

# proposal

การวิเคราะห์ออกแบบระบบขายคอร์สออนไลน์  
ของบริษัท ENGLISH COURSES จำกัด

จัดทำโดย

นาย กฤษฎพล ดีดพิณ

รหัสนักศึกษา 6105003310

เสนอ

ผ.ศ. พรพรรณ ประชาพิพัฒน์

รายงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาวิชา cos3108

ภาคเรียนที่ 2 ปี การศึกษา 2564

ภาควิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์มหาวิทยาลัยรามคำแหง

## คำนำ

รายงานชุดนี้จุดประสงค์เพื่อวิเคราะห์และออกแบบระบบการขายอุปกรณ์คอมพิวเตอร์ออนไลน์บริษัท E  
NGLISHCOURSES จำกัด

เพื่อเพิ่มขยายช่องทางการจำหน่ายให้หลากหลายรูปแบบ, เพิ่มช่องทางการทำการตลาดให้กับสินค้า,  
อำนวยความสะดวกให้กับลูกค้าได้ตลอดและสามารถแข่งขันกับผู้ค้ารายอื่นๆได้

## สารบัญ

	หน้า
คำนำ	2
สารบัญ	3
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของปัญหา	5
1.2 วัตถุประสงค์	6
1.3 ประโยชน์ที่คาดว่าจะรับ	6
1.4 ขอบเขตของการดำเนินงาน	6
1.5 นิยามคำศัพท์	7
1.6 ระเบียบวิธีการดำเนินงาน	8
1.7 เครื่องมือในการพัฒนาระบบ	8
1.8 แผนการดำเนินงาน	9
1.9 งบประมาณที่ใช้	10
บทที่ 2 วรรณกรรมที่เกี่ยวข้อง	10
2.1 ทฤษฎีที่เกี่ยวข้อง	12
บทที่ 3 ขั้นตอนการดำเนินงาน	17
3.1.1 การวิเคราะห์ระบบ	18
- Use Case Diagram(Main)	19
- Use Case Diagram(UC-01)	20
- Use Case Diagram(UC-02)	21
- Use Case Diagram(UC-03)	22
- Use Case Diagram(UC-04)	23
- Use case Description(UC-01)	24

- Use case Description(UC-02)	25
- Use case Description(UC-03)	26
- Use case Description(UC-04)	27
3.1.2 Sequence Diagram	27
- Sequence Diagram(UC-01)	27
- Sequence Diagram(UC-02)	28
- Sequence Diagram(UC-03)	29
- Sequence Diagram(UC-04)	30
3.1.3 Activity Diagram	30
- Activity Diagram(UC-01)	31
- Activity Diagram(UC-02)	32
- Activity Diagram(UC-03)	33
- Activity Diagram(UC-04)	34
3.2 การออกแบบระบบ	
3.2.1 Class Diagram	39
3.2.2 Presentation Logic Subsystem	40
3.2.3 Working Logic Subsystem	41
3.2.4 Database Logic Subsystem	42
3.2.5 System Architecture Design	44
3.2.6 Persistent Data Design	45
4) สรุปและข้อเสนอแนะ	45
บรรณานุกรม	46

# 1.บทนำ

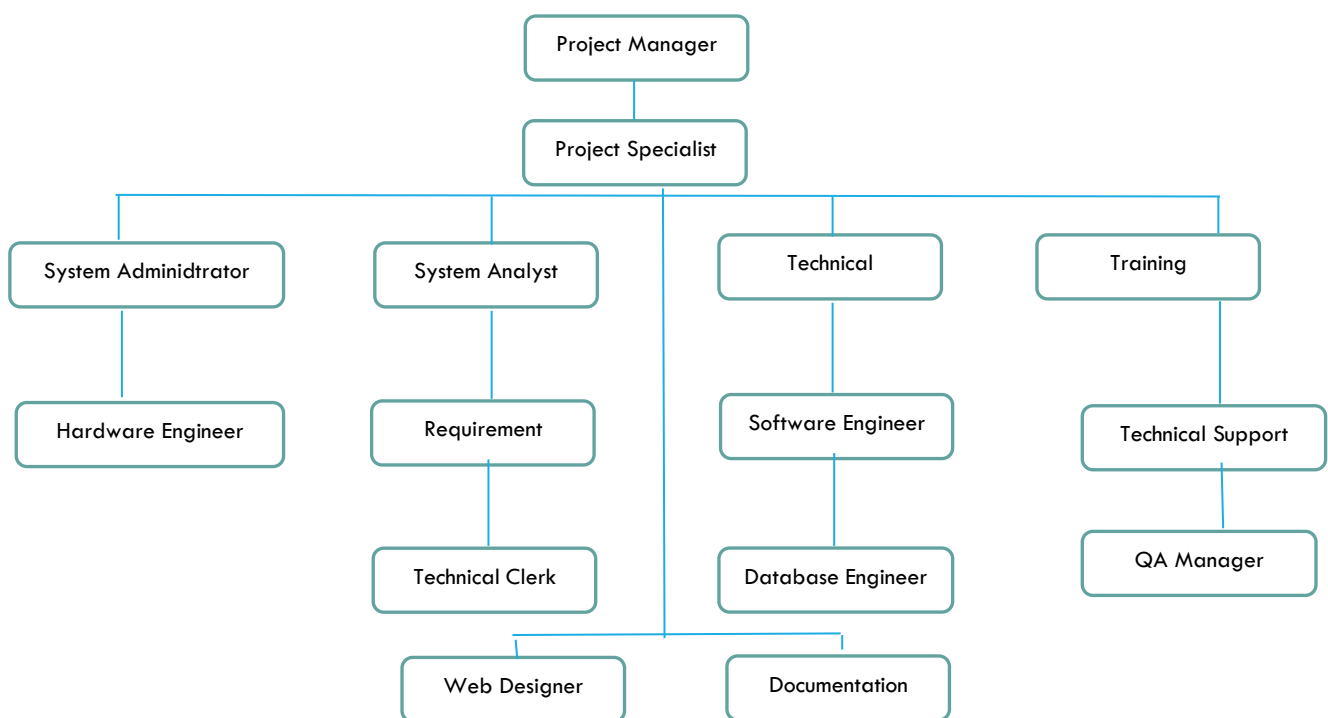
## 1.1 ที่มาและความสำคัญของปัญหา

ชื่อโครงการ : การวิเคราะห์ออกแบบระบบขายคอร์สออนไลน์

ของบริษัท ENGLISH COURSES จำกัด

บริษัทenglishcoursesจำกัดเป็นบริษัทขายคอร์ออนไลน์ทำหน้าที่เป็นหน้าร้านในการขาย

และควบคุมการให้บริการขายชำระเงินจัดส่งสินค้าของอุปกรณ์ในการประกอบการเรียนคอร์สออนไลน์



โครงการวิเคราะห์ออกแบบระบบขายคอร์ออนไลน์ที่นำเสนอนี้ประกอบด้วย3ขั้นตอนในการพัฒนาระบบทางทีมงานDeveloperสามารถนำผลของการออกแบบระบบทั้งหมดของโครงการนี้ไปพัฒนาต่อในขั้นตอนการติดตั้ง

## 1)implementation Phase และขั้นตอนปฏิบัติงานจริง 2)Operation and Maintenance Phase

ตามลำดับเพื่อให้การพัฒนาระบบแล้วเสร็จอย่างสมบูรณ์ถูกต้องรองรับความต้องการตามวัตถุประสงค์ของระบบและเกิดประโยชน์ต่อไปในอนาคต

### ความสำคัญและขอบเขตของปัญหา

1. ลูกค้าซื้อคอร์สเรียนออนไลน์ผ่านเว็บไซต์ได้ตลอดเวลา
2. ลูกค้าเลือกชำระค่าลงทะเบียนเรียนผ่าน mobile banking
3. ลูกค้าสามารถติดต่อกับพนักงานผ่านแชทของระบบความน่าเชื่อถือสูง
4. พนักงานจะแจ้งวันที่ส่งอุปกรณ์ไปทางemail
- 5.ลูกค้าสามารถติดต่อสอบถามข้อสงสัยในการเรียนได้ตามเวลาที่กำหนดผ่านแชทผู้สอน
- 6.ลูกค้าจะได้รับแจ้งเตือนทุกวันทางemailเมื่อถึงเวลาเรียน

### 1.2)วัตถุประสงค์ของระบบ

1. ลูกค้าสามารถหารายละเอียดก่อนซื้อคอร์สได้ตลอดเวลา
2. ลูกค้าสามารถติดต่อกับพนักงานผ่านแชทของระบบความน่าเชื่อถือสูง และปลอดภัย
3. ลูกค้าเลือกชำระค่าลงทะเบียนเรียนผ่านmobilebankingหรือจัดเก็บเงินปลายทาง
4. ลูกค้าสามารถดูวิดีโอการสอนได้ตลอดเวลาผ่านเว็บไซต์
5. ลูกค้าสามารถถามคำถามผู้สอนได้ผ่านแชท

### 1.3)ประโยชน์ที่คาดว่าจะได้รับ

1. รองรับในการซื้อขายออนไลน์ ในการแสดงสินค้า เลือกสินค้า ชำระสินค้า จัดส่งสินค้า รวดเร็ว สะดวก
2. รองรับความต้องการสารสนเทศของผู้บริหาร เพื่อช่วยในการตัดสินใจในการวางแผนกำหนดกลยุทธ์ต่างๆเพื่อการแข่งขันทางธุรกิจได้ตามที่ต้องการ สะดวก รวดเร็ว
3. รองรับการพัฒนากระบวนการเงินภายในองค์กรต่อไปในอนาคต

4. ข้อมูลมีความถูกต้องแม่นยำ เชื่อถือได้ ทำให้การประมวลผลถูกต้อง แม่นยำ รวดเร็ว สะดวก
5. ไม่ต้องมีหน้าร้านสะดวกในการขาย
6. ไม่ต้องใช้พนักงานจำนวนมาก ประหยัดค่าใช้จ่าย

## 1.4 ขอบเขตการดำเนินงานโครงการ

### 1. ผู้ดูแลระบบ

- สามารถเพิ่ม,ปรับปรุง,ลบ,สิทธิ์การเข้าใช้ระบบของพนักงานทุกคน
- สามารถระงับบัญชีการใช้งานของลูกค้าได้
- กำหนด Data Control, System Control, System Security

### 2. พนักงาน

- สามารถเรียกดูรายละเอียดคำสั่งซื้อได้
- สามารถเรียกดูแซทระหว่างลูกค้าได้
- สามารถเรียกดู ข้อมูลในใบสั่งซื้อได้แก่รายการสินค้าในใบสั่งซื้อ, สถานะการชำระเงิน, สถานะเตรียมส่ง, สถานะการส่งสมบูรณ์

### 3. ผู้จัดการ

- สามารถเรียกดูรายละเอียดคำสั่งซื้อ
- สามารถเรียกดูแซทระหว่างลูกค้ากับพนักงานได้
- สามารถเรียกดูรายละเอียดการจัดส่งของ
- สามารถเรียกดูข้อมูลการชำระเงิน
- สามารถเรียกดู ข้อมูลในใบสั่งซื้อได้แก่รายการสินค้าในใบสั่งซื้อ, สถานะการชำระเงิน, สถานะเตรียมส่ง, สถานะการส่งสมบูรณ์

### 5. ลูกค้า

- สามารถดูรายละเอียดคอร์สเรียนได้
- สามารถสอบถามรายละเอียดเพิ่มเติมทางแซทระหว่างพนักงานได้
- สามารถชำระค่าเรียนได้ผ่านหลายช่องทางตามต้องการ

- สามารถแก้ไขข้อมูลส่วนตัวของตัวเอง

#### 6) พนักงานส่งสินค้า

-สามารถเรียกดูข้อมูลลูกค้าที่จะจัดส่งเฉพาะ

-สามารถเรียกดูข้อมูลสินค้าในใบสั่งซื้อที่มีสถานะเตรียมส่ง แต่ไม่มีสถานะส่งสมบูรณ์

-บันทึกยืนยันลูกค้าได้รับสินค้าแล้ว

#### 7) พนักงานธนาคาร

- ส่งข้อมูลTransaction Fileรายละเอียดการชำระเงินของลูกค้าส่งที่mailของผู้จัดการ

## 1.5 นิยามศัพท์

**Data Control,** ควบคุมข้อมูลส่วนบุคคล(Data

Controller)ผู้ควบคุมข้อมูลส่วนบุคคลคือบุคคลหรือนิติบุคคลที่มีอำนาจหน้าที่ตัดสินใจเกี่ยวกับการเก็บรวบรวมใช้หรือเผยแพร่ข้อมูลส่วนบุคคล

**System Control,** ระบบควบคุม(Control System)

หมายถึงการควบคุมระบบหรือสิ่งที่ผู้ออกแบบต้องการควบคุมให้ได้ค่าผลลัพธ์ในรูปแบบของเอาต์พุตที่ต้องการซึ่งทำได้โดยการป้อนค่าอินพุตให้กับระบบ

**System Security,**

คือการป้องกันข้อมูลสารสนเทศรวมถึงองค์ประกอบอื่นๆที่เกี่ยวข้องเช่นระบบและฮาร์ดแวร์ที่ใช้ในการจัดเก็บและถ่ายโอนข้อมูลสารสนเทศนั้นให้รอดพ้นจากอันตรายอยู่ในสถานะที่มีความปลอดภัยไร้ความกังวลและความกลัว

**Transaction File** แฟ้มข้อมูลรายการเปลี่ยนแปลง(transaction file) เช่น

เจ้าหน้าที่งานทะเบียนจะจัดเก็บแฟ้มข้อมูลนักศึกษาไว้ในหน่วยเก็บข้อมูลสำรองภายในเครื่องคอมพิวเตอร์

