

แอปพลิเคชันเฟรมเวิร์กบนเว็บสำหรับระบบยืม-คืน

WEB-BASE APPLICATION FRAMEWORK FOR  
BORROWING-RETURNING SYSTEM



ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

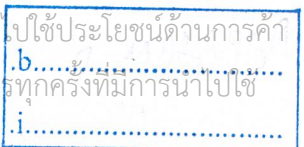
ปีการศึกษา 2546

เลขหมู่.....

เลขทะเบียน 51793

วัน,เดือน,ปี 29 ก.ค. 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**WEB-BASE APPLICATION FRAMEWORK FOR  
BORROWING-RETURNING SYSTEM**



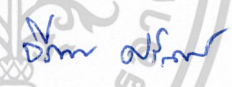


**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE  
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE  
FACULTY OF SCIENCE  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
ACADEMIC YEAR 2003**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ แอปพลิเคชันเฟรมเวิร์คบนเว็บสำหรับระบบยืม-คืน  
WEB-BASE APPLICATION FRAMEWORK FOR BORROWING-  
RETURNING SYSTEM

ชื่อนักศึกษา นางสาวนุกูล ทวีสุข 43050382  
นางสาววิษญู ทวีสิน 43050411  
ปริญญา วิทยาศาสตร์บัณฑิต  
ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์  
สาขาวิชา วิทยาการคอมพิวเตอร์  
ปีการศึกษา 2546  
อาจารย์ที่ปรึกษา ผศ.ดร.ศรัณย์ อินทโกสุม

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระ  
จอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลัก  
สูตรวิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2546

	คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ	ผศ.ดร.จิรพร ศรีสวัสดิ์	
กรรมการ	ดร.นันทิกา เบญจเทพานันท์	
กรรมการและอาจารย์ที่ปรึกษา	ผศ.ดร.ศรัณย์ อินทโกสุม	

(ผู้ช่วยศาสตราจารย์ ดร. วีระ บุญจริง)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	แอปพลิเคชันเฟรมเวิร์คบนเว็บสำหรับระบบยืม-คืน WEB-BASE APPLICATION FRAMEWORK FOR BORROWING- RETURNING SYSTEM	
ชื่อนักศึกษา	นางสาว นุกุล ทวีสุข	43050382
	นางสาว วิชญ ทวีสิน	43050411
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2546	
อาจารย์ที่ปรึกษา	ผศ.ดร. ศรีณย์ อินทโกสุม	

### บทคัดย่อ

การพัฒนาระบบโดยใช้แอปพลิเคชันเฟรมเวิร์ค เป็นแนวคิดหนึ่งของการพัฒนาระบบงานแบบนำกลับมาใช้ใหม่ แอปพลิเคชันเฟรมเวิร์คคือซอฟต์แวร์คอมพิวเตอร์ที่ยังไม่สมบูรณ์ ซึ่งประกอบด้วยโครงสร้างและการจัดการลำดับการทำงาน ซึ่งแอปพลิเคชันที่คล้ายกันต้องมีส่วนร่วม ผู้พัฒนาแอปพลิเคชันสามารถพัฒนาแอปพลิเคชันได้โดยเพียงแค่ปรับเฟรมเวิร์คให้เหมาะสมตามความต้องการของแอปพลิเคชันนั้น ซึ่งผลลัพธ์ก็คือการพัฒนาแอปพลิเคชันทำได้ รวดเร็ว สะดวก และถูกต้องมากขึ้น ปัญหาพิเศษฉบับนี้มีจุดประสงค์เพื่อออกแบบและพัฒนาแอปพลิเคชันเฟรมเวิร์คของระบบยืมคืนบนเว็บ การทดสอบทำโดยใช้ เฟรมเวิร์คในการพัฒนาส่วนของการยืม-คืนของระบบห้องสมุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Special Project Title</b>	WEB-BASE APPLICATION FRAMEWORK FOR BORROWING-RETURNING SYSTEM	
<b>Students</b>	Miss Nugul Tvesuk	43050382
	Miss Witchaya Taveesin	43050411
<b>Degree</b>	Bachelor of Science	
<b>Department</b>	Mathematics and Computer Science, Faculty of Science	
<b>Programme</b>	Computer Science	
<b>Academic Year</b>	2003	
<b>Special Project Advisor</b>	Asst.Prof.Dr. Sarun Intakosum	

### ABSTRACT

Application development using application framework is one of the reusable system development techniques. Application framework is an incomplete software component that holds the common structure and handles the control flow shared by similar applications. Application developer can develop an application by adjusting a framework to complete the requirement of the application. As a consequence, an application can be developed faster, more convenient, and more robust. This special project designs and develops a web-base application framework for borrowing-returning system. Testing of the framework is done by applying it to the implementation of borrowing-returning part of a library system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่อง แอปพลิเคชันเฟรมเวิร์คบนเว็บสำหรับระบบยืม-คืน สามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ อาจารย์ ศรัณย์ อินทโกสุม อาจารย์ผู้รับผิดชอบปัญหาพิเศษนี้ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่างๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนการทำปัญหาพิเศษนี้สำเร็จลุล่วงไปด้วยดี รวมทั้งเพื่อนๆ และพี่ๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ

มีนาคม 2547



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
<b>บทที่ 1 บทนำ.....</b>	<b>1</b>
1.1 ความสำคัญและที่มาของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์.....	1
1.3 ขอบเขตของการศึกษา.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	1
1.5 ขั้นตอนในการดำเนินงาน.....	2
1.6 เอกสารอ้างอิง.....	2
1.7 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ.....	2
1.8 งบประมาณที่คาดว่าจะใช้ในการทำปัญหาพิเศษ.....	2
<b>บทที่ 2 งานวิจัยที่เกี่ยวข้อง.....</b>	<b>3</b>
2.1 บทนำ.....	3
2.2 การพัฒนาเทคโนโลยีเว็บ.....	3
2.3 เว็บแอปพลิเคชัน.....	5
2.3.1 สถาปัตยกรรมไคลเอนต์/เซิร์ฟเวอร์.....	5
2.3.2 เว็บเซิร์ฟเวอร์.....	6
2.3.3 เว็บเบราว์เซอร์.....	6
2.4 แอปพลิเคชันเฟรมเวิร์ค.....	7
2.5 เทคโนโลยีที่นำมาสร้างส่วนติดต่อผู้ใช้งานแอปพลิเคชัน.....	7
2.5.1 HTML.....	8
2.5.2 JSP.....	8
<b>บทที่ 3 แอปพลิเคชันเฟรมเวิร์คสำหรับระบบยืม-คืน.....</b>	<b>10</b>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
3.1 ขอบเขตของระบบ.....	10
3.2 การศึกษาและวิเคราะห์ความต้องการทั่วไปของระบบยืม-คืน.....	10
3.2.1 ความสามารถด้านการจัดการฐานข้อมูลของทรัพยากรในระบบยืม-คืน.....	10
3.2.2 ความสามารถด้านการจัดการฐานข้อมูลสมาชิกในระบบยืม-คืน.....	10
3.2.3 ความสามารถด้านการจัดการระบบยืม-คืนทรัพยากรในระบบยืม-คืน.....	10
3.2.4 ความสามารถด้านการค้นหาทรัพยากรในระบบยืม-คืน.....	10
3.3 ความต้องการเฉพาะของระบบยืม-คืนแต่ละระบบ.....	11
3.4 การออกแบบแอปพลิเคชันเฟรมเวิร์คสำหรับระบบยืม-คืน.....	11
3.4.1 Internal Interface.....	12
3.4.1.1 อินเทอร์เฟซ IConnectDB.....	13
3.4.1.2 อินเทอร์เฟซ IBorrowReturn.....	13
3.4.1.3 อินเทอร์เฟซ IMemberManage.....	13
3.4.1.4 อินเทอร์เฟซ IResourceManage.....	13
3.4.2 Abstract Class.....	13
3.4.2.1 แอปสแตร์กคลาส Abs_BorrowReturn.....	13
3.4.2.2 แอปสแตร์กคลาส Abs_MemberManage.....	13
3.4.2.3 แอปสแตร์กคลาส Abs_ResourceManage.....	13
3.4.3 Controller Class.....	14
3.4.3.1 คลาส ControlMember.....	14
3.4.3.2 คลาส ControlResource.....	16
3.4.3.3 คลาส ControlBorrowReturn.....	18
<b>บทที่ 4 ขั้นตอนการดำเนินงานวิจัย.....</b>	<b>23</b>
4.1 การพัฒนาแอปพลิเคชันเฟรมเวิร์ค.....	23
4.1.1 คลาสอินเทอร์เฟซภายในเฟรมเวิร์ค.....	23
4.1.2 แอปสแตร์กคลาสภายในเฟรมเวิร์ค.....	24
4.1.3 คลาสคอนโทรลเลอร์ภายในเฟรมเวิร์ค.....	26
4.2 ทดสอบแอปพลิเคชันเฟรมเวิร์ค.....	30
4.2.1 การสร้างคลาสที่สืบทอดมาจากแอปสแตร์กคลาสในเฟรมเวิร์ค.....	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
4.2.2 การสร้างส่วนติดต่อผู้ใช้งานของห้องสมุด.....	33
4.2.3 การสร้างโปรแกรมหลักเพื่อเรียกใช้งานเมธอดของคอนโทรลเลอร์.....	35
<b>บทที่ 5 สรุปผลวิจัยและข้อเสนอแนะ.....</b>	<b>38</b>
5.1 สรุปผลการวิจัย.....	38
5.2 ข้อเสนอแนะ.....	38
<b>บรรณานุกรม.....</b>	<b>39</b>
<b>ภาคผนวก ก.คู่มือการติดตั้ง.....</b>	<b>40</b>
การติดตั้ง Tomcat.....	40
<b>ภาคผนวก ข.รายละเอียดของเฟรมเวิร์ค.....</b>	<b>44</b>
<b>ภาคผนวก ค.ตัวอย่างการสร้างคลาสเพื่อใช้งานเฟรมเวิร์ค.....</b>	<b>53</b>



## สารบัญตาราง

ตารางที่	หน้า
ข.1 เมธอดของอินเทอร์เฟซ IConnectDB.....	44
ข.2 เมธอดของอินเทอร์เฟซ IBorrowReturn .....	45
ข.3 เมธอดของอินเทอร์เฟซ IMemberManage .....	45
ข.4 เมธอดของอินเทอร์เฟซ IresourceManage.....	46
ข.5 เมธอดของแอบสแทรกคลาส ชื่อ Abs_BorrowReturn .....	46
ข.6 เมธอดของแอบสแทรกคลาส ชื่อ Abs_MemberManage.....	47
ข.7 เมธอดของแอบสแทรกคลาส ชื่อ Abs_ResourceManage .....	48
ข.8 เมธอดของคอนโทรลเลอร์ ControlMember.....	48
ข.9 เมธอดของคอนโทรลเลอร์ ControlResource .....	49
ข.10 แอททริบิวต์ของคอนโทรลเลอร์ ControlBorrowReturn .....	50
ข.11 เมธอดของคอนโทรลเลอร์ ControlBorrowReturn.....	49



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ

รูปที่	หน้า
2.1 การพัฒนาเว็บยุคแรก.....	4
2.2 การพัฒนาเว็บยุคที่ 2.....	5
2.3 โคลเอนต์ / เซิร์ฟเวอร์ทรานแซคชั่น.....	6
3.1 การออกแบบเฟรมเวิร์คแอปพลิเคชันระบบยืม-คืน.....	12
3.2 ซีเควนซ์ไดอะแกรมของการจัดการข้อมูลสมาชิก.....	15
3.3 ซีเควนซ์ไดอะแกรมของการจัดการข้อมูลทรัพยากร.....	17
3.4 ซีเควนซ์ไดอะแกรมของการยืมทรัพยากร.....	19
3.5 ซีเควนซ์ไดอะแกรมของการคืนทรัพยากร.....	21
4.1 หน้าจอหลักของส่วนติดต่อผู้ใช้งาน.....	33
4.2 หน้าจอส่วนติดต่อผู้ใช้งานสำหรับจัดการข้อมูลสมาชิก.....	34
4.3 หน้าจอส่วนติดต่อผู้ใช้งานสำหรับจัดการข้อมูลทรัพยากร.....	34
4.3 หน้าจอส่วนติดต่อผู้ใช้งานสำหรับค้นหาข้อมูลทรัพยากร.....	35
ก.1 หน้าจอสำหรับกำหนด CLASSPATH Variable.....	41
ก.2 หน้าจอสำหรับกำหนด JAVA_HOME Variable.....	42
ก.3 หน้าจอของ Web Browser ที่แสดงตัวอย่างของ Servlets.....	43
ค.2 ตัวอย่างคลาสที่สืบทอดมาจากแอ็บสแตรคคลาส Abs_MemberManage.....	53
ค.3 ตัวอย่างคลาสที่สืบทอดมาจากแอ็บสแตรคคลาส Abs_ResourceManage .....	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของปัญหา

เฟรมเวิร์คสำหรับ โปรแกรมประยุกต์ เป็นวิธีช่วยให้การพัฒนา โปรแกรมประยุกต์ทำได้ โดยสะดวก ถูกต้อง และรวดเร็วขึ้น

จากคุณสมบัติที่ดีของแอปพลิเคชันเฟรมเวิร์คดังกล่าว จึงมีแนวคิดที่จะพัฒนาเฟรมเวิร์ค สำหรับโปรแกรมประยุกต์ที่สามารถใช้งานบนเว็บได้ โดยเฟรมเวิร์คของโปรแกรมประยุกต์ที่ พัฒนาขึ้นสำหรับปัญหาพิเศษนี้ จะเป็นเฟรมเวิร์คสำหรับระบบยืม-คืนทั่วไป เช่น ระบบห้องสมุด ระบบเช่าวิดีโอ ระบบเช่ารถยนต์ เป็นต้น

### 1.2 ความมุ่งหมายและวัตถุประสงค์

1.2.1 เพื่อพัฒนาเฟรมเวิร์คของโปรแกรมประยุกต์สำหรับระบบยืม-คืนทั่วไป ซึ่งจะช่วยให้โปรแกรมพัฒนาระบบยืม-คืน ทำได้สะดวก รวดเร็ว

1.2.2 เพื่อศึกษาวิธีการพัฒนาเฟรมเวิร์คบนเว็บ

1.2.3 เพื่อขยายขีดความสามารถของเว็บให้สนับสนุนการพัฒนาเฟรมเวิร์คสำหรับโปรแกรมประยุกต์

1.2.4 เพื่อนำความรู้ที่ได้เรียนมาประยุกต์ใช้กับงานจริง

### 1.3 ขอบเขตของการศึกษา

1.3.1 เฟรมเวิร์คสำหรับ โปรแกรมประยุกต์ที่พัฒนาขึ้น ทดสอบจากระบบยืม-คืนของห้องสมุด

1.3.2 เว็บแอปพลิเคชันที่พัฒนาขึ้นจะใช้เทคโนโลยีของจาวา

### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้เฟรมเวิร์คสำหรับ โปรแกรมประยุกต์ที่สามารถนำไปพัฒนาโปรแกรมระบบยืม-คืนทั่วไป

1.4.2 ได้วิธีการขยายความสามารถของเว็บที่จะสนับสนุนการพัฒนาเฟรมเวิร์คสำหรับโปรแกรมประยุกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.5 ขั้นตอนในการดำเนินงาน

- 1.5.1 ศึกษาเฟรมเวิร์คและระบบงานยืม-คืน
- 1.5.2 ศึกษาโปรแกรมที่เกี่ยวกับการแก้ปัญหา
- 1.5.3 ออกแบบเฟรมเวิร์คสำหรับงานยืม-คืน
- 1.5.4 เขียนโปรแกรมและทดสอบการทำงาน
- 1.5.5 สร้างแอปพลิเคชันเพื่อทดสอบเฟรมเวิร์คที่พัฒนาขึ้น
- 1.5.6 ประเมินผลงาน
- 1.5.7 ทำรายงาน
- 1.5.8 ส่งผลงานและนำเสนอผลงาน

## 1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

- 1.6.1 คอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์
- 1.6.2 คอมพิวเตอร์ที่ทำหน้าที่เป็นไคลเอนต์
- 1.6.3 เครื่องมือพัฒนาเว็บซึ่งใช้เทคโนโลยีจาวา
- 1.6.4 ระบบจัดการฐานข้อมูล MySQL
- 1.6.5 ระบบเครือข่าย

## 1.7 งบประมาณที่คาดว่าจะใช้ในการทำปัญหาพิเศษ

- |       |                    |          |
|-------|--------------------|----------|
| 1.7.1 | ค่าหนังสืออ้างอิง  | 2000 บาท |
| 1.7.2 | ค่าถ่ายเอกสาร      | 500 บาท  |
| 1.7.3 | ค่าเข้าปก-เย็บเล่ม | 500 บาท  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### งานวิจัยที่เกี่ยวข้อง

#### 2.1 บทนำ

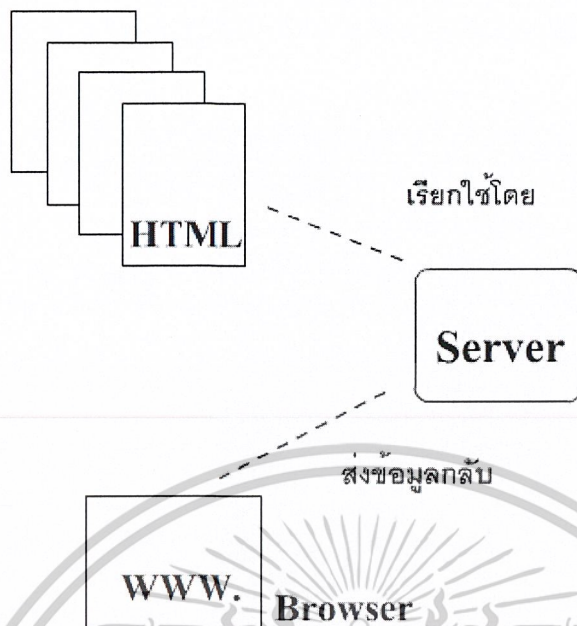
ปัจจุบันมีระบบงานที่มีโครงสร้างการทำงานมาตรฐานหลัก ที่มีความคล้ายคลึงกัน เช่น ระบบการเช่าโรงแรม ระบบการเช่ารถ ระบบห้องสมุด ระบบการเช่าร้านวิดีโอ เป็นต้น เนื่องจากระบบงานเหล่านี้ล้วนมีการทำงานหลักที่เหมือนกัน ได้แก่ การจัดการฐานข้อมูลของสมาชิกและทรัพยากรในระบบยืม-คืน การตรวจสอบสมาชิก การตรวจสอบทรัพยากร การตรวจสอบการเงื่อนไขการให้บริการยืม-คืน และการคิดค่าบริการ

จากขั้นตอนการทำงานของระบบที่มีรูปแบบดังกล่าวนี้ จึงสร้างเฟรมเวิร์คขึ้นมาใช้งาน ในบทนี้จะกล่าวถึงภาพรวมของเว็บและเทคโนโลยีที่เกี่ยวข้องกับการสร้างแอปพลิเคชันเฟรมเวิร์ค ซึ่งเกี่ยวข้องกับหัวข้อปัญหาพิเศษนี้

