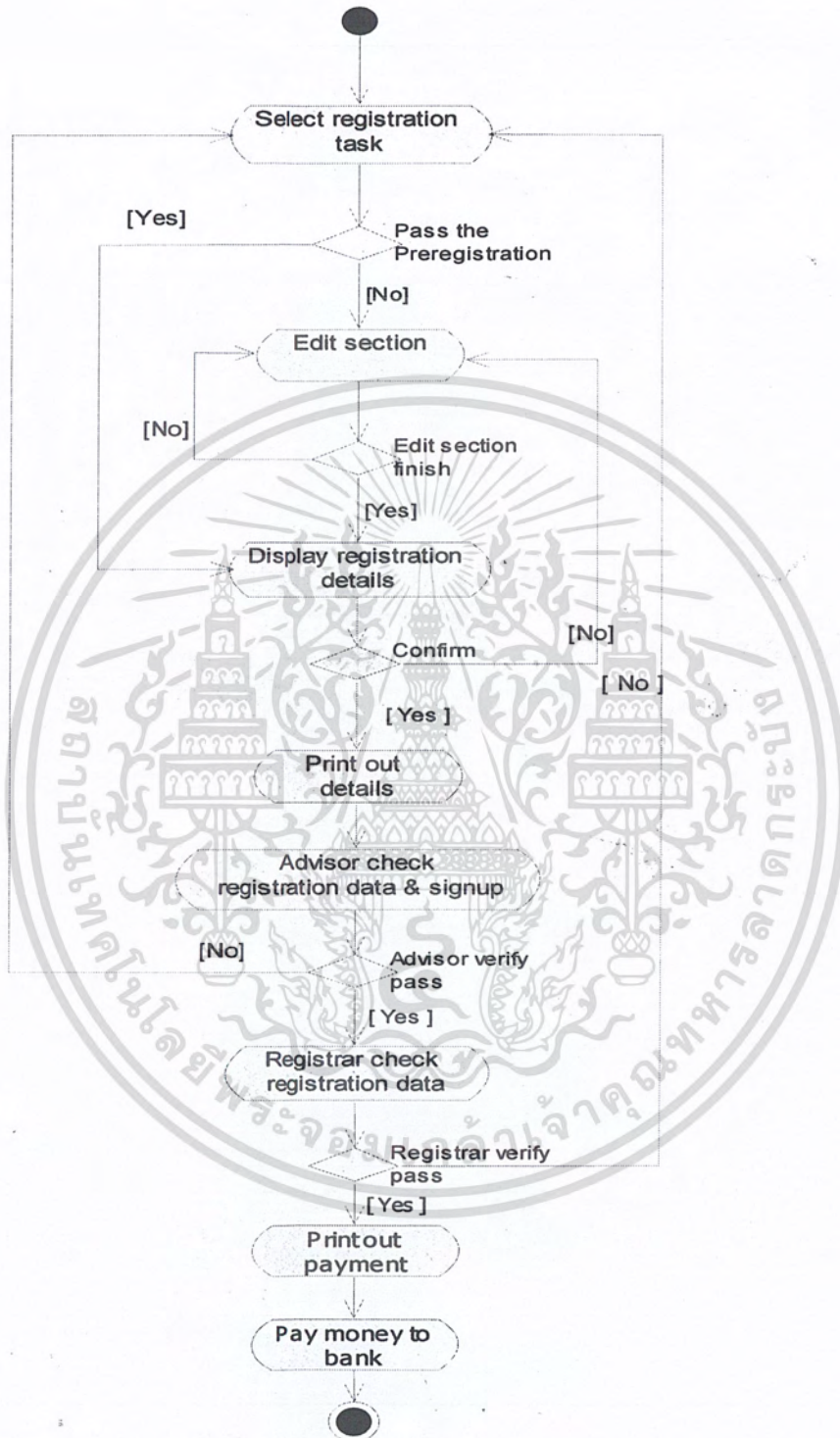


Usecase Diagram

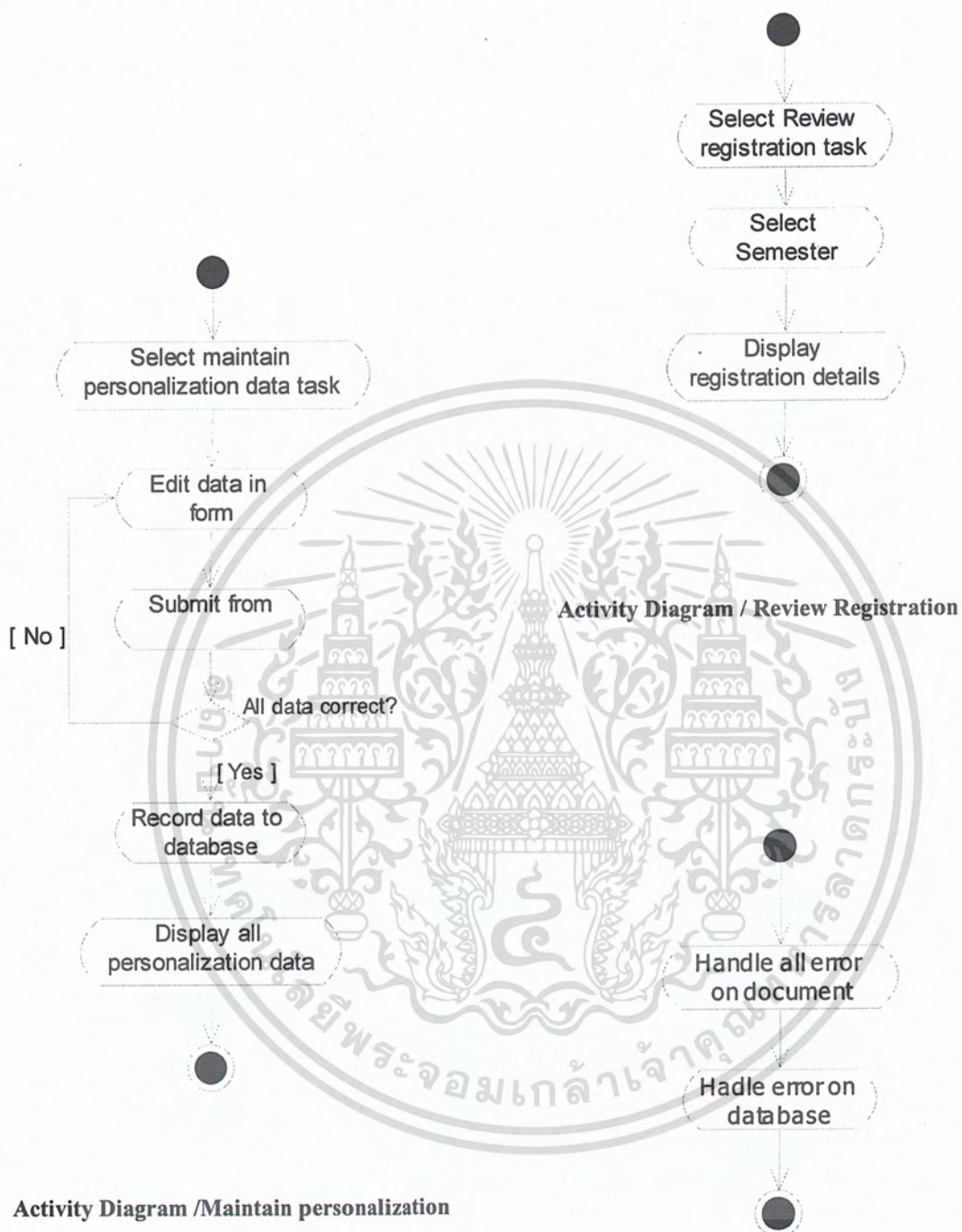


Activity Diagram / Login

Activity Diagram / Preregistration

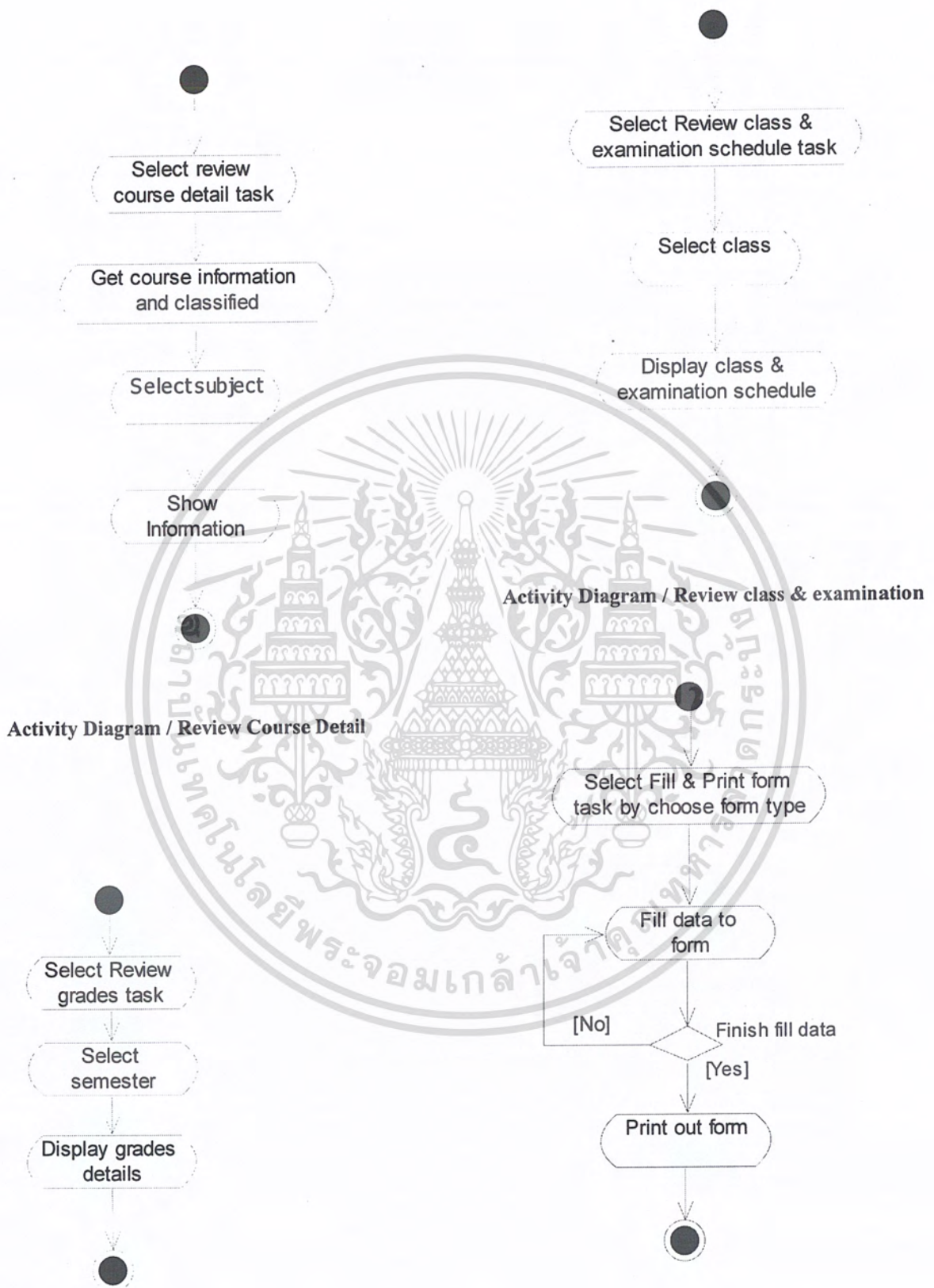


Activity Diagram / Registration



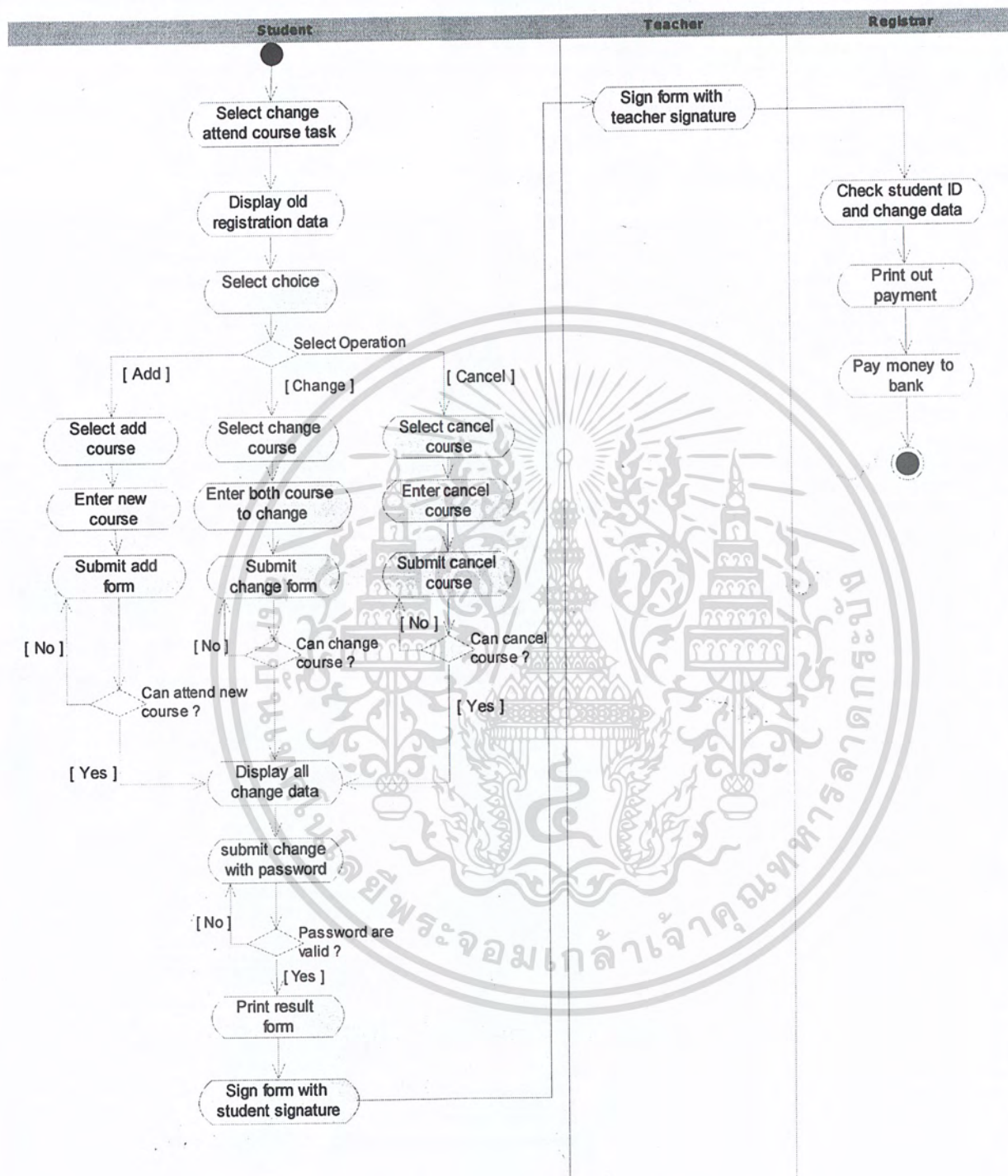
Activity Diagram /Maintain personalization