## 1.6 ระเบียบวิธีการดำเนินงาน

ระเบียบวิธีการดำเนินงานใช้วิธีSystem Development Life Cycle มี5 Phase คือ



Phase	วัตถุประสงค์	กิจกรรม	สิ่งที่ได้
1	<b>Planning</b>	<ul style="list-style-type: none"> <li>- Define problem online sales system</li> <li>- Produce project schedule</li> <li>- Confirm project feasibility</li> <li>- Staff the project online sales system</li> <li>- Launch project online sales system</li> </ul>	<ul style="list-style-type: none"> <li>- Proposal</li> <li>- Gantt chart</li> </ul>
2	<b>Analysis</b>	<ul style="list-style-type: none"> <li>- Gather information to learn problem domain</li> <li>- Define system requirements</li> <li>- Analysis : Data, Use Case, Data Control, System Control, Security</li> <li>- Build prototypes for discovery of requirements - Prioritize requirements</li> <li>- Generate and evaluate alternatives</li> <li>- Review recommendations with management - In-house Development : produce the diagrams</li> </ul>	<ul style="list-style-type: none"> <li>- Use Case Diagrams - Use Case Description</li> <li>- Activity Diagram</li> <li>- Sequence Diagram</li> </ul>
3	<b>Design</b>	<ul style="list-style-type: none"> <li>- Design and integrate the network, the application architecture, User interfaces, System interfaces, database, Design details, System controls</li> </ul>	<ul style="list-style-type: none"> <li>- User Interface I/O</li> <li>- Class Diagram</li> <li>- Architecture Diagram</li> <li>- Working Logic Subsystem</li> <li>- Presentation Logic Subsystem</li> </ul>
4	<b>Implementation</b>	<ul style="list-style-type: none"> <li>- Construct software components</li> <li>- Verify and test - Convert data</li> <li>- Train users and document the system</li> <li>- Install the system</li> </ul>	<ul style="list-style-type: none"> <li>- Source Program</li> <li>- Testing system</li> <li>- Error Verification Report Of Application Program</li> <li>- Documents</li> <li>- Training</li> </ul>
5	<b>Operations and Support</b>	<ul style="list-style-type: none"> <li>- Maintain system online sales system</li> <li>- Enhance system a) Small upgrades or enhancements to expand system capabilities b) Larger enhancements may require separate development project</li> <li>- Support users : Help desk and/or support team</li> </ul>	<ul style="list-style-type: none"> <li>- New System</li> <li>- Operations</li> <li>- Evaluation</li> <li>- Maintenance</li> </ul>

## 1.7) ทรัพยากรหรือเครื่องมือในการวิจัยหรือพัฒนาระบบ

1. ค่า Develop
2. ค่าจด Domain Name
3. ค่า Web Hosting
4. Server ( Google : CPU 32 core, RAM 16 GB )
5. Network พร้อมติดตั้ง 1 set
6. Network O/S ( Linux ) 1 set
7. RedhatEnterprise Linux Server 1 set
8. Document + Training

## 1.8)แผนการดำเนินงานโครงการ

### แผนการดำเนินงาน

กิจกรรม	ธ.ค.				ม.ค.				ก.พ.				มี.ค.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. Planning	✓	✓														
2. Analysis			✓	✓	✓	✓										
3. Design						✓	✓	✓	✓	✓	✓	✓				
4. Implementation										-	-	-	-	-	-	-
5. Operations										-	-	-	-	-	-	-

## 1.9)งบประมาณที่ใช้

### งบประมาณ ( หน่วย : บาท )

1. ค่า Develop	600,000
2. ค่าจด Domain Name	1,000 ต่อปี
3. ค่า Web Hosting	2,000 ต่อปี
4. Server ( Google : CPU 32 core, RAM 16 GB )	50,000 ต่อเดือน
5. Network พร้อมติดตั้ง 1 set	25,000
6. Network O/S ( Linux ) 1 set	40,000
7. Redhat Enterprise Linux Server 1 set	13,000
8. Document + Training	10,000
รวมเงินลงทุน ( Investment Cost )	741,000 บาท

ค่า Maintenance ( ต่อปี )	
ปีที่ 1	Free + Guatantee
ปีที่ 2	200,000 บาท
ปีที่ 3	500,000 บาท
ปีที่ 4	800,000 บาท
ปีที่ 5	1,200,000 บาท

## 2.วรรณกรรมที่เกี่ยวข้อง

**System Development Life Cycle ( SDLC )** ประกอบด้วย5ขั้นตอนดังนี้

### 1. System Planning

- Define business problem and scope
- Produce detailed project schedule
- Confirm project feasibility
- Staff the project (resource management )
- Launch project a official announcement

### 2. System Analysis

- Gather information to learn problem domain
- Define system requirement
- Analysis : Data, Use Case, Data Control, System Control, Security
- Build prototypes for discovery of requirements
- Prioritize requirements
- Generate and evaluate alternatives
- Review recommendations with management
- In-house Development produce the diagrams : Use Case Diagram, Use CaseDescription, Activity Diagram, Sequence Diagram

### 3. System Design

- Design and integrate the network

- Design the application architecture
- Design the user interfaces
- Design the system interfaces
- Design and integrate the database
- Prototype for design details
- Design and integrate system controls

#### **4. Implementation Phase or Development Phase**

- Construct software components
- Verify and test – Convert data
- Train users and document the system
- Install the system

#### **5. Operation and Support ( Operation and Maintenance )**

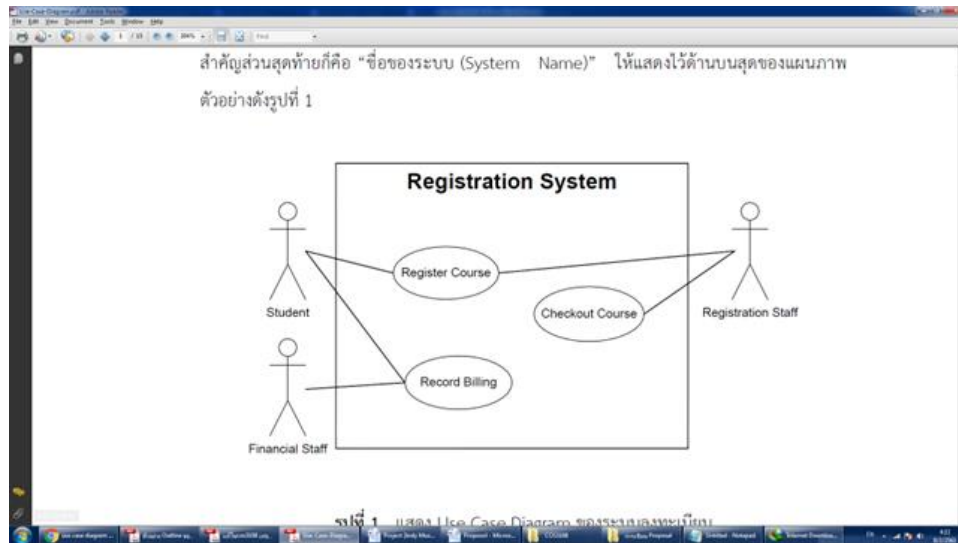
- Maintain system : Small patches, repairs, and updates
- Enhance system a. Small upgrades or enhancements to expand system capabilities  
b. Larger enhancements may require separate development project
- Support users : Help desk and/or support team

### **ทฤษฎีการสร้าง Diagram**

#### **Use Case Diagram**

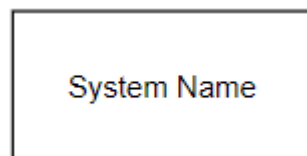
Use Case Diagram เป็นแผนภาพที่ใช้แสดงให้เห็นว่าระบบทำงานหรือมีหน้าที่ใดบ้าง โดยมีสัญลักษณ์รูปวงรีแทน Use Case และสัญลักษณ์รูปคน (Stick Man Icon) แทน Actor สำหรับชื่อ Use Case นั้น ให้ใช้คำกริยาหรือกริยาวลี เช่น ลงทะเบียนเรียน, ตรวจสอบรายวิชา, บันทึกการชำระเงิน, Generate Report, Enter Sales Data, Compute Commission เป็นต้น ส่วนการมีปฏิสัมพันธ์ระหว่าง Use Case และ Actor จะใช้เส้นตรงลากมาเชื่อมต่อกัน โดยที่เส้นตรงจะมีหัวลูกศรหรือไม่ก็ได้ ส่วนเส้นแบ่งขอบเขต ระหว่าง Actor กับ Use Case จะใช้เส้นกรอบสี่เหลี่ยม เรียกว่า

“System Boundary” และสิ่งสำคัญส่วนสุดท้ายก็คือ “ชื่อของระบบ (System Name)”  
ให้แสดงไว้ด้านบนสุดของแผนภาพ

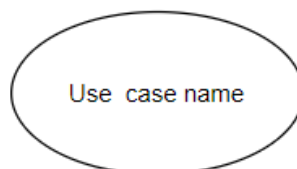


สัญลักษณ์ที่สำคัญของ Use Case Diagram มีดังต่อไปนี้

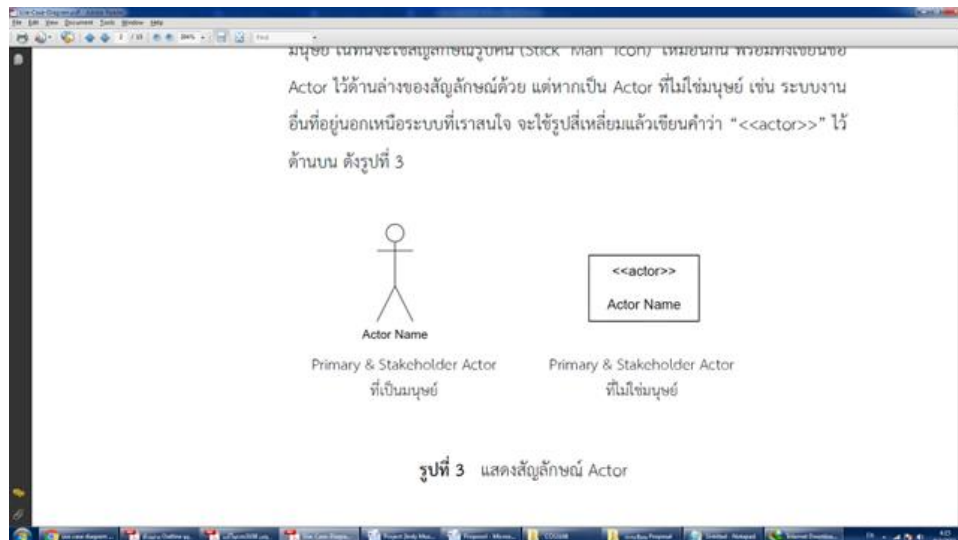
-System Boundaryคือ เส้นแบ่งขอบเขตระหว่างระบบกับผู้กระทำต่อระบบ (Use Case กับ Actor) ใช้รูป สี่เหลี่ยมเป็นสัญลักษณ์ พร้อมทั้งเขียนชื่อระบบไว้ด้านใน



-Use Caseคือ หน้าที่ที่ระบบต้องกระทำ ใช้สัญลักษณ์รูปวงรี พร้อมทั้งเขียนชื่อ Use Case ซึ่งมักใช้เป็นคำกริยา

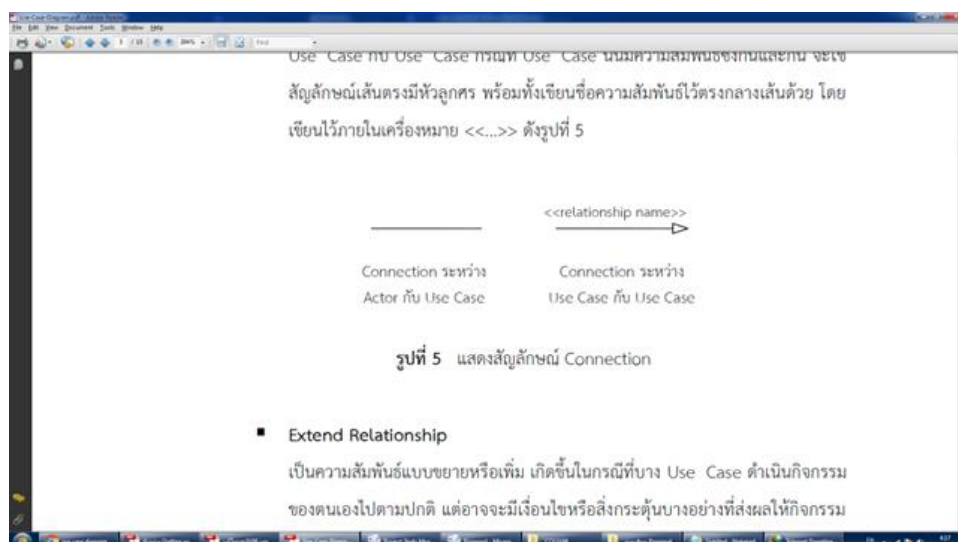


-Actor คือ ผู้เกี่ยวข้องกับระบบ เช่น ผู้จัดการ , พนักงานขาย , Admin  
ในที่นี้จะใช้สัญลักษณ์รูปคน (Stick Man Icon) เหมือนกัน พร้อมทั้งเขียนชื่อ Actor ไว้ด้านล่างของสัญลักษณ์ด้วย



-Connection\_คือ เส้นที่ลากเชื่อมต่อระหว่าง Actor กับ Use Case ที่มีปฏิสัมพันธ์กัน ใช้เส้นตรงที่ไม่มีหัวลูกศรเป็นสัญลักษณ์ของ Connection

ส่วน Connection ที่ใช้เชื่อมต่อระหว่าง Use Case กับ Use Case กรณีที่ Use Case นั้นมีความสัมพันธ์ซึ่งกันและกัน จะใช้ เส้นตรงที่มีหัวลูกศร พร้อมทั้งเขียนชื่อความสัมพันธ์ไว้ตรงกลางเส้นด้วย โดย เขียนไว้ภายในเครื่องหมาย <<...>> โดยหลักๆแล้วจะมี Extend กับ Include