#### 2.2 การพัฒนาเทคโนโลยีเว็บ

อินเทอร์เน็ตยุคแรกเป็นการติดต่อสื่อสารบนเครือข่ายอินเทอร์เน็ต ที่มีการนำเสนอข้อมูลเป็นแบบสถติก (static) ข้อมูลต่างๆ ได้รับการออกแบบและกำหนดให้มีโครงสร้างที่แน่นอน ข้อมูลเหล่านี้จะถูกจัดเก็บไว้ในฟอร์แมตของเอกสาร Hyper Text Markup Language (HTML) เป็นหลัก และถูกจัดเก็บไว้ในเซิร์ฟเวอร์ที่ติดต่อกันผ่านทางโพรโตคอล Hyper Text Transfer Protocol (HTTP) เพื่อนำข้อมูลจากเซิร์ฟเวอร์มาแสดงผลผ่านโปรแกรมเว็บเบราว์เซอร์ ดังรูปที่ 2.1

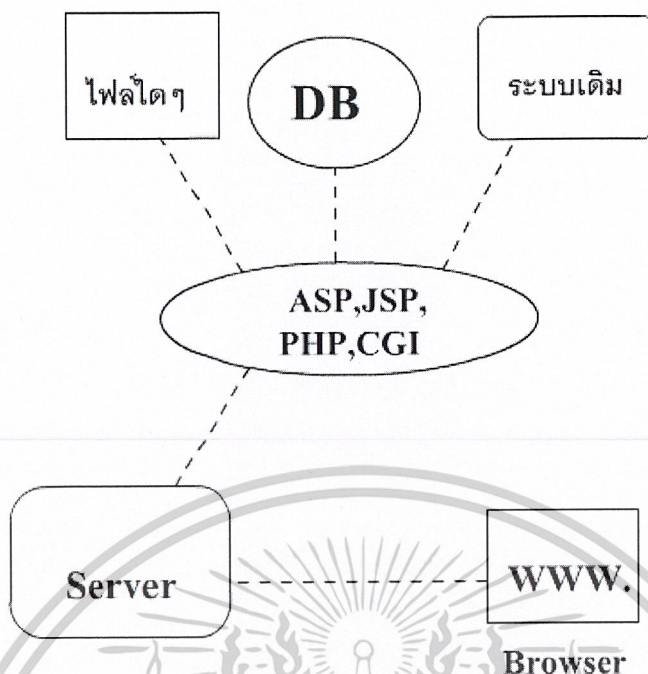
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### รูปที่ 2.1 การพัฒนาเว็บยุคแรก

ยุคต่อมาเป็นยุคที่เว็บมีการเชื่อมต่อเข้ากับระบบฐานข้อมูล และมีการประมวลผลบางอย่าง ทำให้ลักษณะของข้อมูลที่แสดงผลออกมามีลักษณะเป็นแบบไดนามิก มีการใช้งานภาษาสำหรับการโปรแกรมมากขึ้น เช่น Common Gateway Interface (CGI), Active Server Pages (ASP) ของไมโครซอฟต์, JavaServer Pages (JSP) ที่ใช้ภาษาจาวาในการพัฒนา โดยที่โปรแกรมต่างๆที่เขียนขึ้นจะอยู่ที่เซิร์ฟเวอร์ และผู้ใช้งานต้องใช้งาน โปรแกรมผ่านทางเว็บเบราว์เซอร์เพื่อแสดงผลที่ใช้งาน ดังรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 การพัฒนาเว็บยุคที่ 2

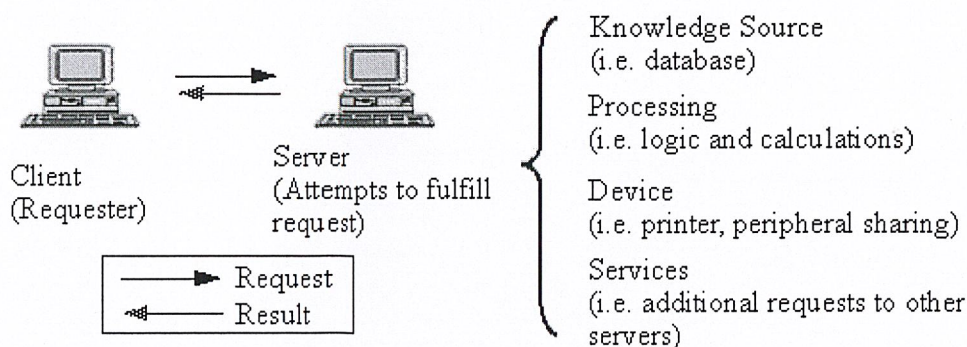
## 2.3 เว็บแอปพลิเคชัน (Web Application)

เว็บแอปพลิเคชัน คือ ระบบงานที่สร้างขึ้นมาในลักษณะเว็บเพจแล้วนำไปเก็บไว้ที่เว็บเซิร์ฟเวอร์ โดยสามารถเรียกใช้งานระบบผ่านโปรแกรมเบราว์เซอร์ ซึ่งการพัฒนาเว็บแอปพลิเคชันจะต้องอาศัยเทคโนโลยีระบบเครือข่ายอินเทอร์เน็ตและโคลเอนต์ / เซิร์ฟเวอร์

### 2.3.1 สถาปัตยกรรมโคลเอนต์ / เซิร์ฟเวอร์

เครือข่ายคอมพิวเตอร์มีความสำคัญในการเชื่อมโยงภายในองค์กร การทำงานขององค์กรเดิมเน้นการทำงานแบบ Host-based System คือ ในหน่วยงานจะมีเครื่องคอมพิวเตอร์ขนาดใหญ่ที่ใช้เป็นทั้งแอปพลิเคชันเซิร์ฟเวอร์และดาต้าเบส/ไฟล์เซิร์ฟเวอร์ และมีเครื่องคอมพิวเตอร์ขนาดเล็กเชื่อมต่อเข้ากับเครื่องเซิร์ฟเวอร์ ทำหน้าที่เป็นเทอร์มินอลในการทำงาน เครื่องเทอร์มินอลจะส่งข้อมูลไปประมวลผลที่เครื่องเซิร์ฟเวอร์ และส่งผลลัพธ์กลับมาแสดงที่เทอร์มินอล ข้อดีของการทำงานแบบนี้ คือ ไม่จำเป็นต้องใช้เครื่องเทอร์มินอลที่มีประสิทธิภาพสูงมาก และการเปลี่ยนแปลงแก้ไขแอปพลิเคชันจะทำได้ง่าย แต่มีข้อเสีย คือ เมื่อระบบมีขนาดใหญ่ มีปริมาณเครื่องเทอร์มินอลที่เชื่อมต่อมาก เครื่องเซิร์ฟเวอร์จะทำงานหนักและอาจมีผลต่อเวลาที่ใช้ในการตอบสนองผู้ใช้งาน ปัจจุบันจึงเกิดแนวความคิดแบบ โดยการพัฒนาระบบแบบโคลเอนต์/เซิร์ฟเวอร์ เครื่องเซิร์ฟเวอร์จะทำหน้าที่เป็นเพียงดาต้าเบส/ไฟล์เซิร์ฟเวอร์เท่านั้น ส่วนการประมวลผลทั้งหมดจะอยู่ที่เครื่องโคลเอนต์ ดังรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### รูปที่ 2.3 ไคลเอนต์ / เซิร์ฟเวอร์ทรานแซกชัน

เซิร์ฟเวอร์ คือ เครื่องคอมพิวเตอร์ที่ให้บริการเมื่อผู้ใช้องขอและให้บริการแก่ผู้ใช้หลายคน

ไคลเอนต์ คือ เครื่องคอมพิวเตอร์ที่ร้องขอบริการจากเซิร์ฟเวอร์

#### การแบ่งประเภทของไคลเอนต์ / เซิร์ฟเวอร์ตามประเภทการใช้งาน

- 1) ดาต้าเบสเซิร์ฟเวอร์ คือ เซิร์ฟเวอร์ที่ทำหน้าที่เป็น Database Management System (DBMS) คอยจัดการกับข้อมูลที่จัดเก็บอยู่ในเซิร์ฟเวอร์เดียวกันนี้ ทำให้ผู้ใช้ที่เป็นไคลเอนต์สามารถเข้ามาใช้ข้อมูลร่วมกันได้
- 2) แอปพลิเคชันเซิร์ฟเวอร์ คือ เซิร์ฟเวอร์ที่ทำหน้าที่จัดการกับ โปรแกรมแอปพลิเคชันในการเชื่อมต่อระหว่างดาต้าเบสเซิร์ฟเวอร์กับผู้ใช้เพื่อให้ทั้งสองส่วนสามารถเชื่อมต่อกันได้อย่างสมบูรณ์
- 3) ไฟล์เซิร์ฟเวอร์ คือ เซิร์ฟเวอร์ที่ทำหน้าที่จัดเก็บไฟล์ที่สร้างขึ้น เพื่อให้ไคลเอนต์อื่นๆ สามารถเข้ามาใช้ไฟล์ร่วมกันได้โดยอาจจะกำหนดสิทธิ์ในการทำงานตามกลุ่มของผู้ใช้

#### 2.3.2 เว็บเซิร์ฟเวอร์

เว็บเซิร์ฟเวอร์คือ แอปพลิเคชันที่ทำหน้าที่รับและประมวลผลข้อมูลที่ถูกร้องขอจากผู้ใช้บริการทางอินเทอร์เน็ต โดยผ่านทางเว็บเบราว์เซอร์ การประมวลผลอาจจะเป็นการคำนวณ ค้นหาหรือวิเคราะห์ข้อมูลก็ได้ หลังจากเว็บเบราว์เซอร์รับคำร้องและประมวลผลแล้ว ผลลัพธ์ที่ได้จะถูกส่งกลับไปยังผู้ใช้โดยแสดงผลในเว็บเบราว์เซอร์ นอกจากนี้เว็บเบราว์เซอร์จะให้บริการในอินเทอร์เน็ตแล้ว เว็บเบราว์เซอร์อาจนำมาประยุกต์ใช้ในระบบเครือข่ายภายในองค์กร หรืออินทราเน็ตได้อีกด้วย

#### 2.3.3 เว็บเบราว์เซอร์

เว็บเบราว์เซอร์ เป็นโปรแกรมหนึ่งที่ทำให้ผู้ใช้ติดต่อหรือดูข้อมูลบน World Wide Web (WWW) และเป็นโปรแกรมบนเครื่องไคลเอนต์ที่ใช้ HTTP ซึ่งเป็นโพรโทคอลที่ใช้ในการร้องขอข้อมูลจากเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์ ข้อมูลที่ผู้ใช้สามารถจะเข้าไปดูได้ มีทั้งข้อมูลที่อยู่ในรูปข้อความหรือกราฟฟิก ตัวอย่างโปรแกรมเว็บเบราว์เซอร์ที่นิยมใช้กันทั่วไป เช่น Internet Explorer และ Netscape เป็นต้น

## 2.4 แอปพลิเคชันเฟรมเวิร์ค

เฟรมเวิร์ค เป็นการออกแบบระบบที่สามารถนำกลับมาใช้ใหม่ได้ ซึ่งจะอธิบายว่าระบบถูกแบ่งออกเป็นกลุ่มๆที่มีการติดต่อระหว่างกันและติดต่อกันอย่างไร รวมถึงการควบคุมลำดับการทำงานภายในเฟรมเวิร์ค ส่วนที่สำคัญที่สุดของเฟรมเวิร์ค คือ การแบ่งระบบออกเป็นคอมโพเนนต์ตามหน้าที่การทำงาน และสามารถนำคอมโพเนนต์เหล่านี้กลับมาใช้ใหม่ได้

เฟรมเวิร์คมักถูกอิมพลีเมนต์ด้วยภาษาโปรแกรมเชิงวัตถุ เช่น C++ , Smalltalk , Eiffel หรือ จาวา เฟรมเวิร์คจะประกอบไปด้วยหลายๆแอบสแตรกคลาส ซึ่งไม่สามารถสร้างออบเจ็กต์ได้ แต่จะทำหน้าที่เป็นคลาสต้นแบบเท่านั้น ภายในแอบสแตรกคลาสจะมีอย่างน้อย 1 เมธอดที่ยังไม่ได้สร้างโค้ดขึ้น เฟรมเวิร์คจะใช้แอบสแตรกคลาสเหล่านี้เพื่อออกแบบคอมโพเนนต์และโครงสร้างของเฟรมเวิร์ค เพื่อให้ผู้ที่นำไปพัฒนาสืบทอดคลาสเหล่านี้ต่อไป

อินเทอร์เฟซและแอบสแตรกคลาสจะมีเมธอดต้นแบบที่ผู้พัฒนานำไปสร้างโค้ดขึ้น โดยไม่กระทบกับโครงสร้างของอัลกอริทึมในเฟรมเวิร์ค ซึ่งแต่ละเมธอดจะทำงานในแต่ละขั้นตอนของอัลกอริทึมนั้นคือแต่ละเมธอดเป็นอิสระต่อกัน บางเมธอดจะมีโค้ดพื้นฐานเตรียมไว้ให้แล้ว (default implementation) โดยผู้พัฒนาสามารถใช้เมธอดเหล่านั้นได้เลย หรือสร้างโค้ดใหม่ทับโค้ดเดิมได้ตามความเหมาะสมกับความต้องการเพิ่มเติมใหม่ๆ เฟรมเวิร์คมีคุณสมบัติของภาษาโปรแกรมเชิงวัตถุ 3 ข้อ ได้แก่

- Abstract data types คือ ข้อมูลที่วิธีและรายละเอียดของข้อมูลต้องไม่มีความสำคัญต่อการใช้งานชนิดข้อมูลนั้น และต้องไม่สามารถอ้างถึงข้อมูลนั้นได้โดยตรง หากต้องการอ่านหรือเขียนข้อมูลบางส่วนจะต้องทำผ่าน operations ที่กำหนดไว้เท่านั้น

- Polymorphism คือ เมธอดที่สามารถจัดการกับ instance ของคลาสที่ต่างกันได้ หากเป็นคลาสที่สืบทอดมาจากคลาสหนึ่งที่กำหนดให้เป็นพารามิเตอร์ของเมธอดนั้น

- Inheritance คือ การนำคลาสใหม่ที่ถูกรสร้างไว้แล้ว มาเพิ่มหรือเปลี่ยนแปลงสมาชิกบางตัวเป็นคลาสใหม่ที่เหมาะสมกับหน้าที่ใหม่ที่ต้องการ คลาสใหม่จะได้สมาชิกของคลาสเดิมมา โดยจะมีสมาชิกใหม่เพิ่มขึ้นหรือสมาชิกของคลาสเดิมบางตัวถูกเปลี่ยนแปลงไปจากเดิม คลาสเดิมสามารถใช้งานได้เช่นเดิม แต่คลาสใหม่สามารถทำงานในหน้าที่เดิมและทำงานในหน้าที่ใหม่ได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นตอนการทำงานของเฟรมเวิร์กจะถูกควบคุมโดยคอนโทรลเลอร์ การใช้งานเฟรมเวิร์ก ผู้พัฒนาจะต้องทราบล่วงหน้าเกี่ยวกับรูปแบบของแอ็บสแตรกคลาสที่อยู่ภายในเฟรมเวิร์ก เพื่อจะเรียกใช้งานเฟรมเวิร์กได้อย่างถูกต้องและเหมาะสมตรงความต้องการของผู้พัฒนา โดยผู้พัฒนาสร้างโปรแกรมหลัก และพิจารณาว่าโปรแกรมที่สร้างขึ้นจำเป็นต้องเรียกใช้งานเฟรมเวิร์กผ่านคอนโทรลเลอร์ในสไลด์บ้าง จากนั้นคอนโทรลเลอร์ในเฟรมเวิร์กจะเป็นผู้เรียกโค้ดที่ผู้พัฒนาสร้างขึ้น เพื่อทำงานตามลำดับต่อไป

## 2.5 เทคโนโลยีที่นำมาสร้างส่วนติดต่อผู้ใช้งานแอปพลิเคชัน

### 2.5.1 HTML

จุดเด่นที่สำคัญของ HTML คือ ความสามารถในการเชื่อมโยง ข้อมูลไปยังเอกสารอื่นได้ เป็นเอกสารที่มีความสามารถมากกว่าเอกสารทั่วไป และมีความสามารถ แบบไฮเปอร์เท็กซ์ คือสามารถเปิดดูได้โดย เท็กซ์เอดิเตอร์ใดๆ ส่วนการเชื่อมโยงข้อมูลไปยัง เอกสารอื่นๆนั้น สามารถทำได้โดยการใส่สัญลักษณ์พิเศษเข้าไปในเอกสาร เรียกว่า แท็ก (Tag) ซึ่งจะถูกรับอ่านโดยโปรแกรมเว็บเบราว์เซอร์ต่างๆ HTML นั้นก็มีคุณสมบัติที่ง่ายต่อการเขียน สามารถนำเสนอข้อมูลที่มีทั้ง ตัวอักษร, ภาพ, เสียง, วิดีโอ และอื่นๆที่ผนวกอยู่ในไฟล์เดียว และสามารถเชื่อมโยงกับเอกสารอื่นๆได้ง่าย

### 2.5.2 JSP

JSP เป็นสคริปต์อีกภาษาหนึ่ง สำหรับการพัฒนาเว็บแอปพลิเคชัน จุดเด่นที่สำคัญของ JSP อยู่ที่การใช้ภาษาจาวา ซึ่งเป็นภาษาเชิงวัตถุที่ช่วยให้สามารถพัฒนาแอปพลิเคชันขนาดใหญ่และซับซ้อน ได้อย่างสะดวกและรวดเร็ว ข้อดีของ JSP มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำงานโดยไม่มีขีดจำกัดแพลตฟอร์มใดๆ JSP ได้สืบทอดคุณสมบัติเด่นของจาวา คือ การทำงานโดยไม่ขึ้นกับแพลตฟอร์มหรือระบบปฏิบัติการใดๆ ดังนั้น เมื่อพัฒนาเว็บด้วย JSP ในแพลตฟอร์มหนึ่งก็สามารถย้ายไปใช้งานกับแพลตฟอร์มอื่นๆ ได้ไม่ยาก JSP สามารถเรียกใช้งานจาวา API ได้หลากหลาย ซึ่งจาวา API คือกลุ่มของคลาสที่ช่วยอำนวยความสะดวกในการใช้งานต่างๆ เช่นการจัดการเกี่ยวกับเน็ตเวิร์ค, การติดต่อฐานข้อมูล, การจัดการทางด้านกราฟิก และการจัดการทางด้านออบเจ็คต่างๆ

- นำคอมโพเนนท์กลับมาใช้ได้ อีก ไม่ต้องเสียเวลาสร้างใหม่สามารถนำจาวาบีบมาใช้งานร่วมกับสคริปต์ JSP เพราะจาวาบีบเป็นคอมโพเนนท์ที่เขียนขึ้นมาเพื่อใช้สำหรับทำงานหรือทำหน้าที่อย่างใดอย่างหนึ่ง และสามารถนำกลับมาใช้งานได้เสมอ ไม่ต้องเขียนสคริปต์ JSP เพื่อทำงานนั้นทุกครั้ง จึงพัฒนาเว็บไซต์เสร็จเร็วขึ้น