Activity Diagram /Correct registrations

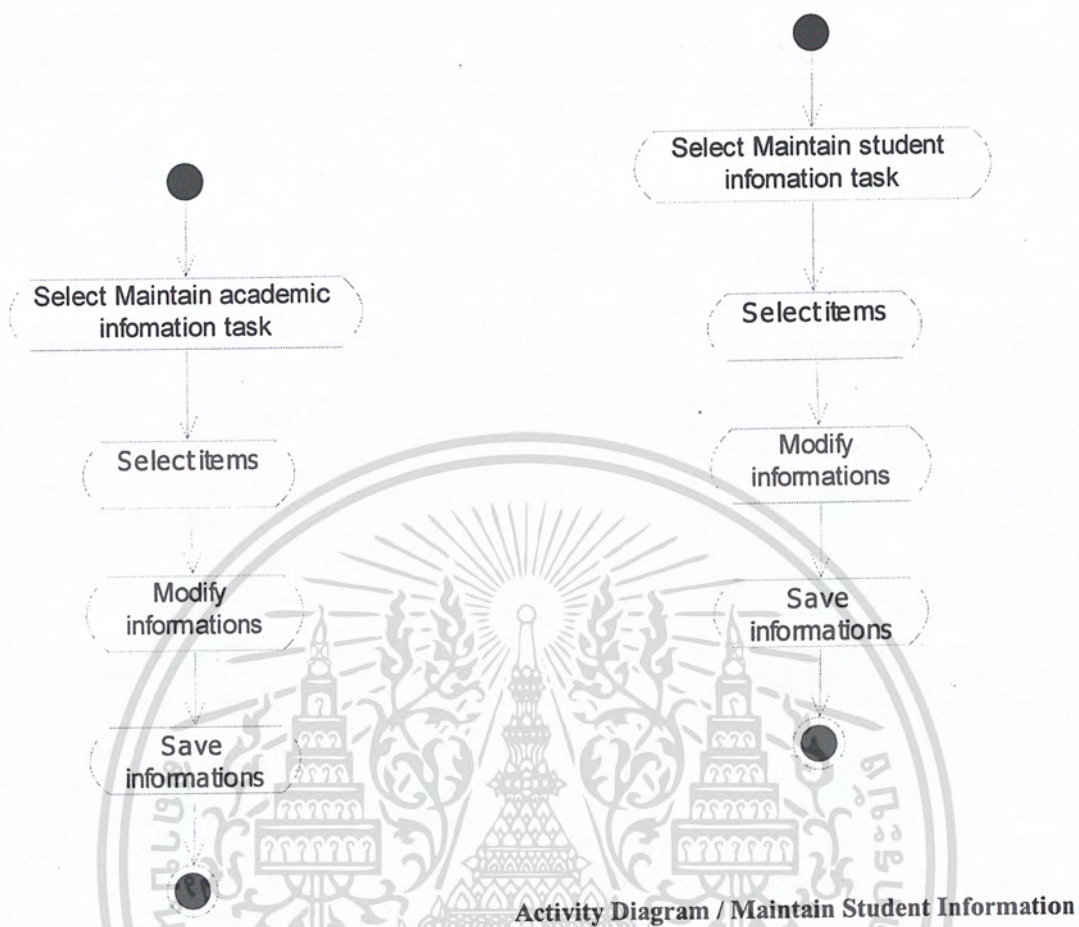


Activity Diagram / Review Grades

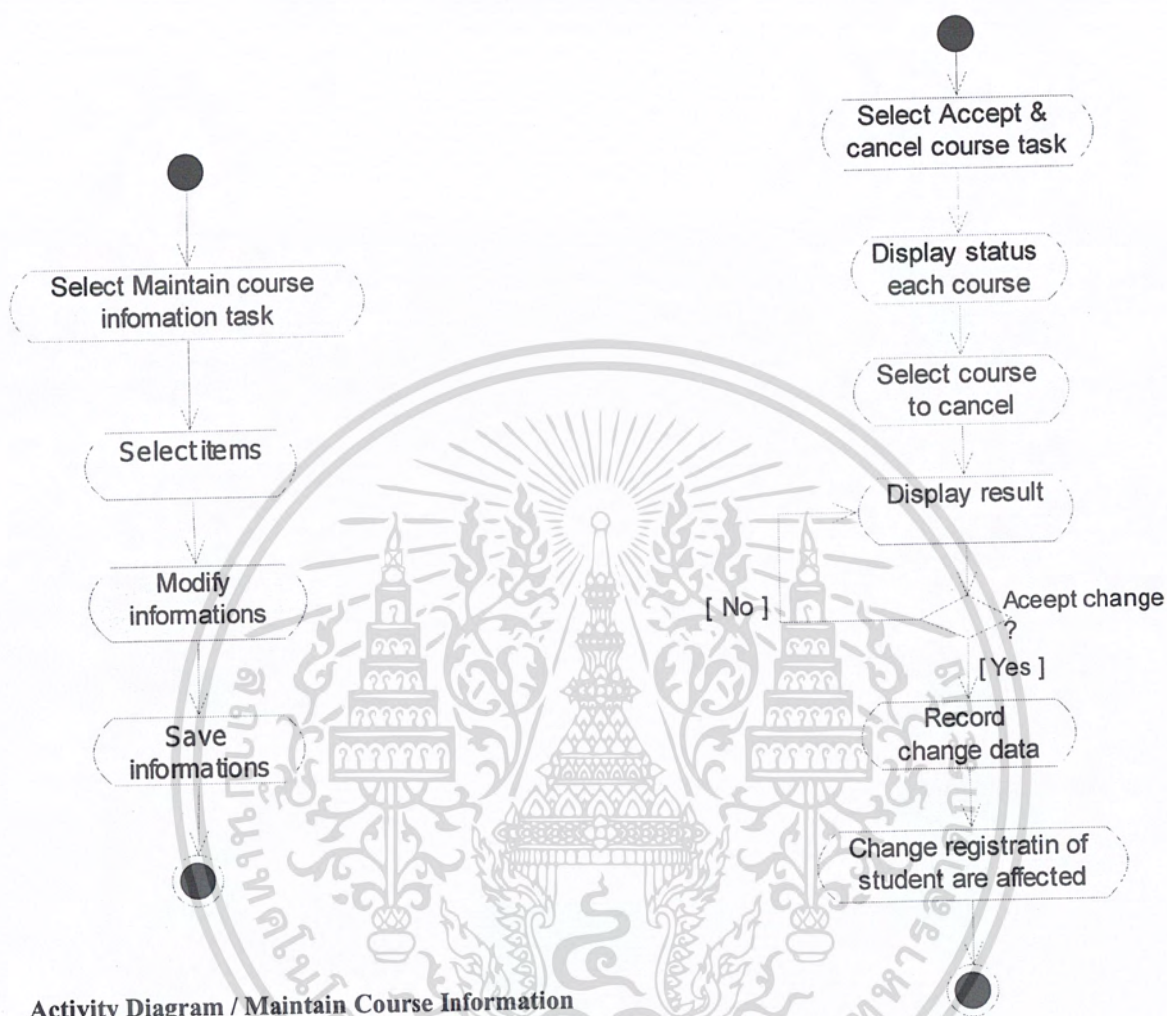
Activity Diagram / Fill & Print Form

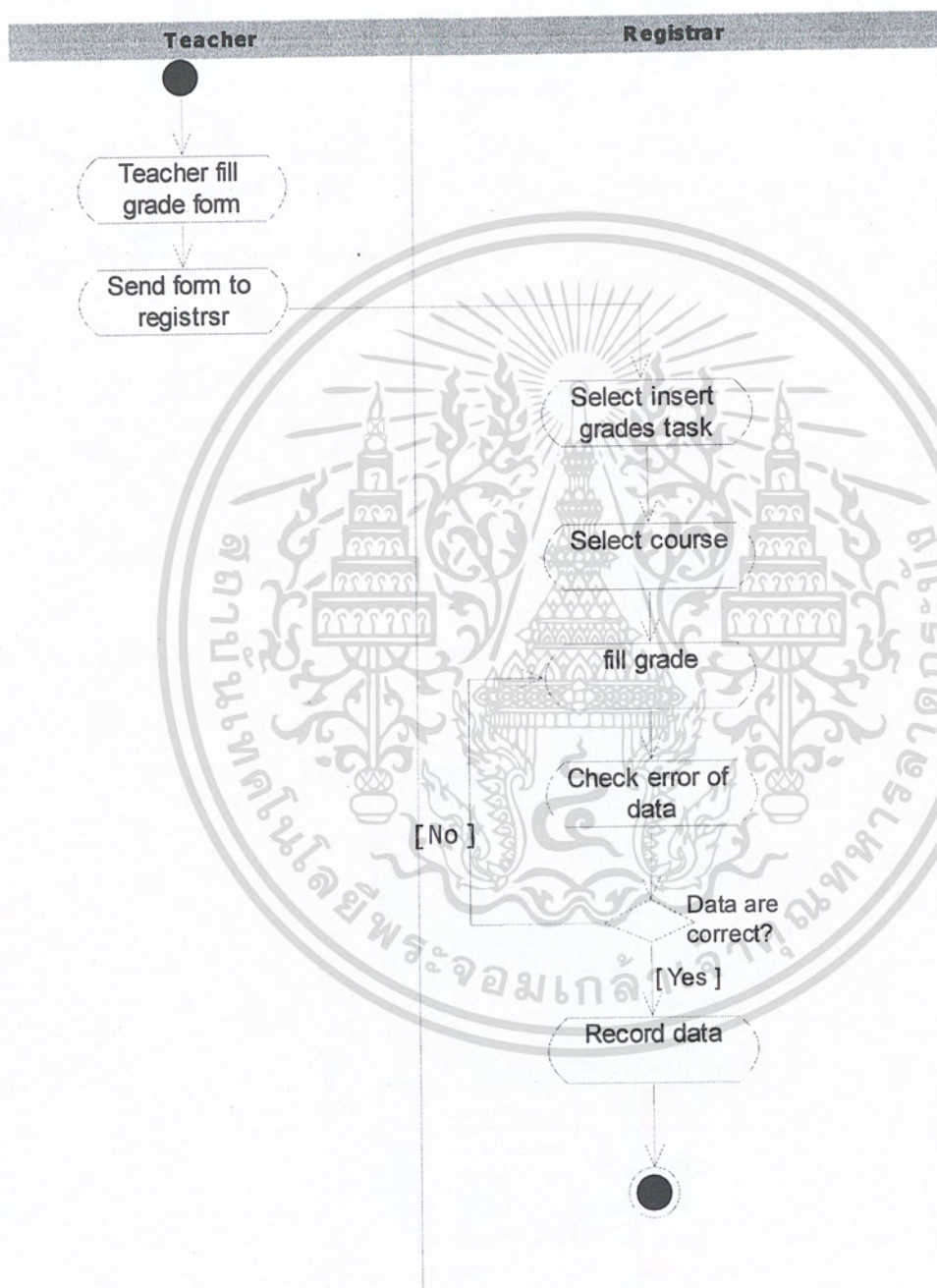


Activity Diagram / Chang Attend Course

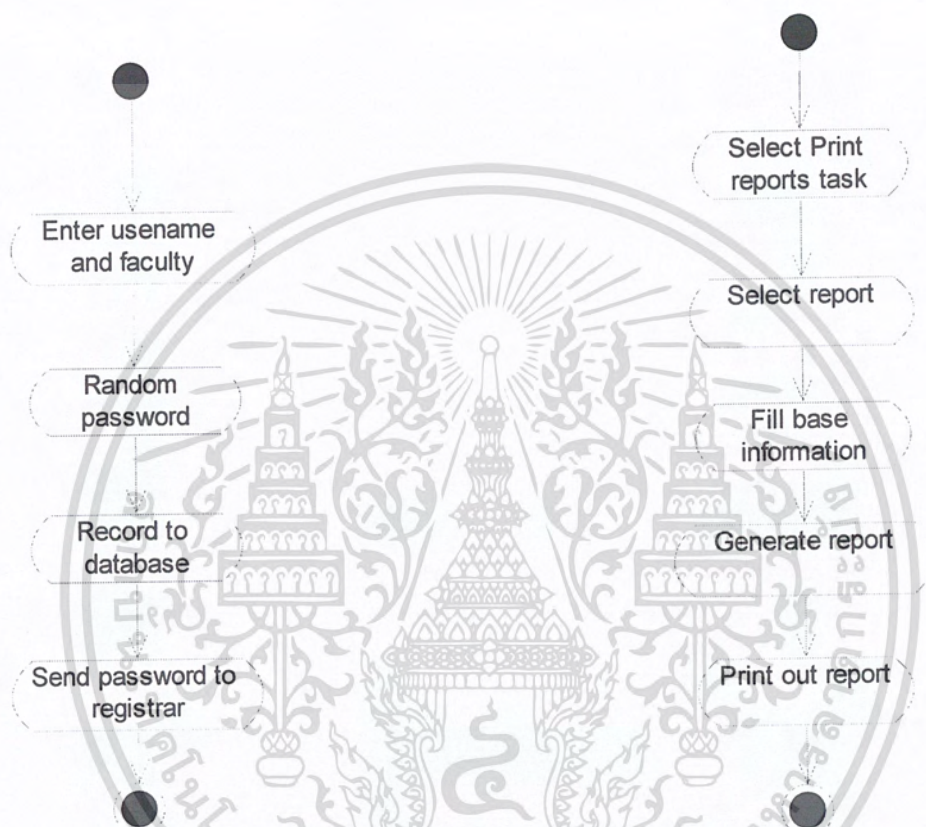


Activity Diagram / Maintain Academic information



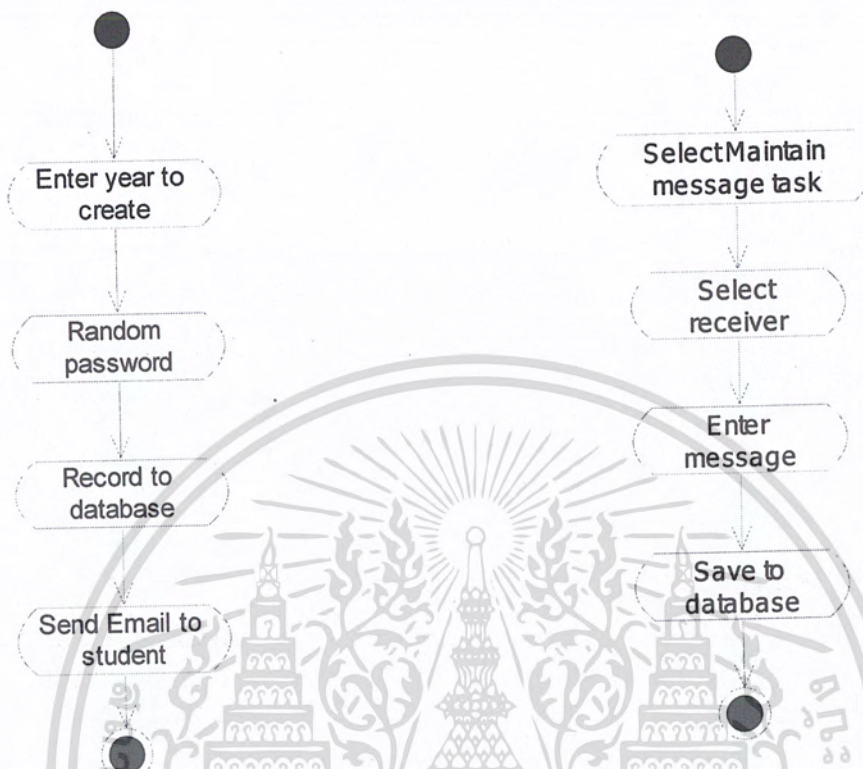


Activity Diagram / Insert Grade



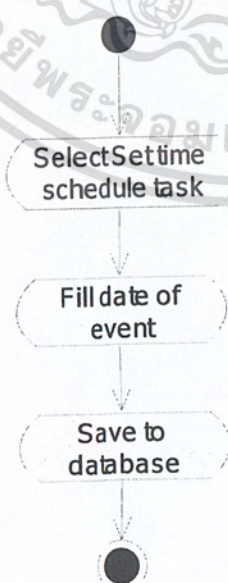
Activity Diagram / Create registrar

Activity Diagram / View & Print Report

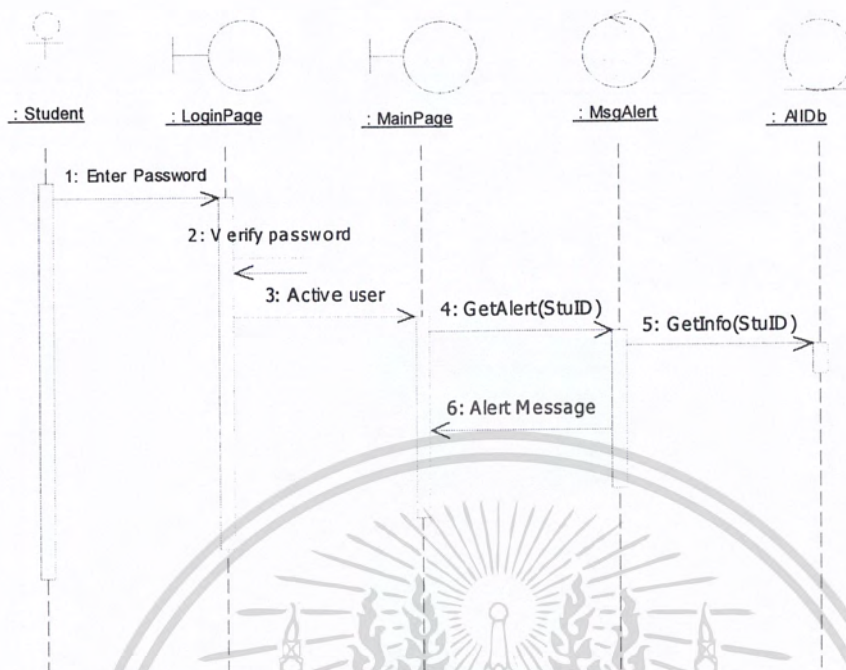


Activity Diagram / Create Student account

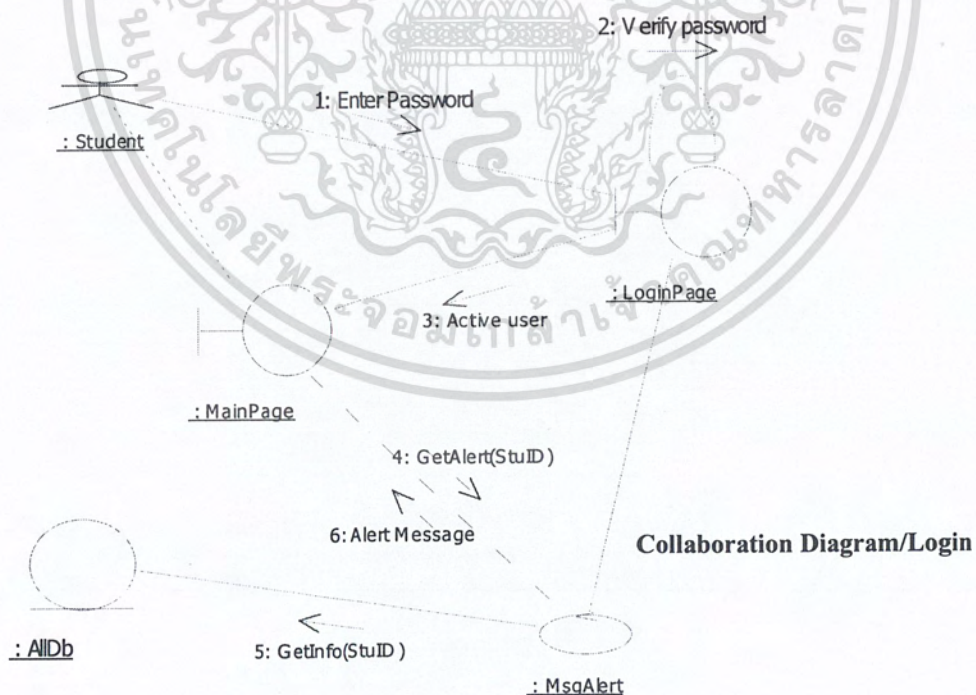
Activity Diagram / Maintain Message



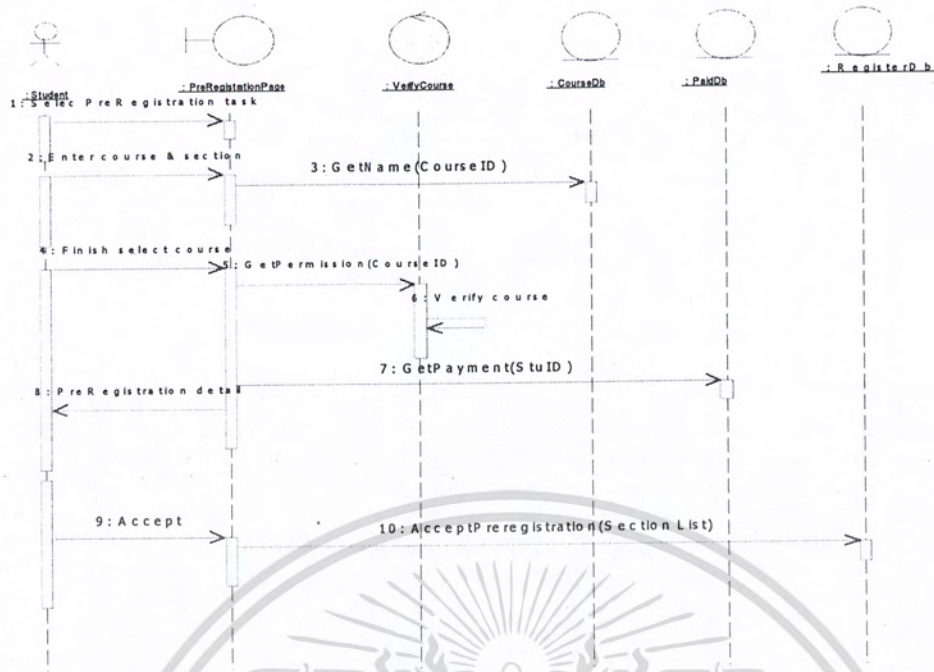
Activity Diagram /Set Time



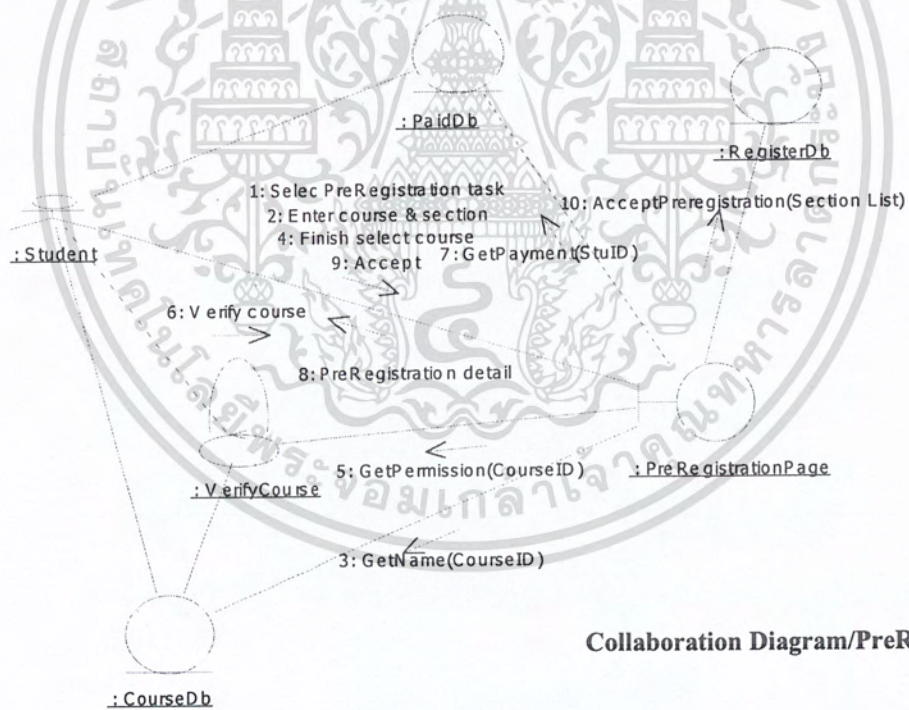
Sequence Diagram/Login



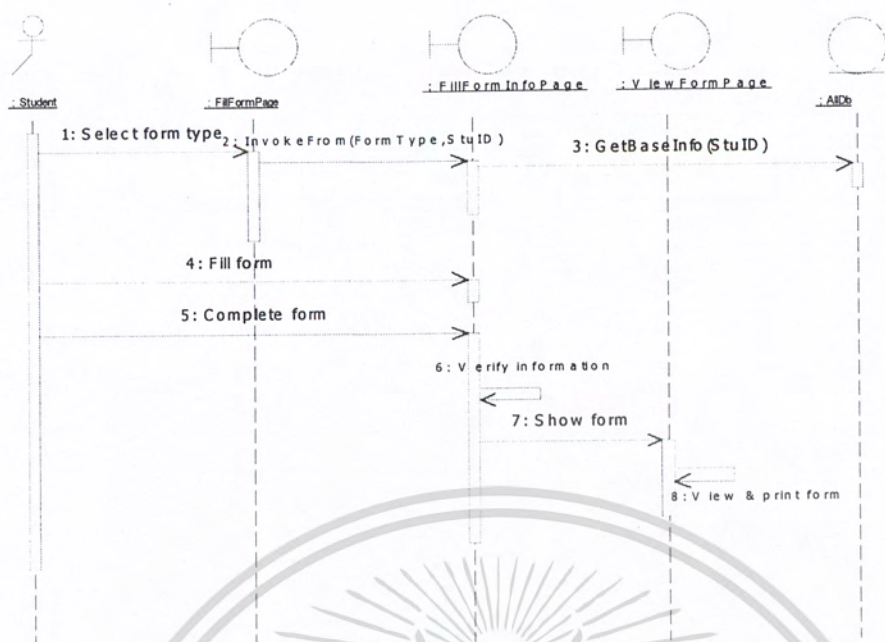
Collaboration Diagram/Login



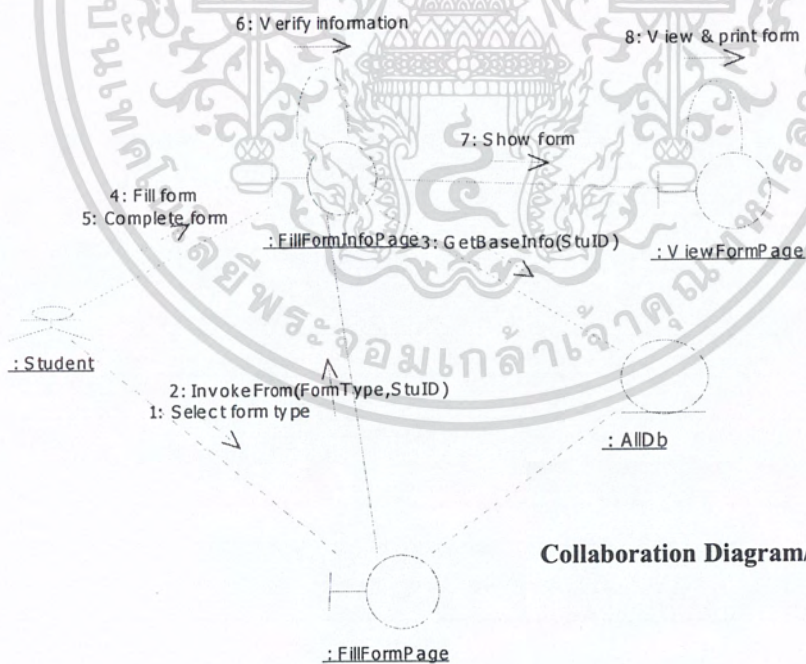
Sequence Diagram/PreRegister



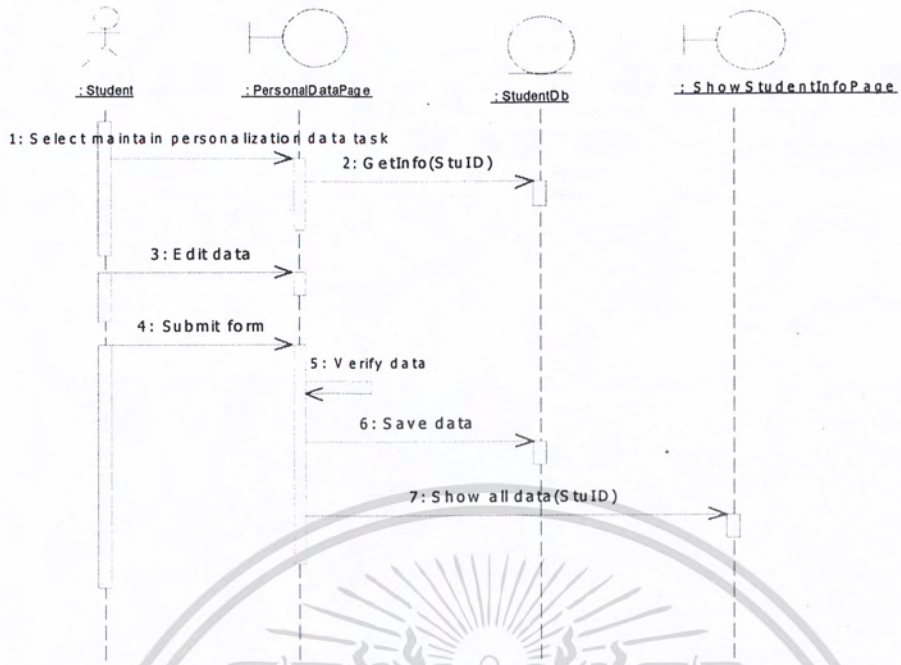
Collaboration Diagram/PreRegister



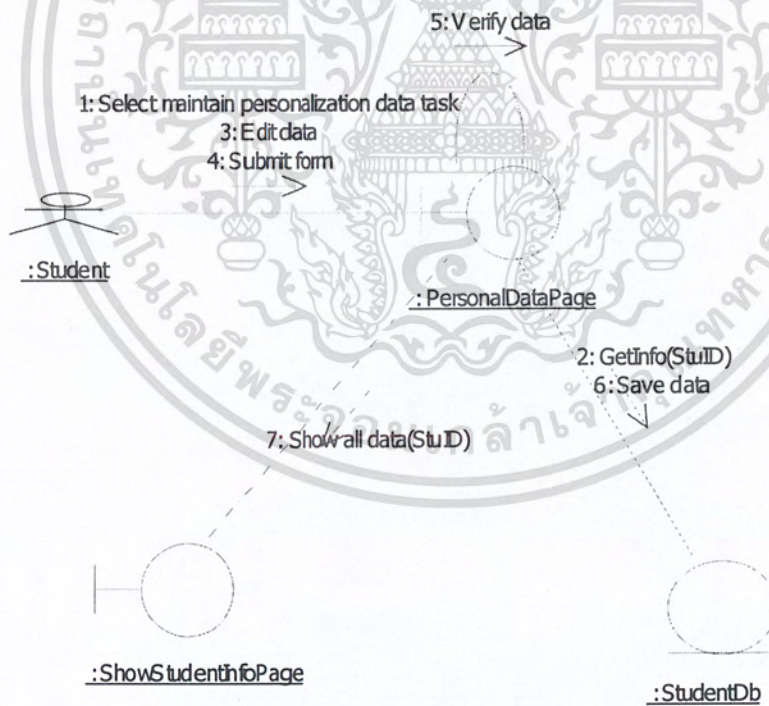
Sequence Diagram/Fill &



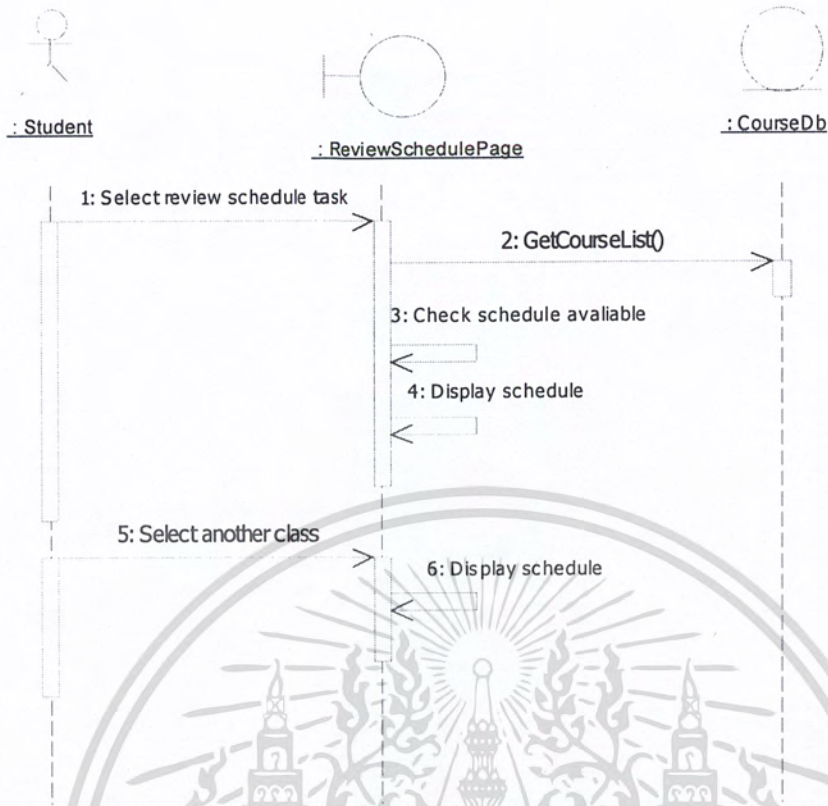
Collaboration Diagram/Fill & print form



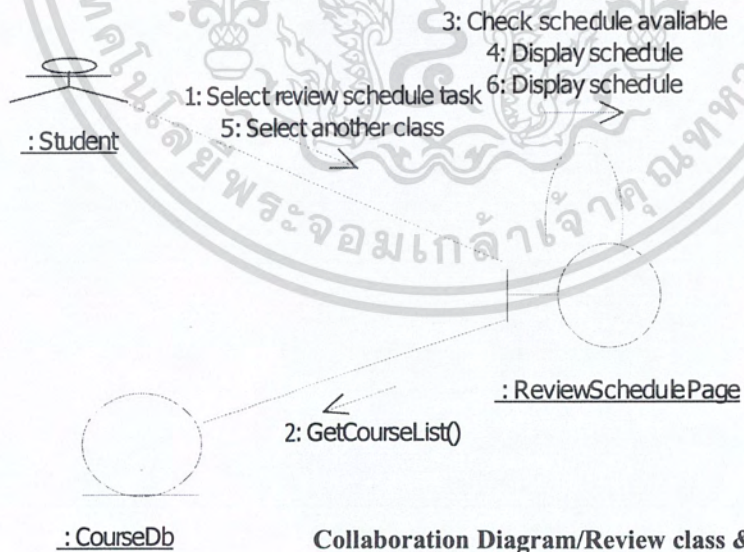
Sequence Diagram/Maintain personalization data