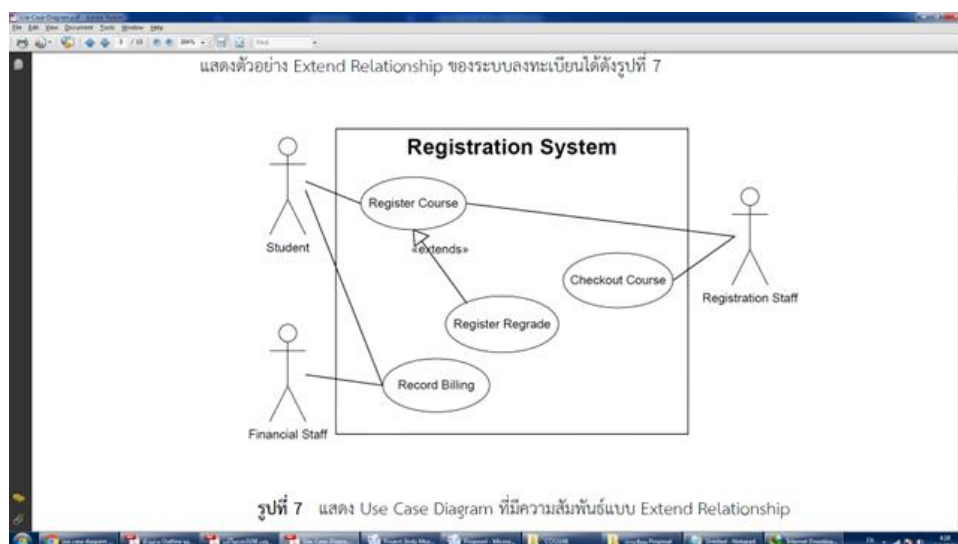
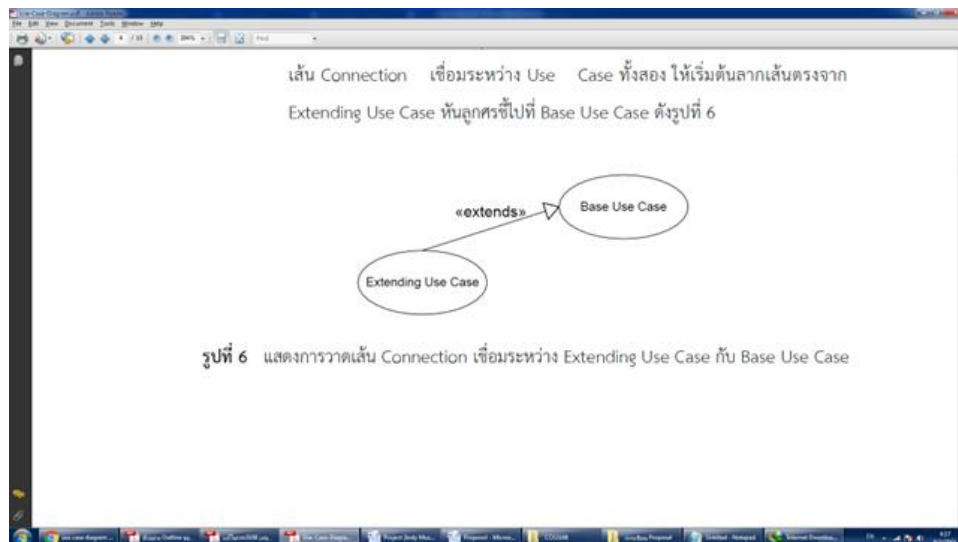
--Extend Relationship เป็นความสัมพันธ์แบบขยายหรือเพิ่ม เกิดขึ้นในกรณีที่บาง Use Case ดำเนินกิจกรรม ของตนเองไปตามปกติ

แต่อาจจะมีเงื่อนไขบางอย่างที่ส่งผลให้กิจกรรมตามปกติของ Use Case

นั้นถูกรบกวนจนเบี่ยงเบนไป เมื่อเกิดเหตุการณ์ผิดปกติขึ้น จะต้องทำหน้าที่พิเศษเพิ่ม

โดยหน้าที่พิเศษที่เพิ่มขึ้นก็คือ “Extending Use Case” นั่นเอง ดังนั้น อาจกล่าวได้ว่า Use Case ที่เป็น Extending Use Case จะเกิดขึ้นเพียงบางครั้งเท่านั้น โดยเรียก Use Case ที่ถูกรบกวนหรือ Use Case ที่ดำเนินงานตามปกติว่า “Base Use Case” และเรียก Use Case ที่ทำ หน้าทีรบกวน Base Use Case ว่า “Extending Use Case”

การวาด เส้น Connection เชื่อมระหว่าง Use Case ทั้งสอง ให้เริ่มต้นลากเส้นตรงจาก Extending Use Case หันลูกศรชี้ไปที่ Base Use Case



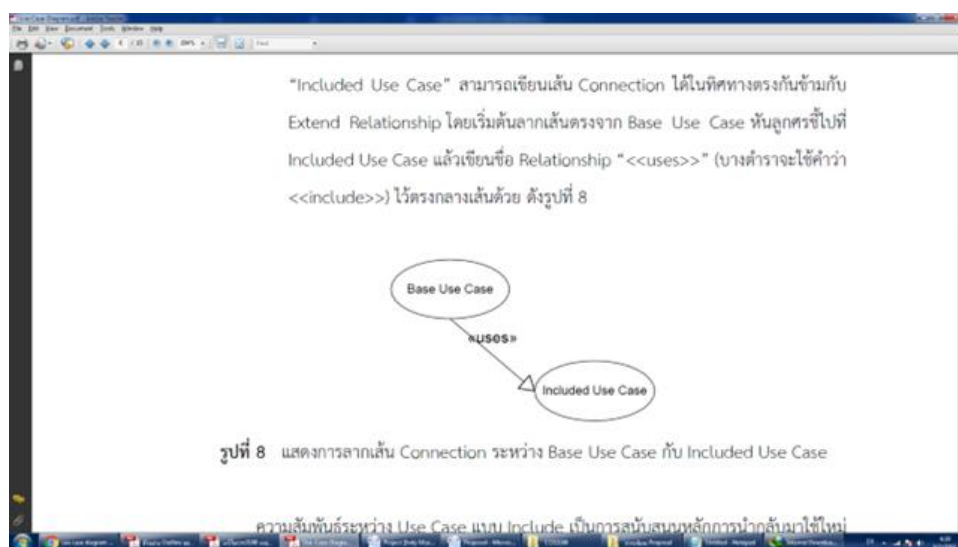
### Extends Relationship

จากรูป สังเกตที่ Use Case “Register Course” ซึ่งเป็น Base Use Case คือ ทาหน้าที่ รับลงทะเบียนตามปกติ แต่เมื่อมีเงื่อนไขเกิดขึ้น คือ “นักศึกษาบางคนอาจมีการ



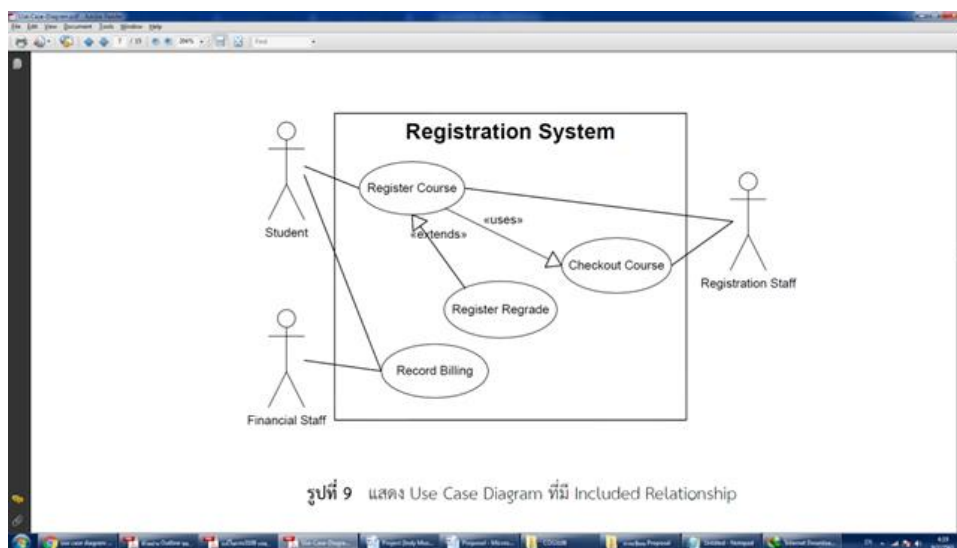
ลงทะเบียนเรียนซ้ำเพื่อ Regrade” จึงได้เพิ่ม Extending Use Case เพื่อมารองรับ  
หน้าที่พิเศษดังกล่าว นั่นคือ “Register Regrade”

--Include Relationship เป็นความสัมพันธ์อีกรูปแบบหนึ่งของ Use Case Diagram ก็คือ  
ความสัมพันธ์แบบเรียกใช้ เกิดขึ้นในกรณีที่ Use Case หนึ่งไปเรียกหรือดึงกิจกรรมของอีก  
Use Case หนึ่งมาใช้ เพื่อให้กิจกรรมนั้นเกิดขึ้นจริงใน Use Case ของตนเอง  
เราเรียกความสัมพันธ์ระหว่าง Use Case ในลักษณะนี้ว่า “Include Relationship” โดย Use  
Case ที่ทำหน้าที่ดึงกิจกรรมมาจาก Use Case อื่นๆ เรียกว่า “Base Use Case” ในขณะที่  
Use Case ที่ถูกเรียก หรือถูกดึงกิจกรรมมาใช้ เรียกว่า “Included Use Case”  
สามารถเขียนเส้น Connection ได้ในทิศทางตรงกันข้ามกับ Extend Relationship  
โดยเริ่มต้นลากเส้นตรงจาก Base Use Case หันลูกศรชี้ไปที่ Included Use Case  
แล้วเขียนชื่อ Relationship “<<uses>>” (บางตำราจะใช้คำว่า <<include>>)  
ไว้ตรงกลางเส้นด้วย



ความสัมพันธ์ระหว่าง Use Case แบบ Include เป็นการสนับสนุนหลักการนำกลับมาใช้ใหม่  
ของ Use Case (Use Case Reusability) กล่าวคือ Use Case หนึ่งสามารถถูก Include  
ได้โดย Base Use Case หลายๆ ตัว และสามารถถูก Include ได้มากกว่าหนึ่งครั้งด้วย เช่น  
ในการทำงาน ของระบบ ATM Use Case “การตรวจสอบผู้เข้าใช้ระบบ (Validate User)”  
สามารถเป็น Included Use Case ให้กับ Base Use Case หลายๆ ตัว ได้แก่ Base Use  
Case “การถอนเงิน

(Withdraw Money)” และ Base Use Case “การโอนเงิน (Transfer Money)”



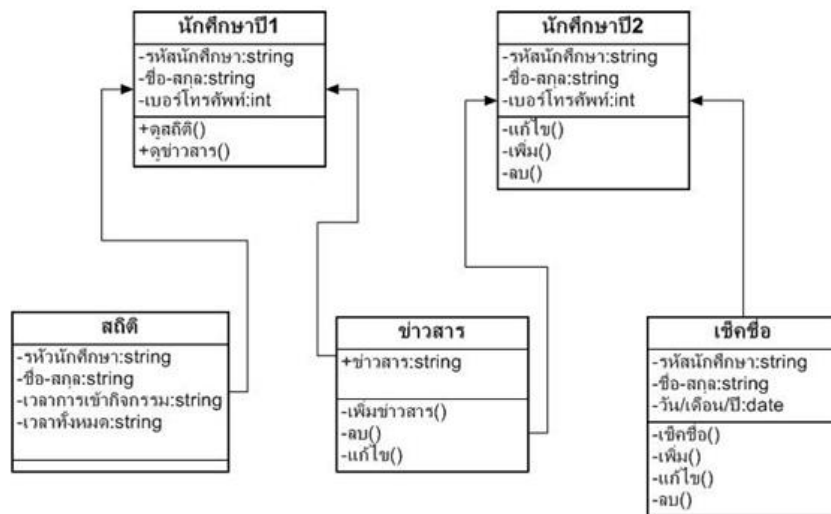
Include

### Relationship

เมื่อพิจารณาแล้ว Use Case “ตรวจสอบรายวิชา (Checkout Course)” สามารถถูกเรียกใช้จาก Use Case “ลงทะเบียนเรียน (Register Course)” ได้ ดังนั้น Use Case “Checkout Course” มีความสัมพันธ์กับ Use Case “Register Course” แบบ Include แสดง Use Case Diagram อีกครั้ง

### Class Diagram

คือ แผนภาพที่ใช้แสดง Class และความสัมพันธ์ในแง่ต่าง ๆ (Relation) ระหว่าง Class เหล่านั้น ซึ่งความสัมพันธ์ที่กล่าวถึงใน Class Diagram นี้ถือเป็นความสัมพันธ์เชิงสถิต (Static Relationship) หมายถึง ความสัมพันธ์ที่มีอยู่แล้วเป็นปกติในระหว่าง Class ต่าง ๆ ไม่ใช่ความสัมพันธ์ที่เกิดขึ้นเนื่องจากกิจกรรมต่างๆ ซึ่งเรียกว่า ความสัมพันธ์เชิงกิจกรรม (Dynamic Relationship) สิ่งปรากฏใน Class Diagram นั้นประกอบด้วยกลุ่มของ Class และกลุ่มของ Relationship โดยสัญลักษณ์ที่ใช้ในการแสดง Class นั้นจะแทนด้วยสี่เหลี่ยมที่แบ่งออกเป็น 3 ส่วน โดยแต่ละส่วนนั้น จะใช้ในการแสดง ชื่อของ Class, Attribute, และฟังก์ชัน ต่าง ๆ ตามลำดับจากบนลงล่าง



