

Proposal

การวิเคราะห์ออกแบบระบบแอพพลิเคชันขายหนังสืออิเล็กทรอนิกส์ออนไลน์

ของบริษัท ลาเต้ อีบุ๊กส์โตร์ จำกัด

จัดทำโดย

นาย ศิชา ダメะลี

รหัสนักศึกษา 6505001948

เสนอ

ผ.ศ. พรพรรณ ประชาพิพัฒ

รายงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาวิชา COS3108

การวิเคราะห์ และออกแบบระบบ (System Analysis and Design)

ภาคเรียนที่ 1 ปีการศึกษา 2567

ภาควิชาการคอมพิวเตอร์

คณะวิทยาศาสตร์ มหาวิทยาลัยรามคำแหง

คำนำ

รายงานฉบับนี้จัดทำขึ้นเพื่อวิเคราะห์และออกแบบระบบแอพพลิเคชันการขายหนังสืออิเล็กทรอนิกส์ประจำทอนิเมะออนไลน์ของบริษัท ลาเต้ อีบุ๊คส์โตร์ จำกัด ซึ่งเป็นบริษัทที่มีความเชี่ยวชาญในการจำหน่ายหนังสือออนไลน์หลายแพลตฟอร์ม ทางออนไลน์ โดยมุ่งหวังที่จะพัฒนาระบบทั้งด้านระบบให้ตอบสนองต่อพฤติกรรมการใช้บริการและความต้องการของลูกค้าที่มีการเปลี่ยนแปลงตามยุคดิจิทัล

สารบัญ

1. บทนำ	
1.1 ความสำคัญและขอบเขตของปัญหา	6
1.2 วัตถุประสงค์ของระบบใหม่	6
1.3 ประโยชน์ที่คาดว่าจะได้รับ	7
1.4 ขอบเขตการดำเนินงานโครงการ	7
1.5 นิยามศัพท์	9
1.6 ระเบียบวิธีการดำเนินงาน	10
1.7 ทรัพยากร หรือ เครื่องมือในการวิจัยหรือพัฒนาระบบ	12
1.8 แผนการดำเนินการ	13
1.9 งบประมาณ	15
2. วรรณกรรมที่เกี่ยวข้อง	
2.1 ทฤษฎีที่เกี่ยวข้อง	16
2.1.1 วัจกรภาพพัฒนาระบบ	16
2.1.2 ระบบฐานข้อมูล (Database System)	19
2.1.3 ทฤษฎีการสร้าง Diagram	20
ตัวอย่าง Activity diagram	
ตัวอย่าง sequence diagram	
3. ขั้นตอนการดำเนินงาน	
3.1 การวิเคราะห์ระบบ	33
3.1.1 Use Case Diagram	33
3.1.2 Use Case Description	44
3.1.3 Sequence Diagram	66
3.1.4 Activity Diagram	79

3.2 การออกแบบระบบ	92
3.2.1 Class Diagram	92
3.2.2 Presentation Logic Subsystem	93
3.2.3 Working Logic Subsystem	95
3.2.4 Database Logic Subsystem	97
3.2.5 System Architecture Design	98
3.2.6 Persistent Data Design	99
3.2.7 Screen Form Design	
104	