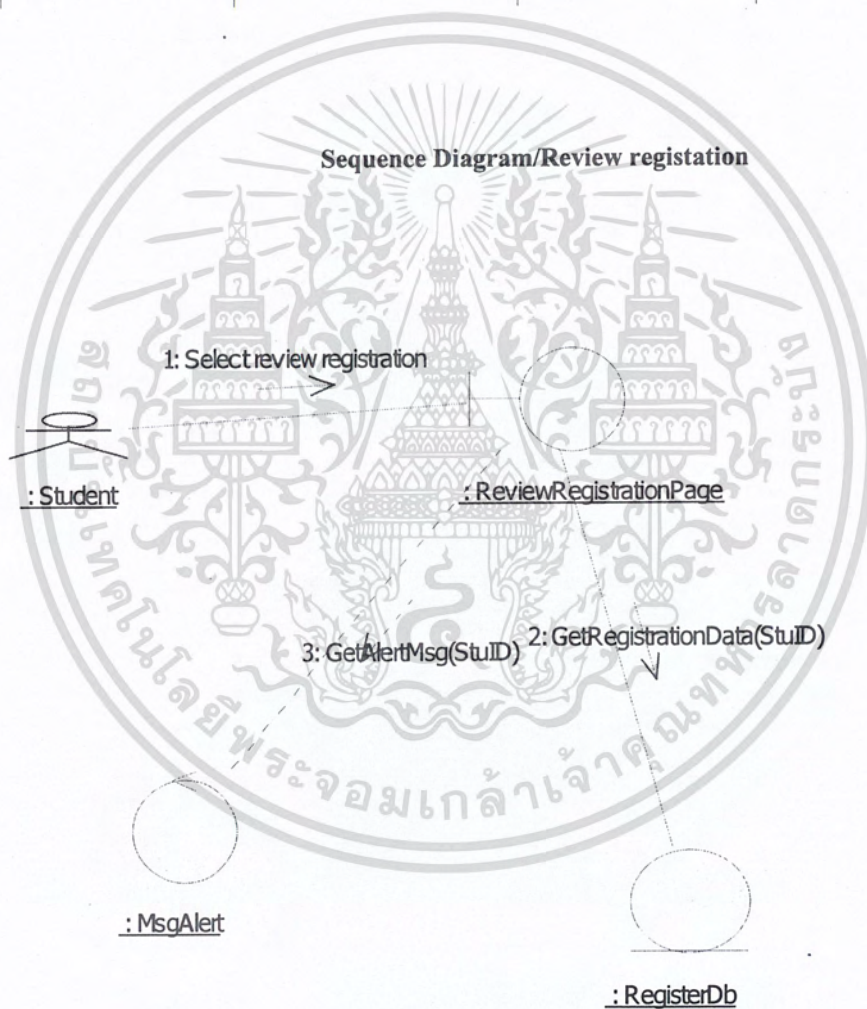
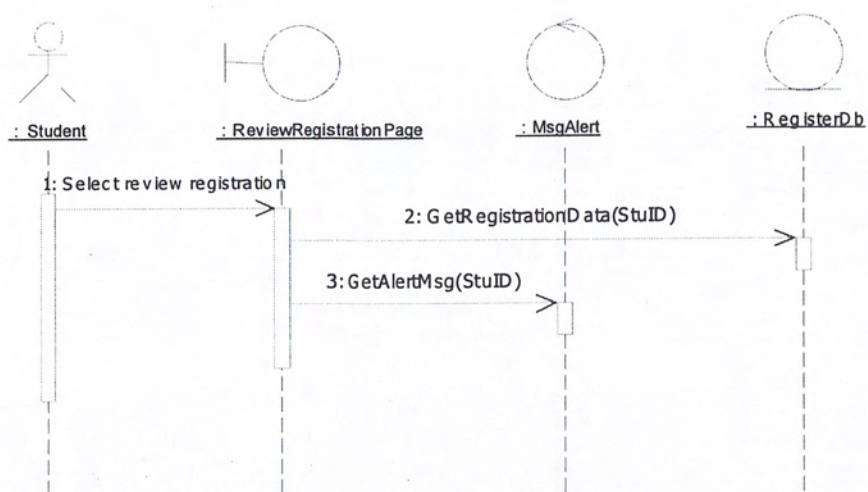
Collaboration Diagram/Maintain personalization data



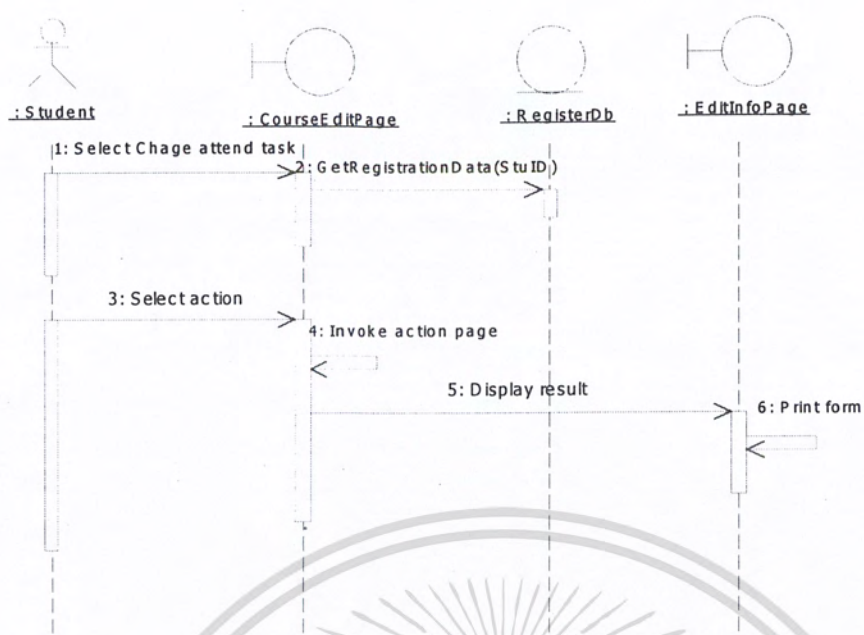
Sequence Diagram/Review class & examination schedule



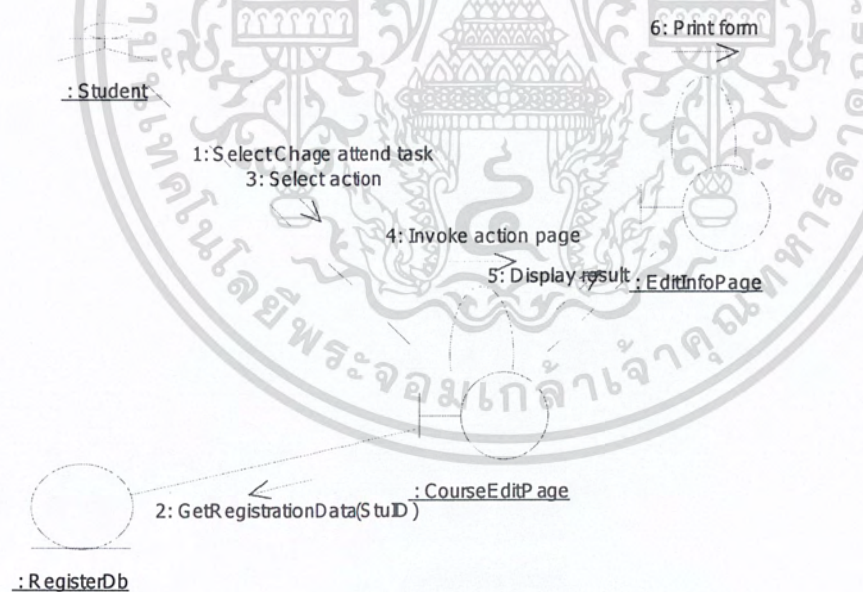
Collaboration Diagram/Review class & examination schedule



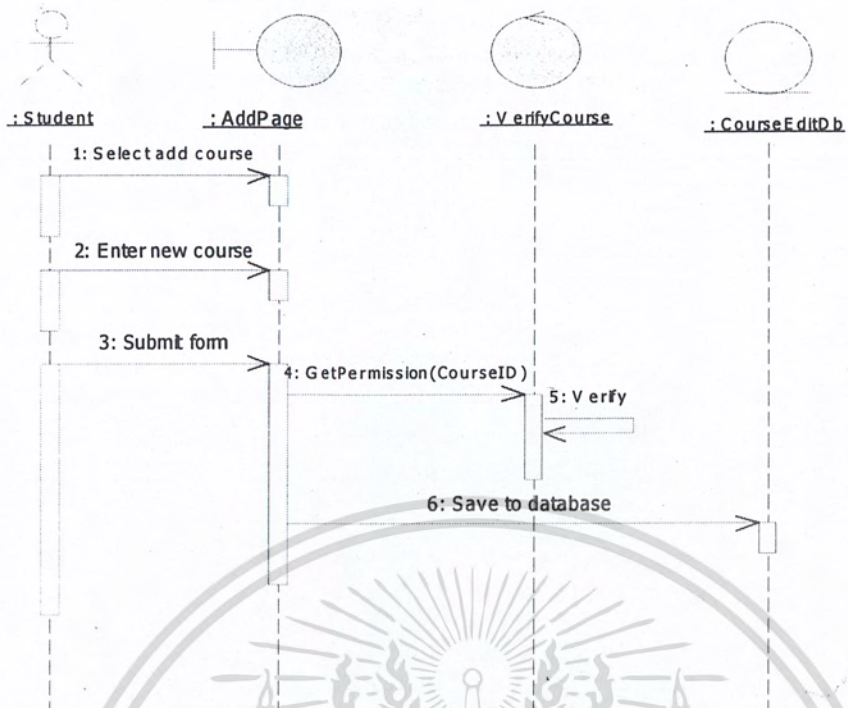
Collaboration Diagram/Review registration



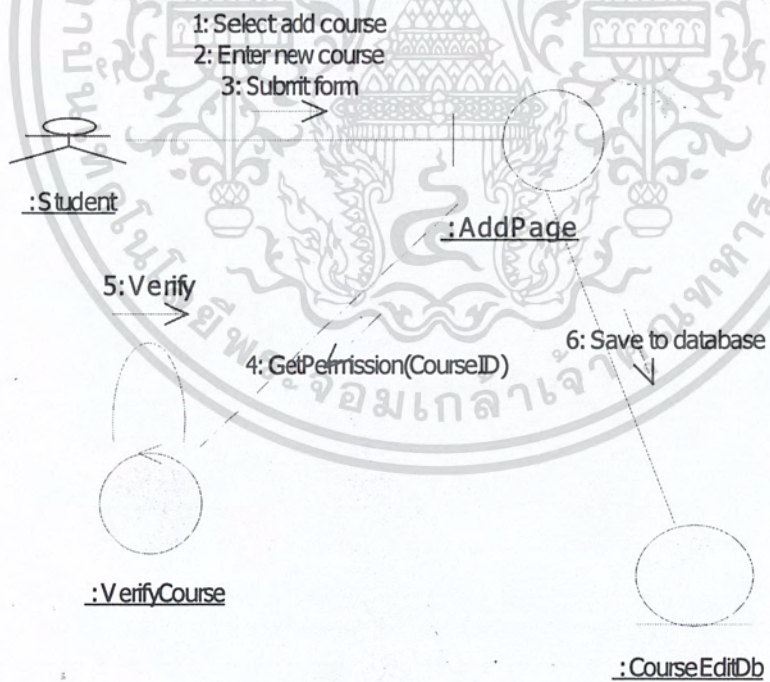
Sequence Diagram/StudentChange



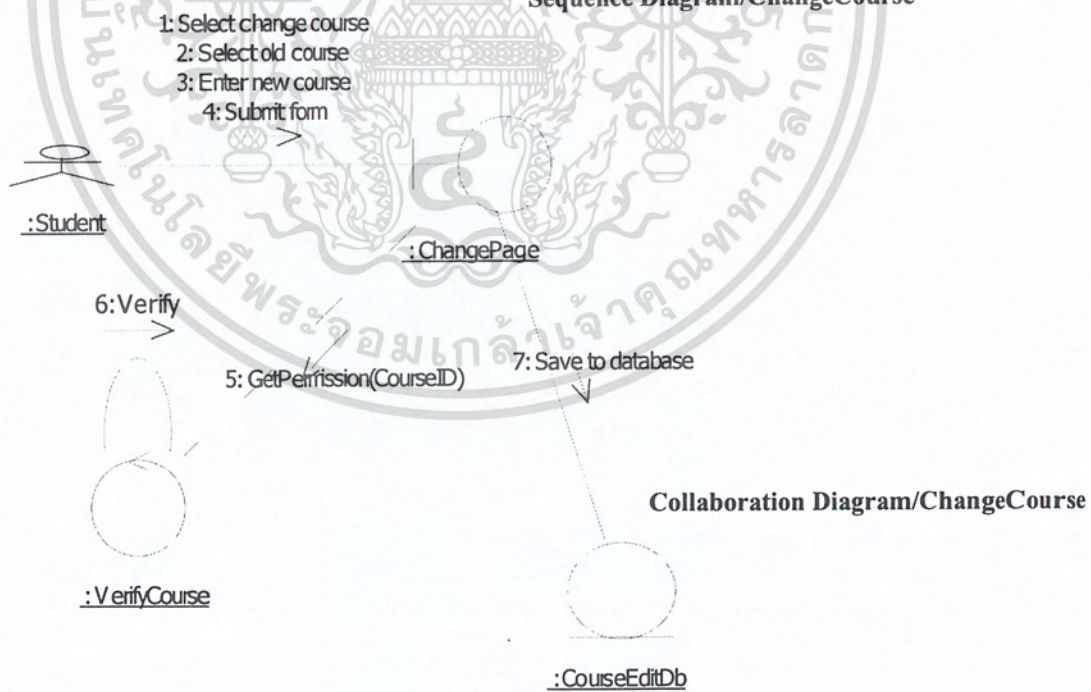
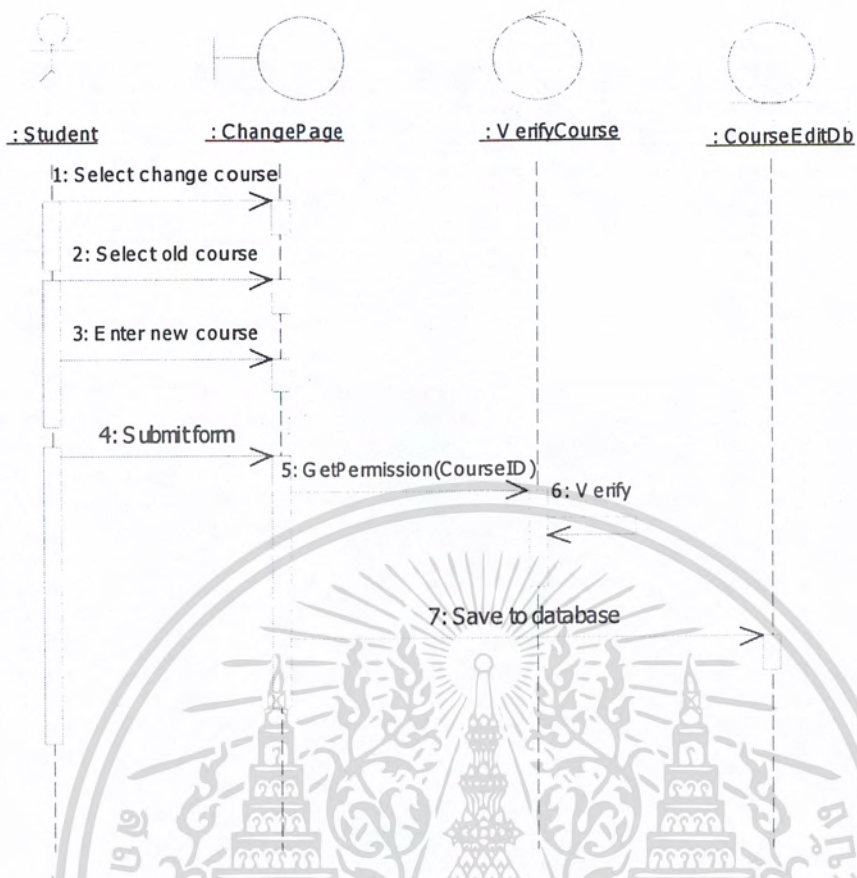
Collaboration Diagram/StudentChange