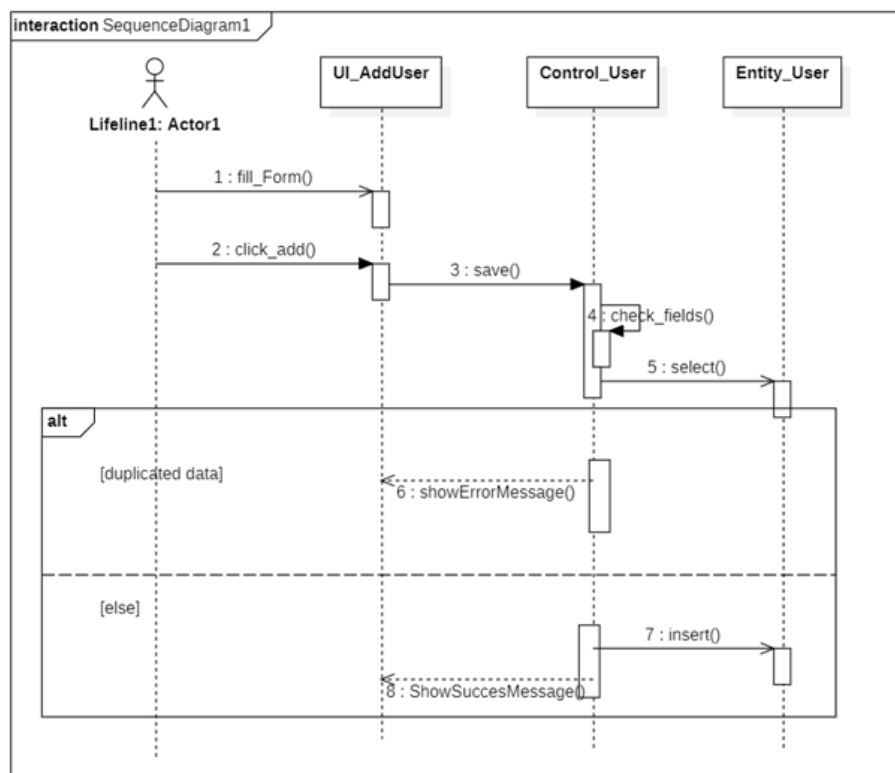
ในการเขียนสัญลักษณ์แทน Class สิ่งที่ต้องคำนึงถึงอีกสิ่งหนึ่งคือ ระดับการเข้าถึงเรียกสัญลักษณ์ที่ใช้แทนการเข้าถึงนี้ว่า Visibility แบ่งออกได้เป็น 3 ประเภท ได้แก่

1. Private เขียนแทนด้วยสัญลักษณ์ - หมายถึง Attribute หรือ ฟังก์ชัน ที่ไม่สามารถมองเห็นได้จากภายนอก แต่สามารถมองเห็นได้จากภายในตัวของ Class เองเท่านั้น
2. Protect เขียนแทนด้วยสัญลักษณ์ # หมายถึง Attribute หรือ ฟังก์ชันเหล่านี้ จะเป็นของ Super class เมื่อทำการ Inheritance แล้ว Attribute หรือ ฟังก์ชัน ที่มี Visibility แบบ Protect จะกลายเป็น Private Attribute/ฟังก์ชัน หรือ Protected ขึ้นอยู่กับภาษา Programming ที่นำไปใช้
3. Public เขียนแทนด้วยสัญลักษณ์ + หมายถึง Attribute หรือ ฟังก์ชัน ที่สามารถมองเห็นได้จากภายนอก และสามารถเข้าไปเปลี่ยนค่า อ่านค่าหรือเรียกใช้งาน **Attribute** หรือ ฟังก์ชัน นั้นได้ทันทีโดยอิสระจากภายนอก โดยปกติแล้ว Visibility แบบ Public มักจะใช้กับ Attribute

## Sequence Diagram

เป็นแผนภาพที่แสดงให้เห็นถึงการปฏิสัมพันธ์ (Interaction) ระหว่างobjectโดยเฉพาะ การส่ง Message ระหว่างobjectตามลำดับของเวลา (Sequence)

ที่เกิดเหตุการณ์ขึ้นจากน้อยไป มาก โดยจะมีสัญลักษณ์แสดงให้เห็นลำดับของการส่ง Message ตามเวลาส่งอย่างชัดเจน แสดง ลักษณะของ Sequence Diagram



ส่วนประกอบของ Sequence Diagram

Actor คือ ผู้กระทำต่อระบบ

Object คือ object ที่ต้องทำหน้าที่

Lifeline คือ เส้นแสดงชีวิตของ object หรือคลาส

Focus of Control / Activation จุดเริ่มต้นและจุดสิ้นสุดของแต่ละกิจกรรมในระหว่างที่มีชีวิตอยู่

A Message คำสั่งหรือฟังก์ชันที่คลาสหนึ่งส่งให้อีกคลาสหนึ่ง ซึ่งสามารถ ส่งกลับได้ด้วย

Callback / Self Delegation คือ การประมวลผลและคือค่าที่ได้ภายใน object เดียวกัน

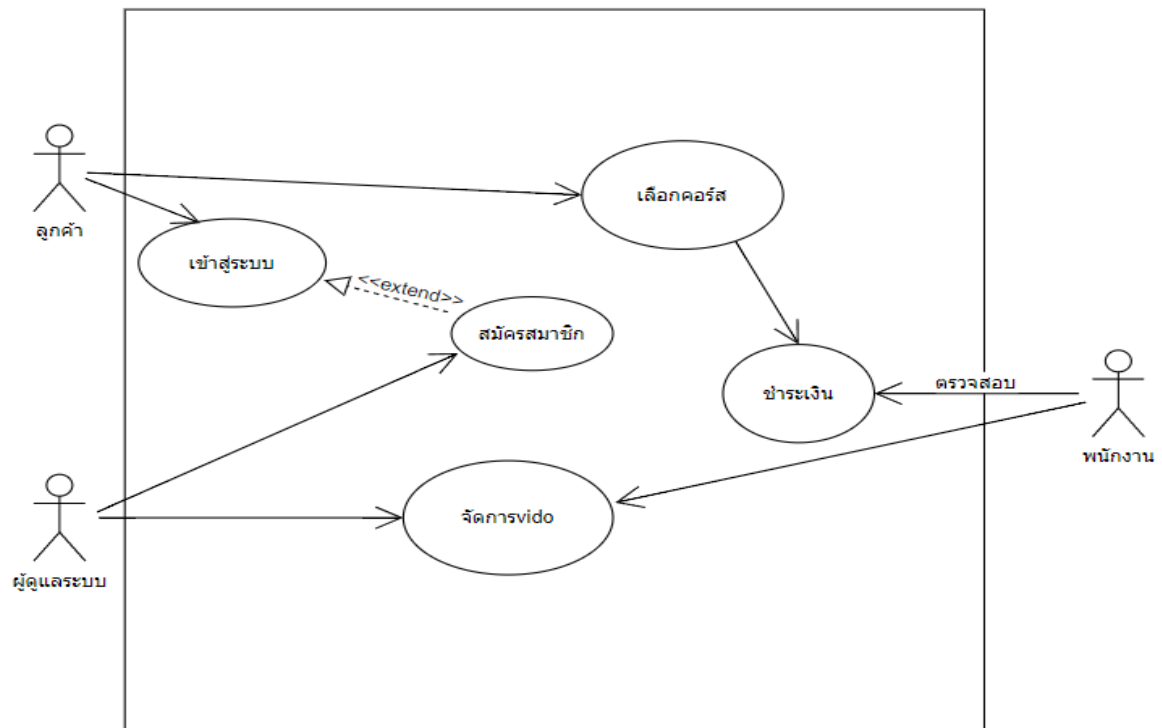
### 3.ขั้นตอนการดำเนินงาน

#### 3.1วิเคราะห์ระบบ

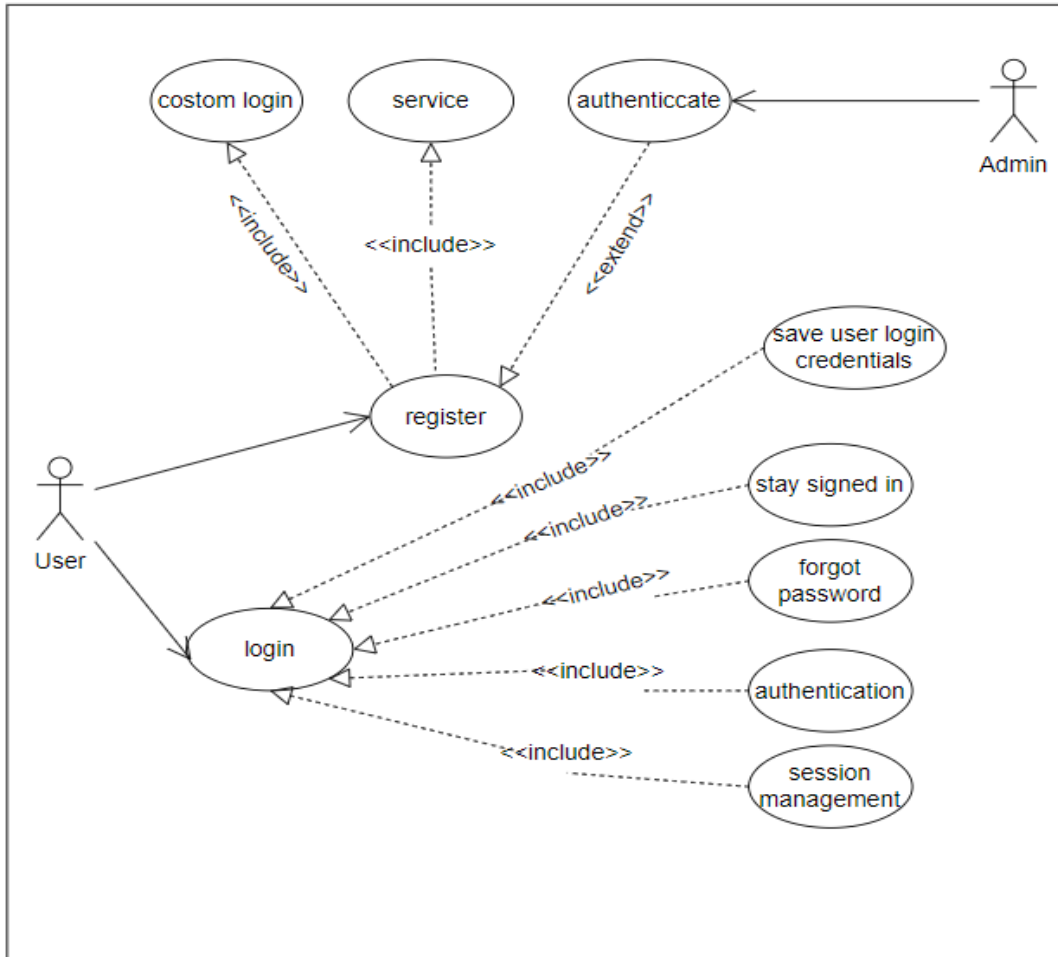
##### 3.1.1 ตาราง Use case diagram

Diagram ID	Diagram name	Diagram Description
UC-main	UC-Main	Use Case diagram
UC-01	UC-001	การเข้าสู่ระบบ
UC-02	UC-002	จัดการสินค้า
UC-03	UC-003	เลือกคอร์สเรียน
UC-04	UC-004	การชำระเงิน

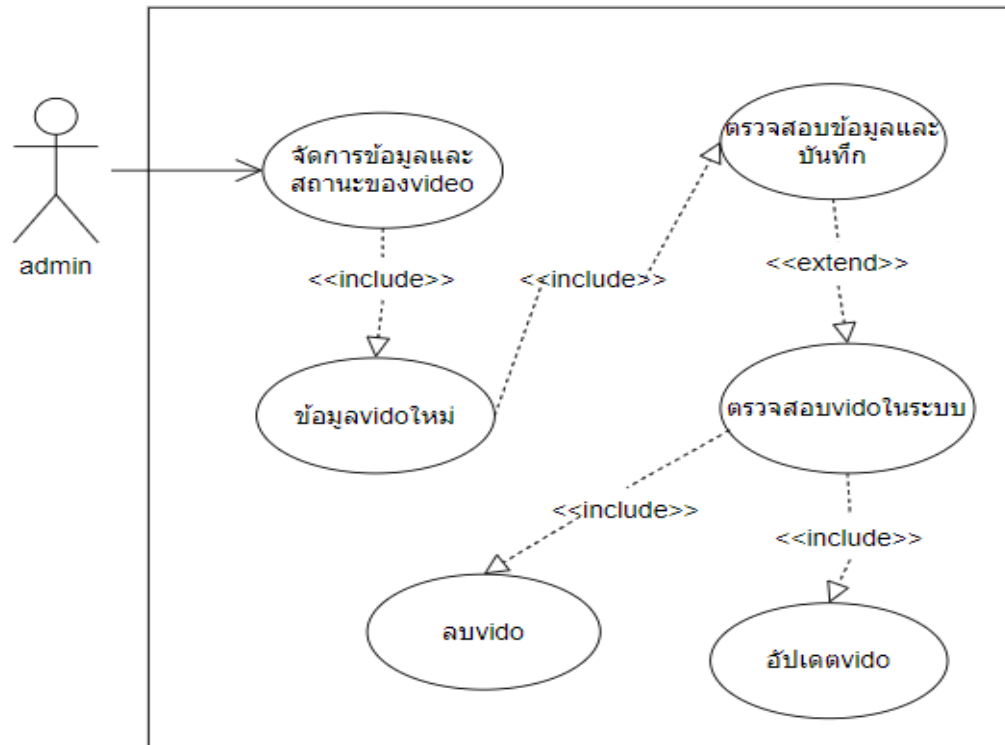
## Use Case Diagram (Main)