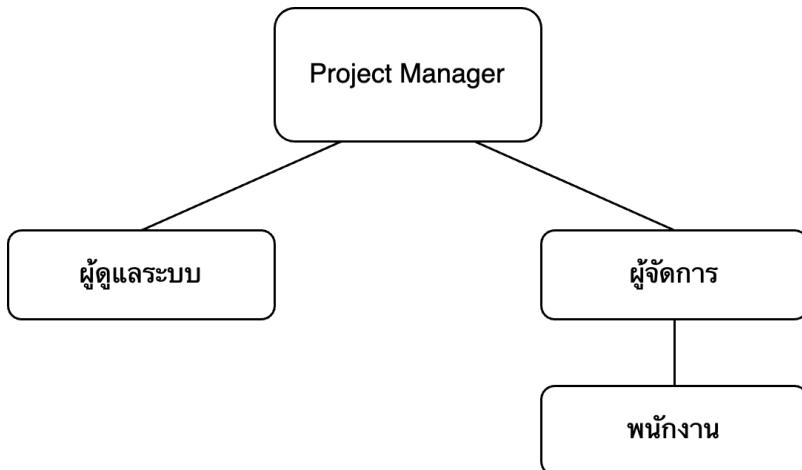
ข้อสรุปและข้อเสนอแนะ

บรรณานุกรม

1. บทนำ

โครงการ การวิเคราะห์ออกแบบระบบแอพพลิเคชันขายหนังสืออิเล็กทรอนิกส์ออนไลน์ ของบริษัท จ่ายแล้วบิด จำกัด

บริษัท ลาเต้ บุ๊กสโตร์ เป็นบริษัทขายหนังสือออนไลน์ ที่ขายหนังสือหลากหลายประเภท
สามารถสั่งซื้อผ่านเว็บไซต์ อย่างไรก็ตาม, ในช่วงไม่กี่ปีที่ผ่านมา บริษัทเริ่มแข่งขันกับปัญหาหลายประการที่ส่งผลกระทบ
ต่อการดำเนินงานของบริษัท การเปลี่ยนแปลงพฤติกรรมของลูกค้า ลูกค้ากำลังเปลี่ยนแปลงความต้องการไป
ยังหนังสืออิเล็กทรอนิกส์ (E-books) ซึ่งสะดวกและเข้าถึงได้ง่ายกว่า, ต้นทุนการดำเนินงานสูง การจัดการสต็อก
หนังสือ, และค่าจ้างพนักงานสูง ส่งผลให้บริษัทมีต้นทุนการดำเนินงานที่สูงขึ้น, ปัญหาด้านโลจิสติกส์ ค่าใช้จ่ายสูง
ในการเช่าพื้นที่ และการบำรุงรักษาคลังสินค้า เนื่องจากการเปลี่ยนแปลงพฤติกรรมลูกค้า, ต้นทุนการดำเนินงานที่
สูง, และปัญหาด้านโลจิสติกส์ บริษัท ลาเต้ บุ๊กสโตร์ จึงตัดสินใจเปลี่ยนเข้าสู่ตลาดอีบุ๊กเพื่อรับมือกับปัญหาดังกล่าว
ลดต้นทุน และตอบสนองความต้องการของลูกค้าในยุคดิจิทัล



โครงการ การวิเคราะห์ออกแบบระบบแอพพลิเคชันขายหนังสืออิเล็กทรอนิกส์(E-book) ออนไลน์
ที่นำเสนอของผลประกอบด้วยขั้นตอน 3 ขั้นตอนของการพัฒนาระบบ ซึ่งทีมงาน Developer สามารถฯ ผล
ของการออกแบบระบบทั้งหมดของโครงการนี้ ไปพัฒนาต่อในขั้นตอนการติดตั้ง (Implementation Phase) และ

ขั้นตอนปฏิบัติงานจริง (Operation and Maintenance Phase) ตามลำดับ เพื่อให้การพัฒนาระบบที่มีแล้ว เสิร์ฟสมบูรณ์ ถูกต้อง สามารถแก้ปัญหาของระบบปัจจุบัน เป็นไปตามวัตถุประสงค์ของระบบใหม่ และเกิด ประโยชน์ต่อไปในอนาคต

1.1 ความสำคัญและขอบเขตของปัญหา

1. ลูกค้าไม่สามารถทดลองอ่านสินค้าได้ จะตัดสินใจซื้อสินค้าได้
2. พนักงานส่งสินค้า
3. ไม่สามารถตรวจสอบการชำระเงินจากลูกค้าได้ทันที
4. ไม่สามารถชำระเงินได้หลายช่องทาง
5. ไม่สามารถเข้าสู่ระบบได้หลากหลายช่องทาง เช่น Google, Apple
6. ไม่สามารถแข่งขันกับบริษัทคู่แข่งได้