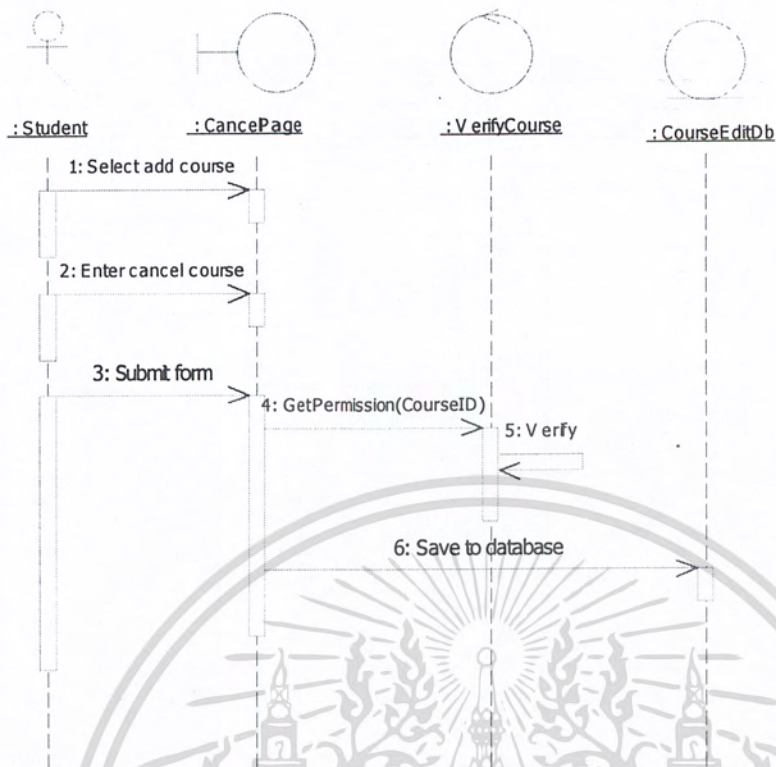


Sequence Diagram/AddCourse

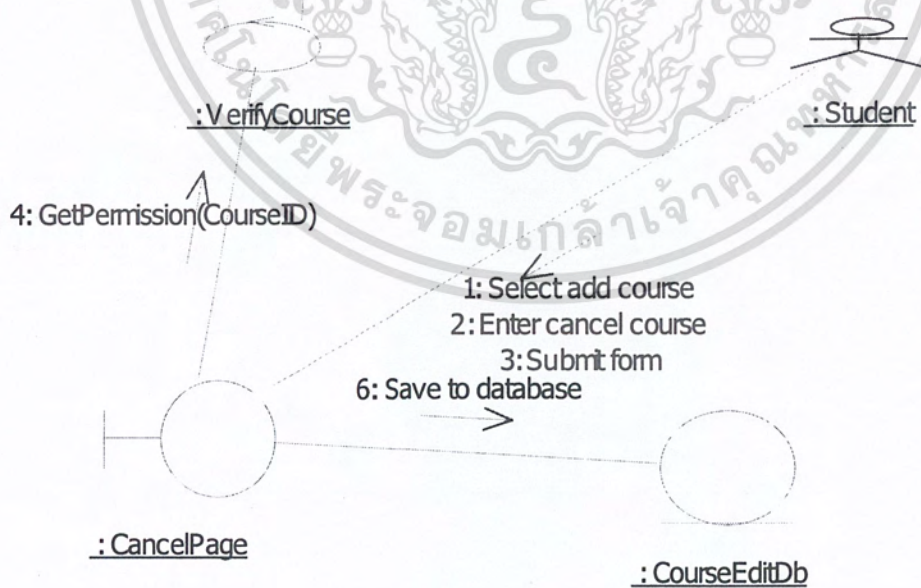


Collaboration Diagram/AddCourse

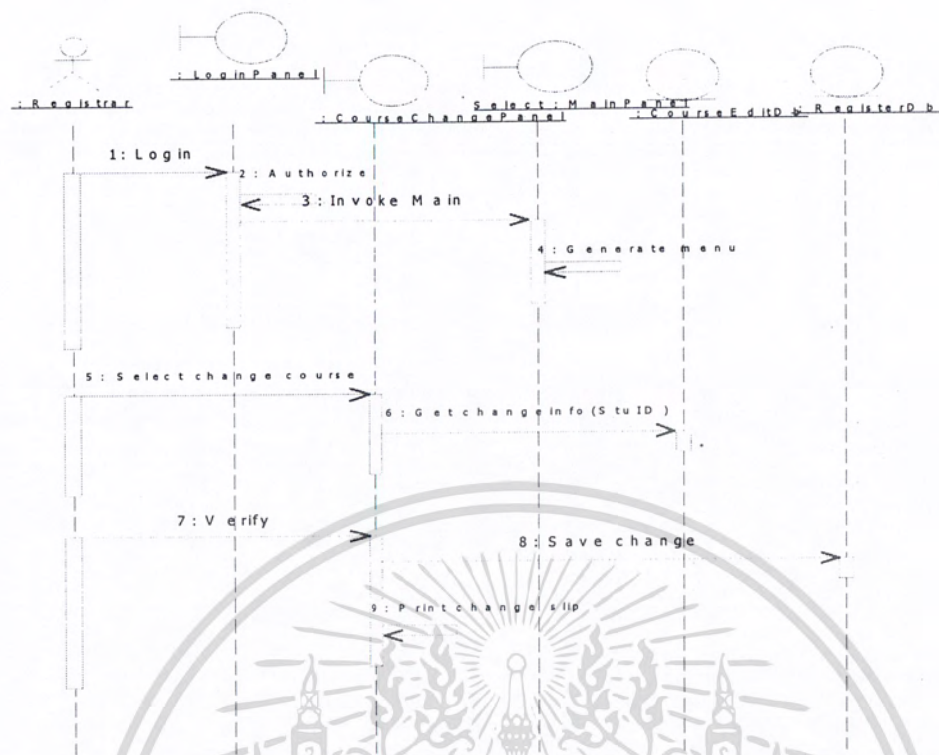




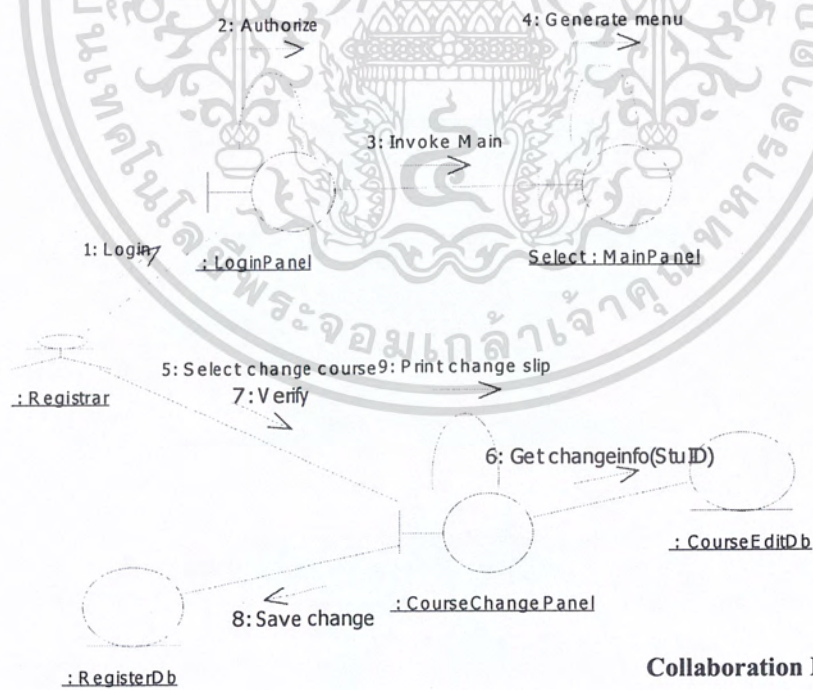
Sequence Diagram/CancelCourse



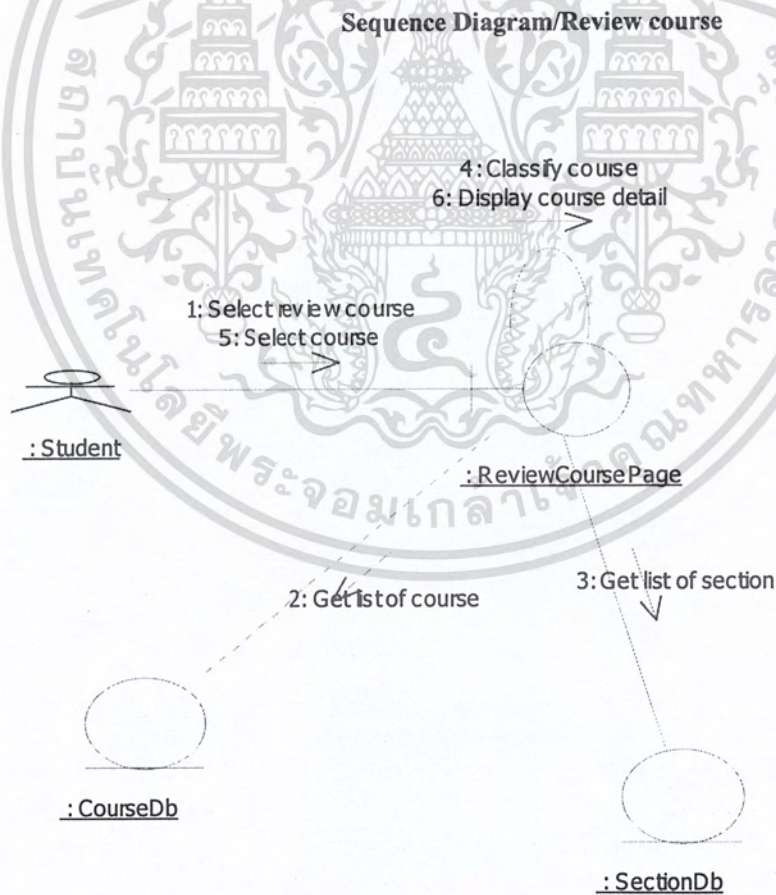
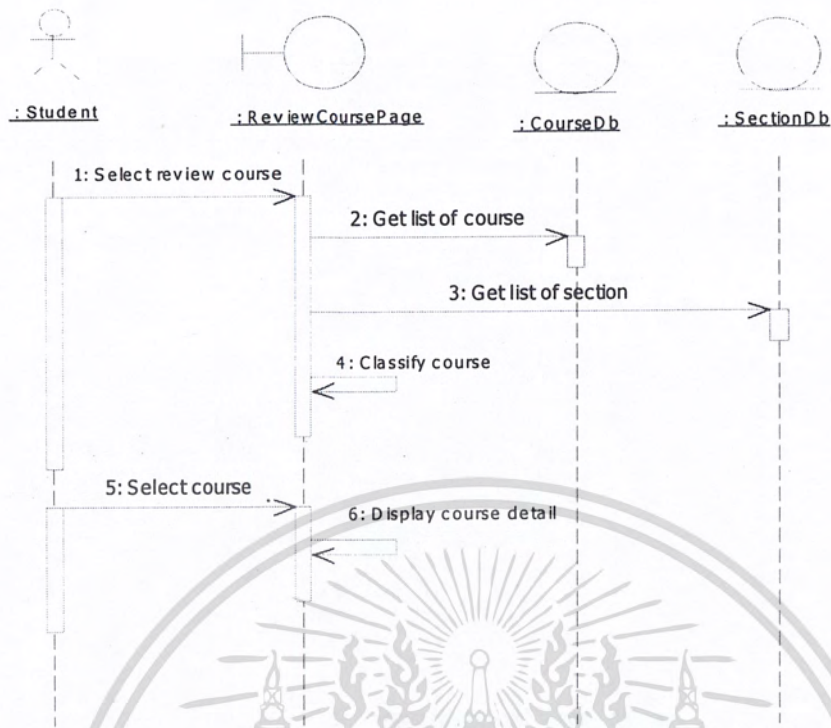
Collaboration Diagram/CancelCourse



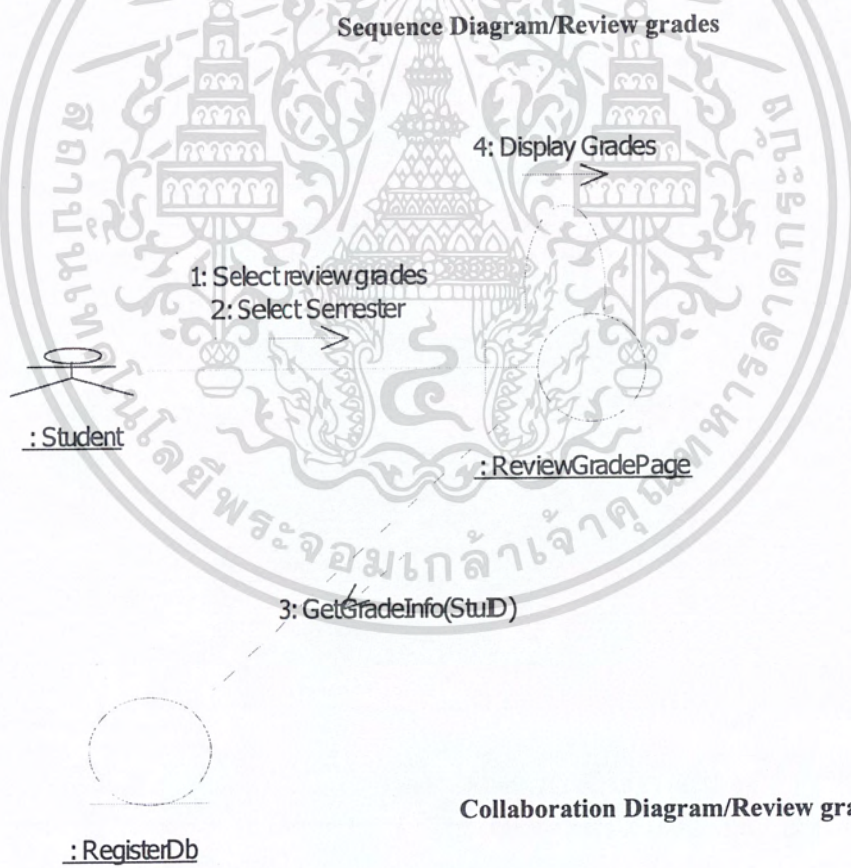
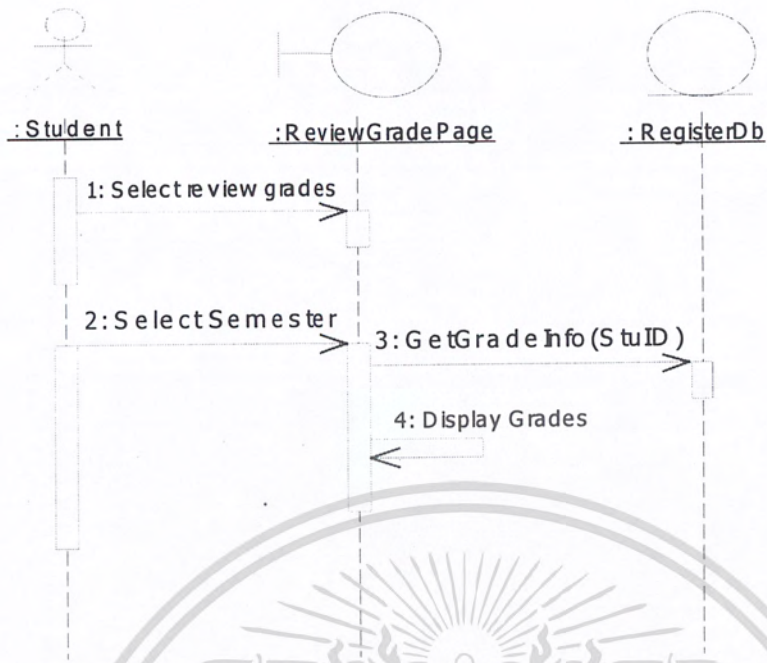
Sequence Diagram/AcceptChange

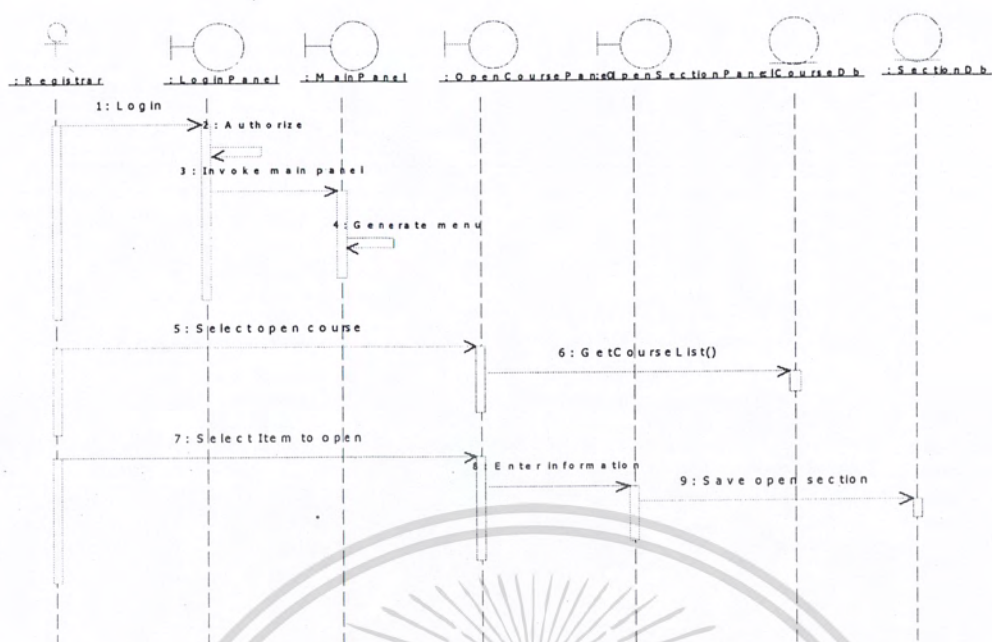


Collaboration Diagram/AcceptChange

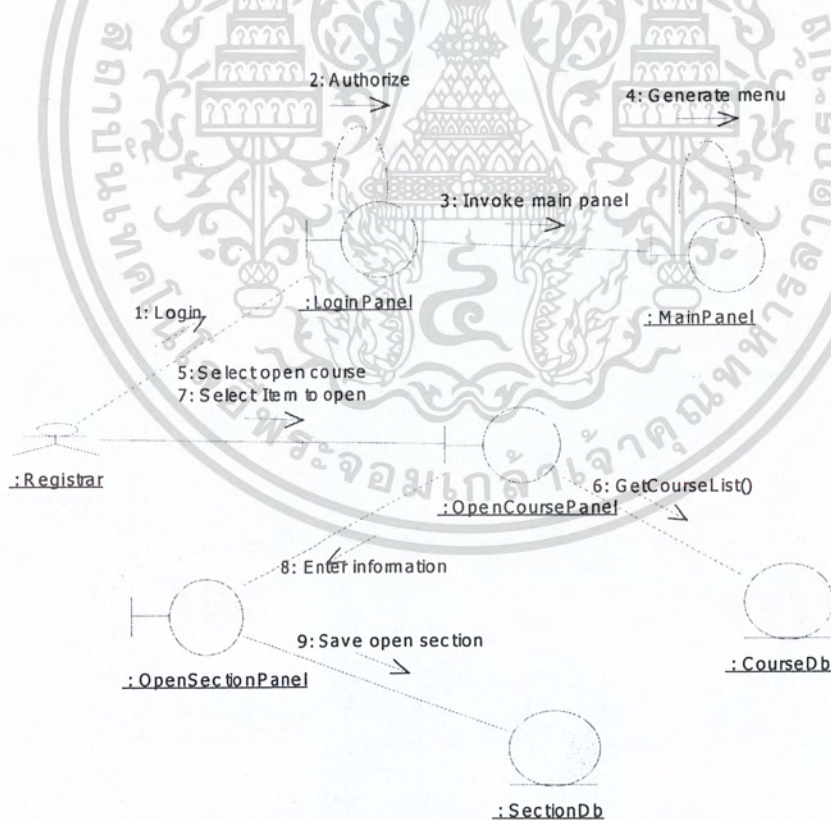


Collaboration Diagram/Review course

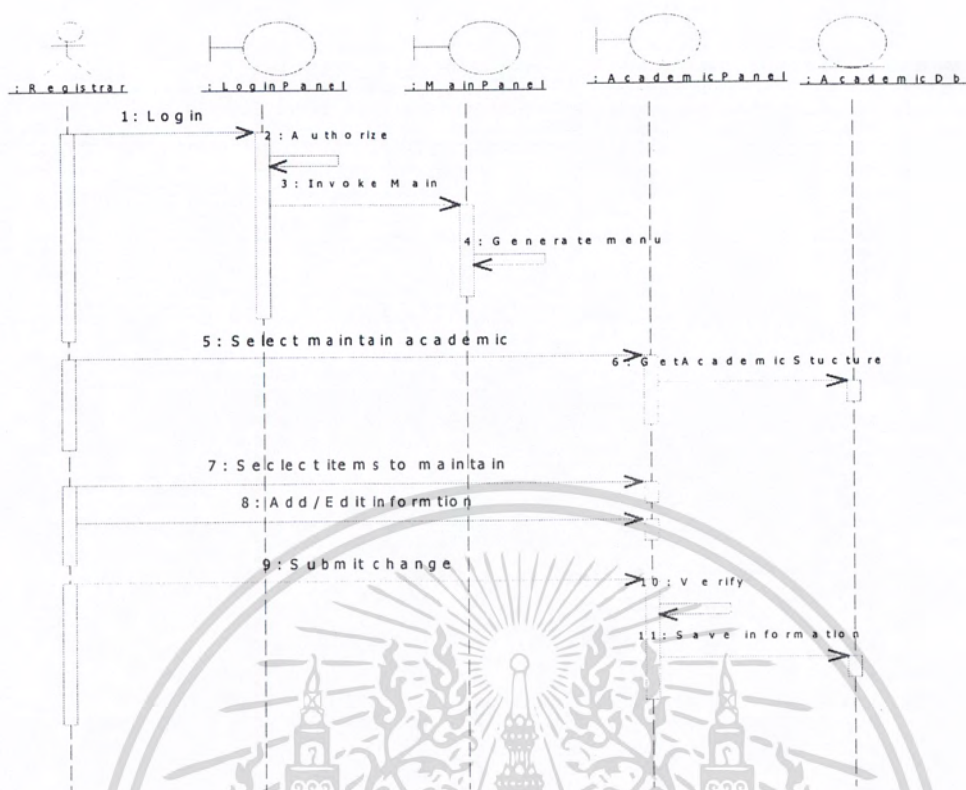




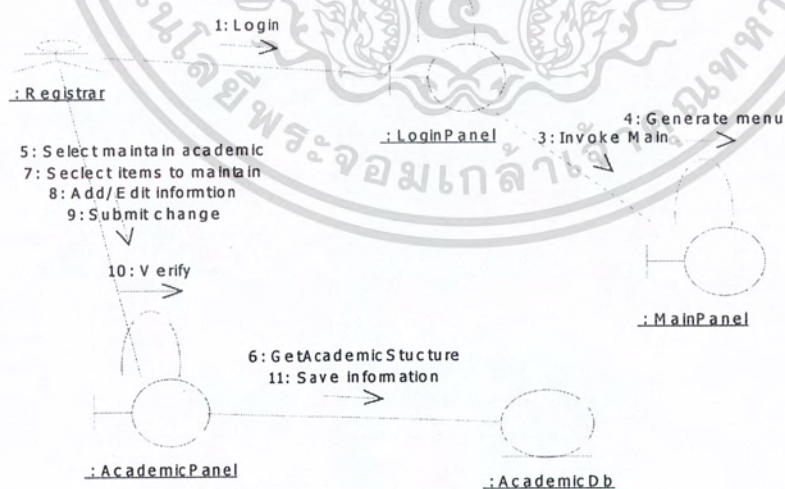
Sequence Diagram/Open course



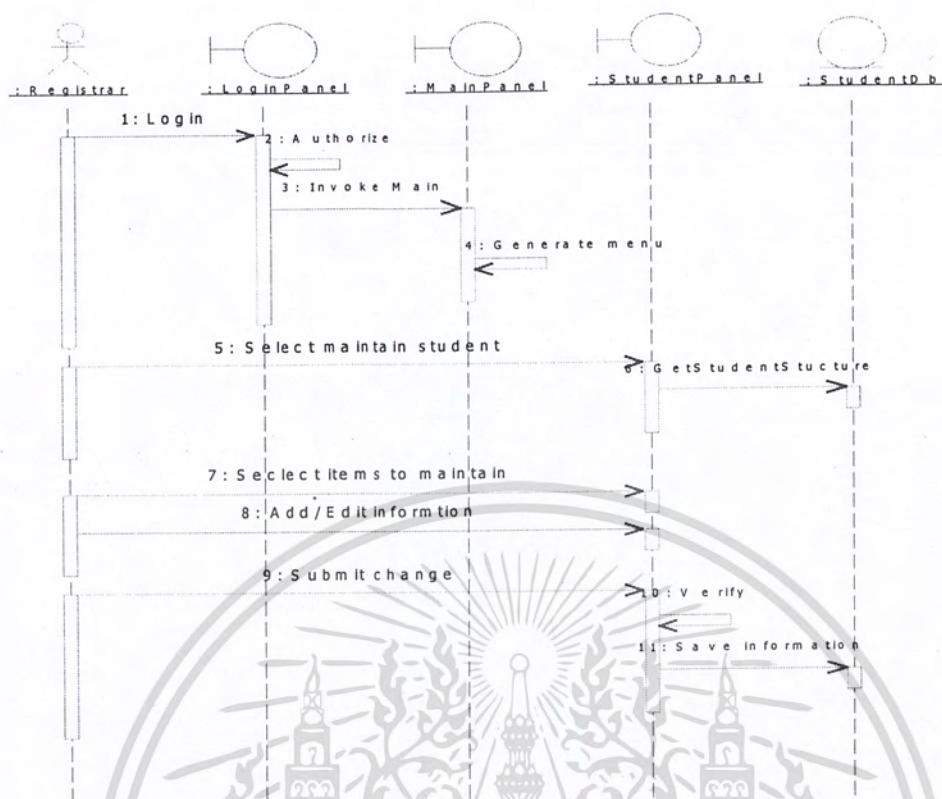
Collaboration Diagram/Open course



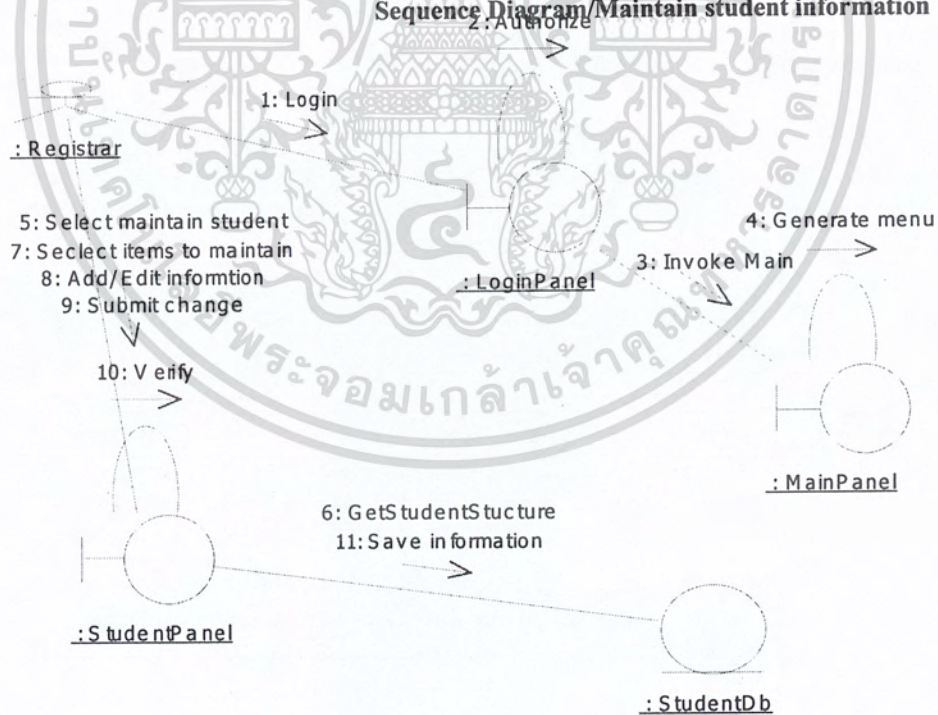
Sequence Diagram/Maintain academic information



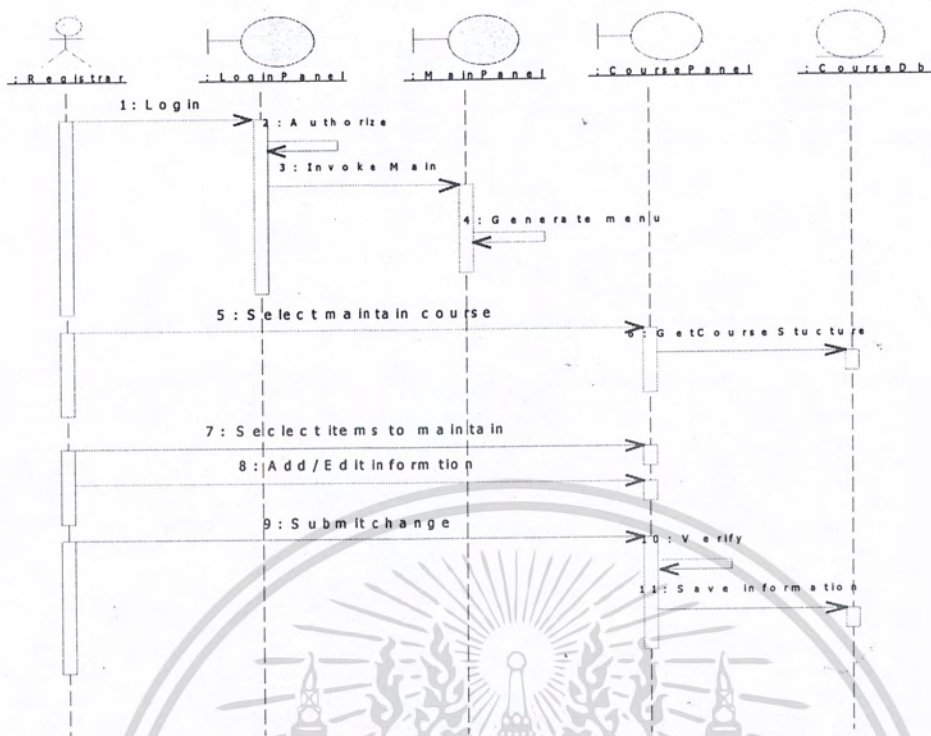
Collaboration Diagram/Maintain academic information