## Use Case Diagram :UC-001(การเข้าสู่ระบบ)

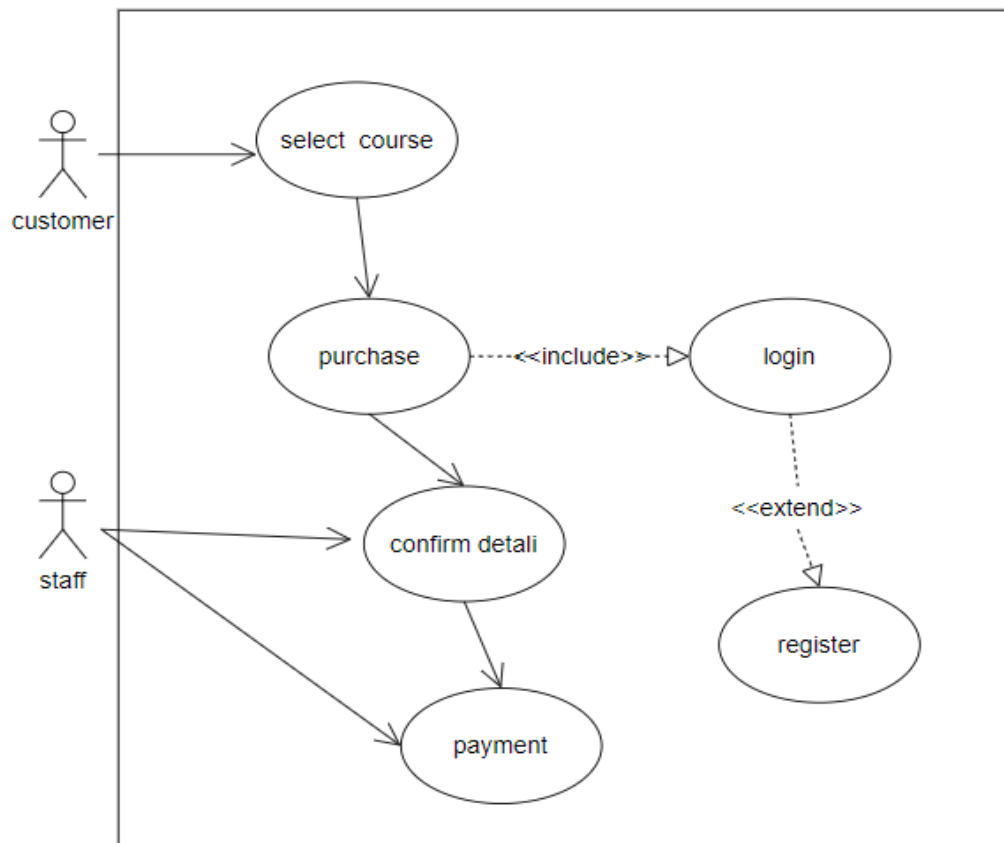


## Use Case Diagram : UC-002 (การจัดการสินค้า)

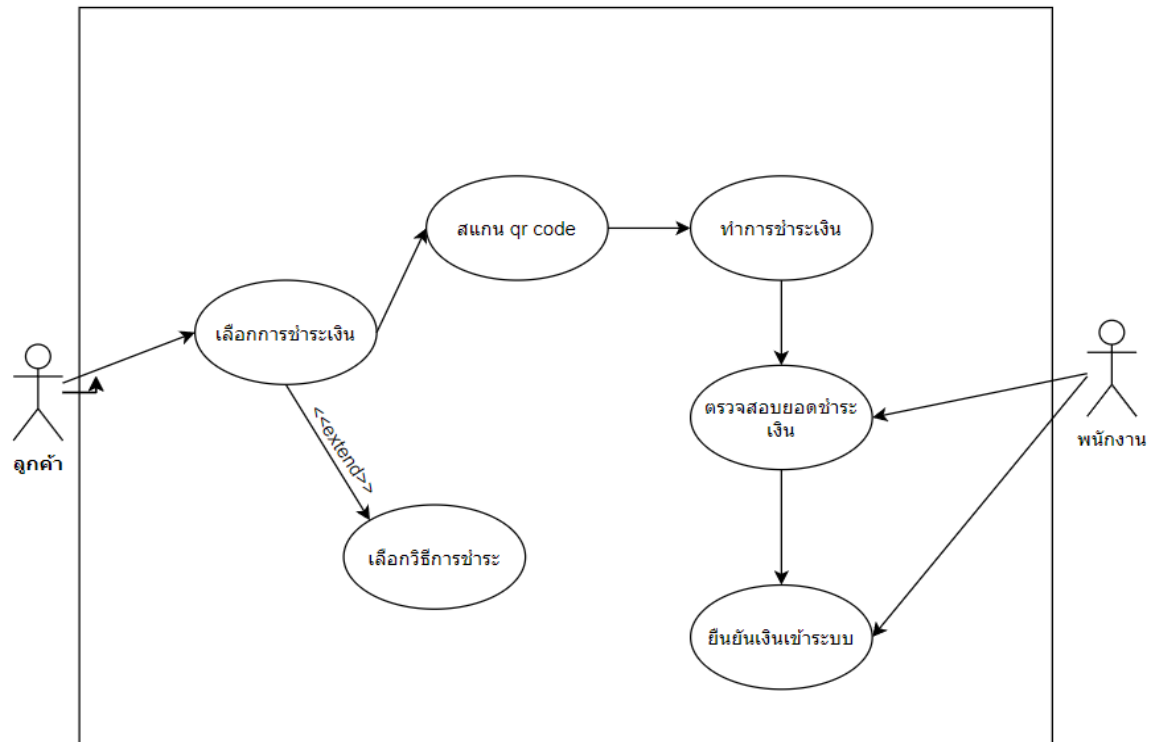




## Use Case Diagram : UC-003 (เลือกคอร์สเรียน)



## Use Case Diagram : UC-004 (การชำระเงิน)



## 1.2 ) USE CASE DESCRIPTION

### - USE CASE DESCRIPTION ของ UC-001

Name	Use Case Description: การเข้าสู่ระบบ
Use Case ID	UC-01
Use Case Name	UC-001
Actor	ลูกค้า
Purpose	เพื่อให้ลูกค้าสามารถเข้าดูวิดีโอการสอนได้
Preconditions	ลูกค้าต้องมีบัญชีผู้ใช้ของตัวเอง
Study Conditions	ลูกค้าทำการสำเร็จ
Main Flows	1. ลูกค้าเลือกเมนูการเข้าสู่ระบบ 2. ลูกค้าตัดสินใจแล้วกดเลือกlogin 3. ลูกค้าต้องมีบัญชีผู้ใช้เพื่อเข้าใช้ระบบ 4. ลูกค้าทำการกดสมัครสมาชิก 5. ลูกค้ายืนยันรายละเอียด ข้อมูลของตัวเอง 6. เสร็จสิ้น
Exceptions	หากลูกค้ายังไม่มีบัญชีผู้ใช้ของตัวเอง ระบบจะให้ลูกค้าทำการสมัครก่อน
Special Requirement	มีการอัปเดตสถานะความถูกต้องของข้อมูล
Technology and Data Variation List	มีการแจ้งเตือนสถานะในเว็บไซต์ของUser
Frequency of Occurrence	ตลอดเวลา
Open Issues	การจะเข้าดูวิดีโอคอร์สการสอน หรือ ทำการซื้อคอร์สออนไลน์จำเป็นต้องมี บัญชีผู้ใช้ในฐานข้อมูลก่อนระบบจะทำการตรวจสอบก่อนว่ามีหรือไม่ ถ้ามีระบบจะทำการเข้าสู่ระบบ หากไม่มีระบบจะลูกค้าไปสมัครบัญชีก่อน

## - USE CASE DESCRIPTION ของ UC-002

Name	Use Case Description: การจัดการสินค้า
Use Case ID	UC-02
Use Case Name	UC-002
Actor	ผู้ดูแลระบบ
Purpose	เพื่อให้ผู้ดูแลสามารถจัดการกับข้อมูลต่างๆในระบบได้
Preconditions	ต้องมีบันชีผู้ดูแลระบบ
Study Conditions	-
Main Flows	<ol style="list-style-type: none"> <li>1.ผู้ดูแลจัดการข้อมูลและสถานะของวิดีโอ</li> <li>2. ผู้ดูแลทำการอัปเดตข้อมูลใหม่</li> <li>3. ผู้ดูแลตรวจสอบก่อนบันทึก</li> <li>4. ระบบตรวจสอบความถูกต้องของข้อมูล ก่อนแสดงผลหน้าจอ</li> <li>5. ระบบทำการอัปเดตข้อมูล เรียบร้อย</li> <li>6. ผู้ดูแลสามารถแก้ไข หรือทำการลบข้อมูลได้</li> <li>7.เสร็จสิ้น</li> </ol>
Exceptions	<p>หากใส่ข้อมูลไม่ครบถ้วน</p> <p>ระบบจะทำการแจ้งเตือนให้ใส่ข้อมูลใหม่อีกครั้งให้ครบถ้วน</p> <p>จึงสามารถทำการบันทึกหรือจัดเก็บข้อมูลได้สำเร็จ</p>
Special Requirement	ดูรายการสินค้าได้ง่ายไม่ log in ก็สามารถดูสินค้าได้
Technology and Data	--
Variation List	
Frequency of Occurrence	--
Open Issues	ต้องเข้าสู่ระบบก่อนถึงสามารถดูข้อมูลสินค้าได้ครบถ้วน

## - USE CASE DESCRIPTION ของ UC-003

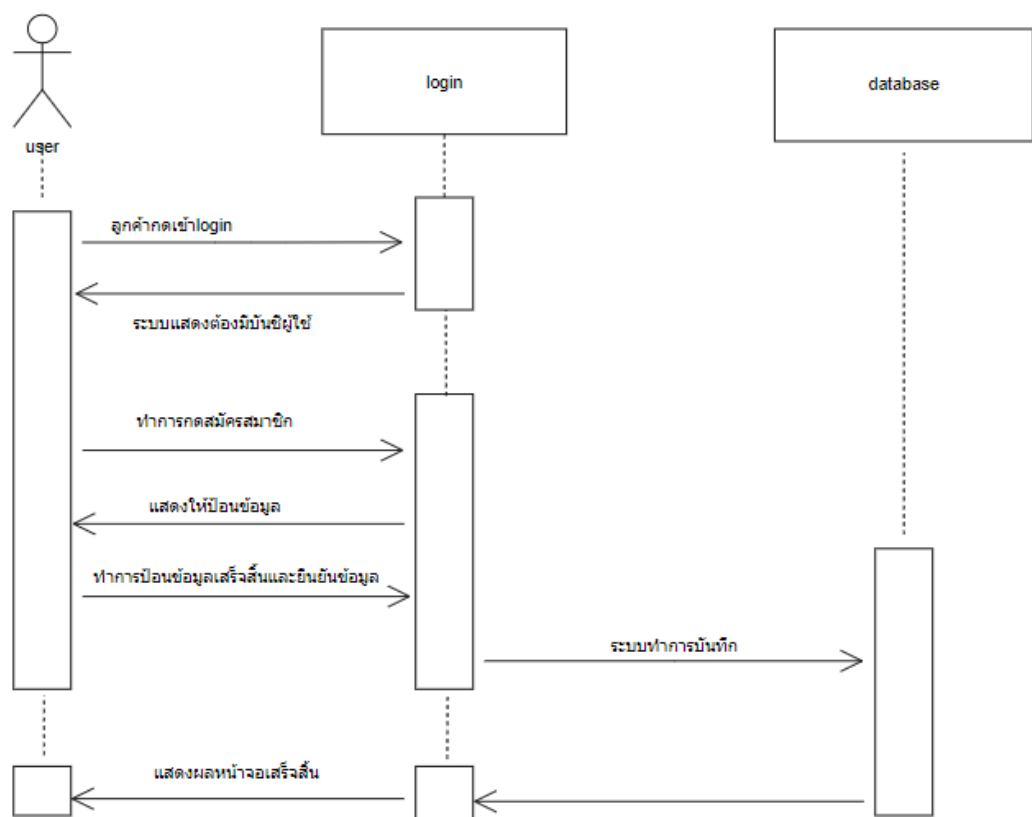
Name	Use Case Description: เลือกคอร์สที่ต้องการเรียน
Use Case ID	UC-03
Use Case Name	UC-003
Actor	ลูกค้า
Purpose	เพื่อให้ลูกค้าสั่งซื้อผ่านระบบออนไลน์ได้
Preconditions	ลูกค้าต้องมีบัญชีผู้ใช้ของตัวเอง
Study Conditions	ลูกค้าทำการสำเร็จ
Main Flows	<ol style="list-style-type: none"> <li>1. ลูกค้าเลือกคอร์สที่ต้องการเรียน</li> <li>2. ลูกค้าตัดสินใจแล้วกดซื้อ</li> <li>3. ลูกค้าต้องมีบัญชีผู้ใช้เพื่อเข้าใช้ระบบ</li> <li>4. ลูกค้ายืนยันรายละเอียด คอร์สที่เลือกที่อยู่จัดส่ง ช่องทางการชำระ</li> <li>5. ลูกค้าชำระเงิน</li> <li>6. เสร็จสิ้น แสดงรายละเอียดคอร์สเรียน วันที่จัดส่งหนังสือประกอบการเรียน</li> </ol>
Exceptions	ลูกค้าต้องมีบัญชีผู้ใช้ของตัวเองก่อนซื้อคอร์สเรียน
Special Requirement	ลูกค้าสามารถดูรายละเอียดคอร์สต่างๆก่อนเลือกซื้อได้
Technology and Data Variation List	จะมีการแจ้งเตือนว่าพนักงานออนไลน์มี
Frequency of Occurrence	ตลอดเวลา
Open Issues	ลูกค้าสามารถสอบถามข้อมูลเพิ่มเติมได้ในแชทกับพนักงานเมื่อออนไลน์ตามเวลาทำงาน