- มีความยืดหยุ่นในการใช้งานในการเขียนสคริปต์ JSP สามารถกำหนดแท็กใหม่ขึ้นมาใช้งานให้เหมาะสมกับความต้องการได้ นอกจากนี้ภาษาที่ใช้เขียนสคริปต์ JSP ไม่ได้จำกัดเฉพาะภาษาจาวาเท่านั้น ตามหลักการแล้วสามารถใช้ภาษาอื่นๆ ในการเขียนสคริปต์ได้ รวมทั้งยังสามารถนำไปใช้งานร่วมกับ Extension Markup Language (XML) ได้อย่างมีประสิทธิภาพอีกด้วย

### สิ่งที่มีเพื่อใช้งาน JSP

1. Java 2 SDK, Standard Edition ซึ่งเป็นคอมไพเลอร์ของภาษาจาวาที่ต้องใช้ในการคอมไพล์โค้ดของ JSP ที่แปลงเป็นเซิร์ฟเลทแล้ว ( ซึ่งก็คือการคอมไพล์ไฟล์เซิร์ฟเลทที่มีนามสกุล .java ให้กลายเป็นไฟล์ .class )

2. ทอมแคท ( Tomcat ) เป็นเว็บเซิร์ฟเวอร์ที่สนับสนุนการทำงานของเซิร์ฟเลทและ JSP โดย JSP Container ซึ่งอยู่ในทอมแคทจะทำหน้าที่แปลงไฟล์ .jsp เป็นไฟล์ .java

3. โปรแกรมเอดิเตอร์ทั่วไป เช่น โน้ตแพด ( Notepad ) ในระบบวินโดวส์

### การใช้งาาบีบเสริม JSP

จาวาบีบเป็น “คอมโพเนนท์ของซอฟต์แวร์” ที่เขียนขึ้นด้วยภาษาจาวา หรือจาวาบีบ เป็นคลาสของภาษาจาวา ซึ่งจาวาบีบที่สร้างขึ้นนี้จะต้องเป็นไปตามข้อกำหนดของจาวาบีบ API

### บทที่ 3

## การวิเคราะห์และออกแบบแอปพลิเคชันเฟรมเวิร์คสำหรับระบบยืม-คืน

### 3.1 ขอบเขตของระบบ

ออกแบบและพัฒนาแอปพลิเคชันเฟรมเวิร์คสำหรับนำไปสร้างคอมโพเนนต์เพื่อใช้ในการพัฒนาโปรแกรมสำหรับระบบงานยืม-คืนทั่วไป เช่น ระบบงานห้องสมุด, การเช่ารถยนต์, การเช่าห้องของโรงแรม โดยผ่านทางเว็บ ซึ่งมีการจัดการพื้นฐานที่คล้ายกันในส่วนที่เป็น การยืม, การคืน, การสืบค้น และการชำระเงินจากการเช่า - ยืม

### 3.2 การศึกษาและวิเคราะห์ความต้องการทั่วไปของระบบยืม-คืน

ความต้องการของระบบยืม-คืน ได้แก่ การให้บริการยืมและคืนทรัพยากรแก่ผู้ยืมที่เป็นสมาชิก โดยทรัพยากรในระบบยืม-คืนต้องลงทะเบียนไว้ในระบบด้วย ซึ่งเจ้าหน้าที่จะเป็นผู้ติดต่อกับผู้ยืมและการใช้งานระบบ ความต้องการทั่วไปสำหรับระบบยืม-คืน แบ่งเป็นประเภทต่างๆ ได้ดังนี้

#### 3.2.1 ความสามารถด้านการจัดการฐานข้อมูลของทรัพยากรในระบบยืม-คืน

- สามารถเพิ่มเติมทรัพยากรเข้าไปในฐานข้อมูลทรัพยากรของระบบยืม-คืนได้
- สามารถปรับปรุงรายละเอียดของทรัพยากรที่มีอยู่ในฐานข้อมูลทรัพยากรของระบบยืม-คืนได้
- สามารถลบทรัพยากรที่มีอยู่ในฐานข้อมูลทรัพยากรของระบบยืม-คืนได้

#### 3.2.2 ความสามารถด้านการจัดการฐานข้อมูลสมาชิกในระบบยืม-คืน

- สามารถเพิ่มข้อมูลสมาชิก เข้าไปในฐานข้อมูลสมาชิกได้
- สามารถปรับปรุงรายละเอียดของสมาชิก ที่อยู่ในฐานข้อมูลสมาชิกได้
- สามารถลบข้อมูลสมาชิกที่มีอยู่ในฐานข้อมูลสมาชิกได้

#### 3.2.3 ความสามารถด้านการจัดการระบบยืม-คืนทรัพยากรในระบบยืม-คืน

- สามารถเพิ่มข้อมูลการยืมทรัพยากร เข้าในฐานข้อมูลการยืม-คืน
- สามารถลบข้อมูลการยืมทรัพยากรในฐานข้อมูลการยืม-คืนทรัพยากรภายในระบบเมื่อสมาชิกทำการคืนทรัพยากร

#### 3.2.4 ความสามารถด้านการค้นหาทรัพยากรในระบบยืม-คืน

- สามารถค้นหาข้อมูลทรัพยากรภายในระบบ โดยใช้คีย์ต่างๆของทรัพยากรได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

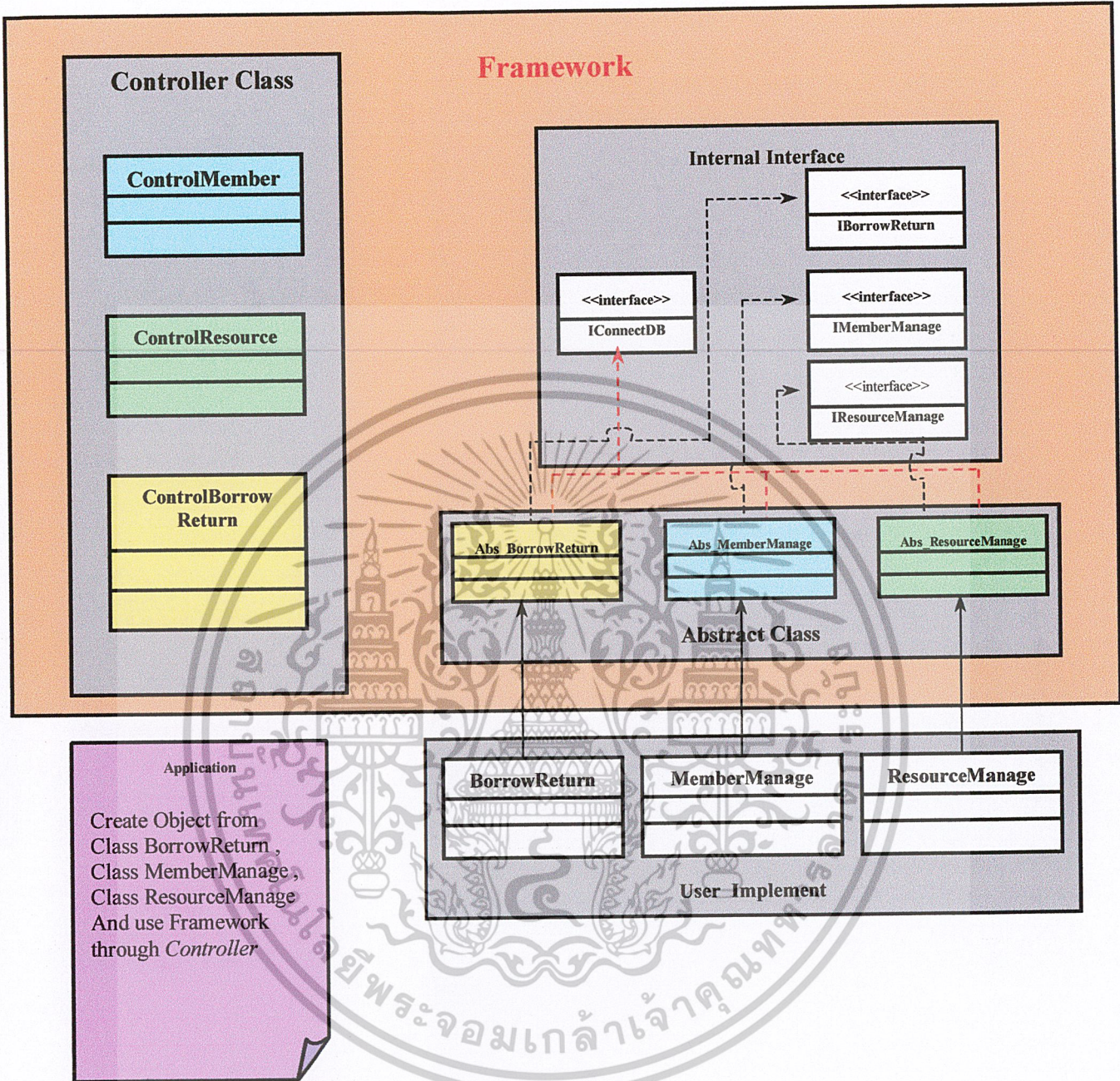
### 3.3 ความต้องการเฉพาะของระบบยืม-คืนแต่ละระบบ

- สามารถกำหนดเงื่อนไขการยืม-คืนได้
- ในบางระบบเมื่อทำการคืนทรัพยากรอาจไม่ต้องการลบข้อมูลการยืมในฐานข้อมูล สามารถตั้งค่าแฟลกในการตัดสินใจว่าจะลบข้อมูลการยืมในฐานข้อมูลนั้นหรือไม่
- สามารถกำหนดอัตราค่าบริการในการยืมได้

### 3.4 การออกแบบแอปพลิเคชันเฟรมเวิร์คสำหรับระบบยืม-คืน

จากการวิเคราะห์ความต้องการของระบบยืม-คืนทั่วไป จะมีกิจกรรมหลักต่างๆ คือ การยืม-คืน การจัดการฐานข้อมูลทรัพยากร การจัดการฐานข้อมูลสมาชิก ส่วนรายละเอียดปลีกย่อยอื่นๆ จะขึ้นอยู่กับนโยบายการดำเนินงานของแต่ละระบบยืม-คืน

ในการออกแบบแอปพลิเคชันเฟรมเวิร์คมีวัตถุประสงค์เพื่อช่วยการทำงานหลักของระบบการยืม-คืนซึ่งเป็นงานเกี่ยวข้องกับฐานข้อมูล ได้แก่ การเพิ่มข้อมูลลงในฐานข้อมูล การค้นหาข้อมูลในฐานข้อมูล การแก้ไขข้อมูลในฐานข้อมูล และการลบข้อมูลในฐานข้อมูล และสามารถควบคุมลำดับการทำงานภายในระบบ โดยในการออกแบบแอปพลิเคชันเฟรมเวิร์คจะนำเสนอการออกแบบโดยใช้ไดอะแกรม ดังรูปที่ 3.1



รูปที่ 3.1 การออกแบบเฟรมเวิร์คแอปพลิเคชันระบบยืม-คืน  
 หมายเหตุ : รายละเอียดของเฟรมเวิร์คที่ ภาคผนวก ข

**3.4.1 Internal Interface**

เป็นคลาสที่มีไว้เพื่อกำหนดรูปแบบของแอ็บสเตรคคลาสภายในเฟรมเวิร์ค เพื่อให้เป็นมาตรฐานของรูปแบบเมธอดต่างๆ ของคลาสที่สืบทอดแอ็บสเตรคคลาสต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1.1 อินเทอร์เฟซ IConnectDB เป็นอินเทอร์เฟซ ที่มีเมธอดต่างๆ ที่ทำหน้าที่จัดการการติดต่อกับฐานข้อมูล

3.4.1.2 อินเทอร์เฟซ IBorrowReturn เป็นอินเทอร์เฟซ ที่มีเมธอดต่างๆ ที่ทำหน้าที่จัดการการยืม-คืน

3.4.1.3 อินเทอร์เฟซ IMemberManage เป็นอินเทอร์เฟซ ที่มีเมธอดต่างๆ ที่ทำหน้าที่จัดการเกี่ยวกับสมาชิก

3.4.1.4 อินเทอร์เฟซ IResourceManage เป็นอินเทอร์เฟซ ที่มีเมธอดต่างๆ ที่ทำหน้าที่จัดการเกี่ยวกับทรัพยากร

### 3.4.2 Abstract Class

แอปสแตรกคลาสจะอิมพลิเมนต์มาจากอินเทอร์เฟซอินเทอร์เฟซเมื่อผู้พัฒนาต้องการใช้เฟรมเวิร์กจะต้องสร้างคลาสที่สืบทอดมาจากแอปสแตรกคลาสเหล่านี้มาพัฒนาเป็นคลาสที่นำไปใช้งานและสร้างออบเจ็กต์ หรือ สร้างส่วนติดต่อกับแอปพลิเคชันต่อไป

3.4.2.1 แอปสแตรกคลาส Abs\_BorrowReturn ได้มาจากการอิมพลิเมนต์ IConnectDB และ IBorrowReturn ซึ่งทำหน้าที่จัดการและการติดต่อกับฐานข้อมูลเกี่ยวกับการยืม-คืน

3.4.2.2 แอปสแตรกคลาส Abs\_MemberManage ได้มาจากการอิมพลิเมนต์ IConnectDB และ IMemberManage ซึ่งทำหน้าที่จัดการและติดต่อกับฐานข้อมูลสมาชิก

3.4.2.3 แอปสแตรกคลาส Abs\_ResourceManage ได้มาจากการอิมพลิเมนต์ IConnectDB และ IResourceManage ซึ่งทำหน้าที่จัดการและติดต่อกับฐานข้อมูลทรัพยากร

เมธอดต่างๆ ภายในแอปสแตรกคลาสที่มีโค้ดมาให้แล้ว ได้แก่ เมธอดที่ทำหน้าที่ติดต่อและจัดการฐานข้อมูล เช่น ติดต่อกับฐานข้อมูล เพิ่มข้อมูล แก้ไข สืบค้น ลบข้อมูลในฐานข้อมูล และเลิกการติดต่อกับฐานข้อมูล ดังนี้

เมธอดเหล่านี้ได้มาจากการอิมพลิเมนต์อินเทอร์เฟซ IConnectDB นั่นเอง ซึ่งผู้พัฒนาสามารถทำการสร้างโค้ดทับเมธอดเหล่านี้ได้ ถ้าต้องการเปลี่ยนแปลงชนิดของฐานข้อมูล เช่น ต้องการให้ทำงานร่วมกับฐานข้อมูล MySQL หรือ Oracle

เมธอดอื่นๆภายในแอปสแตรกคลาสนอกเหนือจากที่กล่าวมาแล้ว ผู้พัฒนาต้องสร้างโค้ดขึ้นเองเนื่องจาก ที่อยู่และชนิดของฐานข้อมูล รวมทั้งคีย์หลักของตารางสมาชิก ตารางทรัพยากร ตารางการยืม-คืน ในฐานข้อมูล เงื่อนไขการยืมของสมาชิกในระบบ เงื่อนไขการยืมทรัพยากรของระบบ เงื่อนไขการชำระเงิน จะขึ้นกับผู้พัฒนาระบบยืม-คืน ได้แก่

**หมายเหตุ :** ดูรายละเอียดของเมธอดในเฟรมเวิร์ก ที่ภาคผนวก ข

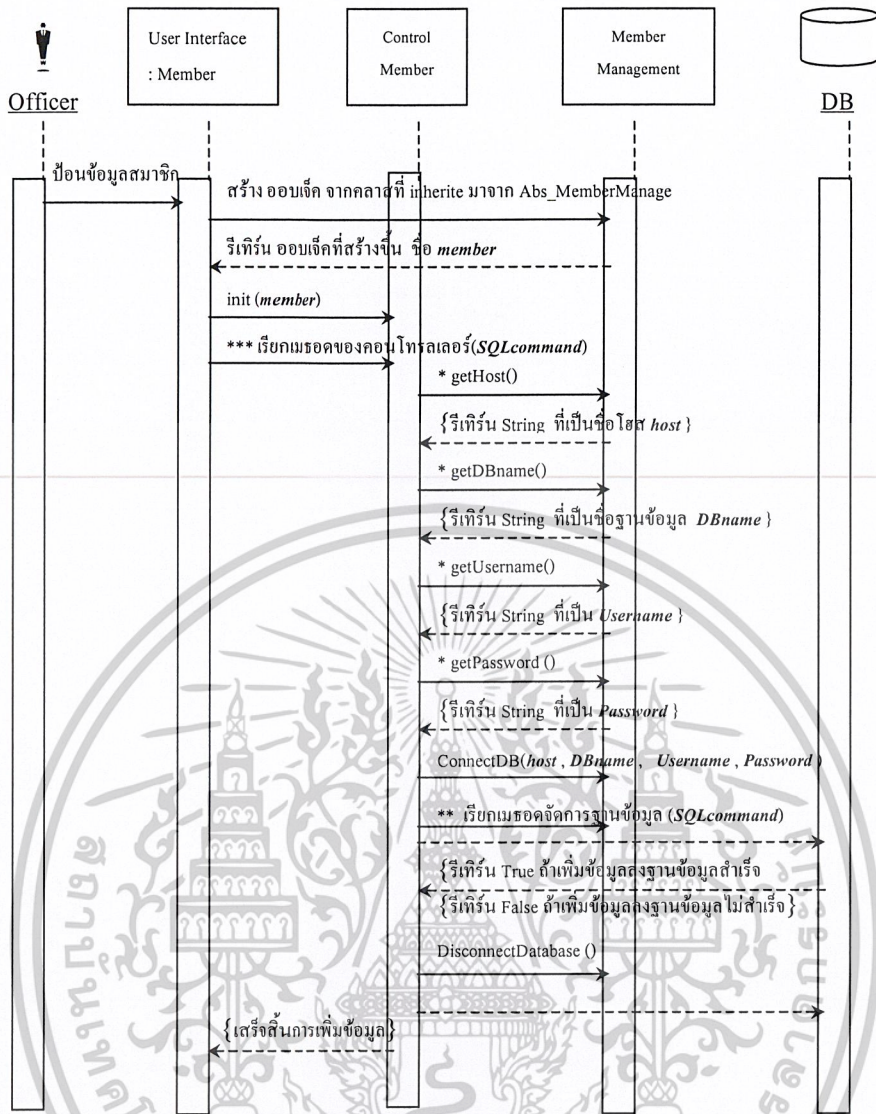
### 3.4.3 Controller Class

การเพิ่ม แก้ไข สืบค้น และลบข้อมูลในฐานข้อมูลโดยทำงานผ่านคอนโทรลเลอร์ ซึ่งคอนโทรลเลอร์จะเป็นผู้เรียกเมธอดของออบเจกต์และดำเนินการตามลำดับ โดยที่การทำงานนี้จำเป็นต้องใช้ที่อยู่ของฐานข้อมูล ชื่อฐานข้อมูล รวมทั้ง ชื่อและรหัสผ่านของผู้ใช้ฐานข้อมูล รวมทั้งคำสั่ง SQL ในการเพิ่ม แก้ไข และ ลบข้อมูลในฐานข้อมูลซึ่งแตกต่างกันออกไป เนื่องจากชื่อคอลัมน์ต่างๆ ในฐานข้อมูลของผู้ใช้แต่ละรายต่างกัน จึงจำเป็นต้องให้ผู้ใช้สร้างคำสั่ง SQL ในการเพิ่ม แก้ไขและลบข้อมูลในฐานข้อมูลแล้วส่งเป็นพารามิเตอร์ที่มีประเภทข้อมูลเป็นสตริงมาให้คอนโทรลเลอร์