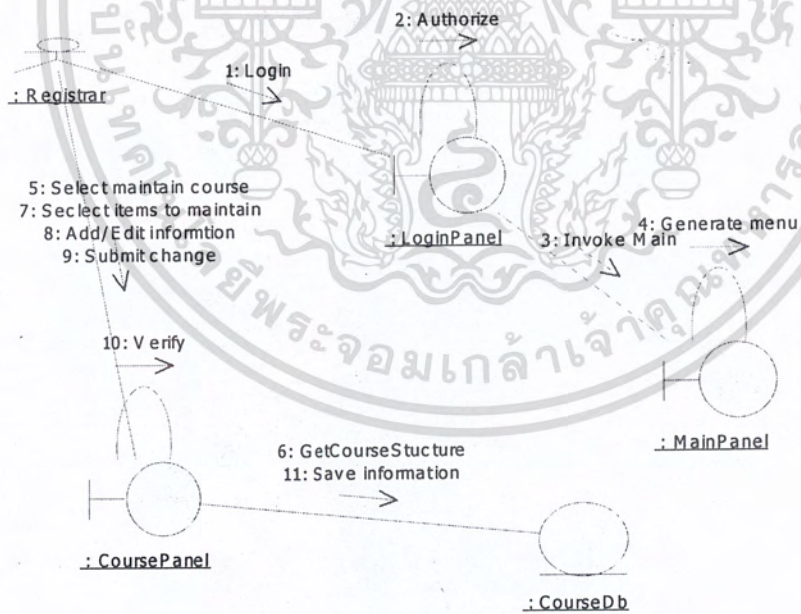
Sequence Diagram/Maintain student information



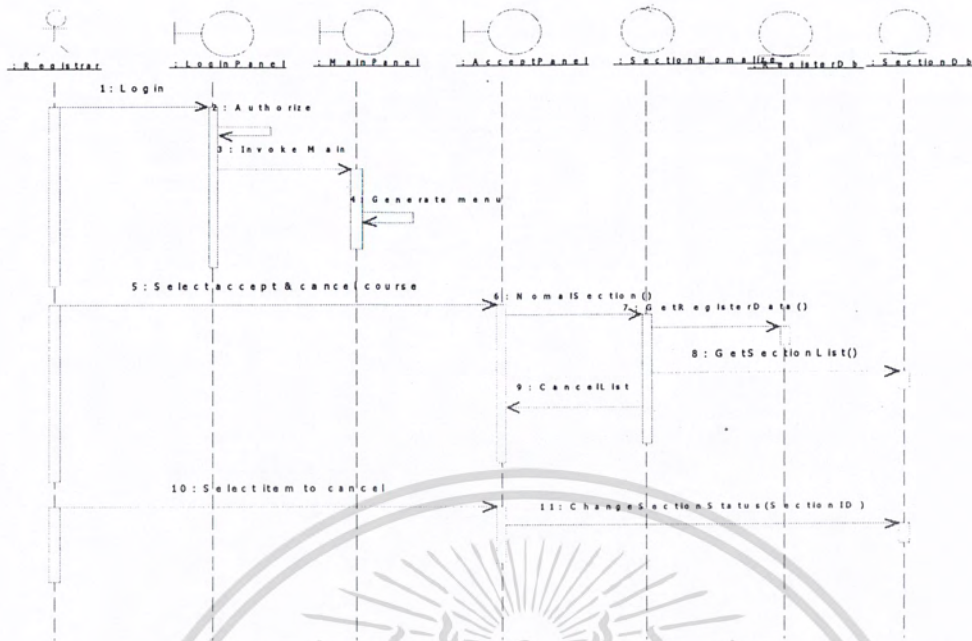
Collaboration Diagram/Maintain student information



Sequence Diagram/Maintain course informations



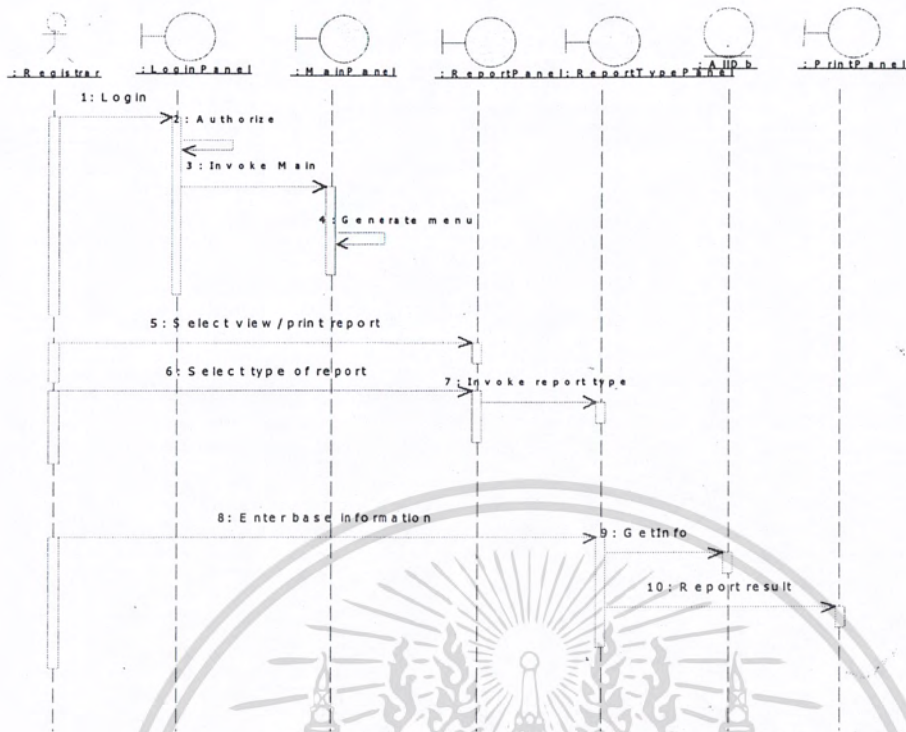
Collaboration Diagram/Maintain course informations



Sequence Diagram/Accept & cancel course



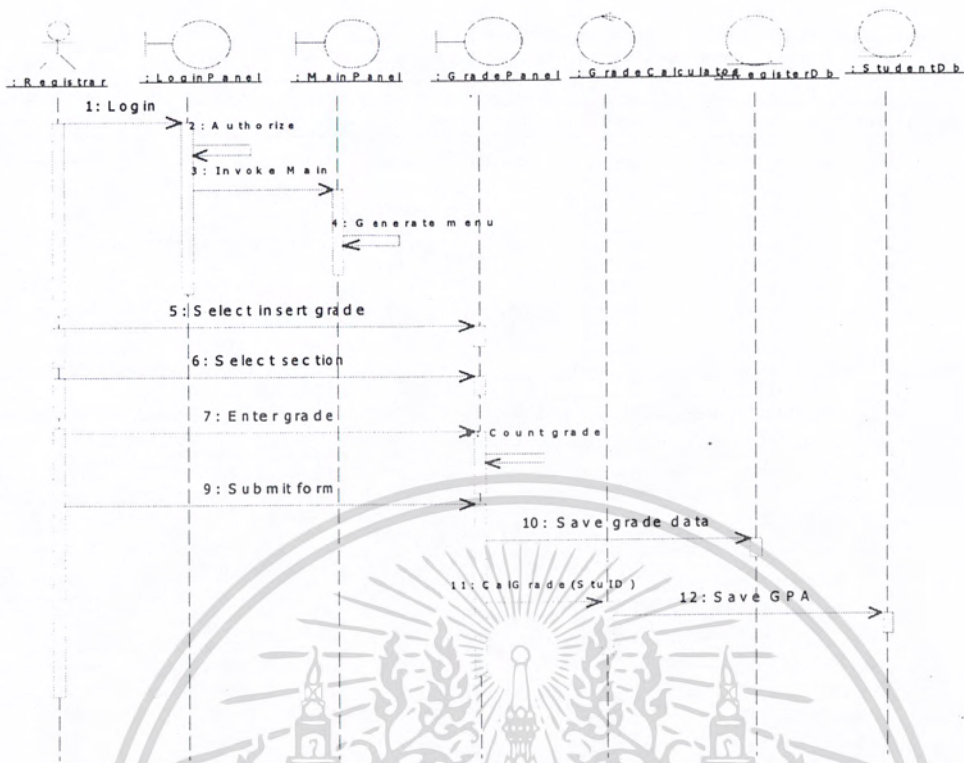
Collaboration Diagram/Accept & cancel course