## - USE CASE DESCRIPTION ของ UC-004

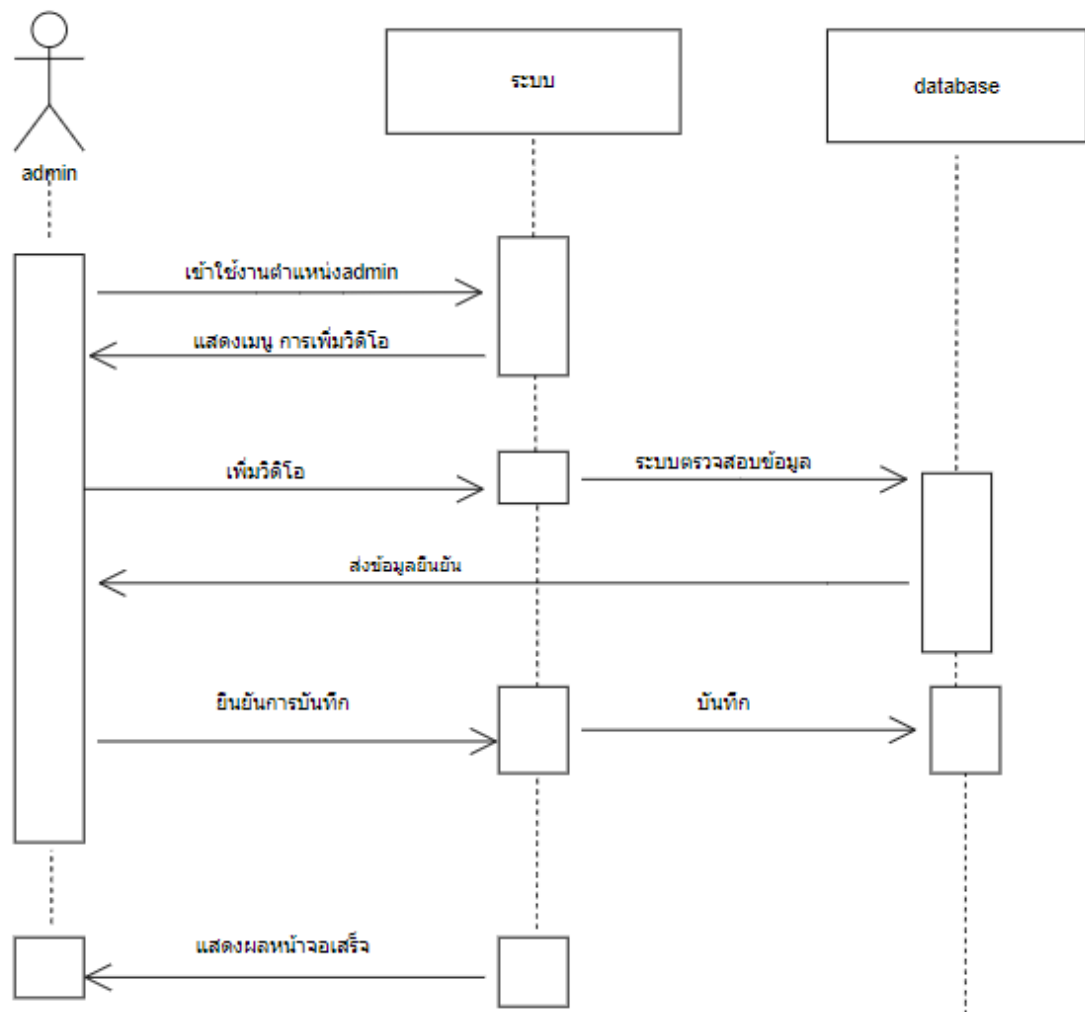
Name	Use Case Description: การชำระเงิน
Use Case ID	UC-04
Use Case Name	UC-004
Actor	ลูกค้าและพนักงาน
Purpose	เพื่อให้ลูกค้าสามารถชำระเงินได้ง่าย
Preconditions	ลูกค้าต้องทำการเลือกคอร์สก่อน
Study Conditions	ลูกค้าทำรายการสำเร็จ
Main Flows	<ol style="list-style-type: none"> <li>1. ลูกค้าเลือกเมนูการชำระเงิน</li> <li>2. ลูกค้าเลือกวิธีการชำระ</li> <li>3. ลูกค้าสแกน qr code การชำระเงินที่ระบบแสดงมาให้ในหน้าจอ</li> <li>4. ลูกค้าทำการชำระ</li> <li>5. ลูกค้ารอการยืนยันการชำระจากระบบ</li> <li>6. เสร็จสิ้น</li> </ol>
Exceptions	ลูกค้ายังไม่มีทำการเลือกคอร์สที่จะซื้อไว้ก่อน
Special Requirement	สามารถตรวจสอบรายการการชำระเงินได้จากการชำระเงินของลูกค้า
Technology and Data Variation List	ต้องมี Transection ในการตรวจสอบยอดเงิน
Frequency of Occurrence	ตามเวลาทำการของพนักงาน
Open Issues	ถ้ายังไม่มีชำระเงินภายในเวลาที่กำหนดระบบจะทำการยกเลิกการสั่งซื้อโดยอัตโนมัติ