ที่อยู่ของฐานข้อมูล ชื่อฐานข้อมูล ชื่อและรหัสผ่านของผู้ใช้ฐานข้อมูล รวมทั้งเงื่อนไขการยืมคืนของผู้ใช้ในแต่ละรายต่างกัน จึงจำเป็นที่ผู้ใช้ต้องสร้างโค้ดในเมธอดต่างๆที่จะวิเทิร์นพารามิเตอร์เหล่านี้มาให้คอนโทรลเลอร์ ได้แก่เมธอดที่มีเครื่องหมาย \* ในซีควนซ์ไคอะแกรม ดังรูปที่จะแสดงต่อไป ส่วนเมธอดจัดการฐานข้อมูลที่มีเครื่องหมาย \*\* ในซีควนซ์ไคอะแกรม จะมีโค้ดอยู่ผู้ใช้ไม่ต้องสร้างโค้ดเอง แต่ถ้าต้องการจะสร้างโค้ดใหม่ทับโค้ดเดิมก็สามารถทำได้

#### 3.4.3.1 คลาส ControlMember

การจัดการสมาชิกจะทำงาน โดยผ่านทางคอนโทรลเลอร์ คอนโทรลเลอร์จะเป็นผู้เรียกเมธอดของออบเจกต์ที่สร้างจากคลาสที่สืบทอดมาจาก Abs MemberManage ซึ่งออบเจกต์จะถูกส่งเข้ามาเก็บไว้ในคลาส ControlMember และดำเนินการทำงานตามลำดับ ภายในคลาส ControlMember มีเมธอดต่างๆ ที่ทำหน้าที่จัดการข้อมูลสมาชิก โดยผู้พัฒนาจะเรียกใช้เมธอดเหล่านี้ แสดงลำดับการทำงานภายใน ControlMember ดังรูปที่ 3.2 ซีควนซ์ไคอะแกรมของการจัดการข้อมูลสมาชิก



รูปที่ 3.2 ซีควเอนซ์ไดอะแกรมของการจัดการข้อมูลสมาชิก  
 หมายเหตุ ผู้ใช้ต้องเขียนโค้ดในเมธอดที่มีเครื่องหมาย \*

จากรูปที่ 3.2 ในส่วนที่มีเครื่องหมาย \*\* ผู้พัฒนาจะต้องเรียกเมธอดที่อยู่ภายในคอนโทรลเลอร์ เพื่อจัดการข้อมูลสมาชิกในฐานข้อมูล โดยส่ง SQLcommand ในการเพิ่ม แก้ไข สืบค้น หรือ ลบข้อมูลในฐานข้อมูล เป็นพารามิเตอร์ไปให้กับคอนโทรลเลอร์ ได้แก่

- runAddMember(insertCMD) เมื่อต้องการเพิ่มข้อมูลสมาชิกลงฐานข้อมูลแล้วคอนโทรลเลอร์ จะเรียกใช้เมธอดที่จัดการฐานข้อมูลคือ insertData(insertCMD) ต่อไป
- runUpdateMember(updateCMD) เมื่อต้องการแก้ไขข้อมูลสมาชิกในฐานข้อมูลแล้ว คอนโทรลเลอร์จะเรียกใช้เมธอดจัดการฐานข้อมูลคือ updateData(updateCMD) ต่อไป

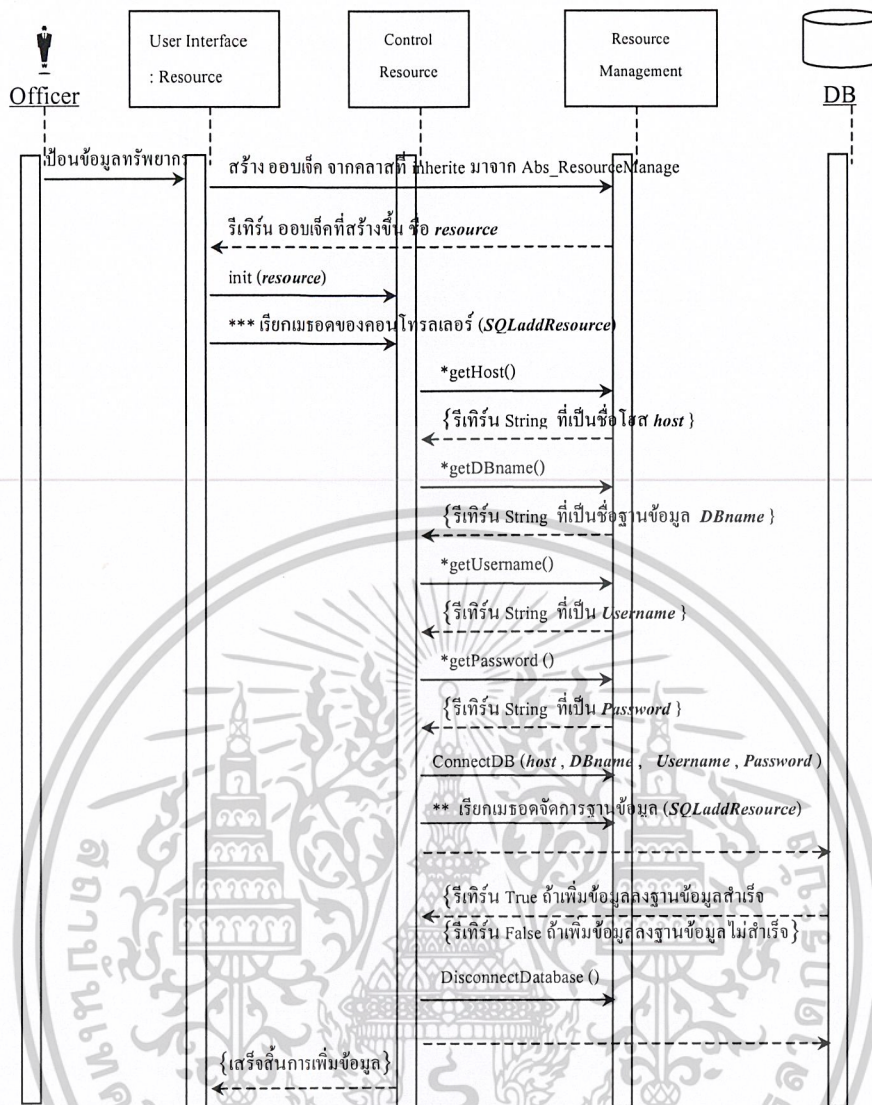
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- `runDeleteMember(deleteCMD)` เมื่อต้องการลบข้อมูลสมาชิกในฐานข้อมูลแล้ว คอนโทรลเลอร์จะเรียกใช้เมธอดที่จัดการฐานข้อมูลคือ `deleteData(deleteCMD)` ต่อไป
- `runSearchMember(searchCMD)` เมื่อต้องการค้นหาข้อมูลสมาชิกในฐานข้อมูลแล้ว คอนโทรลเลอร์จะเรียกใช้เมธอดที่จัดการฐานข้อมูลคือ `queryData(searchCMD)` ต่อไป
- `runSelectMember(searchCMD)` เมื่อต้องการค้นหาและแสดงข้อมูลสมาชิกในฐานข้อมูลแล้ว คอนโทรลเลอร์จะเรียกใช้เมธอดที่จัดการฐานข้อมูลคือ `selectData(searchCMD)` ต่อไป

#### 3.4.3.2 คลาส ControlResource

การจัดการทรัพยากรจะทำงานโดยผ่านทางคอนโทรลเลอร์ คอนโทรลเลอร์จะเป็นผู้เรียกเมธอดของออบเจกต์ที่สร้างจากคลาสที่สืบทอดมาจาก `Abs_ResourceManage` ซึ่งออบเจกต์จะถูกส่งเข้ามาเก็บไว้ในคลาส `ControlResource` และดำเนินการทำงานตามลำดับ ภายในคลาส `ControlResource` มีเมธอดต่างๆ ที่ทำหน้าที่จัดการทรัพยากร โดยผู้พัฒนาจะเรียกใช้เมธอดเหล่านี้ แสดงลำดับการทำงานภายใน `ControlResource` ดังรูปที่ 3.3 ที่เสนอขั้นตอนการของการจัดการข้อมูลทรัพยากร





รูปที่ 3.3 ซีควเอนซ์ไดอะแกรมของการจัดการข้อมูลทรัพยากร

หมายเหตุ ผู้ใช้ต้องเขียนโค้ดในเมธอดที่มีเครื่องหมาย \*

จากรูปที่ 3.3 ในส่วนที่มีเครื่องหมาย \*\* ผู้พัฒนาจะต้องเรียกเมธอดที่อยู่ในคอนโทรลเลอร์ เพื่อจัดการข้อมูลทรัพยากรในฐานข้อมูล โดยส่ง SQLcommand ในการเพิ่ม แก้ไข สืบค้น หรือ ลบข้อมูลในฐานข้อมูล เป็นพารามิเตอร์ไปให้กับคอนโทรลเลอร์ ได้แก่

- runAddResource(insertCMD) เมื่อต้องการเพิ่มข้อมูลทรัพยากรลงฐานข้อมูลแล้วคอนโทรลเลอร์จะเรียกใช้เมธอดที่จัดการฐานข้อมูลคือ insertData(insertCMD) ต่อไป
- runUpdateResource(updateCMD) เมื่อต้องการแก้ไขข้อมูลทรัพยากรในฐานข้อมูลแล้วคอนโทรลเลอร์จะเรียกใช้เมธอดจัดการฐานข้อมูลคือ updateData(updateCMD) ต่อไป

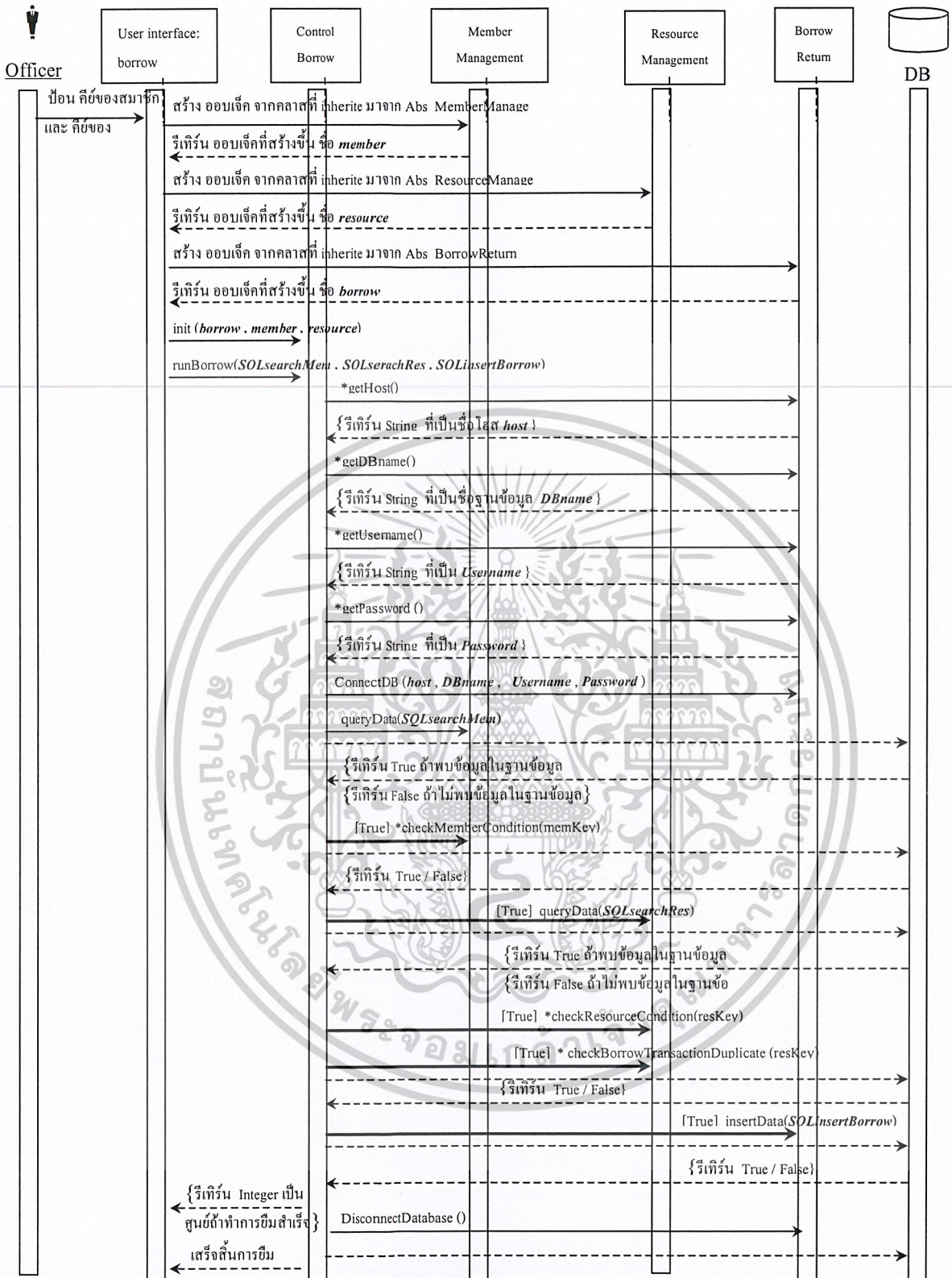
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- `runDeleteResource(deleteCMD)` เมื่อต้องการลบข้อมูลทรัพยากรในฐานข้อมูลแล้ว คอนโทรลเลอร์จะเรียกใช้เมธอดที่จัดการฐานข้อมูลคือ `deleteData(deleteCMD)` ต่อไป
- `runSearchResource(searchCMD)` เมื่อต้องการค้นหาข้อมูลทรัพยากรในฐานข้อมูลแล้ว คอนโทรลเลอร์จะเรียกใช้เมธอดที่จัดการฐานข้อมูลคือ `queryData(searchCMD)` ต่อไป
- `runSelectResource (searchCMD)` เมื่อต้องการค้นหาและแสดงข้อมูลทรัพยากรในฐานข้อมูลแล้ว คอนโทรลเลอร์จะเรียกใช้เมธอดที่จัดการฐานข้อมูลคือ `selectData(searchCMD)` ต่อไป

#### 3.4.3.3 คลาส ControlBorrowReturn

การจัดการยืม-คืนทรัพยากรจะทำงานโดยผ่านทางคอนโทรลเลอร์ คอนโทรลเลอร์จะเป็นผู้เรียกเมธอดของออบเจกต์ที่สร้างจากคลาสที่สืบทอดมาจาก `Abs_BorrowReturn` ซึ่งออบเจกต์จะถูกส่งเข้ามาเก็บไว้ในคลาส `ControlBorrowReturn` และดำเนินการทำงานตามลำดับ ภายในคลาส `ControlBorrowReturn` มีเมธอดต่างๆ ที่ทำหน้าที่จัดการยืม-คืนทรัพยากร โดยผู้พัฒนาจะเรียกใช้เมธอดเหล่านี้ ซึ่งจะแสดงลำดับการทำงานในการจัดการยืม-คืนภายใน `ControlBorrowReturn` ดังรูปที่ 3.4 และ รูปที่ 3.5





รูปที่ 3.4 ซีเควนซ์ไดอะแกรมของการยืมทรัพยากร

หมายเหตุ ผู้ใช้ต้องเขียนโค้ดในเมธอดที่มีเครื่องหมาย \*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

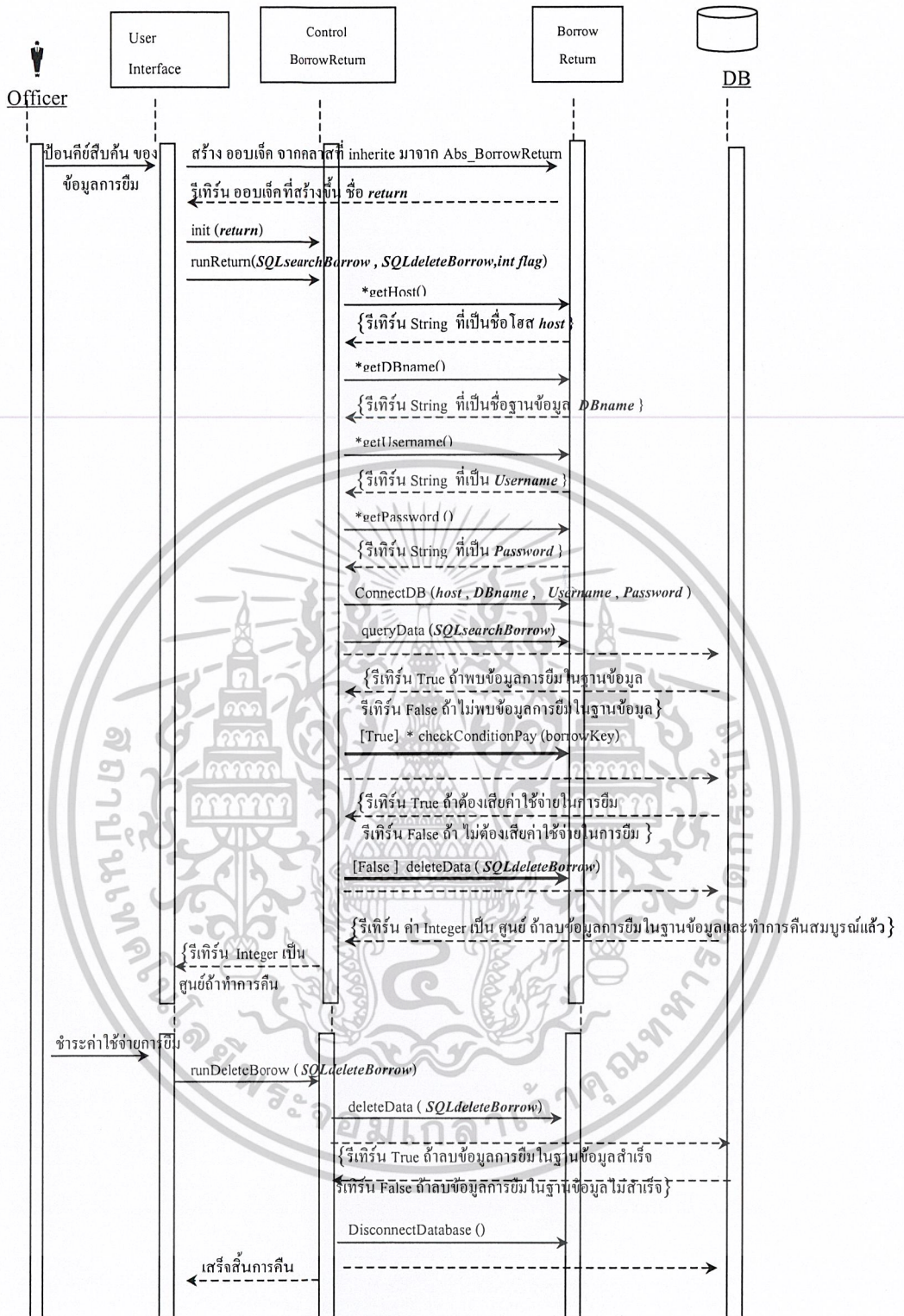
จากรูปที่ 3.4 ผู้พัฒนาจะต้องเรียกเมธอดที่อยู่ภายในคอนโทรลเลอร์เพื่อจัดการยืมทรัพยากรในฐานข้อมูล คือ