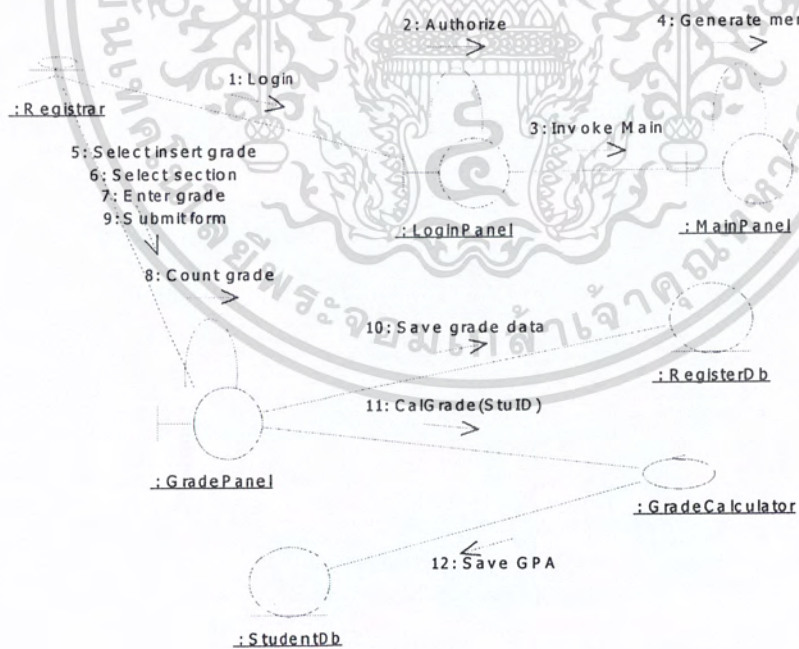
Sequence Diagram/View/Print report



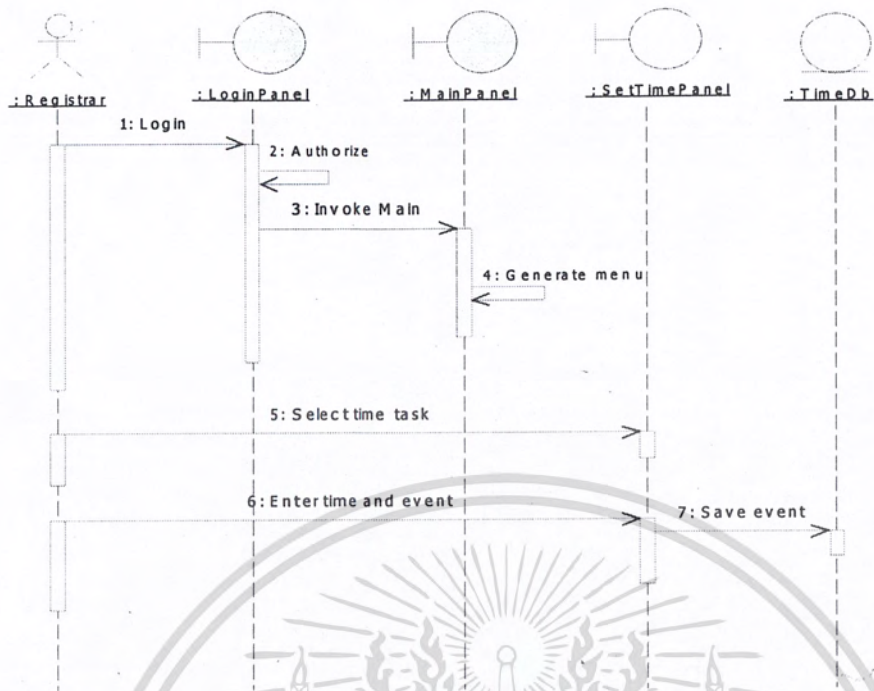
Collaboration Diagram/View/Print report



Sequence Diagram/Insert grades



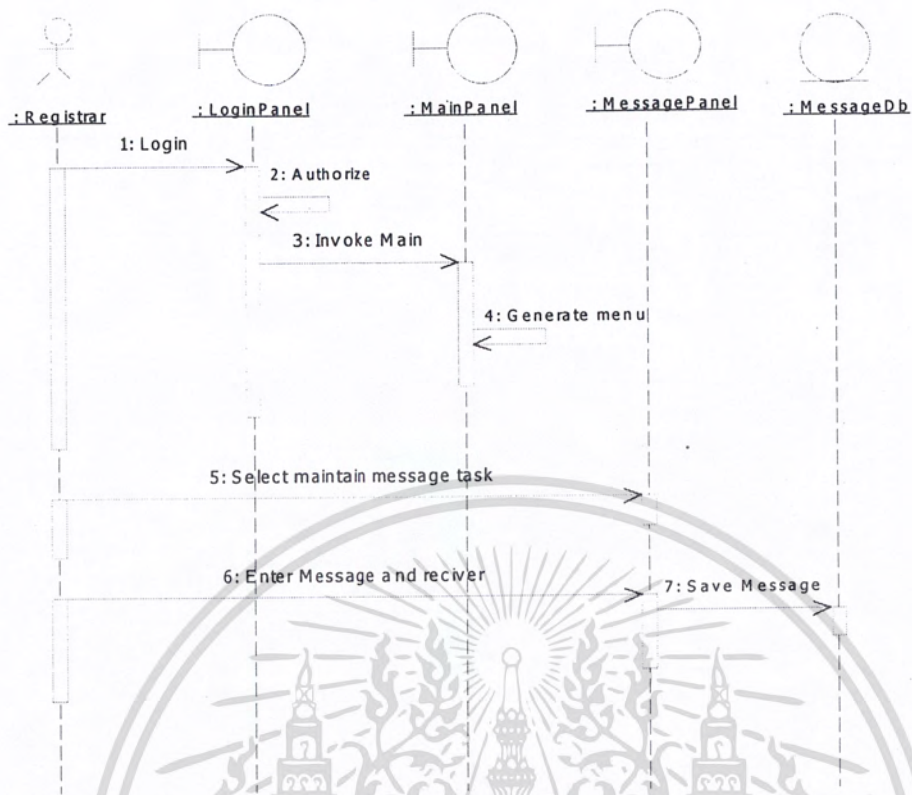
Collaboration Diagram/Insert grades



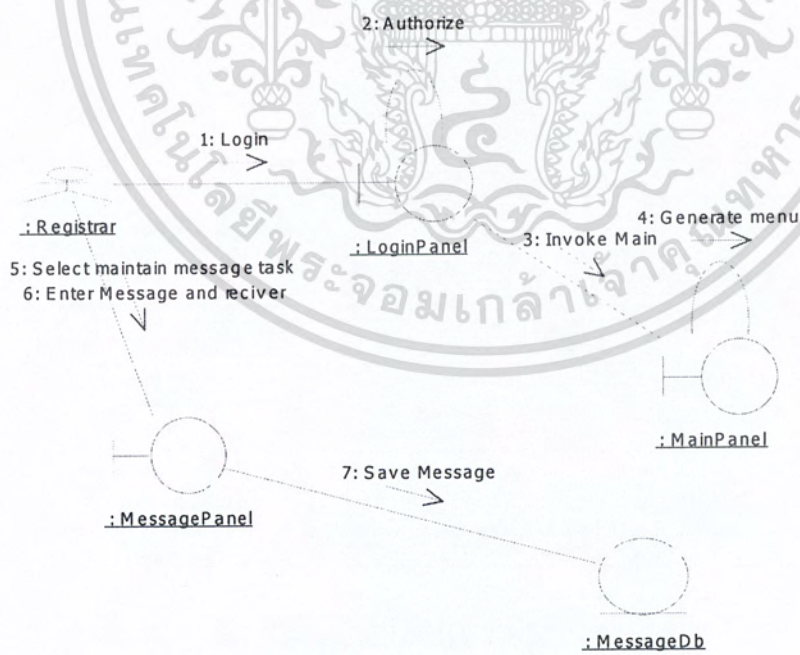
Sequence Diagram/Set time schedule



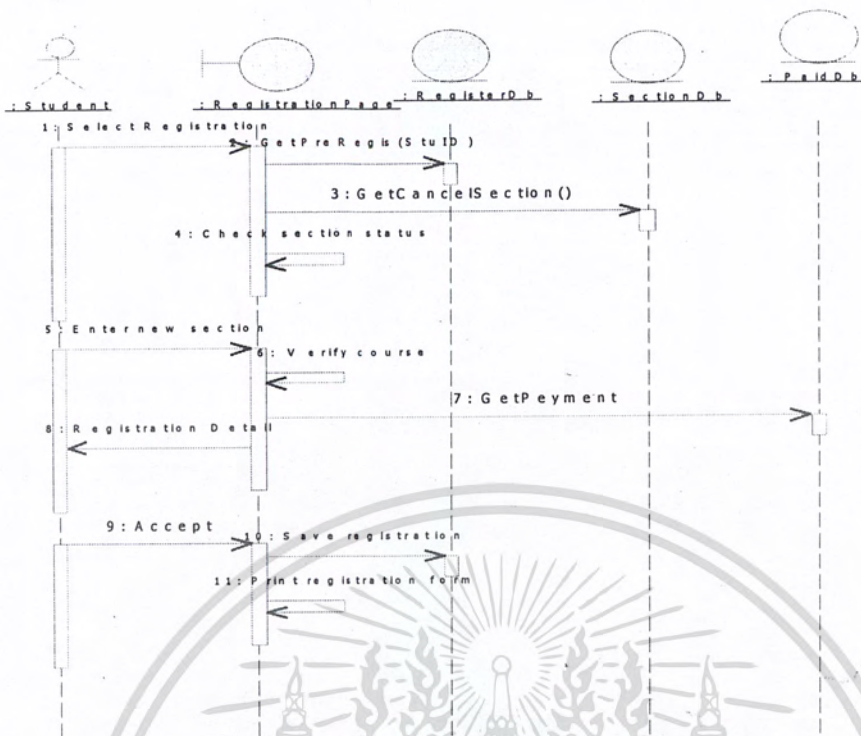
Collaboration Diagram/Set time schedule



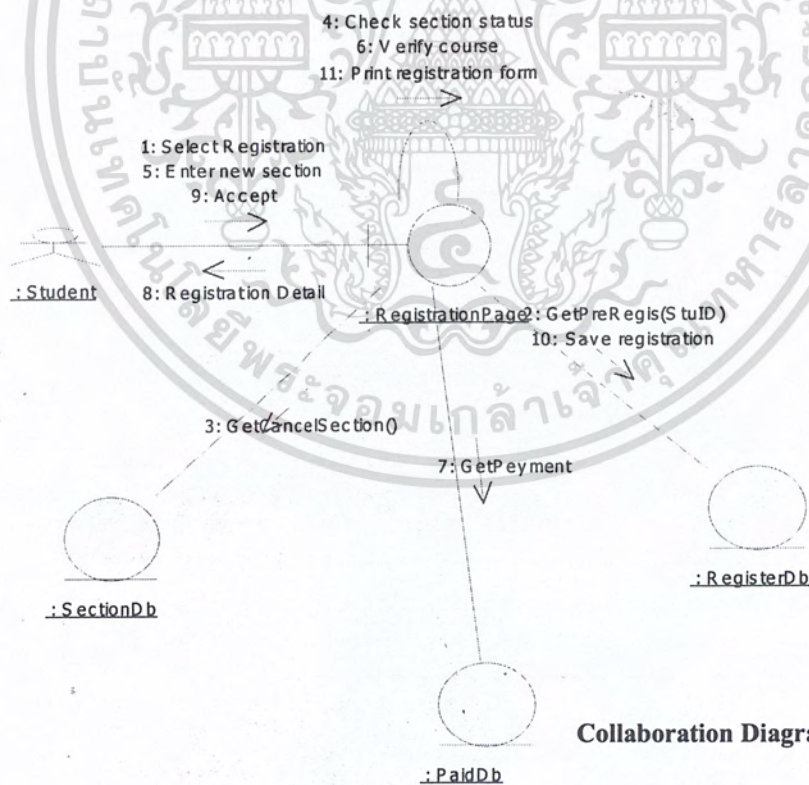
Sequence Diagram/Maintain message



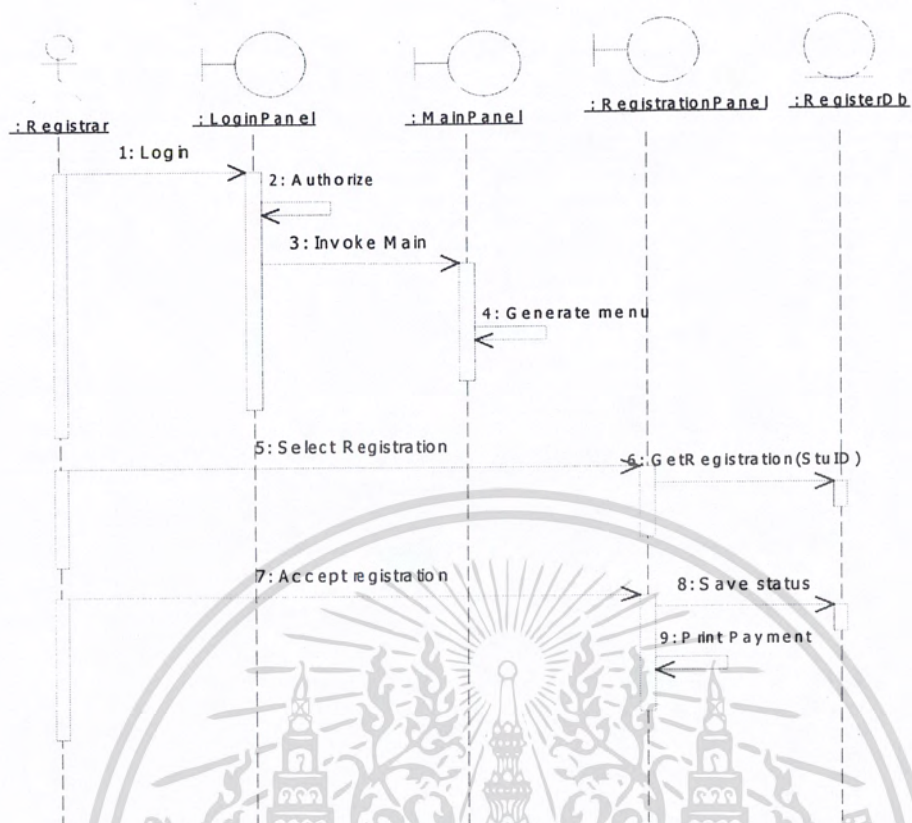
Collaboration Diagram/Maintain message



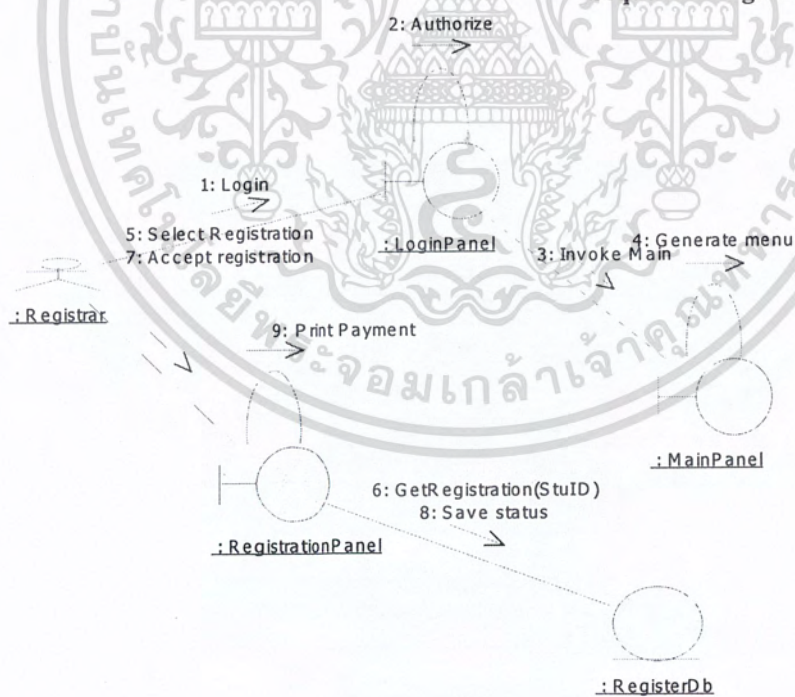
Sequence Diagram/Registration



Collaboration Diagram/Registration

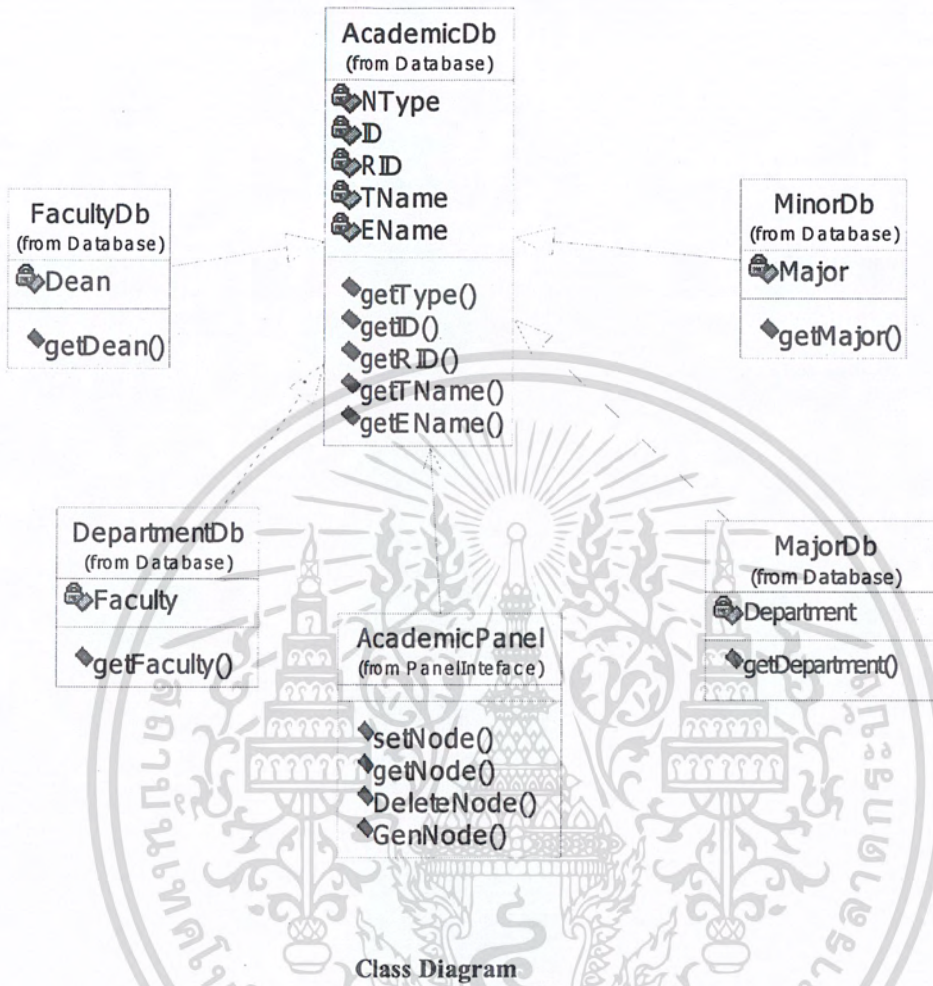


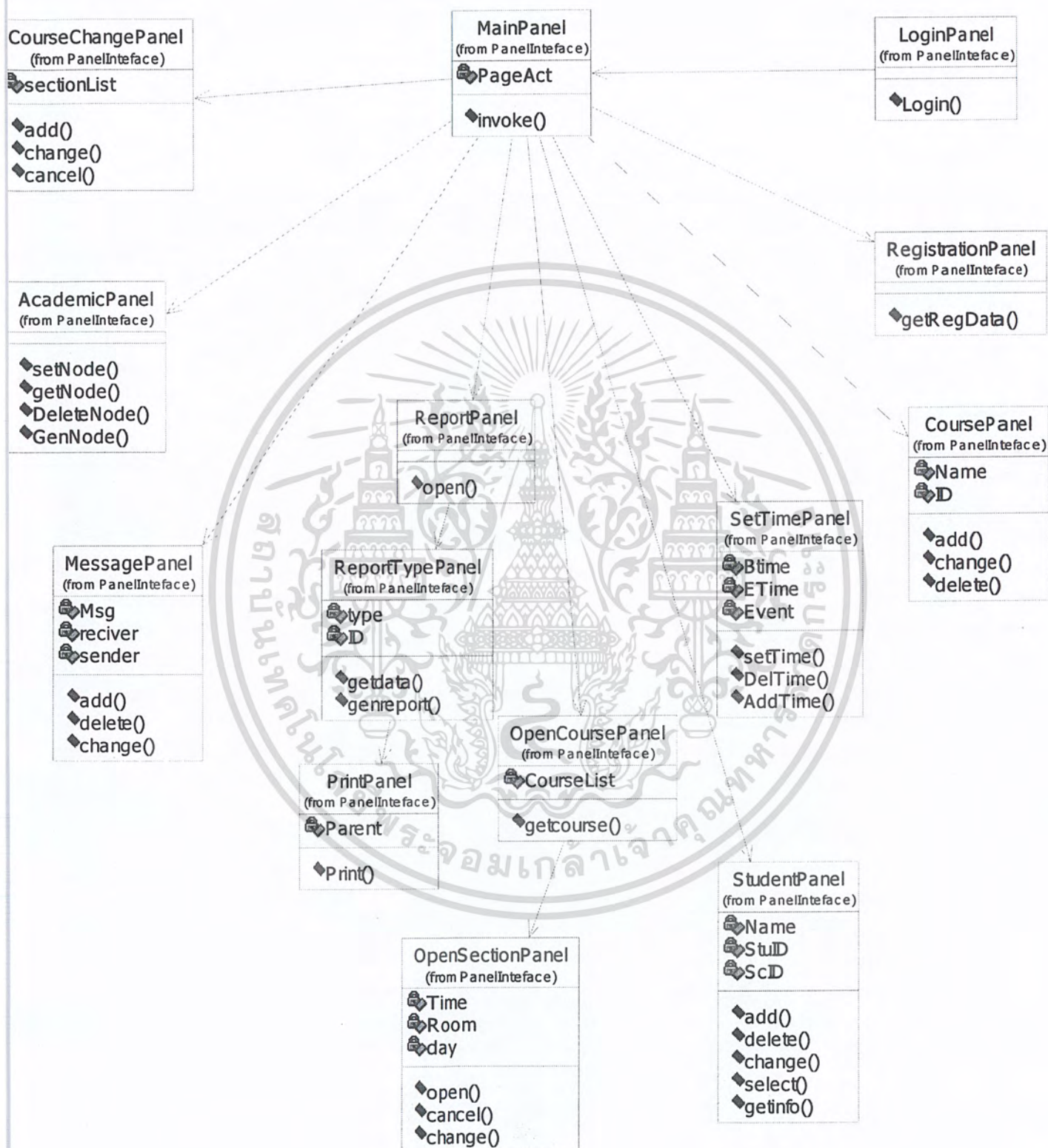
Sequence Diagram/AcceptRegistration



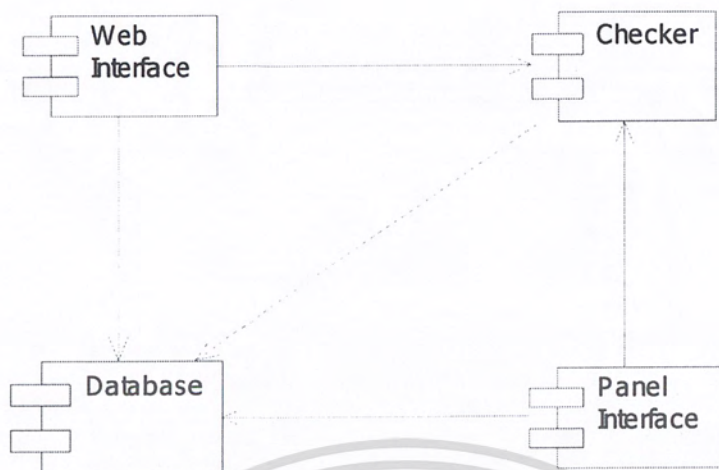
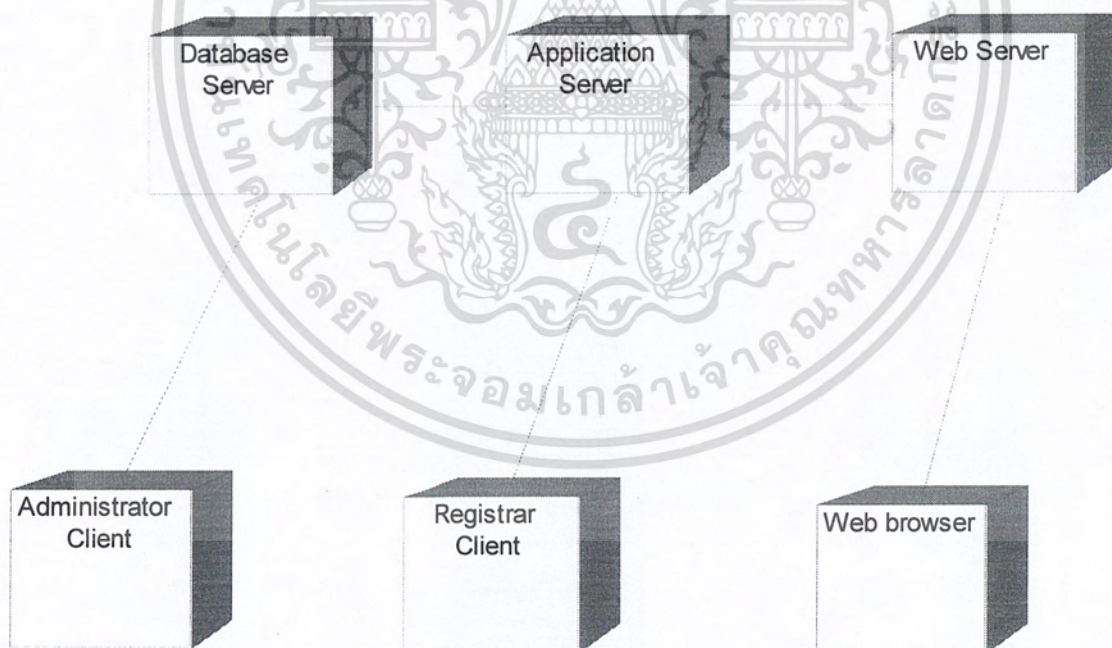
Collaboration Diagram/AcceptRegistration

AcademicDb
(from Database)





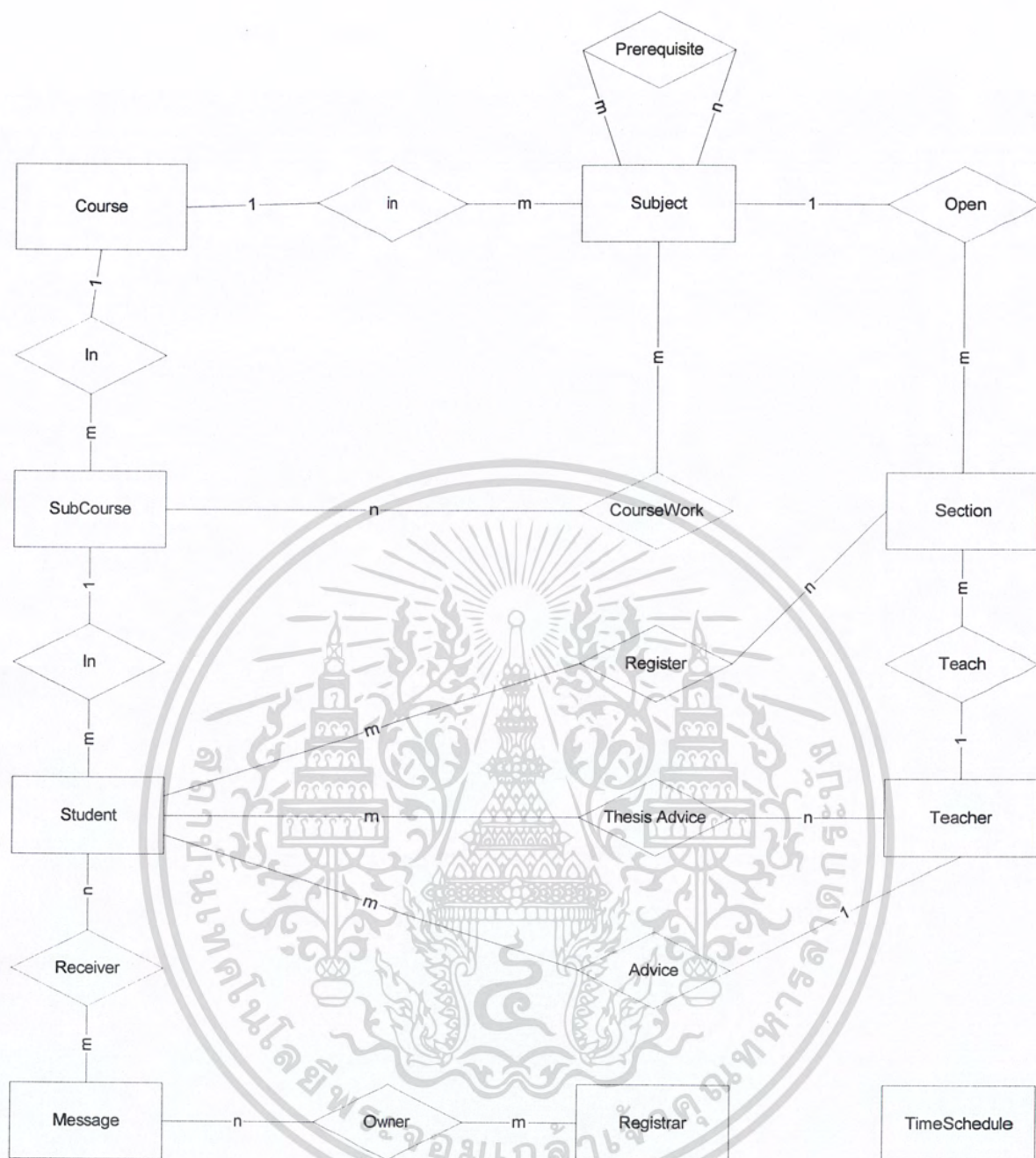
Class Diagram

**Component Diagram****Deployment Diagram**

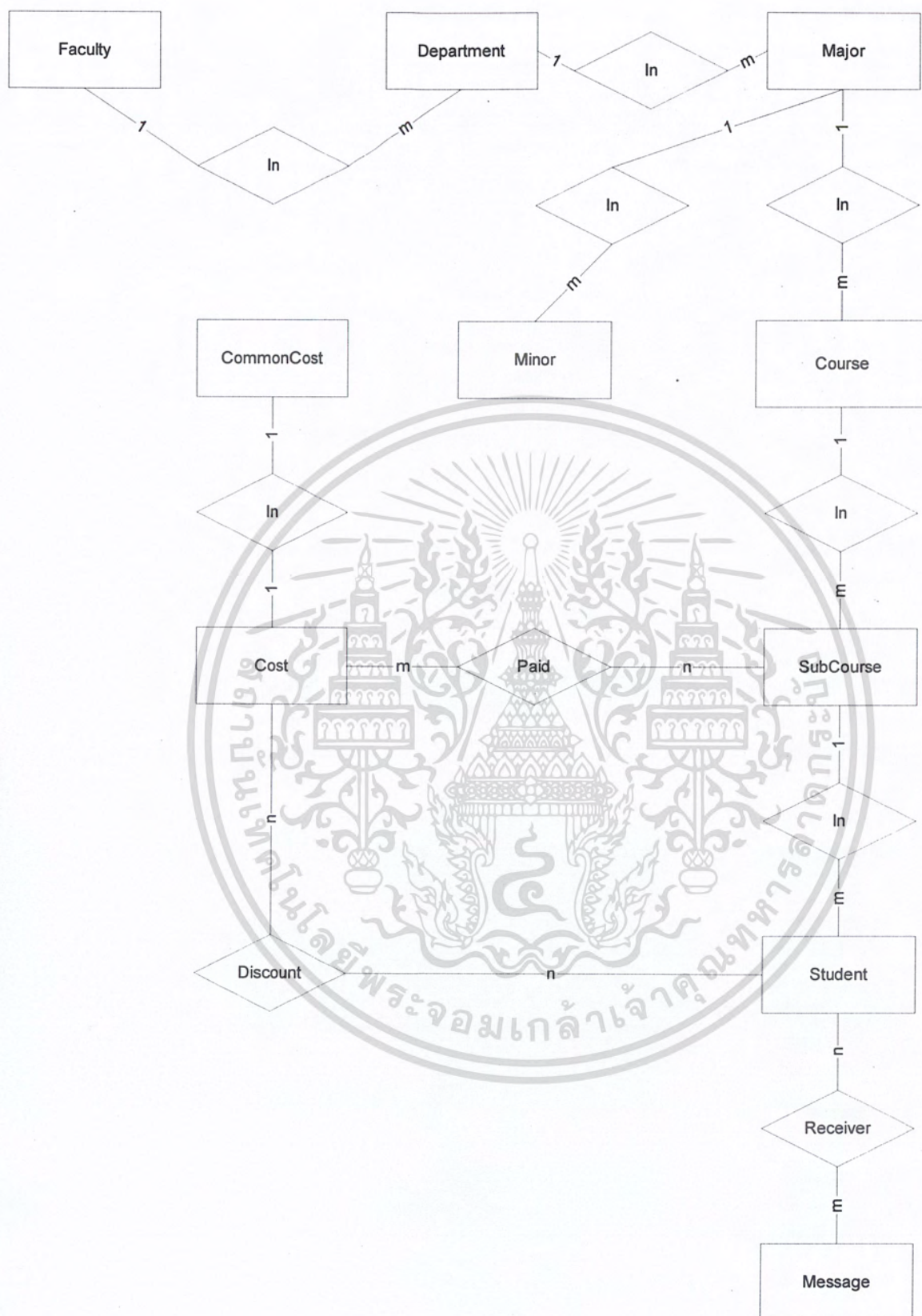


ภาคผนวก ข

ER Diagram และโครงสร้างตารางฐานข้อมูล (Table Schema)



ER-Diagram ส่วนที่ 1



ER-Diagram ส่วนที่ 2

โครงสร้างตารางฐานข้อมูล (Table Schema)

Faculty		
FacID	VARCHAR(2)	รหัสคณะ(PK)
EName	VARCHAR(50)	ชื่อคณะภาษาอังกฤษ
TName	VARCHAR(50)	ชื่อคณะภาษาไทย
DeanPE	VARCHAR(30)	คณบดีภาษาอังกฤษ
DeanNE	VARCHAR(50)	ชื่อคณะภาษาอังกฤษ
DeanFE	VARCHAR(50)	นามสกุลคณะภาษาอังกฤษ
DeanPT	VARCHAR(30)	คณบดีภาษาไทย
DeanNT	VARCHAR(50)	ชื่อคณะภาษาไทย
DeanFT	VARCHAR(50)	นามสกุลคณะภาษาไทย

Department		
DepID	SMALLINT	รหัสภาควิชา(PK)(Gen)
DepRID	VARCHAR(2)	รหัสอ้างอิงภาควิชา
EName	VARCHAR(50)	ชื่อภาควิชาภาษาอังกฤษ
TName	VARCHAR(50)	ชื่อภาควิชาภาษาไทย
FacID	VARCHAR(2)	รหัสคณะ (FK)

Major		
MaID	SMALLINT	รหัสสาขาวิชา(PK)(Gen)
MaRID	VARCHAR(2)	รหัสอ้างอิงสาขาวิชา
EName	VARCHAR(70)	ชื่อสาขาวิชาภาษาอังกฤษ
TName	VARCHAR(70)	ชื่อสาขาวิชาภาษาไทย
DepID	SMALLINT	รหัสภาควิชา (FK)

Minor		
MiID	SMALLINT	รหัสสาขาวิชา(PK)(Gen)
MiRID	VARCHAR(2)	รหัสอ้างอิงสาขาวิชา
EName	VARCHAR(70)	ชื่อสาขาวิชาภาษาอังกฤษ
TName	VARCHAR(70)	ชื่อสาขาวิชาภาษาไทย
MaID	SMALLINT	รหัสสาขาวิชา (FK)

Teacher		
TchID	VARCHAR(4)	รหัสอาจารย์(PK)
PreNE	VARCHAR(20)	คำนำหน้าชื่อภาษาอังกฤษ
NameE	VARCHAR(50)	ชื่อภาษาอังกฤษ
FamilyE	VARCHAR(50)	นามสกุลภาษาอังกฤษ
PreNE	VARCHAR(20)	คำนำหน้าชื่อภาษาไทย
NameE	VARCHAR(50)	ชื่อภาษาไทย
FamilyE	VARCHAR(50)	นามสกุลภาษาไทย
Sex	VARCHAR(1)	เพศ(M,F)

Student		
StuID	VARCHAR(8)	รหัสนักศึกษา(PK)
PreNE	VARCHAR(20)	คำนำหน้าชื่อภาษาอังกฤษ
NameE	VARCHAR(50)	ชื่อภาษาอังกฤษ
FamilyE	VARCHAR(50)	นามสกุลภาษาอังกฤษ
PreNT	VARCHAR(20)	คำนำหน้าชื่อภาษาไทย
NameT	VARCHAR(50)	ชื่อภาษาไทย
FamilyT	VARCHAR(50)	นามสกุลภาษาไทย
NickE	VARCHAR(20)	ชื่อเล่นภาษาอังกฤษ
NickT	VARCHAR(20)	ชื่อเล่นภาษาไทย
InDate	DATE	วันที่เข้าศึกษา
OutDate	DATE	วันที่จบการศึกษา
InTerm	VARCHAR(1)	ภาคเรียนที่เข้าศึกษา
InYear	VARCHAR(4)	ปีที่เข้าศึกษา
Location	VARCHAR(20)	ศูนย์ที่เรียน(ลาดกระบัง,อุดร ..)
Birthday	DATE	วันเกิด
Sex	VARCHAR(1)	เพศ(M,F)
Addr1	LONG VARCHAR	ที่อยู่ปัจจุบัน
LiveWith	VARCHAR(30)	อาศัยอยู่กับ
Addr2	LONG VARCHAR	ที่อยู่ตามทะเบียนบ้าน
Tel1	VARCHAR(20)	เบอร์โทรศัพท์ปัจจุบัน
Tel2	VARCHAR(20)	เบอร์โทรศัพท์ที่ภูมิลำเนา
Mobile	VARCHAR(20)	โทรศัพท์เคลื่อนที่

Student(ต่อ)		
Email	VARCHAR(40)	Email ของนักศึกษา
Religion	VARCHAR(10)	ศาสนา
Origin	VARCHAR(20)	เชื้อชาติ
Citizen	VARCHAR(20)	สัญชาติ
Work	VARCHAR(20)	อาชีพของนักศึกษา
Income	DECIMAL(7,0)	รายได้
OldSch	VARCHAR(100)	สถานศึกษาเดิม
OldGPA	VARCHAR(4)	ผลการศึกษาเฉลี่ยเดิม
OldYrOut	VARCHAR(4)	ปีที่จบการศึกษาจากสถานศึกษาเดิม
OldDegree	VARCHAR(100)	ปริญญาที่ได้จากสถานศึกษาเดิม
OldSDegree	VARCHAR(10)	ชื่อย่อปริญญาที่ได้จากสถานศึกษาเดิม
OldFac	VARCHAR(50)	คณะสถานศึกษาเดิม
OldMajor	VARCHAR(50)	สาขาสถานศึกษาเดิม
OldHonor	VARCHAR(10)	เกียรตินิยมอันดับที่ได้จากสถานศึกษาเดิม
OldSch	SMALLINT	จำนวนปีของหลักสูตรที่เรียนจากสถานศึกษาเดิม
OldUse	SMALLINT	จำนวนปีที่ใช้เรียนจากสถานศึกษาเดิม
Amin	VARCHAR(100)	ชื่อนามสกุลผู้ปกครอง
AmRelate	VARCHAR(10)	ความสัมพันธ์กับนักศึกษา
AmAddr	LONG VARCHAR	ที่อยู่ของผู้ปกครอง
AmTel	VARCHAR(20)	เบอร์โทรศัพท์ของผู้ปกครอง
Contact	VARCHAR(100)	ชื่อนามสกุลผู้ที่ติดต่อถึงนักศึกษาได้สะดวก
CoRelate	VARCHAR(10)	ความสัมพันธ์กับนักศึกษา
CoWorkAddr	LONG VARCHAR	ที่อยู่ของผู้ติดต่อได้
CoTel	VARCHAR(20)	เบอร์โทรศัพท์ของผู้ที่ติดต่อได้
CoPosition	VARCHAR(20)	ตำแหน่งของผู้ที่ติดต่อได้
CoLevel	VARCHAR(10)	ระดับของผู้ที่ติดต่อได้
Father	VARCHAR(100)	ชื่อนามสกุลบิดา
FWork	VARCHAR(20)	อาชีพของบิดา
Mother	VARCHAR(100)	ชื่อนามสกุลมารดา
MWork	VARCHAR(20)	อาชีพของมารดา
MarryS	VARCHAR(10)	สถานะการสมรส