1.2 วัตถุประสงค์ของระบบใหม่

1. ลูกค้าทดลองอ่านก่อนที่จะซื้อสินค้าได้
2. ลูกค้าสามารถชำระเงินได้หลายช่องทาง
3. ลูกค้าสามารถตรวจสอบสถานะการชำระเงินได้
4. พนักงานสามารถตรวจสอบการชำระเงินของลูกค้าได้ถูกต้อง รวดเร็ว ทันที
5. สามารถอ่านหนังสืออิเล็กทรอนิกส์ได้ตลอดเวลา
6. สามารถดาวน์โหลดมาอ่านในโหมดออฟไลน์ได้
7. สมัครสมาชิกได้หลายช่องทาง เช่น Google, Apple

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. สะดวกต่อการสืบค้นหนังสือและเข้าใจเนื้อหาได้เร็ว
2. ลดปัญหาการจัดเก็บหนังสือจากที่เป็นชั้นหนังสือ
3. สามารถอ่านหนังสือผ่านระบบออนไลน์และออฟไลน์ได้
4. ช่วยให้นักเขียนสามารถเผยแพร่ผลงานได้อย่างรวดเร็ว

1.4 ขอบเขตการดำเนินงานโครงการ

1. ผู้ดูแลระบบ
 - สามารถเพิ่ม แก้ไข ลบ สิทธิการเข้าใช้ระบบของลูกค้า, พนักงานทุกคน
 - กำหนด Data control, System control, System security
 - สามารถเรียกดูข้อมูล เพิ่ม ปรับปรุง ข้อมูลการติดต่อระหว่างพนักงานกับลูกค้าได้
2. ผู้จัดการ
 - สามารถเรียกดู เพิ่ม แก้ไข ลบ ข้อมูล หนังสือ (สามารถเรียกดูได้ข้อมูลทุกพิล๊ด)
 - สามารถเรียกดู ข้อมูลลูกค้า เนพาะ อีเมล ชื่อ นามสกุล เบอร์โทรศัพท์
 - สามารถเรียกดู ข้อมูลในใบคำสั่งซื้อ ได้แก่ รายการสินค้าใบสั่งซื้อ, สถานะการชำระเงิน, สถานะการซื้อเสร็จสมบูรณ์
 - สามารถปรับปรุงสถานะการชำระเงินของใบสั่งซื้อ นั่นๆ ที่ลูกค้าชำระเงินแล้ว
 - สามารถยกเลิกใบคำสั่งซื้อของลูกค้าได้
 - สามารถเพิ่มโปรโมชั่นการสั่งซื้อหนังสือได้
 - สามารถดูเงินรายรับ รายวัน/รายเดือน/รายปี
3. พนักงาน
 - สามารถเรียกดู หนังสือ เนพาะ ชื่อ, คำอธิบาย, วันเผยแพร่, isbn, ราคา, ผู้แต่ง, สำนักพิมพ์, ประเภท
 - สามารถเพิ่ม หนังสือ
 - สามารถเรียกดู ข้อมูลลูกค้า เนพาะ อีเมล ชื่อ นามสกุล เบอร์โทรศัพท์

- สามารถเรียกดู ข้อมูลในใบคำสั่งซื้อ ได้แก่ รายการสินค้าใบสั่งซื้อ, สถานะการชำระเงิน, สถานะการซื้อเสร็จสมบูรณ์

4. ลูกค้า

- สามารถเรียกดู แก้ไข ข้อมูลบัญชี เฉพาะ ซื้อ, นามสกุล, เบอร์โทรศัพท์, รหัสผ่าน
- สามารถเรียกดู ตัวอย่างหนังสือได้
 - สามารถเรียกดู หนังสือ เฉพาะ ซื้อ, คำอธิบาย, วันเผยแพร่, isbn, ราคา, ผู้แต่ง, สำนักพิมพ์, ประเภท
 - สามารถสั่งซื้อหนังสือได้
 - สามารถเรียกดู ข้อมูลในใบสั่งซื้อ, ข้อมูลรายการสินค้าใบสั่งซื้อ, สถานะการชำระเงิน, , สถานะการสั่งซื้อสมบูรณ์
 - สามารถเพิ่ม แก้ไข ลบ ข้อมูลใบสั่งซื้อ, ข้อมูลรายการสินค้าใบสั่งซื้อ ก่อนทำการชำระเงิน โดยต้องชำระเงินภายใน 1 วัน หลังจากยืนยันใบคำสั่งซื้อนั้นๆ และ หากพ้นกำหนดระบบจะทำการยกเลิกใบสั่งซื้อนั้นๆ
 - ลูกค้าไม่สามารถจับภาพหน้าจอ หรือ บันทึกวิดีโองานน้ำใจได้ระหว่างอ่านหนังสืออิเล็กทรอนิกส์
 - สามารถดูประโนมัชในการสั่งซื้อหนังสือได้