- `runBorrow(SQLsearchMem , SQLserachRes , SQLinsertBorrow)` โดยส่งพารามิเตอร์เป็น `SQLcommand` ในการสืบค้นสมาชิกและทรัพยากรที่ต้องการยืม รวมทั้ง `SQLcommand` ในการเพิ่มข้อมูลการยืมลงฐานข้อมูล จากนั้นคอนโทรลเลอร์จำเป็นผู้ควบคุมการดำเนินการตามขั้นตอนของการยืมต่อไป โดยการเรียกเมธอดต่างๆตามลำดับดังซีเควนซ์โคโอะแกรม ซึ่งจะอธิบายหน้าที่ของเมธอดที่สำคัญในการจัดการยืมทรัพยากร ดังนี้

หน้าที่ของเมธอด `checkMemberCondition(memKey)` คือ จะตรวจสอบเงื่อนไขการยืมของผู้ยืม เช่น ชื่อกำหนด หรือ ชื่อจำกัด ในการได้รับอนุญาตให้ยืมทรัพยากร เช่น การเป็นสมาชิก ประเภทของสมาชิก จำนวนทรัพยากรขั้นต่ำที่อนุญาตให้สมาชิกแต่ละประเภท จำนวนทรัพยากรที่สมาชิกคนนั้นยังไม่คืน โดยจะส่งพารามิเตอร์ที่เป็นคีย์ของสมาชิกคนนั้นไปให้กับคอนโทรลเลอร์

หน้าที่ของเมธอด `checkResourceCondition(resKey)` คือ จะตรวจสอบเงื่อนไขการยืมของทรัพยากร เช่น ทรัพยากรที่ต้องการยืมชำรุดหรือไม่ว่างให้ยืม โดยจะส่งพารามิเตอร์ที่เป็นคีย์ของทรัพยากรนั้นไปให้กับคอนโทรลเลอร์

หน้าที่ของเมธอด `checkBorrowTransactionDuplicate(resKey)` คือ จะตรวจสอบว่าทรัพยากรนั้นเคยถูกยืมไปแล้วหรือไม่ เพื่อป้องกันการยืมซ้ำ โดยจะส่งพารามิเตอร์ที่เป็นคีย์ของทรัพยากรนั้นไปให้กับคอนโทรลเลอร์



รูปที่ 3.5 ซีเควนซ์ไดอะแกรมของกรณีกินทรัพย์สิน

หมายเหตุ ผู้ใช้ต้องเขียนโค้ดในเมธอดที่มีเครื่องหมาย \*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.5 ผู้พัฒนาจะต้องเรียกเมธอดที่อยู่ภายในคอนโทรลเลอร์เพื่อจัดการยืมทรัพยากรในฐานะข้อมูล คือ

- `runReturn(SQLsearchBorrow , SQLdelete_updateBorrow, int flag)` โดยส่งพารามิเตอร์เป็น SQLcommand ในการสืบค้นข้อมูลการยืม รวมทั้ง SQLcommand ในการลบข้อมูลการยืมในฐานะข้อมูล จากนั้นคอนโทรลเลอร์จำเป็นผู้ควบคุมการดำเนินการตามขั้นตอนของการคืนต่อไป โดยการเรียกเมธอดต่างๆตามลำดับดังซีควเอนซ์ไดอะแกรม ซึ่งจะอธิบายหน้าที่ของเมธอดที่สำคัญในการจัดการคืนทรัพยากร ดังนี้

หน้าที่ของเมธอด `checkConditionPay(borrowKey)` จะตรวจสอบเงื่อนไขการชำระเงิน ได้แก่ ข้อกำหนดของระบบยืม-คืนนั้นๆ เช่น มีการระบุให้มีการเก็บชำระเงินหรือไม่ อัตราค่าบริการเท่าไร รวมทั้งข้อกำหนดอื่นๆ เช่น จะคิดค่าใช้จ่ายหลังการยืมทรัพยากรไปแล้วกี่วัน (บางระบบ เรียกว่า การเสียค่าปรับ หรือ ค่าบริการ ) ฯลฯ ถ้าเงื่อนไขการชำระเงินถูกตรวจสอบแล้วส่งค่ากลับมาเป็น False คอนโทรลเลอร์จะลบข้อมูลการยืมครั้งนั้นถ้าพารามิเตอร์ flag เป็นศูนย์ และเสร็จสิ้นการคืน แต่ถ้าเงื่อนไขการชำระเงินถูกตรวจสอบแล้วส่งค่ากลับมาเป็น True ผู้คืนต้องชำระเงินก่อน จากนั้นผู้พัฒนาจะเรียกใช้เมธอด `runDeleteBorrow` ถ้าต้องการลบข้อมูลการยืม หรือ ผู้พัฒนาจะเรียกใช้เมธอด `runUpdateBorrow` ถ้าต้องการเพียงตั้งค่าสังเกตในบางคอลัมน์ของข้อมูลการยืมในฐานะข้อมูลว่าการยืมครั้งนั้นสมาชิกนำทรัพยากรมาคืนแล้ว สามารถดูรายละเอียดเพิ่มเติมของ 2 เมธอดนี้ที่ภาคผนวก ข

## บทที่ 4

### การพัฒนาและทดสอบแอปพลิเคชันเฟรมเวิร์ค

การดำเนินการวิจัยของปัญหาพิเศษเรื่อง แอปพลิเคชันเฟรมเวิร์คของระบบยืม-คืน ในขั้นตอนการพัฒนาและทดสอบแอปพลิเคชันเฟรมเวิร์ค จะแสดงโค้ดภายในคลาสและอินเทอร์เฟซต่างๆของเฟรมเวิร์คที่ได้ออกแบบไว้และจะทดสอบเฟรมเวิร์คโดยการสร้างระบบห้องสมุดขึ้น ดังต่อไปนี้

#### 4.1 การพัฒนาแอปพลิเคชันเฟรมเวิร์ค

ภายในแอปพลิเคชันเฟรมเวิร์คประกอบไปด้วยส่วนต่างๆ ดังนี้

##### 4.1.1 คลาสอินเทอร์เฟซภายในเฟรมเวิร์ค

ภายในเฟรมเวิร์คมี 4 อินเทอร์เฟซ ได้แก่ IConnectDB IMemberManage IResourceManage และ IBorrowReturn ต่อไปจะแสดงอินเทอร์เฟซ IBorrowReturn โดยใช้ภาษาจาวา ดังตัวอย่างที่ 4.1

#### ตัวอย่างที่ 4.1 อินเทอร์เฟซ IBorrowReturn

```
1 public interface IBorrowReturn
2 {
3     public String getHost();
4     public String getDBname();
5     public String getUsername();
6     public String getPassword();
7     public String getBorrowKey();
8     public void setRate(float rate);
9     /*====Return====*/
10    public boolean checkBorrowTransactionDuplicate(String resKey);
11    public boolean checkConditionPay(String borrowKey);
12 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างที่ 4.1 อธิบายโค้ดได้ ดังนี้

บรรทัดที่ 3 ถึง 6 เป็นเมธอดที่ต้องรีเทิร์นที่อยู่ของฐานข้อมูลผู้ใช้

บรรทัดที่ 8 เป็นเมธอดที่ผู้พัฒนาใช้ในการกำหนดอัตราค่าบริการ

บรรทัดที่ 10 เป็นเมธอดที่ใช้ตรวจสอบเพื่อป้องกันการเกิดทรานแซกชันการยืมทรัพยากร  
ขึ้นเดียวกันซ้ำหลายครั้ง

บรรทัดที่ 11 เป็นเมธอดที่ตรวจสอบเงื่อนไขการชำระเงิน เมธอดต่างๆที่อยู่ภายในอิน  
เทอร์เฟซ IBorrowReturn นี้ถูกกำหนดไว้ในเฟรมเวิร์คเพื่อเป็นต้นแบบให้กับแอ็บสแตรกคลาส Abs\_  
Borrow Return ดังในตัวอย่างที่ 4.2

#### 4.1.2 แอ็บสแตรกคลาสภายในเฟรมเวิร์ค

ภายในเฟรมเวิร์คมี 3 แอ็บสแตรกคลาสได้แก่ Abs\_MemberManage Abs\_Resource Manage  
Abs\_BorrowReturn ต่อไปจะแสดงแอ็บสแตรกคลาส Abs\_BorrowReturn โดยใช้ภาษาจาวา ดังตัว  
อย่าง ที่ 4.2

#### ตัวอย่างที่ 4.2 แอ็บสแตรกคลาส Abs\_BorrowReturn

```

1 abstract public class Abs_BorrowReturn implements IConnectDB , IBorrowReturn
2 { public String username;
3   public String password;
4   public Connection con;
5   public Statement stmt;
6   public ResultSet rs;
7   /*-----Connect DB MySQL-----*/
8   public boolean ConnectDB(String host, String database, String username, String
9   password)
10  {
11  try
12  {
13      Class.forName("com.mysql.jdbc.Driver");
14      con=DriverManager.getConnection("jdbc:mysql://" + host + "/" + database +

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

15     "?user="+username+"&password="+password);
16     stmt = con.createStatement();
17     return true;
18 }
19 catch (java.lang.ClassNotFoundException e)
20 {
21     System.err.println("ClassNotFoundException: ");
22     System.err.println(e.getMessage());
23     return false;
24 }
25 catch (SQLException ex)
26 {
27     System.err.println("SQLException ConnectDB: "+ex.getMessage());
28     return false;
29 }
30 }
31 /*-----*/
32 public void disconnectDatabase()
33 {
34     try
35     {
36         stmt.close();
37         con.close();
38         System.out.println(".....Disconnect DB");
39     }
40     catch (SQLException ex)
41         System.err.println("SQLException DisconnectDB: "+ex.getMessage());
42     }
43 }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอดต่างๆ ที่แสดงดังตัวอย่างที่ 4.2 ภายใน Abs\_BorrowReturn ที่เกี่ยวข้องกับการติดต่อและจัดการฐานข้อมูลซึ่งมีโค้ดมาให้แล้ว โดยจะเป็นฐานข้อมูล MySQL ดังบรรทัดที่ 13 ซึ่งผู้พัฒนาสามารถสืบทอดแอบสแทรกคลาสนี้ แล้วสร้างโค้ดใหม่ทับโค้ดเดิมได้ หากต้องการใช้ฐานข้อมูลอื่น สามารถดูรายละเอียดโค้ดภายในเมธอดเพิ่มเติมได้ใน CD-Rom ของปัญหาพิเศษนี้

#### 4.1.3 คลาสคอนโทรลเลอร์ภายในเฟรมเวิร์ค

ภายในเฟรมเวิร์คมี 3 คลาสคอนโทรลเลอร์ ได้แก่ ControlMember ControlResource ControlBorrowReturn แต่ละคลาสมีเมธอดในการจัดการระบบยืม-คืนคนละส่วน โดยจะแสดงโค้ดของเมธอด runBorrow ในคลาส ControlBorrowReturn ที่จัดการยืมทรัพยากร โดยใช้ภาษาจาวา ดังตัวอย่างที่ 4.3

#### ตัวอย่างที่ 4.3 เมธอด runBorrow ในคลาส ControlBorrowReturn

```

1 public int runBorrow(String CMDsearchMem ,String CMDserachRes ,String
2     CMDinsertBorrow)
3 {
4     System.out.println(".....Runing Control Borrow");
5     if( borrow.ConnectDB(borrow.getHost(),borrow.getDBname(),borrow.getUsername(),
6         borrow.getPassword()))
7     { /*ABOUT MEMBER*/
8         member.ConnectDB(member.getHost(),member.getDBname(),member.getUsername(),
9             member.getPassword());
10        resource.ConnectDB(resource.getHost(),resource.getDBname(),resource.getUsername(),
11            resource.getPassword());
12        System.out.println(".....Connecting DB");
13        if(member.queryData(CMDsearchMem)
14        {
15            System.out.println(".....There is this member in DB");
16            if (member.checkMemberCondition(member.getMemKey()))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

17 {
18     System.out.println(".....This member allow to borrow");
19     /*ABOUT RESOURCE*/
20     if(resource.queryData(CMDserachRes))
21     {
22         System.out.println(".....There is this resource in DB");
23         if (resource.checkResourceCondition(resource.getResKey())) /*#####*/
24         {
25             if(!borrow.checkBorrowTransactionDuplicate(resource.getResKey()))
26             {
27                 System.out.println(".....This resource allow to borrow");
28                 /*ABOUT Borrowing*/
29                 if(borrow.insertData(CMDinsertBorrow))
30                 {
31                     System.out.println(".....Inserted borrow-transaction into DB");
32                     System.out.println(".....Borrow Process Completed");
33                     borrow.disconnectDatabase();
34                     member.disconnectDatabase();
35                     resource.disconnectDatabase();
36                     return 0;
37                 }
38             else
39             {
40                 System.out.println(".....Can not insert borrow-transaction.May be incorrect
41                 insertCMD");
42                 borrow.disconnectDatabase();
43                 member.disconnectDatabase();
44                 resource.disconnectDatabase();
45                 return 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

46     }
47 } // checkBorrowTransactionDuplicate
48 else
49     return 2;
50 }//checkResourceCondition
51 else
52 {
53     System.out.println(".....This resource don't allow to borrow");
54     borrow.disconnectDatabase();
55     member.disconnectDatabase();
56     resource.disconnectDatabase();
57     return 2;
58 }
59 }
60 else
61 {
62     System.out.println(".....There isn't this resource in DB");
63     borrow.disconnectDatabase();
64     member.disconnectDatabase();
65     resource.disconnectDatabase();
66     return 3;
67     }/*=====*/
68 }
69 else
70 {
71     System.out.println(".....This member don't allow to borrow");
72     borrow.disconnectDatabase();
73     member.disconnectDatabase();
74     resource.disconnectDatabase();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

75         return 4;
76     }
77 }
78 else
79 {
80     System.out.println(".....There isn't this member in DB");
81     borrow.disconnectDatabase();
82     member.disconnectDatabase();
83     resource.disconnectDatabase();
84     return 5;
85 }
86 }
87 else
88 {
89     System.out.println(".....Can not connect DB");
90     System.out.println(".....EXIT");
91     return 6;
92 }
93 }

```

จากตัวอย่างที่ 4.3 ภายในเมธอด runBorrow จะจัดลำดับการทำงานเพื่ดำเนินการยืม โดยการเรียกเมธอดต่างๆที่ผู้พัฒนาสร้างขึ้น ดังที่ได้อธิบายลำดับการทำงานไว้ในซีควেনซ์ไดอะแกรมรูปที่ 3.4 ของบทที่ 3 โดยจะอธิบายโค้ดในตัวอย่างที่ 4.3 ดังนี้

บรรทัดที่ 5,6,8,9,10,11 เป็นการติดต่อฐานข้อมูล

บรรทัดที่ 13 เป็นการสืบค้นสมาชิกว่ามีอยู่จริงในฐานข้อมูล

บรรทัดที่ 16 เป็นการตรวจสอบเงื่อนไขการยืมของสมาชิกคนนั้น

บรรทัดที่ 20 เป็นการสืบค้นทรัพยากรว่ามีอยู่จริงในฐานข้อมูล

บรรทัดที่ 23 เป็นการตรวจสอบเงื่อนไขการยืมทรัพยากรชิ้นนั้น

บรรทัดที่ 25 เป็นการตรวจสอบเพื่อป้องกันการเกิดทรานแซกชันยืมทรัพยากรชิ้นเดิมซ้ำกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรทัดที่ 29 เป็นการเพิ่มทรานแซกชันการยืมลงฐานข้อมูล  
 บรรทัดที่ 36 รีเทิร์นค่าเป็นศูนย์เมื่อการยืมเสร็จสิ้นโดยสมบูรณ์  
 บรรทัดที่ 45 รีเทิร์นค่าเป็น 1 เมื่อไม่สามารถเพิ่มทรานแซกชันการยืมลงฐานข้อมูลได้  
 บรรทัดที่ 49 , 57 รีเทิร์นค่าเป็น 2 เมื่อทรัพยากรไม่สามารถให้ยืมได้  
 บรรทัดที่ 66 รีเทิร์นค่าเป็น 3 เมื่อไม่พบทรัพยากรนั้นในฐานข้อมูล  
 บรรทัดที่ 75 รีเทิร์นค่าเป็น 4 เมื่อไม่อนุญาตให้สมาชิกคนนั้นยืมทรัพยากร  
 บรรทัดที่ 84 รีเทิร์นค่าเป็น 5 เมื่อไม่พบสมาชิกคนนั้นในฐานข้อมูล  
 บรรทัดที่ 91 รีเทิร์นค่าเป็น 6 เมื่อไม่สามารถติดต่อฐานข้อมูลได้  
 สามารถดูโค้ดของเมธอดอื่นๆภายในคลาส ControlBorrow Return เพิ่มเติมได้ใน CD- Rom  
 ของปัญหาพิเศษนี้

## 4.2 การทดสอบแอปพลิเคชันเฟรมเวิร์ค

ทดสอบแอปพลิเคชันเฟรมเวิร์คโดยการสร้างระบบห้องสมุดขึ้น มีขั้นตอนดังนี้

### 4.2.1 การสร้างคลาสที่สืบทอดมาจากแอ็บสแตรกคลาสในเฟรมเวิร์ค

การทำงานผ่านคอนโทรลเลอร์ของแอปพลิเคชันเฟรมเวิร์ค ผู้พัฒนาจำเป็นต้องสืบทอดคลาส  
 จากแอ็บสแตรกคลาสในเฟรมเวิร์ค โดยจะต้องสร้างโค้ดภายในเมธอดที่เฟรมเวิร์คกำหนดเครื่องหมาย\*  
 ไว้ดังซีเควนซ์โคแอมแกรมรูปที่ 3.2 3.3 และ 3.4 ในบทที่ 3 ต่อไปจะแสดงตัวอย่างการสร้างคลาสที่  
 สืบทอดมาจากแอ็บสแตรกคลาส Abs\_BorrowReturn โดยใช้ภาษาจาวาได้ดังตัวอย่างที่ 4.4