Student(ต่อ)		
Spouse	VARCHAR(100)	ชื่อนามสกุลภรรยา
SWork	VARCHAR(20)	อาชีพภรรยา
Children	SMALLINT	จำนวนบุตร
PermisStatus	VARCHAR(1)	สถานะสิทธิการลงทะเบียน
Password	VARCHAR(20)	รหัสผ่านของนักศึกษา
GPA	VARCHAR(4)	เกรดเฉลี่ยปัจจุบัน
TchID	VARCHAR(4)	รหัสอาจารย์ที่ปรึกษา(FK)
ScID	SMALLINT	รหัสหลักสูตรย่อยที่สังกัด(FK)

Subject		
SubID	VARCHAR(8)	รหัสวิชา(PK)
EName	VARCHAR(85)	ชื่อวิชาภาษาอังกฤษ
TName	VARCHAR(85)	ชื่อวิชาภาษาไทย
SType	VARCHAR(30)	กลุ่มวิชา
Credit	DECIMAL(2)	หน่วยกิตรวม
Lecture	DECIMAL(2)	เวลาบรรยาย
Practice	DECIMAL(2)	เวลาปฏิบัติ
Research	DECIMAL(2)	เวลาด้านคว้า
OutlineE	LONG VARCHAR	คำอธิบายรายวิชาภาษาอังกฤษ
OutlineT	LONG VARCHAR	คำอธิบายรายวิชาภาษาไทย
CoID	SMALLINT	รหัสหลักสูตร(FK) (PK)

Section		
SecID	SMALLINT	รหัสตอน(PK)(Gen)
SubID	VARCHAR(8)	รหัสวิชา(FK)
ScID	SMALLINT	รหัสหลักสูตรย่อย(0:ทั่วไป,อื่นๆ)
CoID	SMALLINT	รหัสหลักสูตร(FK)
TchID	VARCHAR(4)	รหัสผู้สอน(FK)
Year	VARCHAR(1)	ชั้นปีที่เรียน
Sem	VARCHAR(1)	ภาคที่เรียน
StdBTime	TIME	เวลาเริ่มเรียน
StdEtime	TIME	เวลาเลิกเรียน

Section(ต่อ)		
StdRoom	VARCHAR(8)	ห้องเรียน
StdDay	VARCHAR(10)	วันเรียน
ExDate	DATE	วันสอบ
ExBTime	TIME	เวลาเริ่มสอบ
ExETime	TIME	เวลาเลิกสอบ
ExRoom	VARCHAR(8)	ห้องสอบ
MaxStu	DECIMAL(3,0)	จำนวนนักศึกษาที่รับสูงสุด
CurStu	DECIMAL(3,0)	จำนวนนักศึกษาที่ลงทะเบียน
Sec	SMALLINT	ตอนที่
Status	VARCHAR(1)	Active/Cancel

Course		
CoID	SMALLINT	รหัสหลักสูตร(PK)(Gen)
Ename	VARCHAR(100)	ชื่อหลักสูตรภาษาอังกฤษ
Tname	VARCHAR(100)	ชื่อหลักสูตรภาษาไทย
FDNameT	VARCHAR(100)	ชื่อเต็มปริญญาภาษาไทย
FDNameE	VARCHAR(100)	ชื่อเต็มปริญญาภาษาอังกฤษ
SDNameT	VARCHAR(50)	ชื่อย่อปริญญาภาษาไทย
SDNameE	VARCHAR(50)	ชื่อย่อปริญญาภาษาอังกฤษ
Degree	VARCHAR(1)	ปริญญาโท / เอก (M:Master,D:Doctor)
MaID	SMALLINT	รหัสสาขาวิชา(FK)

SubCourse		
ScID	SMALLINT	รหัสหลักสูตรย่อย(PK)(Gen)
Ename	VARCHAR(100)	ชื่อหลักสูตรภาษาอังกฤษ
Tname	VARCHAR(100)	ชื่อหลักสูตรภาษาไทย
Ctype	VARCHAR(15)	ชนิดหลักสูตร (สามัญ,ทดลองเรียน)
CType2	VARCHAR(10)	ประเภทหลักสูตร (ปกติ,พิเศษ)
CoID	SMALLINT	รหัสหลักสูตร(FK)

Cost		
CID	SMALLINT	รหัสค่าใช้จ่าย(PK)(Gen)
Name	VARCHAR(50)	ชื่อค่าใช้จ่าย
Value	DECIMAL(8,2)	จำนวนเงิน

CommonCost		
CCID	SMALLINT	รหัสค่าใช้จ่ายปกติ(PK)(Gen)
CID	SMALLINT	รหัสค่าใช้จ่าย(FK)

Registrar		
RegID	VARCHAR(4)	รหัสเจ้าหน้าที่(Unique)
FacID	VARCHAR(2)	คณะที่สังกัด
EName	VARCHAR(50)	ชื่อภาษาอังกฤษ
Tname	VARCHAR(50)	ชื่อภาษาไทย
Position	VARCHAR(50)	ตำแหน่ง
UserName	VARCHAR(20)	UserName (PK)
Password	VARCHAR(20)	รหัสผ่าน
Permission	VARCHAR(1)	สิทธิ(N:Normal,S:Special)

TimeSch		
TiID	SMALLINT	รหัสกำหนดเวลา(PK)(Gen)
Name	VARCHAR(50)	ชื่อกำหนดเวลา
BeDate	DATE	วันที่เริ่มต้น
ExDate	DATE	วันที่สิ้นสุด

Message		
MsgID	SMALLINT	รหัสข้อความ(PK)(Gen)
SMsg	VARCHAR(4)	ผู้ส่งข้อความ
RMsg	VARCHAR(8)	ผู้รับข้อความ
MSg	LONG VARCHAR	ข้อความ
PostDate	DATE	วันที่ส่งข้อความ
DelDate	DATE	วันที่ลบข้อความ

Prerequisite		
CoID	SMALLINT	รหัสหลักสูตร(PK)
SubID	VARCHAR(8)	รหัสรายวิชา(PK)
PreID	VARCHAR(8)	รหัสรายวิชาที่ต้องเรียนก่อน(PK)

CourseWork		
ScID	SMALLINT	รหัสหลักสูตรย่อย(PK)
SubID	VARCHAR(8)	รหัสรายวิชา(PK)
EName	VARCHAR(85)	ชื่อวิชาภาษาอังกฤษ
TName	VARCHAR(85)	ชื่อวิชาภาษาไทย
Credit	VARCHAR(10)	หน่วยกิต และ เวลาเรียน
Year	SMALLINT	ปีที่
Sem	SMALLINT	ภาคที่

Paid		
ScID	SMALLINT	รหัสหลักสูตรย่อย(PK)
CID	SMALLINT	รหัสค่าใช้จ่าย(PK)

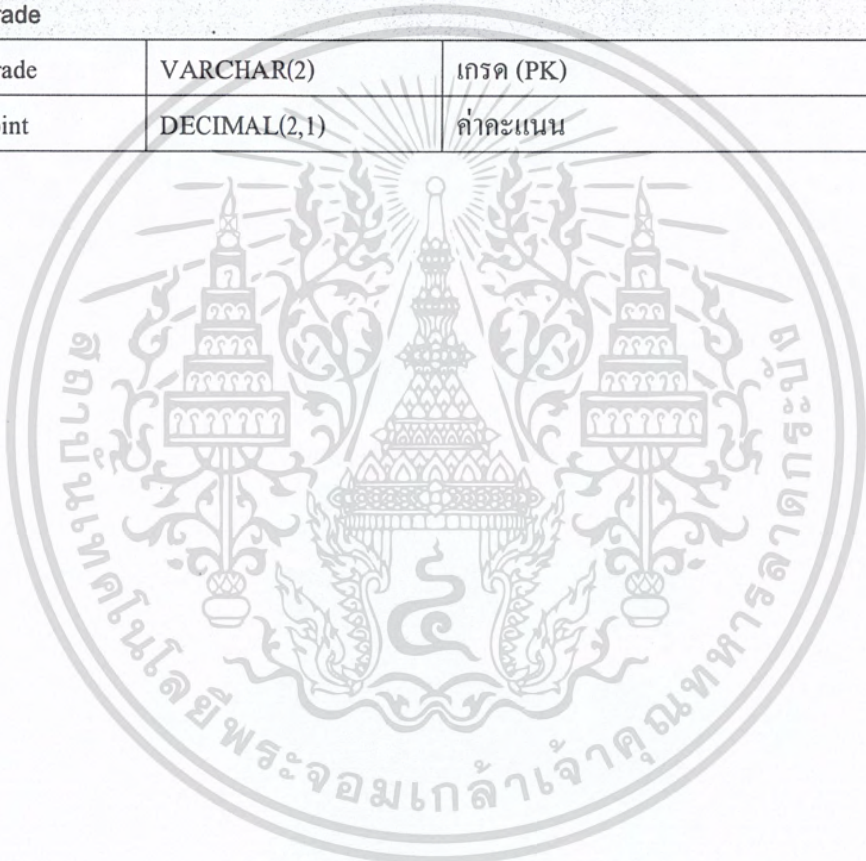
Register		
StuID	VARCHAR(8)	รหัสนักศึกษา(PK)
SecID	VARCHAR(8)	รหัสตอน(PK)
Year	VARCHAR(4)	ปีการศึกษา
Sem	SMALLINT	ภาคการศึกษา
Grade	VARCHAR(1)	เกรดที่ได้
Type	VARCHAR(1) (C,O)	ชนิดการลงทะเบียน(ปกติ,Ordit)
Status	VARCHAR(1) (P,C)	สถานะการลงทะเบียน(ล่วงหน้า,ปกติ)

ThesisAdv		
StuID	VARCHAR(8)	รหัสนักศึกษา(PK)
TchID	VARCHAR(4)	รหัสอาจารย์ที่ปรึกษางานวิจัย(PK)
TName	VARCHAR(100)	ชื่องานวิจัยภาษาไทย
EName	VARCHAR(100)	ชื่องานวิจัยอังกฤษ
Order	SMALLINT	ลำดับอาจารย์คนที่

Owner		
RegID	VARCHAR(4)	รหัสเจ้าหน้าที่ทะเบียนผู้ประกาศ(PK)
MsgID	SMALLINT	รหัสข้อความ(PK)

Reciver		
StuID	VARCHAR(8)	รหัสนักศึกษาผู้รับข้อความ(PK)
MsgID	SMALLINT	รหัสข้อความ(PK)

Grade		
Grade	VARCHAR(2)	เกรด (PK)
Point	DECIMAL(2,1)	ค่าคะแนน



ภาคผนวก ก

แบบฟอร์มคำร้องต่าง ๆ ของนักศึกษาระดับบัณฑิตศึกษา





บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง แบบฟอร์มคำร้องทั่วไป

วันที่.....เดือน.....พ.ศ.....

เรื่อง

เรียน คณบดีบัณฑิตวิทยาลัย

ข้าพเจ้า (นาย,นาง,นางสาว,อื่นๆ)

รหัสประจำตัว.....นักศึกษาชั้นปีที่.....สาขาวิชา.....ภาควิชา

แขนงวิชา.....คณะ.....สถานที่ติดต่อ.....

โทร./มือถือ

อีเมล.....

มีความประสงค์.....

จึงเรียนมาเพื่อโปรดพิจารณาดำเนินการต่อไป

ขอแสดงความนับถือ

ลงชื่อ

(.....)

ความเห็นอาจารย์ที่ปรึกษา

ลงชื่อ

(.....)

ความเห็นของเจ้าหน้าที่	คำสั่ง

ผู้ยื่นคำร้อง รับทราบผลคำร้อง

ลงชื่อ

(.....)



บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
แบบฟอร์มคำร้องขอ TRANSCRIPT

วันที่.....เดือน.....พ.ศ.....

เรียน คณบดีบัณฑิตวิทยาลัย

ข้าพเจ้า (นาย,นาง,นางสาว,อื่น ๆ.....) ชื่อ - นามสกุล.....
(Mr.,Mrs.,Miss,Other.....)(เขียนภาษาอังกฤษตัวพิมพ์ใหญ่)

วัน,เดือน,ปีเกิด.....(เขียนภาษาอังกฤษ ห้ามเขียนด้วยย่อ คัด ค.ศ.โดย พ.ศ.เกิดลบด้วย 543)

รหัสประจำตัวนักศึกษา.....หลักสูตร.....แผน.....

คณะ.....สาขาวิชา.....

แขนงวิชา / วิชาเอก.....

☐ สำเร็จการศึกษาเมื่อภาคเรียนที่.....ปีการศึกษา.....

☐ ยังไม่สำเร็จการศึกษา

มีความประสงค์ขอ Transcript เพื่อวัตถุประสงค์.....

จำนวน.....ชุด (ชุดละ 20 บาท) พร้อมทั้งแนบรูปถ่ายจำนวน.....รูป

หลักฐานประกอบการขอ Transcript

กรณีนักศึกษาสำเร็จการศึกษา

1. รูปถ่ายขนาด 1 นิ้ว ถ่ายมาแล้วไม่เกิน 6 เดือน สวมเสื้อครุย (ภาพสีหรือขาวดำ) 1 ชุดต่อ 1 รูป
2. ใบรับรองวิทยานิพนธ์ (ถ่ายเอกสารแล้วให้ฝ่ายวิจัยฯ รับรองเกรด)

กรณีนักศึกษายังไม่สำเร็จการศึกษา

1. รูปถ่ายขนาด 1 นิ้ว ถ่ายมาแล้วไม่เกิน 6 เดือน (ภาพสีหรือขาวดำ) 1 ชุดต่อ 1 รูป
 - นักศึกษาชาย สวมเสื้อเชิ้ตสีขาวเกลี้ยงไม่มีลายและผูกเนคไทสีดำ หรือสีกรมท่าไม่มีลาย เท่านั้น
 - นักศึกษาหญิง สวมเสื้อเชิ้ตสีขาวเกลี้ยงไม่มีลาย แขนสั้น เท่านั้น

จึงเรียนมาเพื่อโปรดพิจารณาดำเนินการ

ลงชื่อ.....

ที่อยู่ติดต่อได้.....

โทรศัพท์(บ้าน).....โทรศัพท์(ที่ทำงาน).....มือถือ.....

อีเมล.....

หมายเหตุ : 1. ผู้ขอจะมารับ Transcript ได้หลังจากวันที่ขออย่างน้อย 10 วันทำการ

2. ผู้ขอต้องกรอกข้อความข้างต้นด้วยตัวบรรจงให้ชัดเจน หากผิดพลาดบัณฑิตวิทยาลัยจะไม่รับผิดชอบ

3. ผู้ขอต้องยื่นเอกสารประกอบการขอ Transcript ให้ครบถ้วน มิฉะนั้นบัณฑิตวิทยาลัยจะไม่พิจารณาดำเนินการ

4. กรุณานำบัตรนักศึกษามาด้วยทุกครั้ง เพื่อติดต่อขอรับ Transcript



บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
แบบฟอร์มคำร้องขอหนังสือรับรองผลการศึกษา

วันที่.....เดือน.....พ.ศ.....

เรียน คณะบดีบัณฑิตวิทยาลัย

ข้าพเจ้า (นาย,นาง,นางสาว,อื่นๆ)

รหัสประจำตัว.....หลักสูตร ☐ วศ.ด. ☐ คอ.ด. ☐ วศ.ม. ☐ ผ.ม. ☐ สด.ม. ☐ วท.ม. ☐ คอ.ม.

แผน.....นักศึกษาชั้นปีที่.....สาขาวิชา.....ภาควิชา

แขนงวิชาคณะ.....

มีความประสงค์ขอหนังสือรับรองการศึกษา ☐ ทุกภาคการศึกษา (ชุดละ 20 บาท) ☐ เฉพาะภาคเรียนที่.....
ปีการศึกษา.....จำนวน.....ชุด เพื่อวัตถุประสงค์.....

จึงเรียนมาเพื่อโปรดพิจารณาดำเนินการต่อไป

ขอแสดงความนับถือ

ลงชื่อ.....
(.....)

ที่อยู่ติดต่อได้.....

โทรศัพท์(บ้าน).....โทรศัพท์(ที่ทำงาน)มือถือ.....

อีเมล.....



บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
แบบฟอร์มคำร้องขอหนังสือรับรองผลการศึกษา

วันที่.....เดือน.....พ.ศ.....

เรียน คณบดีบัณฑิตวิทยาลัย

ข้าพเจ้า (นาย,นาง,นางสาว,อื่นๆ)

รหัสประจำตัว.....หลักสูตร ☐ วศ.ด. ☐ คอ.ด. ☐ วศ.ม. ☐ ผ.ม. ☐ สด.ม. ☐ วท.ม. ☐ คอ.ม.

แผน.....นักศึกษาชั้นปีที่.....สาขาวิชา.....ภาควิชา

แขนงวิชาคณะ.....

มีความประสงค์ขอหนังสือรับรองการศึกษา ☐ ทุกภาคการศึกษา (ชุดละ 20 บาท) ☐ เฉพาะภาคเรียนที่.....

ปีการศึกษา.....จำนวน.....ชุด เพื่อวัตถุประสงค์.....

จึงเรียนมาเพื่อโปรดพิจารณาดำเนินการต่อไป

ขอแสดงความนับถือ

ลงชื่อ
(.....)

ที่อยู่ติดต่อได้.....

โทรศัพท์(บ้าน).....โทรศัพท์(ที่ทำงาน)มือถือ.....

อีเมล.....



บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
แบบฟอร์มคำร้องขอลาพักการศึกษา

วันที่.....เดือน.....พ.ศ.....

เรียน คณบดีบัณฑิตวิทยาลัย

ข้าพเจ้า (นาย,นาง,นางสาว,อื่นๆ)

รหัสประจำตัว..... เป็นนักศึกษาชั้นปีที่..... ปีการศึกษา.....หลักสูตร.....

แผน.....สาขาวิชา.....ภาควิชา.....

แขนงวิชา.....คณะ.....

มีความประสงค์ขอลาพักการศึกษาในภาคเรียนที่.....ปีการศึกษา.....เนื่องจาก.....

.....
.....
.....

จึงเรียนมาเพื่อโปรดพิจารณา

ขอแสดงความนับถือ

ลงชื่อ.....

(.....)

(ผู้ยื่นคำร้อง)

ที่อยู่ติดต่อได้.....

โทรศัพท์(บ้าน).....โทรศัพท์(ที่ทำงาน).....มือถือ.....

อีเมล.....

(1) ความเห็นของอาจารย์ที่ปรึกษา

- ☐ เห็นควรให้ลาพักการศึกษาต่อไปได้
- ☐ ไม่เห็นควรให้ลาพักการศึกษา เนื่องจาก.....

.....

.....

.....

.....

.....

.....

(3) คำสั่ง (คณบดีบัณฑิตวิทยาลัย)

- ☐ อนุมัติลาพักการศึกษาได้และดำเนินการต่อไป
- ☐ ไม่อนุมัติให้ลาพักการศึกษา เนื่องจาก.....

.....

.....

.....

.....

.....

.....

ลงชื่อ.....

(.....)

..... / /

ลงชื่อ.....

(.....)

..... / /

(2) ความเห็นของเจ้าหน้าที่บัณฑิตวิทยาลัย

ได้ตรวจสอบเรียบร้อยแล้ว

- ☐ เห็นควรพิจารณาอนุมัติได้ตามระเบียบ
- ☐ ไม่เห็นควรพิจารณาอนุมัติ เนื่องจาก.....

.....

.....

- ☐ อื่นๆ.....

.....

.....

ลงชื่อ

(.....)

..... / /



บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
แบบฟอร์มคำร้องขอลาออก

วันที่.....เดือน.....พ.ศ.....

เรียน คณบดีบัณฑิตวิทยาลัย

ข้าพเจ้า (นาย,นาง,นางสาว,อื่นๆ).....
รหัสประจำตัว..... เป็นนักศึกษาชั้นปีที่..... ปีการศึกษา.....หลักสูตร.....
แผน.....สาขาวิชา.....ภาควิชา.....
แขนงวิชา.....คณะ.....

มีความประสงค์ขอลาออกจากการเป็นนักศึกษาของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เนื่องจาก.....

จึงเรียนมาเพื่อโปรดพิจารณา

ขอแสดงความนับถือ

ลงชื่อ.....

(.....)

(ผู้ยื่นคำร้อง)

ที่อยู่ติดต่อได้.....

โทรศัพท์(บ้าน).....โทรศัพท์(ที่ทำงาน).....มือถือ.....

อีเมล.....

(1) ความเห็นของอาจารย์ที่ปรึกษา

- ☐ เห็นควรให้ นศ. ลาออกได้
- ☐ ไม่เห็นควรให้ นศ. ลาออก เนื่องจาก.....
-

ลงชื่อ.....
(.....)
..... / /

(2) ความเห็นของเจ้าหน้าที่ที่เกี่ยวข้อง

- บรรณารักษ์ห้องสมุดของคณะฯ

- ☐ นศ. ไม่ได้ติดค้างหนังสือของห้องสมุด
- ☐ นศ. ติดค้างหนังสือของสำนักหอสมุดกลาง
จำนวน.....เล่ม

ลงชื่อ.....
(.....)
..... / /

- บรรณารักษ์สำนักหอสมุดกลาง

- ☐ นศ. ไม่ได้ติดค้างหนังสือของห้องสมุด
- ☐ นศ. ติดค้างหนังสือของสำนักหอสมุดกลาง
จำนวน.....เล่ม

ลงชื่อ.....
(.....)
..... / /

- เจ้าหน้าที่ฝ่ายทุนของการศึกษาของบัณฑิตวิทยาลัย

- ☐ นศ. ไม่มีภาระในการรับทุนการศึกษา
- ☐ นศ. มีภาระในการรับทุนการศึกษา.....
-

ลงชื่อ.....
(.....)
..... / /

(3) ความเห็นของเจ้าหน้าที่บัณฑิตวิทยาลัย

ได้ตรวจสอบเรียบร้อยแล้ว

- ☐ นศ. มีภาระในการรับทุนการศึกษา
- ☐ นศ. ติดค้างหนังสือของ.....
-

จำนวน เล่ม

- ☐ นศ. ไม่มีภาระหนี้สินใดๆ ต่อสถาบันฯ
ดังนั้นจึง

☐ เห็นควรพิจารณาอนุมัติให้ลาออก

☐ ไม่เห็นควรพิจารณาอนุมัติให้ลาออก
เนื่องจาก

.....

ลงชื่อ.....
(.....)
..... / /

(4) คำสั่ง (คณบดีบัณฑิตวิทยาลัย)

- ☐ อนุมัติให้ลาออกตามที่เสนอ
- ☐ ไม่อนุมัติให้ลาออก เนื่องจาก
-

ลงชื่อ.....
(.....)
..... / /



บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
แบบฟอร์มคำร้องขอปรับสภาพเป็นนักศึกษาสามัญ

วันที่.....เดือน.....พ.ศ.....

เรียน คณบดีบัณฑิตวิทยาลัย

ข้าพเจ้า (นาย,นาง,นางสาว,อื่นๆ).....
รหัสประจำตัว..... เป็นนักศึกษาชั้นปีที่..... ปีการศึกษา.....หลักสูตร.....
แผน.....สาขาวิชา.....ภาควิชา.....
แขนงวิชา.....คณะ.....

มีความประสงค์จะขอปรับสภาพเป็นนักศึกษาสามัญในภาคเรียนที่.....ปีการศึกษา.....ทั้งนี้ได้ปฏิบัติตามเงื่อนไข
ที่คณะฯ กำหนดแล้ว ดังรายละเอียดต่อไปนี้

1. ได้เรียนผ่านรายวิชาที่คณะฯ กำหนดแล้ว จำนวน วิชา
2. เทียบโอนรายวิชาได้ จำนวน วิชา
3. ได้เสนอบทความเรื่อง.....
ลงตีพิมพ์ในวารสาร.....
4. อื่นๆ.....
.....
.....