1.5 นิยามศัพท์

1.5.1 Proposal คือ แผนงาน แผนโครงการ หรือข้อเสนอที่ยื่นเพื่อขออนุมัติ ซึ่งแผนนี้จะได้รับการอนุมัติหรือไม่ก็ได้

1.5.2 Gantt Chart คือ แผนผังคุณกำหนดงาน มักใช้ในด้านการจัดการโครงการต่าง ๆ

1.5.3 Use Case Diagram คือ แผนภาพที่แสดงการทำงานของผู้ใช้ระบบ (User) และความสัมพันธ์ กับระบบย่อย (Sub systems) ภายในระบบใหญ่

1.5.4 Use Case Description คือ รายละเอียดของแต่ละแผนภาพ ที่แสดงการทำงานของผู้ใช้ ระบบ (User) และความสัมพันธ์กับระบบย่อย (Sub systems) ภายในระบบใหญ่

1.5.5 Sequence Diagram คือ Diagram ที่ประกอบไปด้วย Class หรือ Object เส้นที่ใช้เพื่อแสดงลำดับเวลา และเส้นที่ใช้เพื่อแสดงกิจกรรมที่เกิดขึ้นจาก Object หรือ Class ใน Diagram

1.5.6 Activity Diagram คือ แผนภาพที่ใช้ที่แสดงขั้นตอนการเกิดกิจกรรมต่างๆของ Use Case Diagram

1.5.7 Class Diagram คือ แผนภาพที่ใช้แสดง Class และความสัมพันธ์ในแบบต่างๆ (Relation) ระหว่าง Class เหล่านั้น

1.5.8 Architecture Design คือ กระบวนการกำหนดสถาปัตยกรรม ส่วนประกอบ ส่วนประสาน และลักษณะด้านอื่นๆ ของระบบ หรือสรุปประกลับของระบบโดยการออกแบบซอฟต์แวร์ ยังมี ความหมายรวมถึงสิ่งที่ได้จากการออกแบบ

1.5.9 Presentation Logic Subsystem คือ ระบบรองที่เกี่ยวข้องกับระบบย่อยที่ติดต่อกับ ผู้ใช้งาน ระบบโดยตรง

1.5.10 Working Logic Subsystem คือ ส่วนของระบบที่เกี่ยวข้องกับการทำงานที่เกิดขึ้นจริงๆใน CPU ของเครื่องคอมพิวเตอร์

1.5.11 Screen Form Design คือ การออกแบบหน้าจอ

1.6 ระเบียบวิธีการดำเนินงาน

ใช้วิธี System Development Life Cycle มี 5 Phase คือ

Phase ที่	วัตถุประสงค์	กิจกรรม	สิ่งที่ได้
1	Planning	<ul style="list-style-type: none">- Define Problem online system- Produce project schedule- Confirm project feasibility- Staff the project online system- Launch the project online system	<ul style="list-style-type: none">- Proposal- Gantt chart
2	Analysis	<ul style="list-style-type: none">- Gather information to learn problem domain- Define system requirements- 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	<ul style="list-style-type: none">- Use Case Diagram- Use Case Description- Sequence Diagram