ตัวอย่างที่ 4.4 ตัวอย่างคลาส BorrowReturn ที่สืบทอดมาจากแอ็บสแตรกคลาส Abs\_BorrowReturn

```

1    import java.sql.*;
2    import java.io.*;
3    import java.util.*;
4    public class BorrowReturn extends Abs_BorrowReturn
5    {
6        private BorrowReturn BR;
7        private MemberManage member;
8        private ControlBorrowReturn conBR;
9        private String command;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

10     private String host = "localhost";
11     private String database = "frameworkproj";
12     private String username = "project";
13     private String password = "framework";
14     private String borrowReturnTable = "BorrowReturnTable" ;
15     private String resID;
16     private float rate;
17     private String payCondition ;
18     public String getHost()                /*...*/
19         return host;
20     public String getDBname()              /*...*/
21         return database;
22     public String getUsername()           /*...*/
23         return username;
24     public String getPassword()           /*...*/
25         return password;
26     public String getBorrowKey()          /*...*/
27         return resID;
28     public void setRate(float rate)       /*...*/
29         this.rate = rate;
30     public boolean checkConditionPay(String resKey) /*...*/
31     { BR = new BorrowReturn();
32       conBR = new ControlBorrowReturn();
33       conBR.init(BR);
34       command = "select * from BorrowReturnTable where ResID = '"+resID+"'";
35       result = conBR.runSelectBorrow(command);
36     try
37     {   if(result.next())
38         payCondition = result.getString("PayCondition");}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

38         payCondition = result.getString("PayCondition");
39     catch (SQLException ex)
40     { System.err.println("SQLException in checkConditionPay: "+ex.getMessage()); }
41     if(payCondition.equals("T"))         return true;
42     else         return false;
43 }
44 public boolean checkBorrowTransactionDuplicate(String resKey)     /*...*/
45 { BR = new BorrowReturn();
46   conBR = new ControlBorrowReturn();
47   conBR.init(BR);
48   command = "select * from BorrowReturnTable where ResID = '"+resKey+"'";
49   result = conBR.runSelectBorrow(command);
50   try
51   {   if(result.next())         return true;
52       else         return false;   }
53   catch (SQLException ex)
54   {System.err.println("SQLException in BorrowReturnRunDB: "+ex.getMessage());
55     return false;   }
56 }

```

หมายเหตุ : สามารถดูตัวอย่างการสร้างคลาสเพื่อใช้งานเฟรมเวิร์คเพิ่มเติมได้ที่ ภาคผนวก ก

จากตัวอย่างที่ 4.4 เมธอดภายในคลาสที่สืบทอดมาจากแอบสแตรกคلاس Abs\_BorrowReturn ผู้พัฒนาจำเป็นต้องสร้างโค้ดขึ้นให้เหมาะสมกับระบบยืม-คืนของผู้พัฒนาเอง โดยจะอธิบายโค้ดในตัวอย่าง 4.4 ดังนี้

- บรรทัดที่ 18 ถึง 25 เป็นเมธอดที่ใช้รีเทิร์นที่อยู่ ชื่อและรหัสผู้ใช้ของฐานข้อมูล
- บรรทัดที่ 26 เป็นเมธอดที่ใช้รีเทิร์นคีย์หลักของตารางยืม-คืนในฐานข้อมูล
- บรรทัดที่ 28 เป็นเมธอดที่ใช้ตั้งอัตราค่าบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

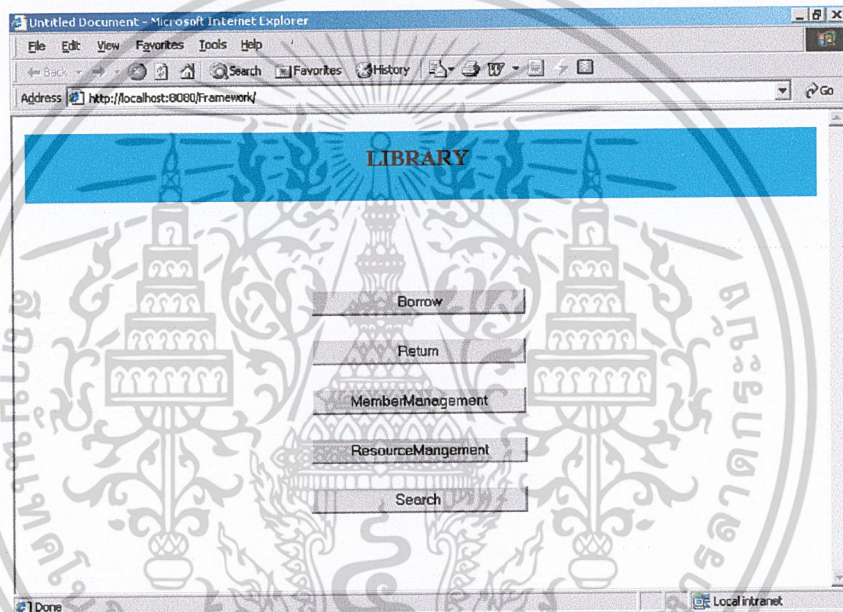
บรรทัดที่ 35 , 49 เป็นการเรียกเมธอด runSelectBorrow -ที่อยู่ภายคลาส ControlBorrow

Return

บรรทัดที่ 44 ตัวอย่างนี้เมธอด checkBorrowTransactionDuplicate จะทำการตรวจสอบทรัพยากรนั้นในฐานข้อมูลการยืม เพื่อป้องกันการเกิดทรานแซกชันของการยืมทรัพยากรที่ยังไม่คืนในฐานข้อมูล ซ้ำอีกครั้ง

#### 4.2.2 การสร้างส่วนติดต่อผู้ใช้งานของห้องสมุด

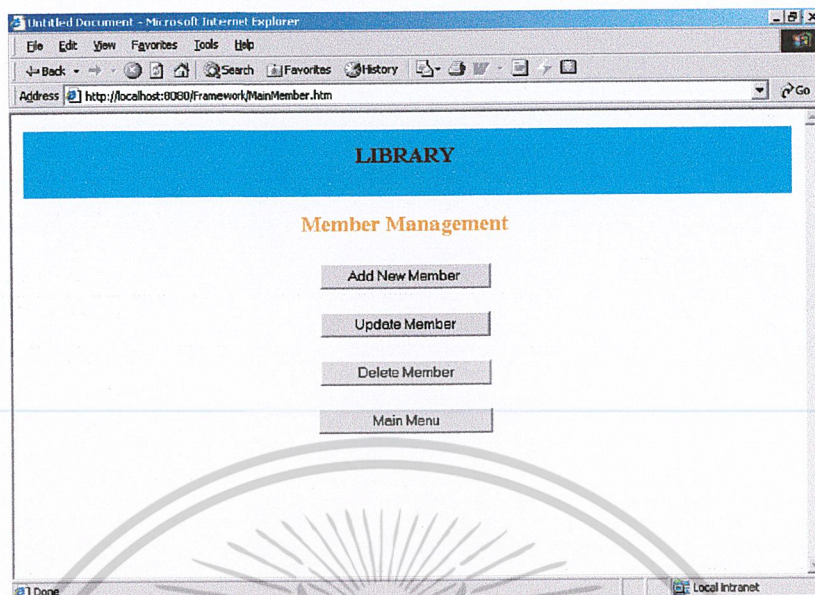
ทำการสร้างส่วนติดต่อผู้ใช้งาน ในที่นี้จะใช้มาโครมีเดียครีมีฟเวอร์เอ็มเอ็ช (Macromedia Dreamweaver MX) ร่วมกับ JSP ดังนี้



รูปที่ 4.1 หน้าจอหลักของส่วนติดต่อผู้ใช้งาน

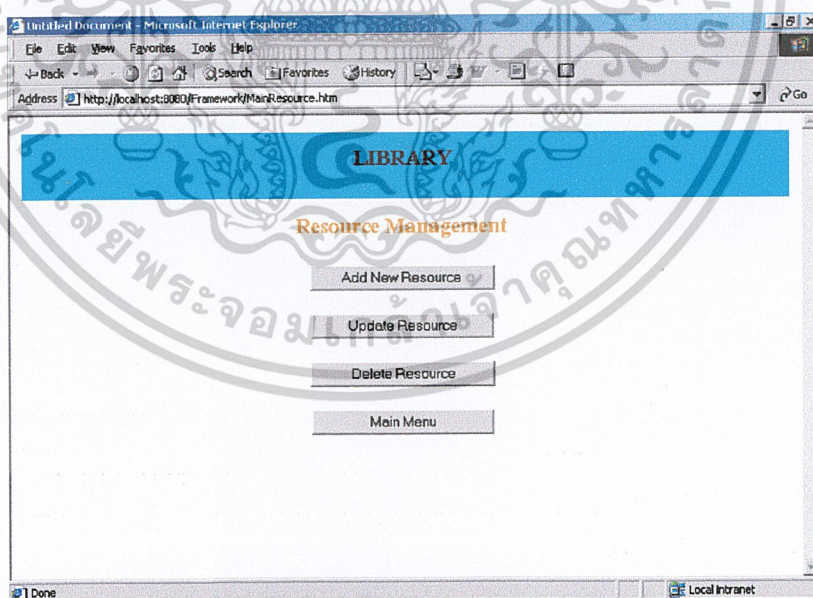
จากรูปที่ 4.1 หน้าจอหลักของส่วนติดต่อผู้ใช้งาน สามารถเลือกการทำงานต่างๆในห้องสมุด ได้แก่ การยืม การคืน การจัดการสมาชิก การจัดการทรัพยากร และการสืบค้นทรัพยากร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 หน้าจอส่วนติดต่อผู้ใช้งานสำหรับจัดการข้อมูลสมาชิก

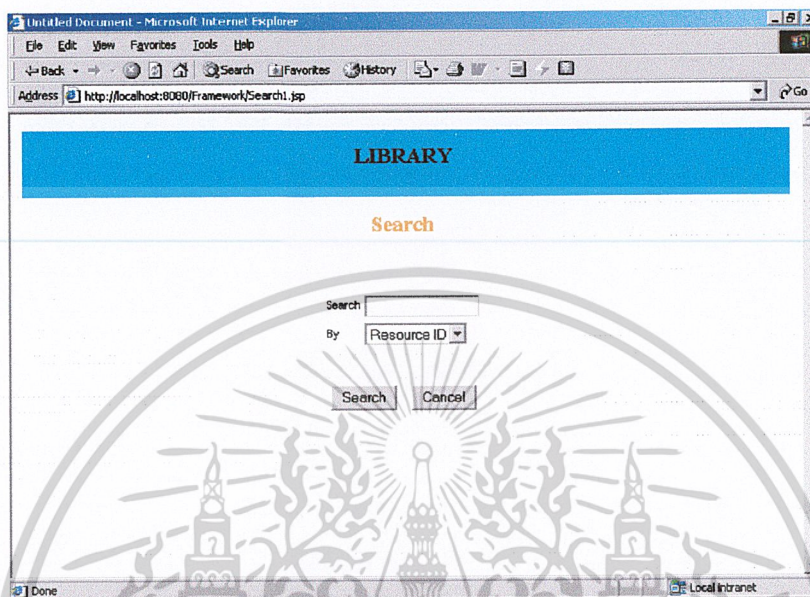
จากรูปที่ 4.2 หน้าจอส่วนติดต่อผู้ใช้งานสำหรับการจัดการข้อมูลสมาชิก ได้แก่ การเพิ่ม การแก้ไข การลบ ข้อมูลสมาชิกในฐานข้อมูล และมีปุ่มสำหรับกลับไปหน้าจอหลัก



รูปที่ 4.3 หน้าจอส่วนติดต่อผู้ใช้งานสำหรับจัดการข้อมูลทรัพยากร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.3 หน้าจอส่วนติดต่อผู้ใช้งานสำหรับการจัดการข้อมูลทรัพยากร ได้แก่ การเพิ่ม การแก้ไข การลบ ข้อมูลทรัพยากรในฐานข้อมูล และมีปุ่มสำหรับกลับไปหน้าจอหลัก



รูปที่ 4.4 หน้าจอส่วนติดต่อผู้ใช้งานสำหรับค้นหาข้อมูลทรัพยากร

จากรูปที่ 4.4 หน้าจอส่วนติดต่อผู้ใช้งานสำหรับการสืบค้นทรัพยากรในห้องสมุด โดยสามารถสืบค้นโดยใช้ รหัสทรัพยากร ชื่อเรื่อง ชื่อผู้แต่ง และ ISBN

#### 4.2.3 การสร้างโปรแกรมหลักเพื่อเรียกใช้งานเมธอดของคอนโทรลเลอร์

การเรียกใช้งานเมธอดภายในคอนโทรลเลอร์ จำเป็นต้องส่งพารามิเตอร์ตามที่ได้ระบุไว้ดังที่แผนผังไดอะแกรมรูปที่ 3.4 และ 3.5 ในบทที่ 3 ในที่นี้จะยกตัวอย่างการสร้างโปรแกรมหลักเพื่อเรียกใช้งานเมธอด runBorrow และ runReturn ของคอนโทรลเลอร์ โดยจะส่งพารามิเตอร์เป็น SQLcommand และ ค่าแฟลก ดังตัวอย่างที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### ตัวอย่างที่ 4.5 การสร้างโปรแกรมหลักเพื่อเรียกการใช้งานเมธอดของคอนโทรลเลอร์

```

1   class MainBorrow
2   {
3   public static void main(String[] args)
4   {
5   BorrowReturn borrow = new BorrowReturn();
6   MemberManage member = new MemberManage();
7   ResourceManage resource = new ResourceManage();
8   String searchMemCMD = "select * from MemberTable where MemID = '00000001' ";
9   String searchResCMD = "select * from ResourceTable where ResID ='00000010' ";
10  String insertBorrowCMD = "insert into BorrowReturnTable values ('00000010'
11  , '00000001' , '2004-02-05' , '2004-02-12' ,5,0 , 'T','T')";
12  ControlBorrowReturn conObj = new ControlBorrowReturn();
13  conObj.init(borrow,member,resource);
14  int ans = conObj.runBorrow(searchMemCMD,searchResCMD,insertBorrowCMD);
15  if(ans == 0) System.out.println("Borrow Complete");
16  String CMDsearchBorrow = "select * from BorrowReturnTable where ResID =
17  '00000010'";
18  String CMDdeleteBorrow = "delete from BorrowReturnTable where ResID =
19  '00000010'";
20  ans = conObj.runReturn(CMDsearchBorrow ,CMDdeleteBorrow , 0);
21  if (ans == 0 ) System.out.println("Return Complete");
22  }
23  }

```

จากตัวอย่างที่ 4.5 เมื่อผู้พัฒนาสร้างโปรแกรมหลักและเรียกใช้งานเมธอดของคอนโทรลเลอร์แล้ว คอนโทรลเลอร์จะจัดลำดับการทำงานต่างๆ ซึ่งจะแสดงไค้ดการทำงานภายในเมธอด runBorrow และ runReturn ไว้ใน CD-Rom ของปัญหาพิเศษนี้ ต่อไปจะอธิบายไค้ดในตัวอย่างที่ 4.5 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรทัดที่ 7 ถึง 10 และ 15 ถึง 18 เป็นการสร้าง SQLcommand โดยตัวอย่างโค้ดนี้จะใช้  
ทรัพยากรที่มีรหัสเป็น '0000010' และสมาชิกมีรหัสเป็น '00000001'

บรรทัดที่ 13 เป็นเรียกเมธอด runBorrow ของคอนโทรลเลอร์ โดยส่งพารามิเตอร์เป็น  
SQLcommand และจะรีเทิร์นค่าเป็นศูนย์ถ้าการยืมเสร็จสิ้นโดยสมบูรณ์

บรรทัดที่ 19 เป็นเรียกเมธอด runReturn ของคอนโทรลเลอร์ โดยส่ง พารามิเตอร์เป็น  
SQLcommand และค่าเฟลก จากนั้นจะรีเทิร์นค่าเป็นศูนย์ถ้าการคืนเสร็จสิ้นโดยสมบูรณ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลวิจัยและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

แอปพลิเคชันเฟรมเวิร์คเป็นแนวคิดในการนำเอาการออกแบบ และตัวโปรแกรมกลับมาใช้ใหม่ แอปพลิเคชันเฟรมเวิร์คสำหรับระบบยืม-คืนที่สร้างขึ้น สามารถนำไปพัฒนาสร้างคอมพิวเตอร์เพื่อใช้ในการพัฒนาแอปพลิเคชันสำหรับระบบยืม-คืนทั่วไปได้ยกตัวอย่างเช่น ระบบงานห้องสมุด, การเช่ารถยนต์, การเช่าห้องของโรงแรม ซึ่งมีการทำงานพื้นฐานที่คล้ายกันในส่วนของ ระบบการยืม-คืน ทรัพยากร ระบบจัดการฐานข้อมูลสมาชิก ระบบจัดการฐานข้อมูลทรัพยากร และระบบสืบค้นทรัพยากร ผู้ใช้สามารถสร้างแอปพลิเคชัน โดยเรียกใช้เมธอดที่มีอยู่ในแอปพลิเคชันเฟรมเวิร์คของระบบยืม-คืน และนำไปพัฒนาเพิ่มเติมในส่วนที่ระบบยืม-คืนแต่ละระบบมีการทำงานเป็นลักษณะเฉพาะ ไม่ต้องทำการเขียน โปรแกรมใหม่ทั้งหมด ทำให้การพัฒนาแอปพลิเคชันทำได้รวดเร็วขึ้น

#### 5.2 ข้อเสนอแนะ

ปัญหาพิเศษเรื่องแอปพลิเคชันเฟรมเวิร์คบนเว็บสำหรับระบบยืม-คืน สามารถใช้เป็นแนวทางสำหรับศึกษาเรื่องเฟรมเวิร์ค รวมถึงเรื่องเทคโนโลยีของการโปรแกรมเชิงวัตถุ ต้นแบบเฟรมเวิร์คสำหรับระบบยืม-คืนที่สร้างขึ้น ยังไม่สมบูรณ์ครอบคลุมทุกการทำงาน ยังขาดส่วนของฟังก์ชันต่างๆ เช่น การจองทรัพยากร การสร้างรายงานต่างๆในระบบ เป็นต้น และส่วนที่ผู้ใช้ทำการอิมพลีเมนต์ต้องใช้ภาษาจาวาซึ่งเป็นเป็นภาษาเดียวกับที่ใช้พัฒนาแอปพลิเคชันเฟรมเวิร์คในการพัฒนา อีกทั้งในส่วนของแอปพลิเคชันเฟรมเวิร์คและส่วนติดต่อผู้ใช้งานต้องอยู่ในเครื่องเดียวกัน ซึ่งอาจเป็นข้อจำกัดในการใช้งาน ผู้ที่สนใจสามารถพัฒนาเฟรมเวิร์คดังกล่าวต่อไป โดยอาจนำเทคโนโลยีเว็บเซอร์วิสเข้ามาใช้ โดยทำให้เป็นแอปพลิเคชันเฟรมเวิร์คบนเว็บเซอร์วิส เพื่อให้สามารถนำมาใช้งานได้อย่างสมบูรณ์และกว้างขวางมากขึ้น

## บรรณานุกรม

คุณากร นามวงศ์พรหม และ ศราวุฒิ อัยภูมิงคล . 2545 .