## 1.3) SEQUENCE DIAGRAM

### – Sequence Diagram ของ UC-01

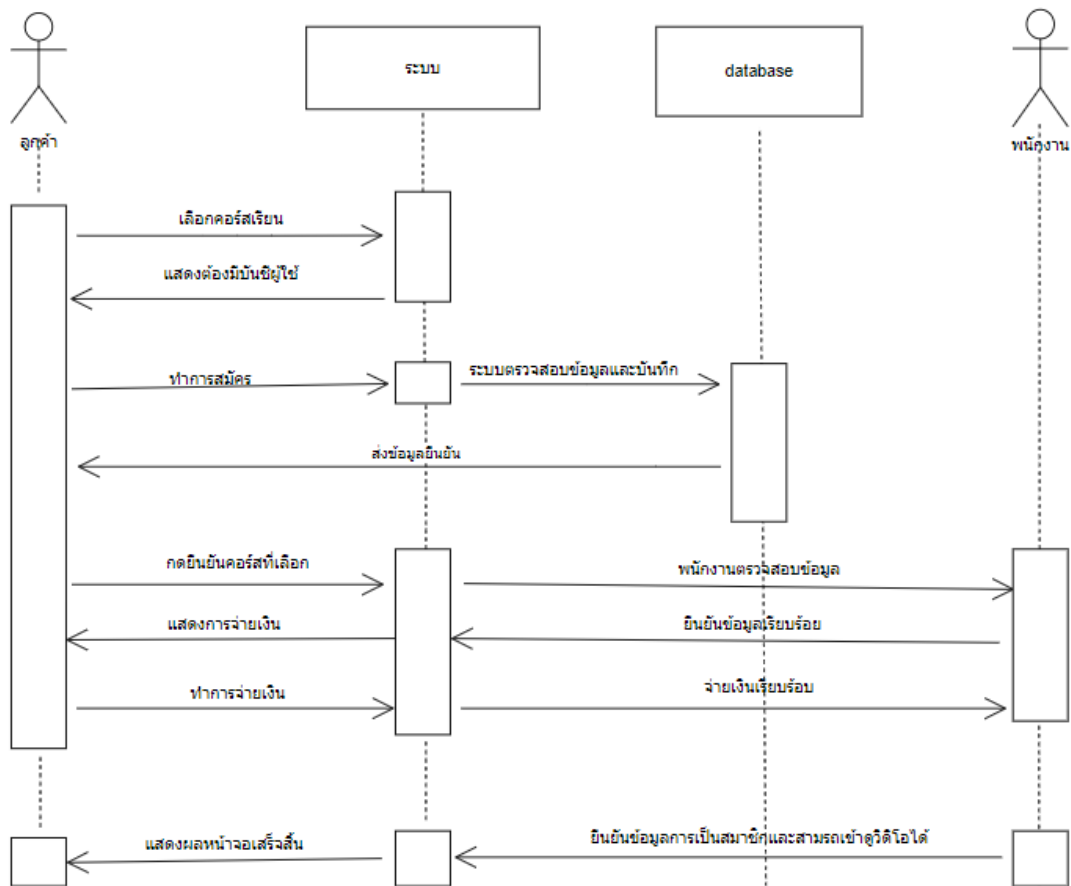


– Sequence Diagram ของ UC-02

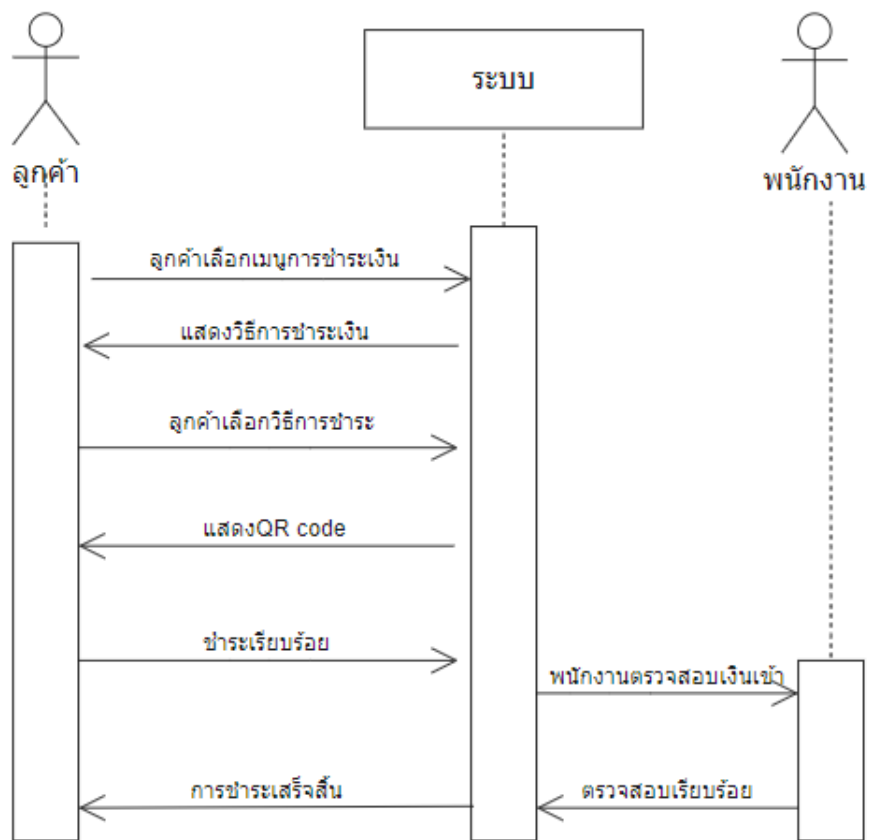




## – Sequence Diagram ของ UC-03

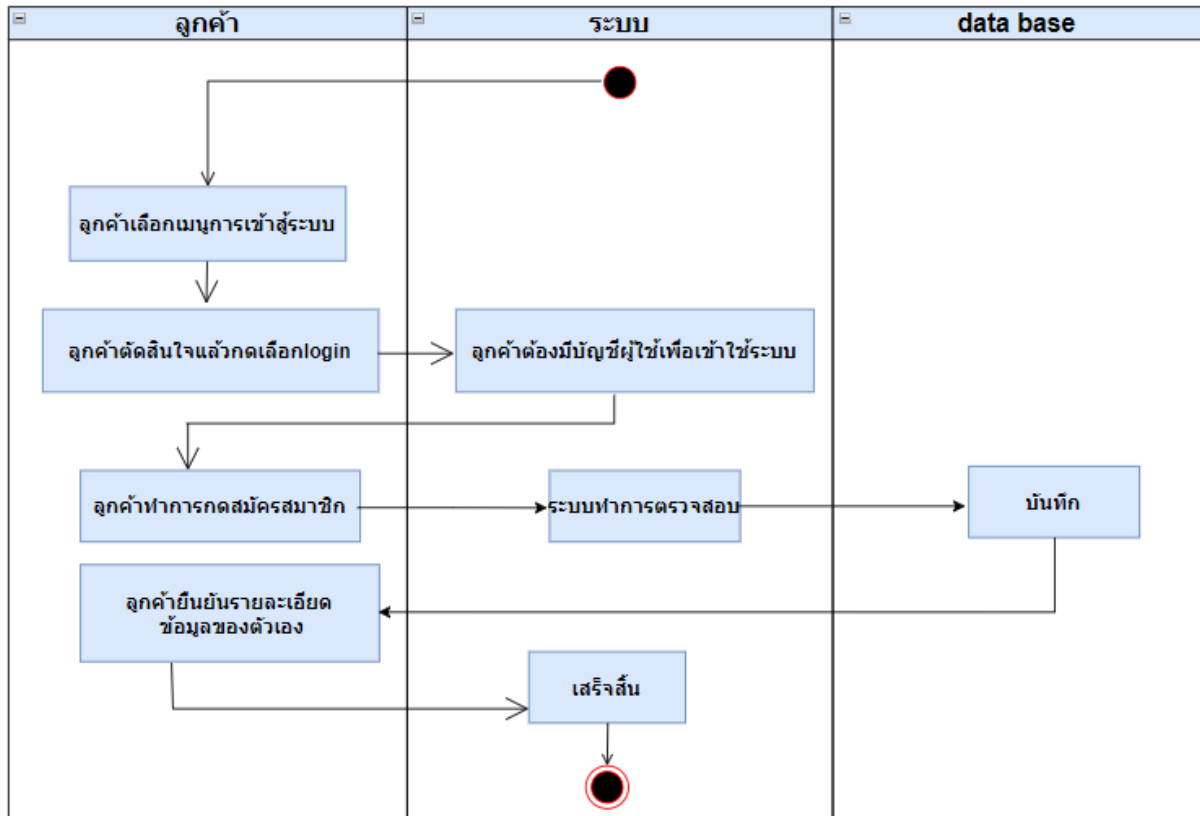


– Sequence Diagram ของ UC-04

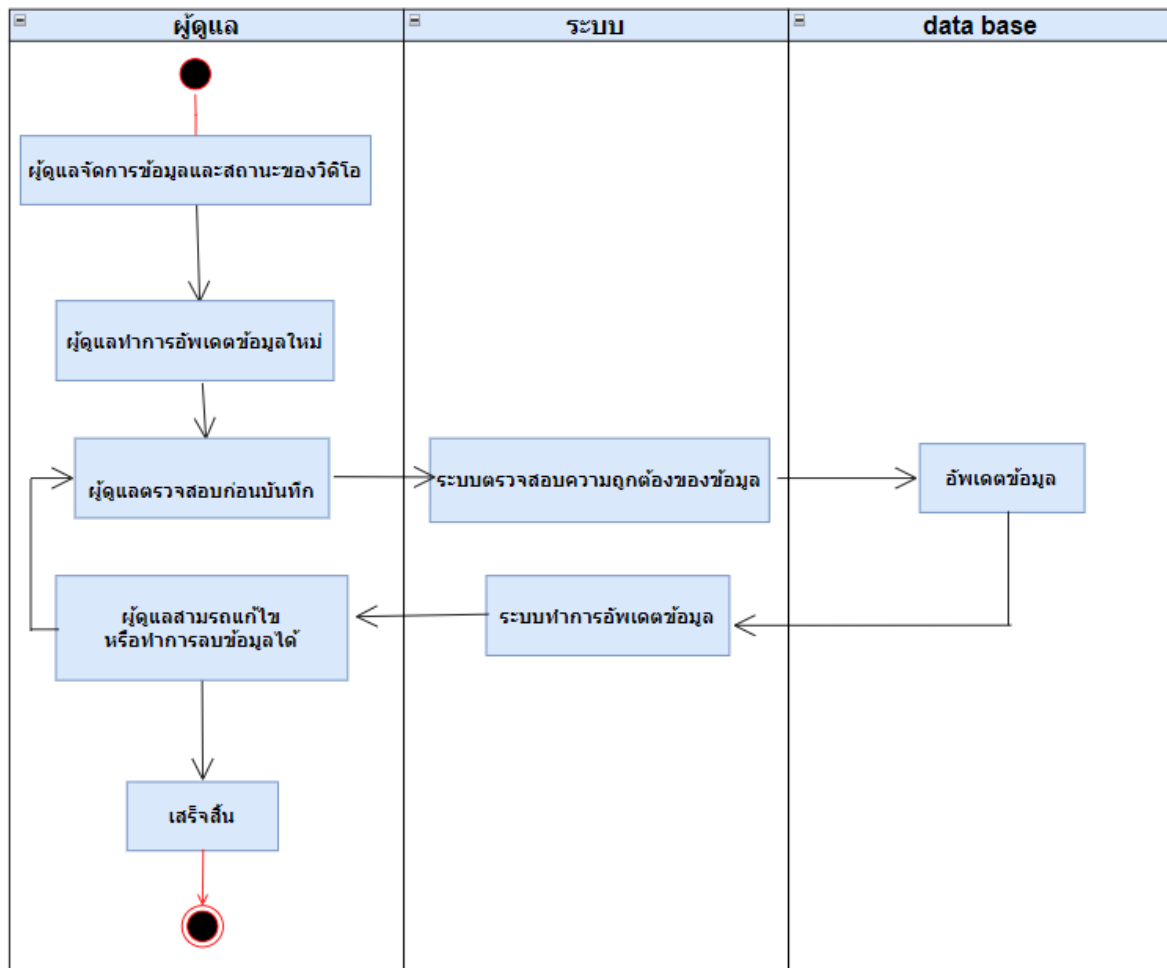


## 1.4 ACTIVITY DIAGRAM

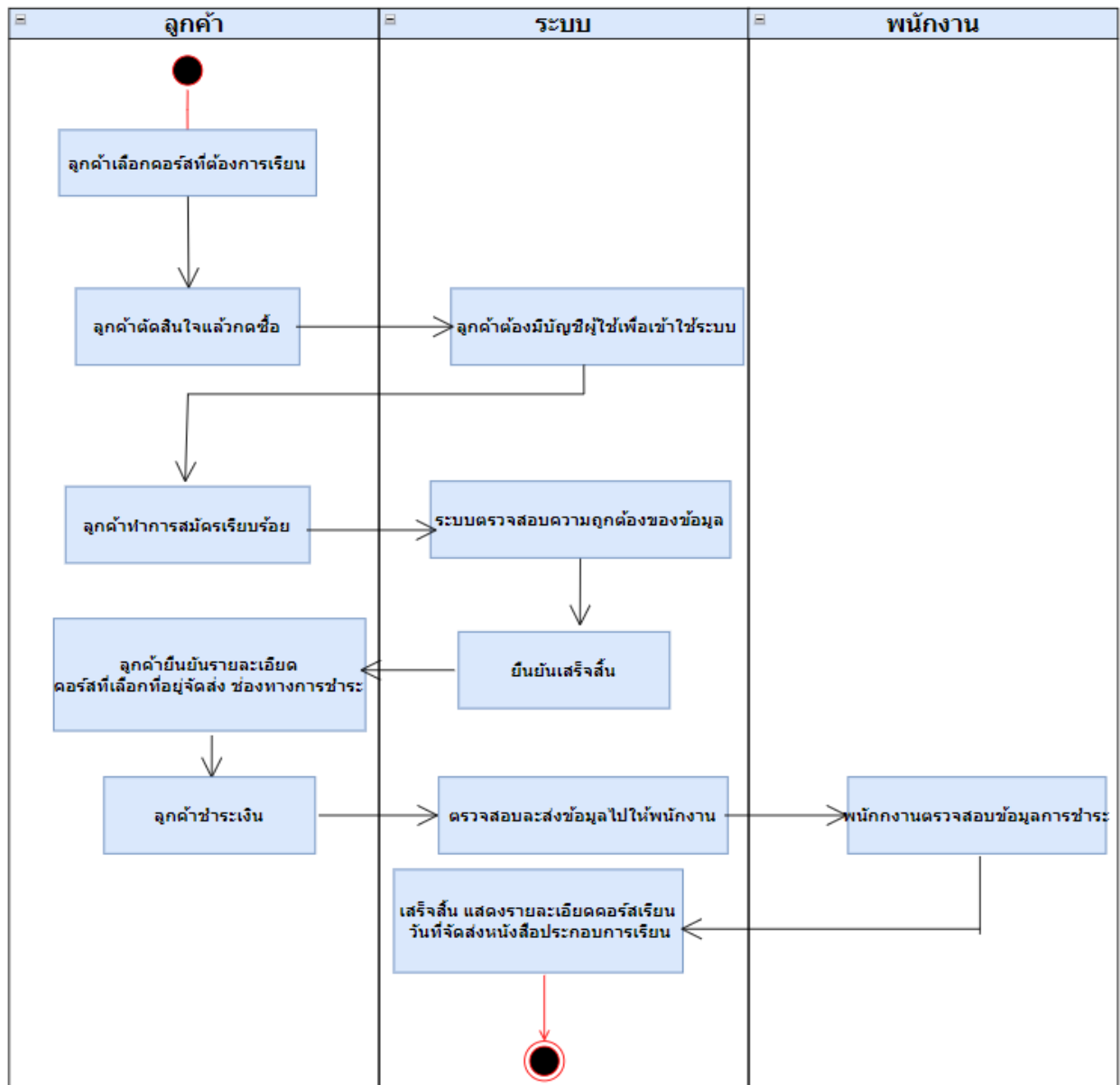
- Activity Diagram ของ UC-01



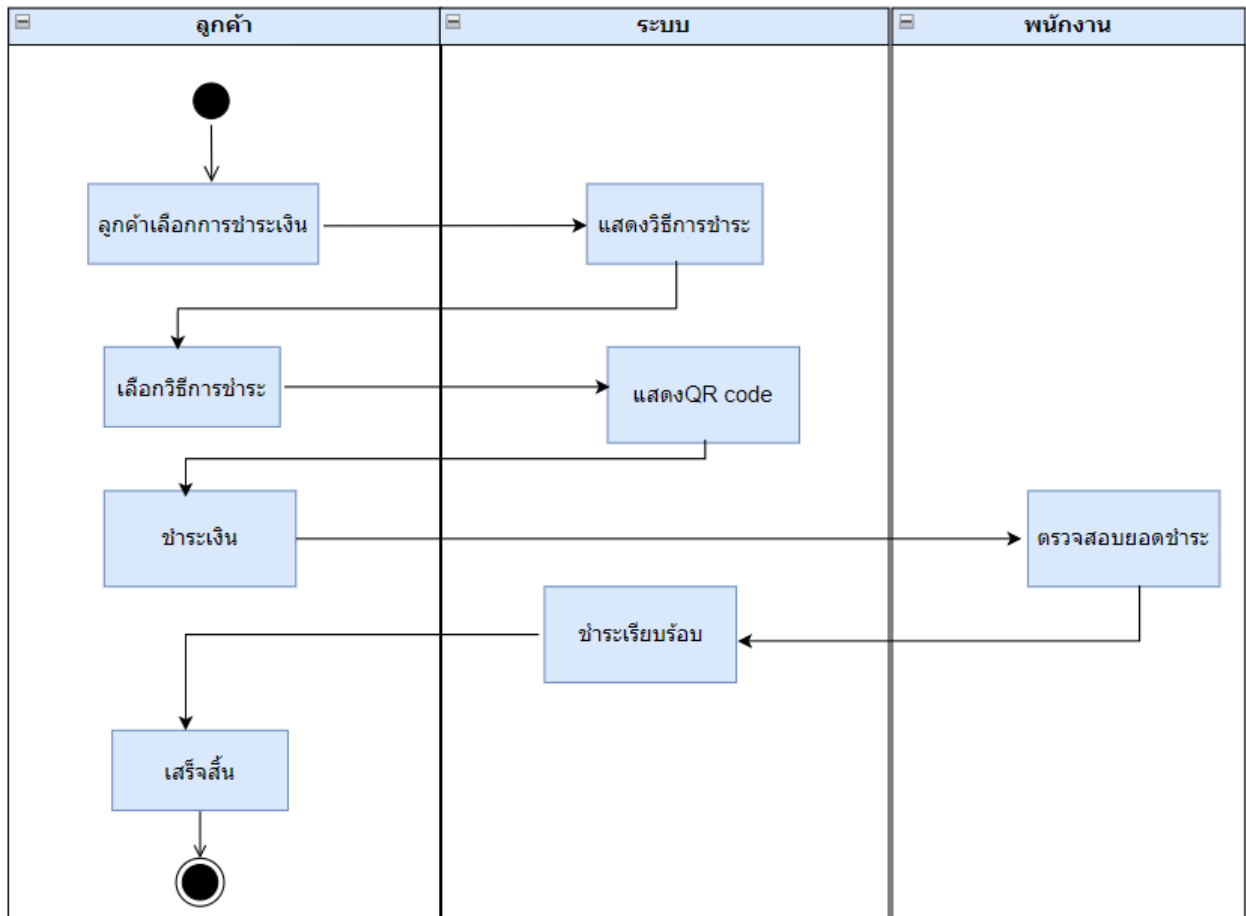
## - Activity Diagram ของ UC-02



## - Activity Diagram ของ UC-03

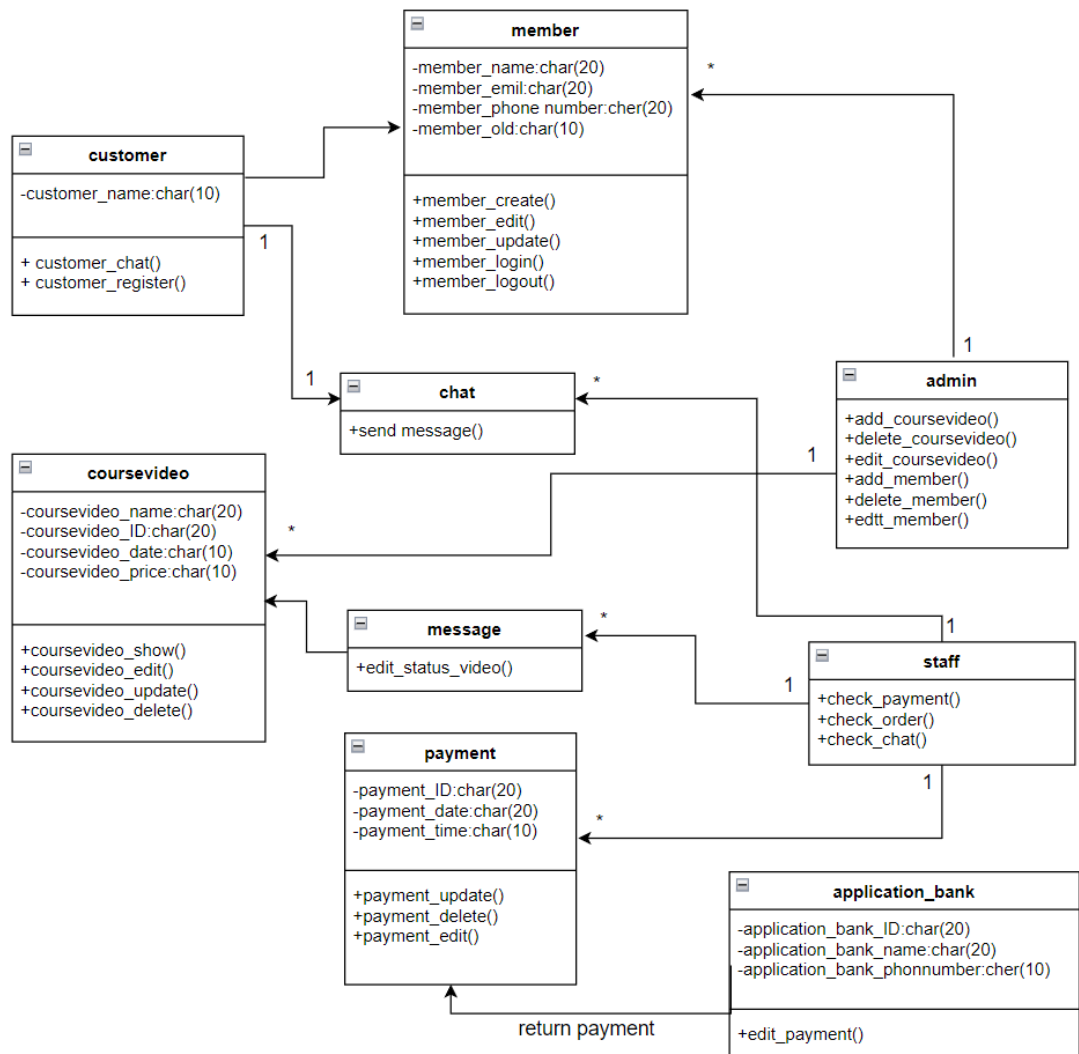


## - Activity Diagram ของ UC-04



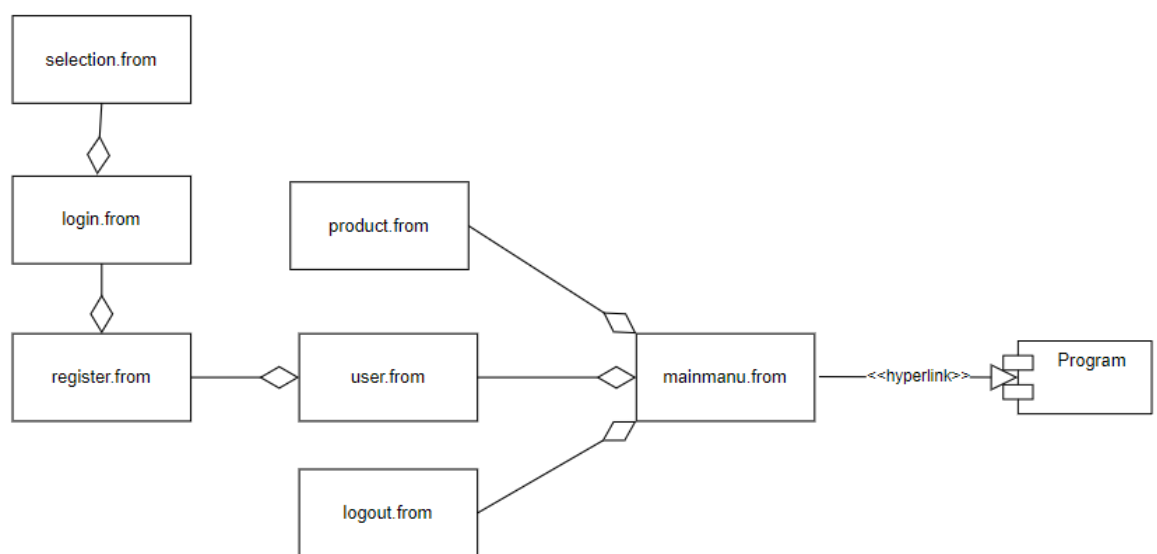
## 3.2 การออกแบบระบบ

### 3.2.1 CLASS DIAGRAM



## 3.2.2 Presentation Logic Subsystem

แสดง User Interface เป็น Form ย่อยจำนวน 3 Form ทั้งหมดอยู่ใน Mainmenu.form



- mainmanu Form เป็น Form ติดต่อกับผู้ใช้งานระบบ มี Form ย่อยได้แก่

- product.form เป็น User Interface

ในส่วนของการแสดงข้อมูลรายการละเอียดต่างๆของคอร์สเรียน

- User.Form เป็น การจัดการข้อมูลผู้ใช้เป็น User Interface ในส่วนของการจัดการข้อมูลของ User ที่ได้ทำการกรอกข้อมูลลงทะเบียนไว้

- Logout.form เป็น การออกจากระบบของการใช้งาน



### 3.2.3 Working Logic Subsystem

แสดง โครงสร้างการทำงานย่อยต่างๆ เมื่อถูก Mainmenu เรียกใช้งาน - ระบบจัดการข้อมูลพื้นฐาน (Basic.vb) เป็นโปรแกรมสำหรับ User Interface ในส่วนของการจัดการ อีก 3 ระบบย่อย

- ระบบจัดการข้อมูลวิดีโอ ( product.vb)

เป็นโปรแกรมสำหรับ User Interface ในส่วนของการจัดการ อัปเดตวิดีโอการสอน หรือปรับเปลี่ยนสถานะของวิดีโอ