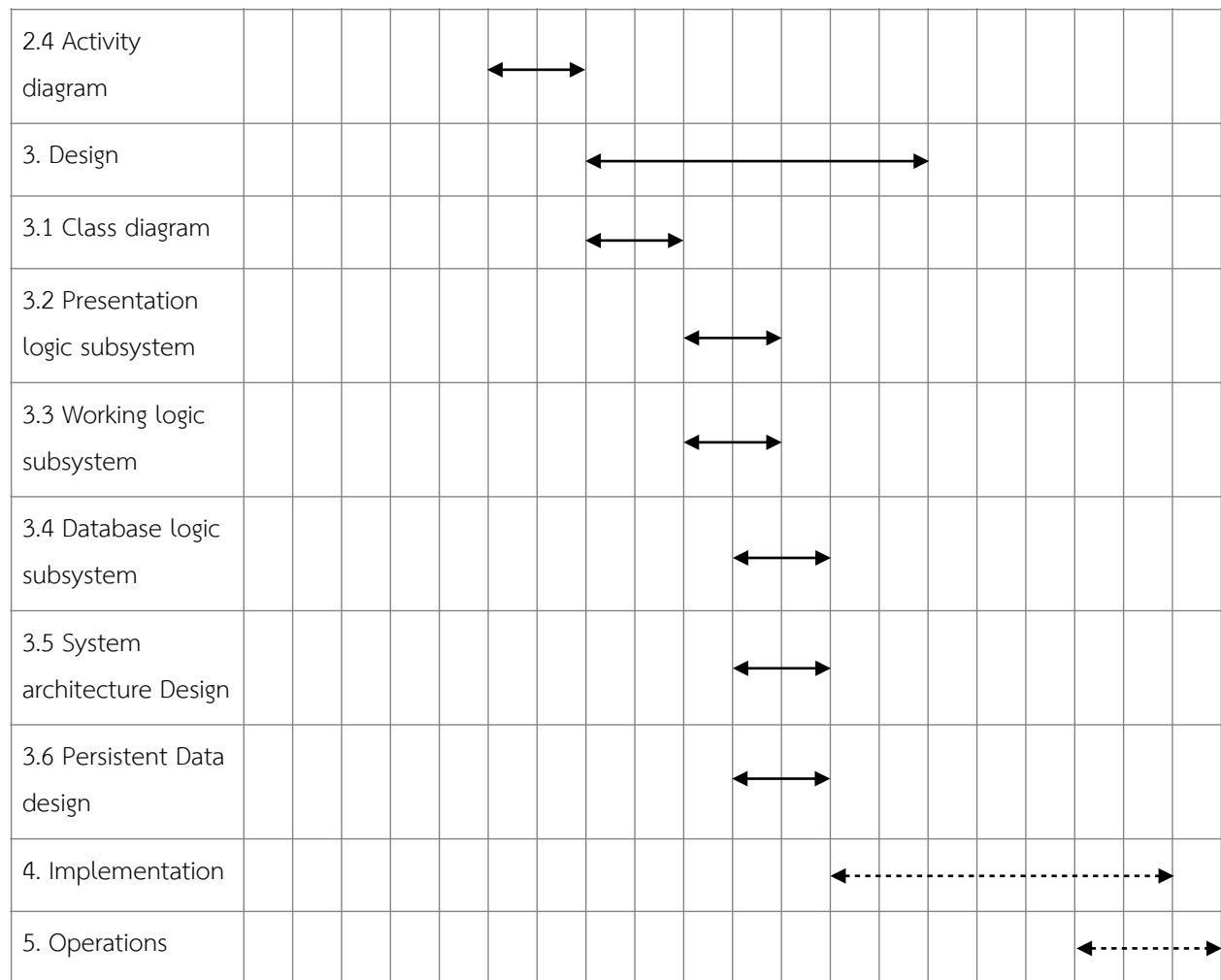
3	Design	<ul style="list-style-type: none"> - Design and integrate the network, the application architecture, User interfaces, System interfaces, database, Design details, System controls 	<ul style="list-style-type: none"> - Class Diagram - Architecture Diagram (Deployment Diagram) - Components Diagram
4	Implementation	<ul style="list-style-type: none"> - Construct software components - Verify and test - Convert data - Train users and document the system - Install the system 	<ul style="list-style-type: none"> - Source Program - Testing, Unit test - Error Verification Report Of Application Program - Documents - Training
5	Operations and Support	<ul style="list-style-type: none"> - Maintain system Anime E-Book store - Enhance system: <ul style="list-style-type: none"> a) Small upgrades or enhancements to expand system capabilities b) Larger enhancements may require separate development project - Support users : Help desk, support team 	<ul style="list-style-type: none"> - New System - Operations - Evaluation - Maintenance