จึงเรียนมาเพื่อโปรดพิจารณา

ขอแสดงความนับถือ

ลงชื่อ.....
(.....)
(ผู้ยื่นคำร้อง)

พร้อมกันนี้ข้าพเจ้าได้ส่งหลักฐาน มาประกอบการพิจารณา ดังนี้

- ☐ บทความที่ได้รับการตีพิมพ์ (ถ่ายหน้าปกของวารสารแนบมาด้วย กรณีที่ได้รับการตีพิมพ์แล้ว)
- ☐ หนังสือการรับรองผลงานวิจัยทางวิชาการ (ในกรณีที่มีผู้ร่วมทำการวิจัยมากกว่า 2 คน)
- ☐ ใบตอบรับการลงตีพิมพ์พร้อมบทความ (กรณีที่ยังไม่ได้ลงตีพิมพ์)

ที่อยู่ติดต่อได้.....
โทรศัพท์(บ้าน).....โทรศัพท์(ที่ทำงาน).....มือถือ.....
อีเมล.....

(1) ความเห็นของอาจารย์ที่ปรึกษา

- ☐ เห็นควรให้ปรับสภาพได้
- ☐ ไม่เห็นควรให้ปรับสภาพ เนื่องจาก.....
-
-

ลงชื่อ.....

(.....)

..... / /

(2) ความเห็นของเจ้าหน้าที่บัณฑิตวิทยาลัย

ได้ตรวจสอบวิชาเรียนและผลการศึกษาเรียบร้อยแล้ว

- ☐ นักศึกษาได้ผ่านเงื่อนไขครบถ้วนสมบูรณ์
- ☐ นักศึกษายังไม่ผ่านเงื่อนไขตามที่คณะ กำหนด
- เนื่องจาก.....
-

ลงชื่อ.....

(.....)

..... / /

(3) ความเห็นเจ้าหน้าที่บัณฑิตศึกษา

ได้ตรวจสอบบทความ / หนังสือรับรองทางวิชาการ / เงื่อนไขของ

หลักสูตร

- ☐ ถูกต้องครบถ้วนสมบูรณ์
- ☐ ไม่ถูกต้อง เห็นควรนำเข้าสู่ที่ประชุมคณะกรรมการบัณฑิตศึกษา
- พิจารณา เนื่องจาก.....
-
- ☐ ไม่ถูกต้อง ไม่ควรนำเข้าสู่ที่ประชุมคณะกรรมการบัณฑิตศึกษา
- พิจารณา เนื่องจาก.....
-
- ☐ อื่น ๆ.....

ลงชื่อ.....

(.....)

..... / /

(4) ผลการพิจารณากรณีต้องนำเข้าสู่ที่ประชุมคณะกรรมการบัณฑิตศึกษา

- ☐ ปรับสภาพนักศึกษาได้ ตั้งแต่ภาคเรียนที่.....
- ปีการศึกษา.....
- ☐ ปรับสภาพนักศึกษาไม่ได้ เนื่องจาก.....
-
-

ลงชื่อ.....

(.....)

รองคณบดีฝ่ายบัณฑิตศึกษา

..... / /

(5) คำสั่ง คณบดีบัณฑิตวิทยาลัย

- ☐ อนุมัติ
- ☐ ไม่อนุมัติ เนื่องจาก.....
-
-

ลงชื่อ.....

(.....)

คณบดีบัณฑิตวิทยาลัย

..... / /



บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
แบบฟอร์มคำร้องขอขยายเวลาการศึกษา(นักศึกษาสามัญ รหัส 38...., 39....)

วันที่.....เดือน.....พ.ศ.....

เรียน คณบดีบัณฑิตวิทยาลัย

ข้าพเจ้า (นาย,นาง,นางสาว,อื่นๆ).....

รหัสประจำตัว.....เป็นนักศึกษาสามัญตั้งแต่ภาคเรียนที่.....ปีการศึกษา.....หลักสูตร.....

แผน.....สาขาวิชา.....ภาควิชา.....

แขนงวิชา.....คณะ.....

มีความประสงค์ขอขยายเวลาการศึกษาในภาคเรียนที่.....ปีการศึกษา.....เป็นครั้งที่.....เนื่องจากอยู่
ในระหว่างการทำวิทยานิพนธ์ในหัวข้อเรื่อง.....

โดยได้รับอนุมัติหัวข้อวิทยานิพนธ์จากบัณฑิตวิทยาลัยเรียบร้อยแล้ว เมื่อวันที่.....และได้ลงทะเบียน
ครบหน่วยกิตและรายวิชาตามหลักสูตรแล้ว จำนวน.....หน่วยกิต ได้ระดับคะแนนเฉลี่ยสะสม.....

จึงเรียนมาเพื่อโปรดพิจารณา

ขอแสดงความนับถือ

ลงชื่อ.....

(.....)

(ผู้ยื่นคำร้อง)

ที่อยู่ติดต่อได้.....

โทรศัพท์(บ้าน).....โทรศัพท์(ที่ทำงาน).....มือถือ.....

อีเมล.....

(1) ความเห็นของอาจารย์ผู้ควบคุมวิทยานิพนธ์

- ☐ เห็นควรให้ขยายเวลาการศึกษาต่อไปได้
- ☐ ไม่เห็นควรให้ขยายเวลาการศึกษา เนื่องจาก.....
-
-
-
-
-

ลงชื่อ.....

(.....)

..... / /

(2) ความเห็นของเจ้าหน้าที่บัณฑิตวิทยาลัย

1. การขอขยายเวลครั้งที่ 1

- ☐ มีประกาศอนุมัติหัวข้อเรียบร้อยแล้ว
- ☐ เรียนครบตามหลักสูตร
- ☐ ระดับคะแนนเฉลี่ย.....

2. การขอขยายเวลครั้งที่ 2, 3, 4

- ☐ รายงานความก้าวหน้าวิทยานิพนธ์เพิ่มเติม

ผลการตรวจสอบ

- ☐ ถูกต้องตามระเบียบ เห็นควรแจ้งคณะฯ เพื่อทราบ
- ☐ ไม่ถูกต้องตามระเบียบ เห็นควรนำเสนอคณะกรรมการ
- บัณฑิตศึกษาคณะ.....พิจารณา
- เนื่องจาก.....
-
-

ลงชื่อ.....

(.....)

..... / /

(3) ความเห็นของคณะฯ

กรณีถูกต้องตามระเบียบ

- ☐ คณะฯรับทราบ/เสนอคณะบดีบัณฑิตวิทยาลัยอนุมัติ
- กรณีไม่ถูกต้องตามระเบียบ

ผลการพิจารณาของคณะกรรมการบัณฑิตศึกษา

- ☐ เห็นควรให้ขยายเวลาการศึกษาได้ในภาคเรียนที่.....
- ปีการศึกษา.....
- ☐ ไม่เห็นควรให้ขยายเวลาการศึกษาในภาคเรียนที่.....
- ปีการศึกษา.....

เนื่องจาก.....

ลงชื่อ.....

(.....)

..... / /

(4) คำสั่ง (คณบดีบัณฑิตวิทยาลัย)

- ☐ อนุมัติ
- ☐ ไม่อนุมัติ เนื่องจาก.....
-
-

ลงชื่อ.....

(.....)

..... / /

แบบรายงานความก้าวหน้าในการทำวิทยานิพนธ์

หัวข้อวิทยานิพนธ์(ภาษาไทย).....
 (ภาษาอังกฤษ)
 อาจารย์ควบคุมวิทยานิพนธ์.....
 รายงานความก้าวหน้าของวิทยานิพนธ์ มีรายละเอียดดังต่อไปนี้



ลงชื่อ.....
 (.....)

นักศึกษา

..... / /

ลงชื่อ.....
 (.....)

อาจารย์ผู้ควบคุม

..... / /



แบบรายงานแผนดำเนินการขยายเวลาการศึกษาเพื่อขอสอบวิทยานิพนธ์ (สำหรับนักศึกษาคณะวิศวกรรมศาสตร์)

ชื่อ - นามสกุล.....รหัสประจำตัว.....

หลักสูตร ☐ วศ.ม ไฟฟ้า ☐ วศ.ม เครื่องกล ☐ วศ.ม ปิโตรเคมี ☐ วศ.ด ไฟฟ้า

ชื่ออาจารย์ที่ปรึกษา.....

ได้ขอขยายเวลาการศึกษาเป็นครั้งที่..... ภาคการศึกษาที่..... ปีการศึกษา.....

เหตุผลที่ขอขยายเวลา

.....

.....

.....

.....

.....

.....

.....

.....

.....

แผนดำเนินการเพื่อขอสอบวิทยานิพนธ์ (ให้ระบุรายละเอียดและแสดงช่วงเวลาอย่างชัดเจน)

รายละเอียดของงาน	เดือน	เดือน	เดือน	เดือน	เดือน	เดือน
1. ดำเนินการวิจัย						
2. รวบรวมและวิเคราะห์ข้อมูลงานวิจัย						
3. จัดทำวิทยานิพนธ์						
การขอสอบวิทยานิพนธ์						

รายละเอียดของงาน(ตัวอย่าง)	มิ.ย.	ก.ค.	ส.ค.	ก.ย.		
1. ดำเนินการวิจัย	↔					
2. รวบรวมและวิเคราะห์ข้อมูลงานวิจัย	←	→				
3. จัดทำวิทยานิพนธ์		←	→			
การขอสอบวิทยานิพนธ์				↔		

ลงชื่อ.....

(.....)

นักศึกษา

..... / /

ลงชื่อ.....

(.....)

อาจารย์ที่ปรึกษา

..... / /



บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
แบบฟอร์มคำร้องขอลงทะเบียนวิชา (Audits)

วันที่.....เดือน.....พ.ศ.....

เรียน คณบดีบัณฑิตวิทยาลัย

ข้าพเจ้า (นาย,นาง,นางสาว,อื่นๆ).....

รหัสประจำตัว..... หลักสูตร..... แผน..... สาขาวิชา.....

ภาควิชา.....แขนงวิชา.....คณะ.....

มีความประสงค์ขอลงทะเบียนรายวิชาที่เปิดสอนในระดับบัณฑิตศึกษา เป็นวิชา Audits ดังมีรายวิชาต่อไปนี้ โดยบันทึกผลการเรียนเป็น S หรือ U ตามประกาศของบัณฑิตวิทยาลัย

รหัสวิชา

ชื่อวิชา

.....
.....
.....

จึงเรียนมาเพื่อโปรดพิจารณา

ขอแสดงความนับถือ

ลงชื่อ.....

(.....)

(ผู้ยื่นคำร้อง)

ที่อยู่ติดต่อได้.....

โทรศัพท์(บ้าน).....โทรศัพท์(ที่ทำงาน).....มือถือ.....

อีเมล.....

(1) ความเห็นของอาจารย์ที่ปรึกษา

☐ เห็นควรให้ น.ศ. ลงทะเบียนในวิชาดังกล่าวเป็นวิชา Audits

☐ ไม่เห็นควรให้ น.ศ. ลงทะเบียนในวิชาดังกล่าว เป็นวิชา Audits เนื่องจาก.....

.....
.....
.....
.....
.....

(3) ความเห็นของเจ้าหน้าที่บัณฑิตวิทยาลัย

☐ เห็นควรพิจารณาอนุมัติได้ตามระเบียบ

☐ ไม่เห็นควรพิจารณาอนุมัติ เนื่องจาก.....

.....
.....
.....
.....
.....

ลงชื่อ.....

(.....)

..... / /

ลงชื่อ.....

(.....)

..... / /

(2) ความเห็นของอาจารย์ผู้สอน

☐ เห็นควรให้ น.ศ. ลงทะเบียนในวิชาดังกล่าว เป็นวิชา Audits

☐ ไม่เห็นควรให้ น.ศ. ลงทะเบียนในวิชาดังกล่าว เป็นวิชา Audits เนื่องจาก.....

.....
.....
.....
.....

(4) คำสั่ง (คณบดีบัณฑิตวิทยาลัย)

☐ อนุมัติตามที่เสนอ

☐ ไม่อนุมัติ เนื่องจาก.....

.....
.....
.....
.....

ลงชื่อ.....

(.....)

..... / /

ลงชื่อ.....

(.....)

..... / /



บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
แบบฟอร์มคำร้องขอถอนวิชาเรียน (เกินกำหนด)

วันที่.....เดือน.....พ.ศ.....

เรียน คณบดีบัณฑิตวิทยาลัย

ข้าพเจ้า (นาย,นาง,นางสาว,อื่นๆ).....

รหัสประจำตัว..... หลักสูตร..... แผน..... สาขาวิชา.....

ภาควิชา.....แขนงวิชา.....คณะ.....

มีความประสงค์ขอถอนวิชาเรียนเกินกำหนด เป็นกรณีพิเศษ ในภาคเรียนที่..... ปีการศึกษา.....

เนื่องจาก.....

ดังมีรายวิชาต่อไปนี้

รหัสวิชา

ชื่อวิชา

จึงเรียนมาเพื่อโปรดพิจารณา

ขอแสดงความนับถือ

ลงชื่อ.....

(.....)

(ผู้ยื่นคำร้อง)

ที่อยู่ติดต่อได้.....

โทรศัพท์(บ้าน).....โทรศัพท์(ที่ทำงาน).....มือถือ.....

อีเมล.....

(1) ความเห็นของอาจารย์ที่ปรึกษา

- ☐ เห็นควรให้ น.ศ. ถอนวิชาเรียนดังกล่าวได้
- ☐ ไม่เห็นควรให้ น.ศ. ถอนวิชาเรียนดังกล่าว เนื่องจาก.....

.....

.....

.....

.....

.....

(3) คำสั่ง (คณบดีบัณฑิตวิทยาลัย)

- ☐ อนุมัติตามที่เสนอ
- ☐ ไม่อนุมัติ เนื่องจาก.....

.....

.....

.....

.....

.....

ลงชื่อ.....

(.....)

..... / /

ลงชื่อ.....

(.....)

..... / /

(2) ความเห็นของเจ้าหน้าที่บัณฑิตวิทยาลัย

- ☐ เห็นควรพิจารณาอนุมัติได้ตามระเบียบ
- ☐ ไม่เห็นควรพิจารณาอนุมัติ เนื่องจาก.....

.....

.....

.....

.....

.....

ลงชื่อ.....

(.....)

..... / /



บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระเจ้าเกล้าเจ้าคุณทหารลาดกระบัง
แบบฟอร์มประวัตินักศึกษาที่สำเร็จการศึกษา

วันที่.....เดือน.....พ.ศ.....

รูปถ่าย 1 นิ้ว
(สวมชุดครุย)

ชื่อ - นามสกุล.....รหัสประจำตัว.....

หลักสูตร.....สาขาวิชา.....

คณะ.....

☐

ชื่อวิทยานิพนธ์

☐

ชื่อโครงการพัฒนาระบบแผนงาน (นศ. แผน ข)

☐

ชื่อโครงการศึกษาระณพิเศษ (นศ. แผน ข)

(ภาษาไทย).....

(ภาษาอังกฤษ).....

อาจารย์ผู้ควบคุมวิทยานิพนธ์/อาจารย์ที่ปรึกษาโครงการ.....

อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม.....

อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม.....

ประวัติส่วนตัว

เกิดวันที่.....เดือน.....พ.ศ.....สถานที่เกิด จังหวัด.....

วุฒิการศึกษาปริญญาตรี.....สถานศึกษา.....ปีที่สำเร็จ.....

วุฒิการศึกษาปริญญาโท.....สถานศึกษา.....ปีที่สำเร็จ.....

ผลงานวิชาการที่ได้รับการตีพิมพ์.....

รางวัลหรือทุนที่เคยได้รับ.....

ประสบการณ์ในการทำงาน.....

อาชีพปัจจุบัน.....สถานที่ทำงาน.....

โทรศัพท์.....

ที่อยู่ติดต่อได้.....

โทรศัพท์(บ้าน).....โทรศัพท์(ที่ทำงาน).....มือถือ.....

อีเมล.....

- หมายเหตุ**
1. โปรดกรอกข้อมูลประวัติและส่งคืนบัณฑิตวิทยาลัยภายในวันที่นักศึกษาส่งมอบวิทยานิพนธ์ฉบับสมบูรณ์หรือ
สอบผ่านประมวลความรู้
 2. นักศึกษาที่ทำวิทยานิพนธ์(Thesis) ให้ส่งแผ่นดิสก์บทคัดย่อ ภาษาไทย-ภาษาอังกฤษ ภายในวันที่
ส่งวิทยานิพนธ์ฉบับสมบูรณ์



บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง แบบฟอร์มขอถอนเงินประกันของเสียหยา

สถาบันจะคืนเงินค่าประกันทรัพย์สินเสียหายให้นักศึกษา เมื่อพ้นสภาพการเป็นนักศึกษาแล้ว เต็มจำนวนคงเหลือ
เมื่อได้หักหนี้สินที่ต้องชำระแล้ว หากนักศึกษาไม่ขอรับคืนภายใน 6 เดือน หลังจากพ้นสภาพการเป็นนักศึกษาให้ถือว่าค่า
ประกันทรัพย์สินเสียหายนี้เป็นสมบัติของสถาบันฯ

หลักฐานที่ต้องใช้

ใบเสร็จรับเงินที่ระบุค่าประกันทรัพย์สินเสียหาย (ใบเสร็จที่ได้รับเมื่อเข้าศึกษาในภาคเรียนแรก)

ข้อปฏิบัติในการกรอกข้อความ

1. กรอกข้อความในแบบฟอร์มที่แนบมาให้ครบด้วยตัวบรรจง ยกเว้นในส่วนที่ได้รับเงินค่าประกันทรัพย์สิน
เสียหายคืนจากสถาบัน
2. ให้นักศึกษากรอกที่อยู่ตามแบบฟอร์มด้านล่างนี้ให้ชัดเจน

ชื่อ.....นามสกุล.....รหัสประจำตัว.....
ที่อยู่ติดต่อได้.....
โทรศัพท์(บ้าน).....โทรศัพท์(ที่ทำงาน).....มือถือ.....
อีเมล.....

สถาบันเทคโนโลยีพระจอมเกล้า-
เจ้าคุณทหารลาดกระบัง กรุงเทพฯ

วันที่.....เดือน.....พ.ศ. 25.....

เรื่อง ขอดอนเงินค่าประกันของเสียหาย

เรียน คณบดีบัณฑิตวิทยาลัย

ข้าพเจ้า(ชื่อ-สกุล).....รหัสประจำตัว.....นักศึกษาชั้นปีที่.....
หลักสูตร..... สาขาวิชา.....คณะ..... เข้าเรียนเมื่อปีการ
ศึกษา 25..... ได้ชำระเงินค่าประกันของเสียหายไว้เป็นเงิน.....บาท (.....)

บัดนี้ข้าพเจ้าได้ ☐ ขอลาออก ☐ พ้นสภาพการเป็นนักศึกษา
☐ สำเร็จการศึกษา ☐ ปรับสภาพนักศึกษา

เมื่อวันที่..... เดือน..... พ.ศ. 25..... จึงใคร่ขอดอนเงินค่าประกันของเสียหาย จำนวนดังกล่าวคืนตามระเบียบ
จึงเรียนมาเพื่อโปรดพิจารณาอนุมัติ

ขอแสดงความนับถือ

ลงชื่อ.....
(.....)

ได้ตรวจสอบทะเบียนนักศึกษาแล้ว (นาย, นาง, นางสาว).....
ได้(ขอลาออก, พ้นสภาพการเป็นนักศึกษา, สำเร็จการศึกษา, ปรับสภาพนักศึกษา) แล้ว เมื่อวันที่.....เดือน.....
พ.ศ. 25.....

ลงชื่อ.....
(.....)

นายทะเบียน

...../...../.....
เสนอคณบดีคณะฯ พิจารณาอนุมัติ

ลงชื่อ.....
(.....)

คณบดี บัณฑิตวิทยาลัย

...../...../.....

สถาบันเทคโนโลยีพระจอมเกล้า-
เจ้าคุณทหารลาดกระบัง กรุงเทพฯ

วันที่..... เดือน..... พ.ศ. 25....

เรื่อง ขอลงเงินค่าประกันของเสียหายคืน

เรียน อธิการบดี

ข้าพเจ้า.....นามสกุล.....รหัสประจำตัว.....
หลักสูตร.....สาขาวิชา.....คณะ.....
มีความประสงค์จะขอเงินค่าประกันของเสียหายคืน เป็นเงินจำนวน.....บาท (.....)

บัดนี้ข้าพเจ้าได้ ☐ ขอลาออก ☐ พ้นสภาพการเป็นนักศึกษา

☐ สำเร็จการศึกษา ☐ ปรับสภาพนักศึกษา

จึงเรียนมาเพื่อโปรดพิจารณาอนุมัติ

ขอแสดงความนับถือ

ลงชื่อ.....
(.....)

อนุมัติให้จ่ายได้

ลงชื่อ.....
(.....)

คนบดึกณะ.....
...../...../.....

ข้าพเจ้าได้รับเงินค่าประกันคืนจากสถาบันฯ เป็นจำนวนเงิน.....บาท (.....)

ตั้งแต่วันที่..... เดือน..... พ.ศ. 25.....

จึงได้ลงลายมือชื่อไว้เป็นหลักฐาน

ลงชื่อ.....
(.....)

ผู้รับเงิน

สถาบันเทคโนโลยีพระจอมเกล้า-
เจ้าคุณทหารลาดกระบัง กรุงเทพฯ

วันที่..... เดือน..... พ.ศ. 25....

เรื่อง ขอลงเงินค่าประกันของเสียหายคืน

เรียน อธิการบดี

ข้าพเจ้า.....นามสกุล.....รหัสประจำตัว.....
หลักสูตร.....สาขาวิชา.....คณะ.....
มีความประสงค์จะขอเงินค่าประกันของเสียหายคืน เป็นเงินจำนวน.....บาท (.....)

บัดนี้ข้าพเจ้าได้ ☐ ขอลาออก ☐ พ้นสภาพการเป็นนักศึกษา

☐ สำเร็จการศึกษา ☐ ปรับสภาพนักศึกษา

จึงเรียนมาเพื่อโปรดพิจารณาอนุมัติ

ขอแสดงความนับถือ

ลงชื่อ.....
(.....)

อนุมัติให้จ่ายได้

ลงชื่อ.....
(.....)

คณบดีคณะ.....
...../...../.....

ข้าพเจ้าได้รับเงินค่าประกันคืนจากสถาบันฯ เป็นจำนวนเงิน.....บาท (.....)
ตั้งแต่วันที่..... เดือน..... พ.ศ. 25....

จึงได้ลงลายมือชื่อไว้เป็นหลักฐาน

ลงชื่อ.....
(.....)

ผู้รับเงิน

บรรณานุกรม

- Martin Fowler with Kendall Scott, *UML Distilled*, Second Edition: A brief guide to the standard Object Modeling Language, Addison - Wesley Longman, 2000
- Ali Bahrami, Boeing Applied Research & Technology, *Object Oriented Systems Development*, International Edition 1999, Irwin McGraw-Hill, 1976
- Ivar Jacobson, Grady Booch, James Rumbaugh, *The Unified Software Development Process* Rational Software Corporation, Addison -Wesley Longman, 1998
- Terry Quatrani, *Visual Modeling with Rational Rose 2000 and UML*, Addison-Wesley, Pearson Education Asia Pte Ltd., 1998
- อาจารย์อภิเนตร อุณาภูฏ, *Object-Oriented Analysis and Design*, แผนกตำรา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 1999
- Marty Hall, *Core Servlet and Java Server Page*, Prentice Hall PTR Upper Saddle River, 2000
- ดร. วีระศักดิ์ ชิงฉาว, *Fundamental of JAVA Programming Volume 1 and Volume 2*, SUM Publishing Department, SUM System Company Limited, 1998
- 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
- Osamu Takagiwa, Frederik Haesbrouck, Veronique Quilblier, Sarah Poger, *Programming with VisualAge for Java, Version 3.5*, Third Edition, Prentice Hall PTR, Upper Saddle River, New Jersey 07458, April 2001
- กิตติ ภัคดีวัฒนกุล, *Java ฉบับโปรแกรมเมอร์*, KTP COMP & CONSULT, 2001