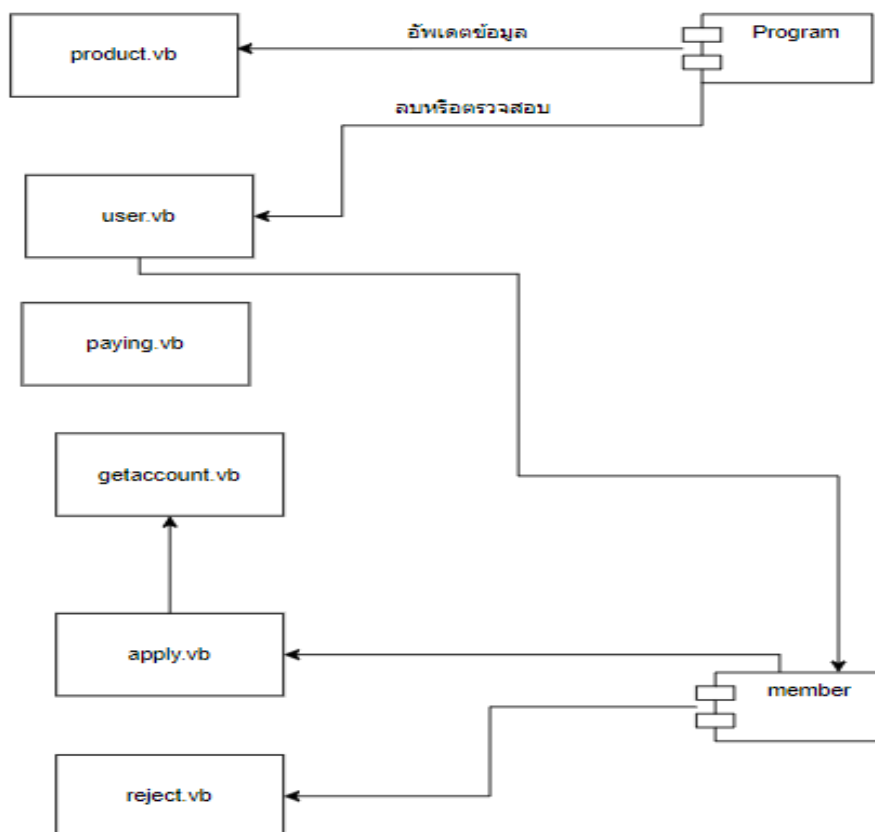
- ระบบจัดการการบันทึกผู้ใช้ (User.vb)

เป็นโปรแกรมสำหรับ User Interface ในส่วนของการจัดการ

ลบหรือตรวจสอบบันทึกข้อมูลผู้ใช้ได้

- ระบบออกสู่ระบบ (Logout.vb)

เป็นส่วน User Interface สำหรับออกจากระบบ

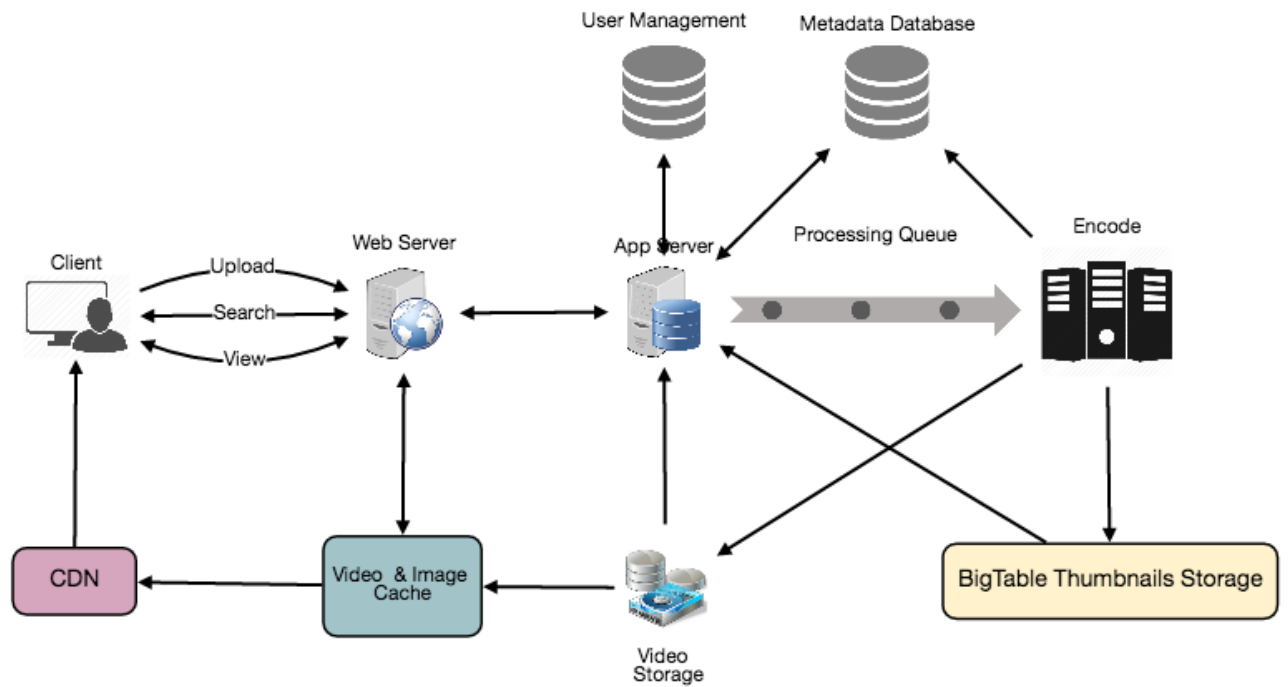


### 3.2.4 Database Logic Subsystem

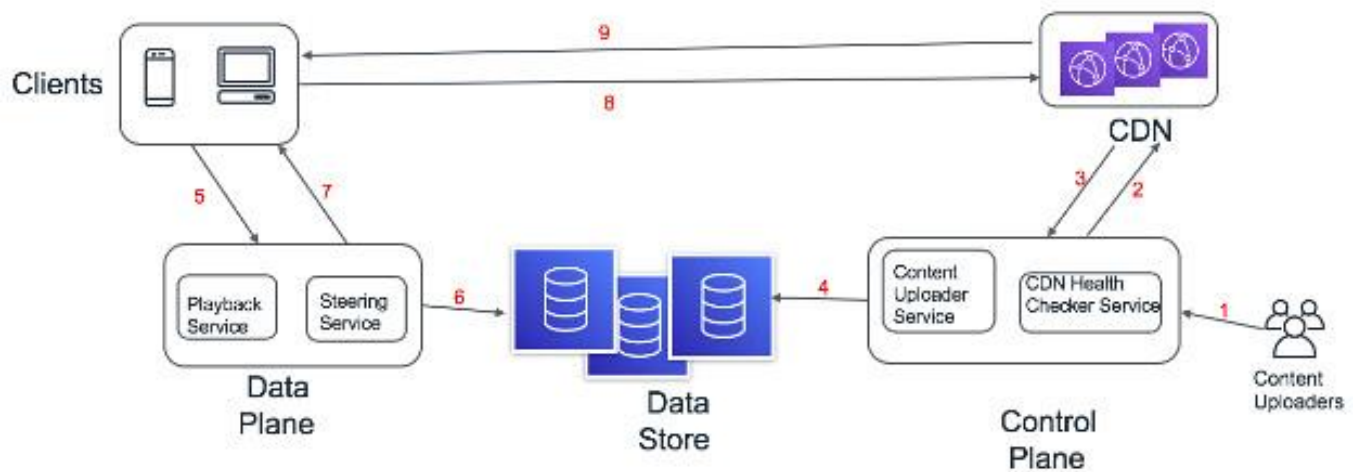
Data	Attribute	Type(size)	Discription
<b>member</b>	-member_name	char(20)	ชื่อผู้ใชั
	-member_email	char(20)	ที่อยู่ที่สามารถติดต่อได้
	-member_phone number	cher(20)	หมายเลขโทรศัพท์มือถือ
	-member_old	char(10)	อายุ
<b>customer</b>	-customer_name	char(10)	ชื่อสมมุติก่อนที่ยังไม่ได้ลงทะเบียน
<b>coursevideo</b>	-coursevideo_name	char(20)	ชื่อวิชา
	-coursevideo_ID	char(20)	หลายเลขวิดีโอ
	-coursevideo_date	char(10)	วันที่อัปเดตวิดีโอ
	-coursevideo_price	char(10)	ราคา

<b>payment</b>	-payment_ID	Char(20)	เลขที่การจ่ายเงิน
	-payment_date	char(20)	วันที่จ่าย
	-payment_time	char(10)	เวลาที่จ่าย

### 3.2.5 System Architecture Design



### 3.2.6 Persistent Data Design



## 4.สรุป

การพัฒนาระบบเว็บไซต์ขายคอร์สออนไลน์ ใน Project นี้พัฒนาถึงขั้นตอนออกแบบ ( Design ) ซึ่งขั้นตอนที่จะพัฒนา ต่อไปคือ Implementation และ Operation

จึงจะสมบูรณ์และสามารถแก้ปัญหาของระบบ

เดิมที่ต้องมีหน้าร้านลดค่าใช้จ่ายได้มากด้วยงบประมาณเจ็ดแสนบาทระบบใหม่มีประโยชน์มากในด้านการเข้าถึงของผู้ใช้ง่ายขึ้น จากทรัพยากรของระบบใหม่

องค์กรสามารถพัฒนาระบบเว็บไซต์ขายคอร์สออนไลน์ ต่อไปในอนาคตได้

บรรณานุกรม

<https://slideplayer.in.th/slide/17710447/>

<https://app.diagrams.net/>