1.7 ทรัพยากร หรือ เครื่องมือในการวิจัยหรือพัฒนาระบบ

การวิจัยและพัฒนาระบบนี้ใช้เครื่องมือและทรัพยากรหลายประเภทเพื่อให้กระบวนการออกแบบและพัฒนาเป็นไปอย่างมีประสิทธิภาพและตอบโจทย์การดำเนินงาน ดังนี้

1. **เครื่องมือวิเคราะห์ระบบ:** ใช้สำหรับการวิเคราะห์โครงสร้างและความต้องการของระบบ เช่น เครื่องมือสร้าง Use Case Diagram, Activity Diagram และ Sequence Diagram เพื่อเข้าใจขั้นตอนการทำงานของระบบ
2. **เครื่องมือออกแบบระบบ:** เช่น [draw.io](#), Canva, Lucidchart สำหรับการออกแบบ Class Diagram, System Architecture Design และ Database Logic Subsystem
3. **เครื่องมือพัฒนาซอฟต์แวร์:** ใช้สำหรับเขียนโปรแกรม เช่น Visual Studio Code รวมถึงเครื่องมือช่วยในการจัดการโค้ด เช่น Git จัดการเรื่อง Version Control
4. **ฐานข้อมูล:** เลือกใช้ฐานข้อมูลที่เหมาะสมกับระบบ เช่น MySQL, PostgreSQL หรือ MongoDB เพื่อจัดเก็บข้อมูลลูกค้า สินค้า และรายการสั่งซื้อ
5. **เซิร์ฟเวอร์และบริการไฮสตั๊ด:** ใช้ VM Server หรือ Cloud Server เช่น AWS, Google Cloud หรือ Microsoft Azure สำหรับการทดสอบและรันระบบในสภาพแวดล้อมจริง
6. **เครื่องมือทดสอบระบบ:** ใช้เครื่องมือ Automated Testing อย่าง Selenium หรือ Playwright เพื่อทดสอบการทำงานของระบบ รวมถึงการทดสอบความปลอดภัยของระบบ
7. **เทคโนโลยีคอนเทนเนอร์ (Container Technology):** เพื่อช่วยในการจัดการและปรับแต่งสภาพแวดล้อมของแอปพลิเคชันให้สามารถพัฒนาและทดสอบได้ง่ายขึ้น โดยใช้ Docker หรือ Kubernetes เพื่อจัดการกับคอนเทนเนอร์ ทำให้การปรับใช้และทดสอบแอปพลิเคชันบนเครื่องที่แตกต่างกันเป็นไปอย่างราบรื่น ลดปัญหาที่เกิดจากความไม่เข้ากันของสภาพแวดล้อมการพัฒนาและการใช้งานจริง

1.8 แผนการดำเนินการ



1.9 งบประมาณ

ลำดับ	รายการ	
1	ค่า Development	200,000 บาท
2	ค่าจด Domain Name	1,000 บาท/ต่อปี
3	ค่าบริการ Apple App Store	3,500 บาท/ต่อปี
4	ค่าบริการ Google Play Store	1,000 บาท
5	ค่าบริการ VM Server (vCPU 8 Core, RAM 16GB, Storage 100GB)	72,000 บาท/ต่อปี
6	Document + Training	8,500 บาท
7	อื่น ๆ	14,000 บาท
	รวมเงินลงทุน (Investment Cost)	300,000 บาท

ค่าบำรุงรักษา (ต่อปี)