แอปพลิเคชันเฟรมเวิร์คบนพื้นฐานของเว็บเซิร์ฟวิส . กรุงเทพฯ :

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

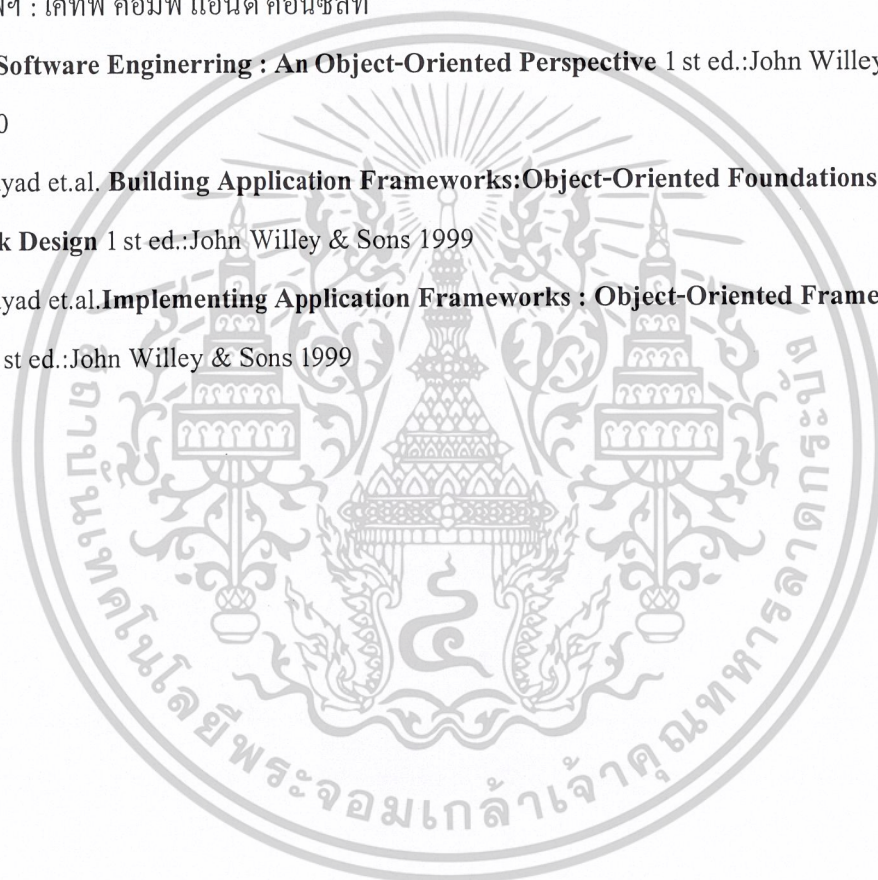
ดวงพร ขอเจริญพร . 2546 . เขียนโปรแกรมจาวาบนเว็บด้วยเซิร์ฟเลทและ JSP

กรุงเทพฯ : เคทีพี คอมพ์ แอนด์ คอนซัลท์

Eric J. Braude. **Software Engineering : An Object-Oriented Perspective** 1 st ed.:John Willey & Sons 2000

Mohamed E. Fayad et.al. **Building Application Frameworks: Object-Oriented Foundations of Framework Design** 1 st ed.:John Willey & Sons 1999

Mohamed E. Fayad et.al. **Implementing Application Frameworks : Object-Oriented Frameworks at Work** 1 st ed.:John Willey & Sons 1999



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

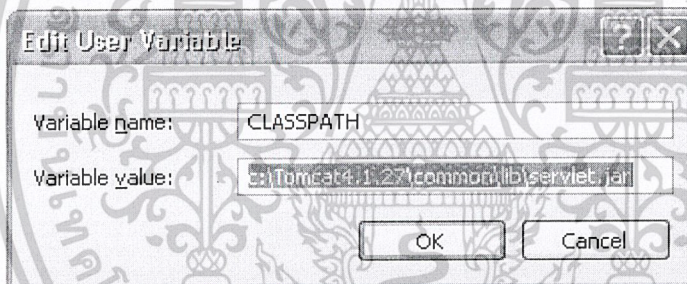
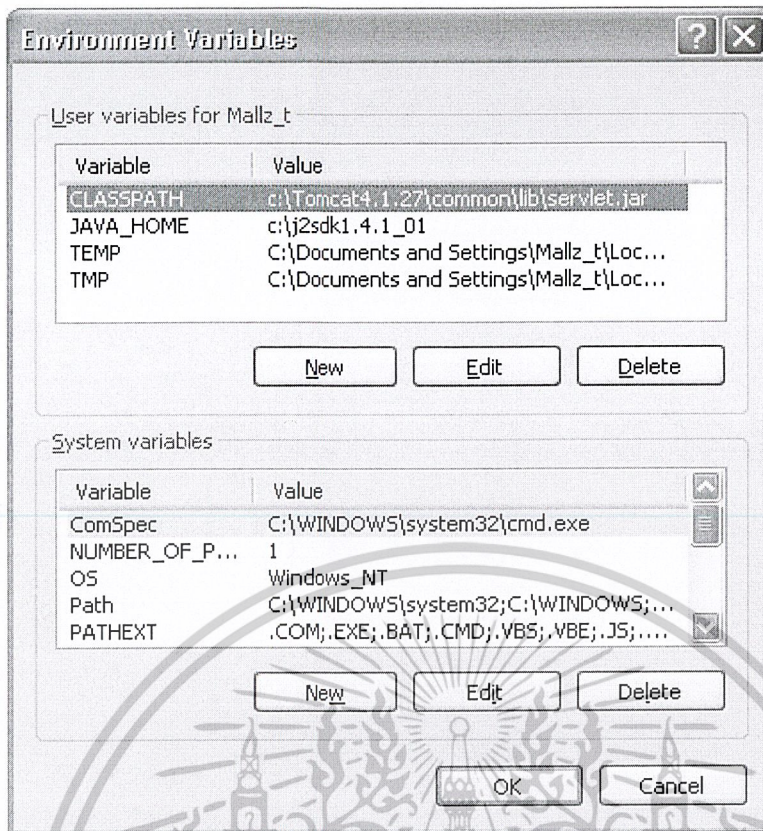
## ภาคผนวก ก

### คู่มือการติดตั้ง

#### การติดตั้ง Tomcat

1. ไปที่ <http://mirror.ehpg.net/apache/jakarta/tomcat-4/>
2. เลือก download ที่ [tomcat-4.1.27.zip](#)
3. Extract .zip File ลงใน c:\Drive หรือ Drive อื่นๆ ซึ่งจะตั้งชื่อ Directory ที่ให้ Tomcat อยู่เป็น Directory ชื่อ c:\Tomcat4.1.27
4. กำหนด Environment Variables การระบุค่าของ Environment Variables สำหรับ Windows 2000 server ที่ Control Panel เลือก System เลือก Advanced Tab เลือก Environment Variables

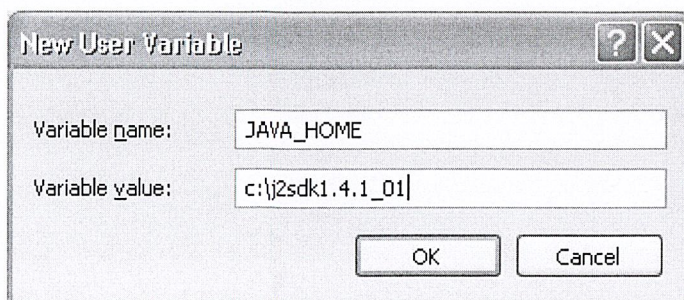
4.1 กำหนด Classes สำหรับ Java Compiler หลังจากได้มีการติดตั้ง Software ที่จำเป็นแล้ว ผู้พัฒนาจำเป็นต้องบอก Java Compiler (javac) ถึงตำแหน่งที่จะมี Servlet และ JSP Class files ที่ Java Compiler จำเป็นที่จะต้องใช้ในการ Compile Program ด้วย โดยทั่วไป Class Files เหล่านี้มักจะอยู่ใน lib subdirectory ของ Directory ที่มี Server install อยู่ Servlet classes จะอยู่ใน servlet.jar และ JSP Classes จะอยู่ใน jsp.jar หรือ jspengine.jar หรือ jasper.jar วิธีที่จะทำให้ javac รู้ถึงตำแหน่งของ classes เหล่านี้คือ การระบุ Jar File เหล่านี้ที่ CLASSPATH Variable กำหนดค่าของ Environment Variable CLASSPATH เป็น c:\Tomcat4.1.27\common\lib\servlet.jar ดังรูปที่ 2 (ในกรณีนี้ Installation Directory ของ Tomcat คือ Tomcat4.1.27)



รูปที่ ก.1 หน้าจอสำหรับกำหนด CLASSPATH Variable

4.2 JAVA\_HOME Setting ผู้พัฒนาจำเป็นต้องกำหนด JAVA\_HOME Environment Variable เพื่ออ้างอิง JDK Installation Directory ถ้าผู้พัฒนาใช้ JDK version 1.2 ขึ้นไป กำหนดค่าของ Environment Variable JAVA\_HOME เป็น c:\j2sdk1.4.1\_02 (ในกรณีนี้ Installation Directory ของ JDK คือ j2sdk1.4.1\_02) ดังรูปที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

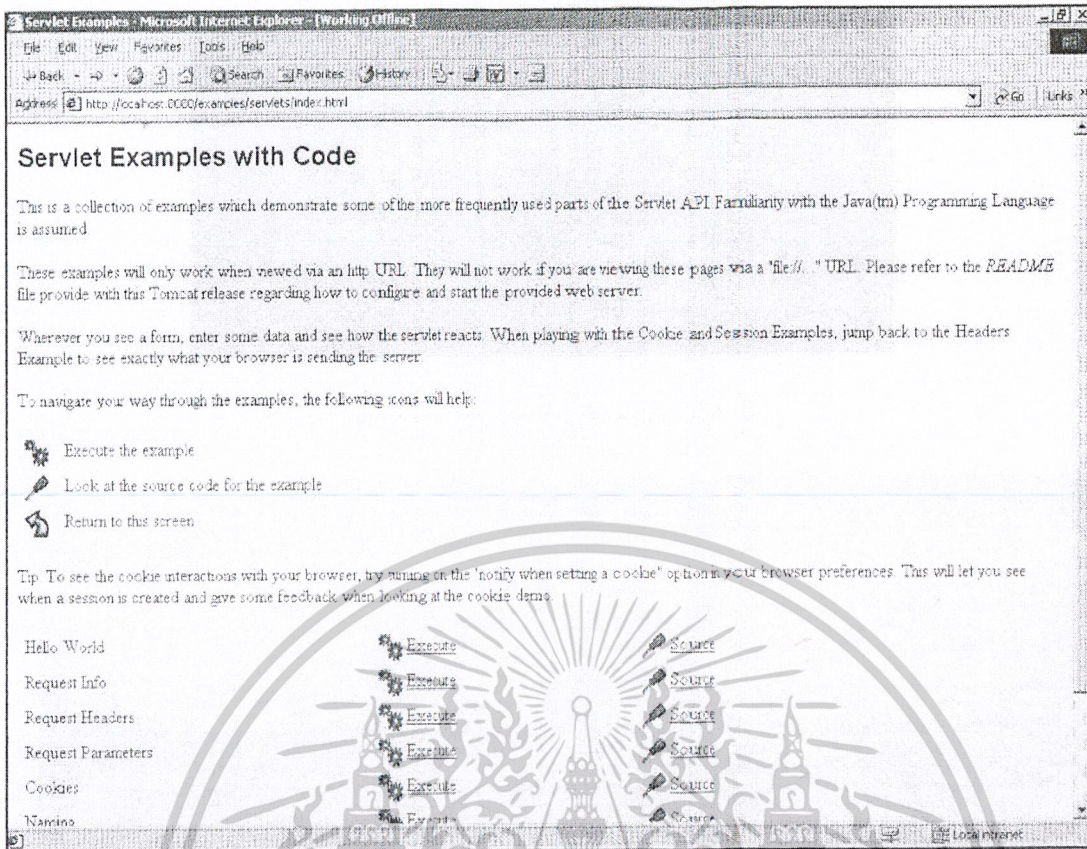


รูปที่ ก.2 หน้าจอสำหรับกำหนด JAVA\_HOME Variable

4.3 Port Number Setting โดย Default แล้ว Tomcat run บน Port 8080 ผู้พัฒนาไม่จำเป็นต้องเปลี่ยนเป็น Port อื่น นอกจากว่ามี Application อื่นใช้ Port 8080 บนคอมพิวเตอร์แล้ว ถ้าผู้พัฒนาต้องการที่จะเปลี่ยน Port Number ให้ทำการแก้ไขไฟล์ server.xml ที่อยู่ที่ Directory Tomcat4.1.18/conf (กรณีนี้ Installation Directory ของ Tomcat คือ tomcat4) โดยเปลี่ยน Port Number ตรงตำแหน่งที่ไว้ใช้ระบุ Port Number เป็น Number ที่ต้องการ

**ขั้นตอนในการทดสอบการติดตั้ง Tomcat version 4.1.27**

1. Start Tomcat โดยการ start Command Prompt แล้วไปที่ Directory c:\Tomcat4.1.18\bin
2. พิมพ์คำว่า Startup จะพบว่า Tomcat จะถูก start ที่หน้าต่างใหม่ของ Command Prompt
3. ทดลอง run ตัวอย่างของ Servlets Page โดยการเปิด Web Browser แล้วพิมพ์ URL Address <http://ipaddress:8080/examples/servlets> (ipaddress คือ หมายเลข IP Address ของเครื่องคอมพิวเตอร์นั้นๆ ทราบได้จากการใช้คำสั่ง ipconfig ที่ Command Prompt) จะพบตัวอย่าง Servlets ให้ทดลอง run ตัวอย่างดูเพื่อทดสอบว่า Tomcat run โดยถูกต้อง ดังรูปที่ 4



รูปที่ ก.3 หน้าจอของ Web Browser ที่แสดงตัวอย่างของ Servlets

4. กรณีที่ต้องการหยุดการทำงานของ Tomcat ให้พิมพ์ `>shutdown` ที่ Directory เดียวกับที่ทำการ Start Tomcat

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ภาคผนวก ข**  
**รายละเอียดของเฟรมเวิร์ค**

ตารางที่ ข.1 เมธอดของอินเทอร์เฟซ IConnectDB

ชื่อ	หน้าที่
ConnectDB(String host, String database, String username, String password)	ใช้สำหรับติดต่อกับฐานข้อมูล โดยจะรับพารามิเตอร์ประเภทสตริง ได้แก่ โฮส (หมายเลข IP), ชื่อฐานข้อมูล, username และ password และจะรีเทิร์นค่า Boolean กลับมา
disconnectDatabase()	ใช้สำหรับยกเลิกการติดต่อกับฐานข้อมูล
queryData(String cmdQuery)	ใช้สำหรับค้นหาข้อมูลในฐานข้อมูล โดยจะรับพารามิเตอร์ประเภทสตริง คือ คำสั่ง SQL Command ที่จะใช้สืบค้นฐานข้อมูล และจะรีเทิร์นค่า Boolean กลับมา
selectData(String cmdSelect)	ใช้สำหรับค้นหาข้อมูลในฐานข้อมูล โดยจะรับพารามิเตอร์ประเภทสตริง คือ คำสั่ง SQL Command ที่จะใช้สืบค้นฐานข้อมูล และจะรีเทิร์นค่า ผลลัพธ์การสืบค้น เป็น ResultSet กลับมา
updateData(String cmdUpdate)	ใช้สำหรับเปลี่ยนแปลงข้อมูลในฐานข้อมูล โดยจะรับพารามิเตอร์ประเภทสตริง คือ คำสั่ง SQL Command ที่จะใช้ในการอัปเดตฐานข้อมูล และจะรีเทิร์นค่าเป็น Boolean กลับมา
deleteData(String cmdDelete)	ใช้สำหรับลบข้อมูลในฐานข้อมูล โดยจะรับพารามิเตอร์ประเภทสตริง คือ คำสั่ง SQL Command ที่จะใช้ในการลบ 1 แถวในฐานข้อมูล และจะรีเทิร์นค่าเป็น Boolean กลับมา
insertData(String cmdInsert)	ใช้สำหรับเพิ่มข้อมูลในฐานข้อมูล โดยจะรับพารามิเตอร์ประเภทสตริง คือ คำสั่ง SQL Command ที่จะใช้ในการเพิ่ม 1 แถวในฐานข้อมูล และจะรีเทิร์นค่าเป็น Boolean กลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.2 เมธอดของอินเทอร์เฟซ IBorrowReturn

ชื่อ	หน้าที่
getHost()	ใช้สำหรับรีเทิร์นสตริงชื่อโฮสของฐานข้อมูล
getDBname()	ใช้สำหรับรีเทิร์นสตริงชื่อฐานข้อมูล
getUsername()	ใช้สำหรับรีเทิร์นสตริงที่เป็น username ของฐานข้อมูล
getPassword()	ใช้สำหรับรีเทิร์นสตริงที่เป็น password ของฐานข้อมูล
getBorrowKey()	ใช้สำหรับรีเทิร์นสตริงที่เป็น PrimaryKey ของฐานข้อมูลเทเบิลข้อมูลการยืม-คืน
setRate(float rate)	ใช้สำหรับกำหนดอัตราค่าบริการ
checkBorrowTransactionDuplicate(String resKey)	ใช้สำหรับตรวจสอบว่าทรัพยากรที่ต้องการจะยืมเคยถูกยืมไปแล้วหรือไม่ โดยใช้พารามิเตอร์ประเภทสตริงที่เป็น Key ของ ทรัพยากรที่ต้องการจะยืม และจะรีเทิร์นเป็นค่า Boolean กลับมาเป็น True ถ้าทรัพยากรนั้นถูกยืมไปแล้ว
checkConditionPay(String borrowKey)	ใช้สำหรับตรวจสอบว่าการคิมนั้นต้องการการชำระเงินหรือไม่ โดยใช้พารามิเตอร์ประเภทสตริงที่เป็น PrimaryKey ของการยืมครั้งนั้น เพื่อสืบค้นเงื่อนไขนั้นจากข้อมูลในฐานข้อมูล และจะรีเทิร์นเป็นค่า Boolean กลับมา

ตารางที่ ข.3 เมธอดของอินเทอร์เฟซ IMemberManage

ชื่อ	หน้าที่
getHost()	ใช้สำหรับรีเทิร์นสตริงชื่อโฮสของฐานข้อมูล
getDBname()	ใช้สำหรับรีเทิร์นสตริงชื่อฐานข้อมูล
getUsername()	ใช้สำหรับรีเทิร์นสตริงที่เป็น username ของฐานข้อมูล
getPassword()	ใช้สำหรับรีเทิร์นสตริงที่เป็น password ของฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

getMemKey()	ใช้สำหรับรีเทิร์นสตริงที่เป็น PrimaryKey ของเทเบิลข้อมูลสมาชิกในฐานข้อมูล
checkMemberCondition(String memKey)	ใช้สำหรับตรวจสอบเงื่อนไขการยืมของสมาชิก โดยใช้พารามิเตอร์ประเภทสตริงที่เป็น PrimaryKey ของเทเบิลของสมาชิกในฐานข้อมูลเพื่อสืบค้นเงื่อนไขนั้น และจะรีเทิร์นค่า เป็น Boolean กลับมา

ตารางที่ ข.4 เมธอดของอินเทอร์เฟซ IResourceManage

ชื่อ	หน้าที่
getHost()	ใช้สำหรับรีเทิร์นสตริงชื่อโฮสต์ของฐานข้อมูล
getDBname()	ใช้สำหรับรีเทิร์นสตริงชื่อฐานข้อมูล
getUsername()	ใช้สำหรับรีเทิร์นสตริงที่เป็น username ของฐานข้อมูล
getPassword()	ใช้สำหรับรีเทิร์นสตริงที่เป็น password ของฐานข้อมูล
getResKey()	ใช้สำหรับรีเทิร์นสตริงที่เป็น PrimaryKey ของเทเบิลข้อมูลทรัพยากรในฐานข้อมูล
checkResourceCondition(String resKey)	ใช้สำหรับตรวจสอบเงื่อนไขการยืมของทรัพยากร โดยใช้พารามิเตอร์ประเภท สตริง ที่เป็น PrimaryKey ของเทเบิลทรัพยากร เพื่อสืบค้นเงื่อนไขนั้นจากข้อมูลในฐานข้อมูล และจะรีเทิร์นค่าเป็น Boolean กลับมา

ตารางที่ ข.5 เมธอดของแอสแทรกคลาส ชื่อ Abs\_BorrowReturn

ชื่อ	หน้าที่
ConnectDB(String host, String database, String username, String password)	(หน้าที่เหมือนใน Interface IConnectDB)
disconnectDatabase()	(หน้าที่เหมือนใน Interface IConnectDB)
queryData(String cmdQuery)	(หน้าที่เหมือนใน Interface IConnectDB)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

selectData(String cmdSelect)	(หน้าที่เหมือนใน Interface IConnectDB)
updateData(String cmdUpdate)	(หน้าที่เหมือนใน Interface IConnectDB)
deleteData(String cmdDelete)	(หน้าที่เหมือนใน Interface IConnectDB)
insertData(String cmdInsert)	(หน้าที่เหมือนใน Interface IConnectDB)
getHost()	(หน้าที่เหมือนใน Interface IBorrowReturn)
getDBname()	(หน้าที่เหมือนใน Interface IBorrowReturn)
getUsername()	(หน้าที่เหมือนใน Interface IBorrowReturn)
getPassword()	(หน้าที่เหมือนใน Interface IBorrowReturn)
getBorrowKey()	(หน้าที่เหมือนใน Interface IBorrowReturn)
checkBorrowTransactionDuplicate(String resKey)	(หน้าที่เหมือนใน Interface IBorrowReturn)
checkConditionPay(String borrowKey)	(หน้าที่เหมือนใน Interface IBorrowReturn)
setRate(float rate)	(หน้าที่เหมือนใน Interface IBorrowReturn)

ตารางที่ ข.6 เมธอดของแอบสแทรกคลาส ชื่อ Abs\_MemberManage

ชื่อ	หน้าที่
ConnectDB(String host, String database, String username, String password)	(หน้าที่เหมือนใน Interface IConnectDB)
disconnectDatabase()	(หน้าที่เหมือนใน Interface IConnectDB)
queryData(String cmdQuery)	(หน้าที่เหมือนใน Interface IConnectDB)
selectData(String cmdSelect)	(หน้าที่เหมือนใน Interface IConnectDB)
updateData(String cmdUpdate)	(หน้าที่เหมือนใน Interface IConnectDB)
deleteData(String cmdDelete)	(หน้าที่เหมือนใน Interface IConnectDB)
insertData(String cmdInsert)	(หน้าที่เหมือนใน Interface IConnectDB)
getHost()	(หน้าที่เหมือนใน Interface IMemberManage)
getDBname()	(หน้าที่เหมือนใน Interface IMemberManage)
getUsername()	(หน้าที่เหมือนใน Interface IMemberManage)
getPassword()	(หน้าที่เหมือนใน Interface IMemberManage)
getMemKey()	(หน้าที่เหมือนใน Interface IMemberManage)
checkMemberCondition(String memKey)	(หน้าที่เหมือนใน Interface IMemberManage)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.7 เมธอดของแอบสแทรกคลาส ชื่อ Abs\_ResourceManage

ชื่อ	หน้าที่
ConnectDB(String host, String database, String username, String password)	(หน้าที่เหมือนใน Interface IConnectDB)
disconnectDatabase()	(หน้าที่เหมือนใน Interface IConnectDB)
queryData(String cmdQuery)	(หน้าที่เหมือนใน Interface IConnectDB)
selectData(String cmdSelect)	(หน้าที่เหมือนใน Interface IConnectDB)
updateData(String cmdUpdate)	(หน้าที่เหมือนใน Interface IConnectDB)
deleteData(String cmdDelete)	(หน้าที่เหมือนใน Interface IConnectDB)
insertData(String cmdInsert)	(หน้าที่เหมือนใน Interface IConnectDB)
getHost()	(หน้าที่เหมือนใน Interface IResourceManage)
getDBname()	(หน้าที่เหมือนใน Interface IResourceManage)
getUsername()	(หน้าที่เหมือนใน Interface IResourceManage)
getPassword()	(หน้าที่เหมือนใน Interface IResourceManage)
getResKey()	(หน้าที่เหมือนใน Interface IResourceManage)
checkResourceCondition(String resKey)	(หน้าที่เหมือนใน Interface IResourceManage)

ตารางที่ ข.8 เมธอดของคอนโทรลเลอร์ ControlMember

ชื่อ	หน้าที่
init(Abs_MemberManage objMember)	ใช้สำหรับรับพารามิเตอร์เป็นออบเจกต์ที่สร้างมาจาก คลาส Abs_MemberManage มาเก็บไว้เพื่อใช้ออบเจกต์นั้นภายในคลาส ControlMember
runAddMember( String insertCMD)	ใช้สำหรับ เรียกเมธอดติดต่อฐานข้อมูล และเพิ่มสมาชิกใหม่ลงฐานข้อมูล โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการเพิ่มข้อมูลสมาชิกในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Boolean กลับมา
runUpdateMember(String updateCMD)	ใช้สำหรับ เรียกเมธอดติดต่อฐานข้อมูล และแก้ไขข้อมูลสมาชิกในฐานข้อมูล โดยจะรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	พารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการแก้ไขข้อมูลสมาชิกในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Boolean กลับมา
runDeleteMember(String deleteCMD)	ใช้สำหรับ เรียกเมธอดคิดต่อฐานข้อมูล และลบข้อมูลสมาชิกในฐานข้อมูล โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการลบข้อมูลสมาชิกในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Boolean กลับมา
runSearchMember(String searchCMD)	ใช้สำหรับ เรียกเมธอดคิดต่อฐานข้อมูล และสืบค้นข้อมูลสมาชิกในฐานข้อมูล โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการสืบค้นข้อมูลในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Boolean กลับมา
runSelectMember(String searchCMD)	ใช้สำหรับ เรียกเมธอดคิดต่อฐานข้อมูล และสืบค้นข้อมูลสมาชิกในฐานข้อมูล โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการสืบค้นข้อมูลในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น ResultSet จากการสืบค้นฐานข้อมูล

ตารางที่ ข.9 เมธอดของคอนโทรลเลอร์ ControlResource

ชื่อ	หน้าที่
init(Abs_ResourceManage objResource)	ใช้สำหรับรับพารามิเตอร์เป็นออบเจกต์ที่สร้างมาจาก คลาส Abs_ResourceManage มาเก็บไว้เพื่อใช้ออบเจกต์นั้นภายในคลาส ControlResource
runAddResource (String insertCMD)	ใช้สำหรับ เรียกเมธอดคิดต่อฐานข้อมูล และเพิ่มทรัพยากรใหม่ลงฐานข้อมูล โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการเพิ่มข้อมูลทรัพยากรในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Boolean กลับมา
runUpdateResource (String updateCMD)	ใช้สำหรับ เรียกเมธอดคิดต่อฐานข้อมูล และแก้ไขข้อมูลทรัพยากรในฐานข้อมูล โดยจะรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	พารามิเตอร์เป็นสตริงได้แก่ คำสั่งSQLCommand ในการแก้ไขข้อมูลทรัพยากรในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Boolean กลับมา
runDeleteResource (String deleteCMD)	ใช้สำหรับ เรียกเมธอดติดต่อฐานข้อมูล และลบข้อมูลทรัพยากรในฐานข้อมูล โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการลบข้อมูลทรัพยากรในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Boolean กลับมา
runSearchResource(String searchCMD)	ใช้สำหรับ เรียกเมธอดติดต่อฐานข้อมูล และสืบค้นข้อมูลทรัพยากรในฐานข้อมูล โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการสืบค้นข้อมูลทรัพยากรในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Boolean กลับมา
runSelectResource (String searchCMD)	ใช้สำหรับ เรียกเมธอดติดต่อฐานข้อมูล และสืบค้นข้อมูลทรัพยากรในฐานข้อมูล โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการสืบค้นข้อมูลทรัพยากรในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น ResultSet จากการสืบค้นฐานข้อมูล

ตารางที่ ข.10 แอททริบิวต์ของคอนโทรลเลอร์ ControlBorrowReturn

ชื่อ	หน้าที่
borrow	ใช้เก็บออบเจ็กต์ที่สร้างมาจาก คลาส Abs_BorrowReturn เพื่อที่จะให้สามารถเรียกใช้เมธอดของออบเจ็กต์นั้น ภายในคลาส ControlBorrowReturn
member	ใช้เก็บออบเจ็กต์ที่สร้างมาจากคลาส Abs_MemberManage เพื่อที่จะให้สามารถเรียกใช้เมธอดของออบเจ็กต์นั้น ภายในคลาส ControlBorrowReturn
resource	ใช้เก็บออบเจ็กต์ที่สร้างมาจาก คลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Abs_ResourceManage เพื่อที่จะให้สามารถเรียกใช้เมธอดของออบเจกต์นั้น ภายในคลาส ControlBorrowReturn
result	ใช้เก็บ ResultSet ที่ได้จากการสืบค้นในฐานข้อมูล

ตารางที่ ข.11 เมธอดของคอนโทรลเลอร์ ControlBorrowReturn

ชื่อ	หน้าที่
init(Abs_BorrowReturn objBorrow,Abs_MemberManage objMember,Abs_ResourceManage objResource)	ใช้สำหรับรับพารามิเตอร์เป็นออบเจกต์ที่สร้างมาจาก คลาส Abs_BorrowReturn Abs_MemberManage และ Abs_ResourceManage มาเก็บไว้ เพื่อใช้ออบเจกต์นั้นภายในคลาส ControlBorrowReturn
init(Abs_BorrowReturn objBorrow)	ใช้สำหรับรับพารามิเตอร์เป็นออบเจกต์ที่สร้างมาจาก คลาส Abs_BorrowReturn มาเก็บไว้เพื่อใช้ออบเจกต์นั้นในคลาส ControlBorrowReturn
runBorrow(String CMDsearchMem , String CMDserachRes , String CMDinsertBorrow)	ใช้สำหรับ เรียกเมธอดติดต่อฐานข้อมูล เมธอดค้นหาสมาชิก/ทรัพยากร และเมธอดตรวจสอบเงื่อนไขการยืมของสมาชิก/ทรัพยากร และเพิ่มข้อมูลการยืมลงฐานข้อมูล ตามลำดับ โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่ง SQLCommand ในการสืบค้นผู้ยืม , ทรัพยากร และ คำสั่ง SQLCommand ในการเพิ่มข้อมูลการยืมลงฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Integer กลับมาเป็น ศูนย์ ถ้าทำการยืมสำเร็จ รีเทิร์นค่าเป็น 2 ถ้าทรัพยากรนั้นไม่อนุญาตให้ยืม รีเทิร์นค่าเป็น 4 ถ้าสมาชิกคนนั้นไม่อนุญาตให้ยืม
runSearchBorrow(String queryCMD)	ใช้สำหรับ เรียกเมธอดติดต่อฐานข้อมูล และสืบค้นข้อมูลการยืมในฐานข้อมูล โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการสืบค้นข้อมูลการยืมในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Boolean กลับมา
runSelectBorrow(String queryCMD)	ใช้สำหรับ เรียกเมธอดติดต่อฐานข้อมูล และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<p>สืบค้นข้อมูลการยืมในฐานข้อมูล โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่ง SQLCommand ในการสืบค้นข้อมูลการยืมในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น ResultSet จากการสืบค้นฐานข้อมูล</p>
runDeleteBorrow(String deleteCMD)	<p>ใช้สำหรับ เรียกเมธอดติดต่อฐานข้อมูล และลบข้อมูลการยืมในฐานข้อมูล โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการลบข้อมูลการยืมในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Boolean กลับมา</p>
runUpdateBorrow(String updateCMD)	<p>ใช้สำหรับ เรียกเมธอดติดต่อฐานข้อมูล และแก้ไขข้อมูลการยืมในฐานข้อมูล โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการแก้ไขข้อมูลการยืมในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Boolean กลับมา</p>
runReturn( String CMDsearchBorrow , String CMDdelete_updateBorrow , int flag)	<p>ใช้สำหรับ เรียกเมธอดติดต่อฐานข้อมูล เมธอดในการค้นหาข้อมูลการยืมและเมธอดตรวจสอบเงื่อนไขการชำระเงิน และลบข้อมูลการยืมในฐานข้อมูลตาม โดยจะรับพารามิเตอร์เป็นสตริง ได้แก่ คำสั่งSQLCommand ในการลบหรือแก้ไขข้อมูลการยืมในฐานข้อมูล โดยอาศัย พารามิเตอร์ ที่เป็น Integer ว่าต้องการลบทรานแซกชันการยืมหรือต้องการแก้ไขบางคอลัมน์ของข้อมูลการยืมในฐานข้อมูล จากนั้นจะรีเทิร์นค่าเป็น Integer กลับมาเป็น ศูนย์ ถ้าทำการคืนสำเร็จ รีเทิร์นค่าเป็น Integer กลับมาเป็น 1 ถ้าการคืนนั้นต้องชำระเงิน รีเทิร์นค่าเป็น Integer กลับมาเป็น 3 ถ้าไม่พบข้อมูลการยืมนั้นในฐานข้อมูล</p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

### ตัวอย่างการสร้างคลาสเพื่อใช้งานเฟรมเวิร์ค

#### 1. การสร้างคลาสที่สืบทอดมาจากแอ็บสแตรคคลาส Abs\_MemberManage

```
1 package LibraryFramework;
2
3 import java.sql.*;
4 import java.io.*;
5 import java.util.*;
6
7 public class MemberManage extends Abs_MemberManage
8 {
9     private String host = "localhost";
10    private String database = "frameworkproj";
11    private String username = "project";
12    private String password = "framework";
13    private String memID;
14    private String memType;
15    private int count;
16    private int maxBorrow;
17
18    public void setMemType(String memType)
19    {
20        this.memType = memType;
21    }
22    public void setCount(int count)
23    {
24        this.count = count;
25    }
26    public void setMaxBorrow(int maxBorrow)
27    {
28        this.maxBorrow = maxBorrow;
29    }
30    /**Must Write This Method*/
31    public String getHost() /**/
32    {
33        return host;
34    }
35    public String getDBname() /**/
36    {
37        return database;
38    }
39    public String getUsername() /**/
40    {
41        return username;
42    }
43    public String getPassword() /**/
44    {
45        return password;
46    }
47    public String getMemKey() /**/
48    {
49        return memID;
50    }
51
52    /**Must Write This Method*/
53    public boolean checkMemberCondition(String ID) /**/
54    {
55        if(memType.equals("VIP"))
56            return true;
57        else if (count < maxBorrow)
58            return true;
59        else
60            return false;
61    }
62
63    /**Must Write This Method*/
64 }
```

รูปที่ ก.1 ตัวอย่างคลาสที่สืบทอดมาจากแอ็บสแตรคคลาส Abs\_MemberManage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. การสร้างคลาสที่สืบทอดมาจากแอ็บสแตรกคลาส Abs\_ResourceManage

```

1 package LibraryFramework;
2
3 import java.sql.*;
4 import java.io.*;
5 import java.util.*;
6
7 public class ResourceManage extends Abs_ResourceManage
8 {
9     private String host = "localhost";
10    private String database = "frameworkproj";
11    private String username = "project";
12    private String password = "framework";
13    private String resID;
14    private String status;
15
16    public void setResID(String resID)
17    {
18        this.resID = resID;
19    }
20    public void setStatus(String status)
21    {
22        this.status = status;
23    }
24
25    /******Must Write This Method*/
26    public String getHost()
27    { return host; }
28    public String getDBName()
29    { return database; }
30    public String getUsername()
31    { return username; }
32    public String getPassword()
33    { return password; }
34    /******Must Write This Method*/
35
36    public String getResKey()
37    { return resID; }
38
39    public boolean checkResourceCondition(String ID)
40    {
41        if(status.equals("T"))
42        {
43            return true;
44        }
45        else
46        {
47            return false;
48        }
49    }
50    /******Must Write This Method*/
51 }

```

รูปที่ ค.2 ตัวอย่างคลาสที่สืบทอดมาจากแอ็บสแตรกคลาส Abs\_ResourceManage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ค

### ตัวอย่างการสร้างคลาสเพื่อใช้งานเฟรมเวิร์ค

ตัวอย่างคลาส MemberManage ที่สืบทอดมาจากแอ็บสแตรกคลาส Abs\_MemberManage

```
1    import java.sql.*;
2    import java.io.*;
3    import java.util.*;
4    public class MemberManage extends Abs_MemberManage
5    {
6        private MemberManage member;
7        private ControlMember conMem;
8        private String command;
9        private ResultSet result;
10       private String host = "localhost";
11       private String database = "frameworkproj";
12       private String username = "project";
13       private String password = "framework";
14       private String memID;
15       private String memType;
16       private int count;
17       private int maxBorrow;
18
19       public String getHost()           /**/
20     {
21         return host;
22     }
23
24     public String getDBname()          /**/
25     {
26         return database;
27     }
28
29     public String getUsername()        /**/
30     {
31         return username;
32     }
33
34     public String getPassword()        /**/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

24     {     return password;     }
25     public String getMemKey()           /**/
26     {     return memID;     }
27     public boolean checkMemberCondition(String memKey) /**/
28     { member = new MemberManage();
29       conMem = new ControlMember();
30       conMem.init(member);
31       command = "select * from MemberTable where memID = '"+ memKey +"'";
32       result = conMem.runSelectMember(command);
33       try
34       { if(result.next())
35         memType = result.getString("memType");
36         count = result.getString ("Count");
37         maxBorrow = result.getString("MaxBorrow");
38         if(memType.equals("V.I.P"))     return true;
39         else if (count < maxBorrow)     return true;
40         else                             return false; }
41       catch (SQLException ex)
42       {System.err.println("SQLException in CheckMemberCondition: "+ex.getMessage());
43         return false; }
44     }
45 }

```

จากตัวอย่างเมธอดภายในคลาสที่สืบทอดมาจากแอ็บสแตรกคลาส Abs\_MemberManage ผู้พัฒนาจำเป็นต้องสร้างโค้ดขึ้นให้เหมาะสมกับระบบยืม-คืนในส่วนการจัดการสมาชิกของผู้พัฒนาเอง โดยจะอธิบายโค้ดในตัวอย่างดังนี้

บรรทัดที่ 17 ถึง 24 เป็นเมธอดที่ใช้รีเทิร์นที่อยู่ ชื่อและรหัสผู้ใช้ของฐานข้อมูล

บรรทัดที่ 25 เป็นเมธอดที่ใช้รีเทิร์นคีย์หลักของตารางสมาชิกในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรทัดที่ 27 ตัวอย่างนี้เมธอด `checkMemberCondition` รีเทิร์นค่าเป็น `true` เมื่อประเภทของสมาชิก (`memType`) เป็น V.I.P หรือ จำนวนหนังสือที่สมาชิกเคยยืมไปแล้ว (`count`) มีค่าน้อยกว่าจำนวนสูงสุดที่สมาชิกคนนั้นสามารถยืมได้ (`maxBorrow`)

บรรทัดที่ 32 เป็นการเรียกเมธอด `runSelectMember` ที่อยู่ภายในคลาส `ControlMember`



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้