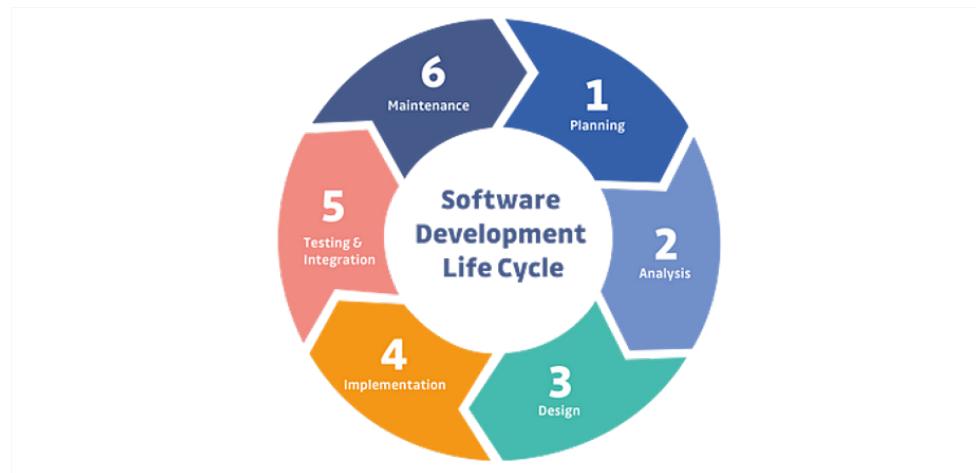
ปีที่	
1	ฟรี + ระบบประกัน
2	100,000 บาท
3	120,000 บาท
4	170,000 บาท
5	200,000 บาท

2. วรรณกรรมที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 วงจรการพัฒนาระบบ

Software Development Lifecycle (วงจรการพัฒนาซอฟต์แวร์หรือ SDLC) คือกระบวนการที่คุ้มค่าและประหยัดเวลาในการออกแบบและสร้างซอฟต์แวร์คุณภาพสูงสำหรับทีมพัฒนา เป้าหมายของ SDLC คือการลดความเสี่ยงของโครงการด้วยการวางแผนล่วงหน้า เพื่อให้ซอฟต์แวร์ตอบสนองตามความคาดหวังของลูกค้าในระหว่างการใช้งานจริงและหลังจากนั้น ระเบียบวิธีการนี้จะแสดงชุดขั้นตอนที่แบ่งกระบวนการพัฒนาซอฟต์แวร์ออกเป็นงานที่คุณสามารถมองเห็นได้



ระยะของการพัฒนาระบบ โดยทั่วไป จะประกอบไปด้วย

ระยะที่ 1 : การวางแผนโครงการ (Project Planning Phase)

ระยะที่ 2 : การวิเคราะห์ (Analysis Phase)

ระยะที่ 3 : การออกแบบ (Design Phase)

ระยะที่ 4 : การนำไปใช้ (Implementation Phase)

ระยะที่ 5 : การทดสอบการรวมระบบ (Testing and Integration Phase)

ระยะที่ 6 : การบำรุงรักษา (Maintenance Phase)

ระยะที่ 1 : การวางแผนโครงการ (Project Planning Phase)

เป็นขั้นตอนการรวบรวม และวิเคราะห์ความต้องการจากผู้ที่มีส่วนได้ส่วนเสีย เพื่อรับ requirement ของลูกค้าและกำหนดขอบเขตของโครงการ ทีมนักพัฒนาจะต้องสื่อสารกับลูกค้าและเก็บข้อมูลที่เกี่ยวข้องทั้งหมด เพื่อทำความเข้าใจวัตถุประสงค์ทางธุรกิจ ฟังก์ชันใช้งานที่ต้องการ และข้อจำกัดของระบบ รวมถึงรายละเอียดงบประมาณกับระยะเวลาของโครงการ เพื่อจัดทำเอกสารข้อกำหนดความต้องการ หรือ Software Requirements Specification (SRS) สำหรับใช้เป็นข้อตกลงร่วมกันระหว่างผู้พัฒนาซอฟต์แวร์กับผู้ว่าจ้าง

ระยะที่ 2 : การวิเคราะห์ระบบ (Analysis Phase)

เป็นขั้นตอนวิเคราะห์ความต้องการ (Requirement) ทั้งในส่วนของกรอบเวลาของโครงการ และ Feature ที่ต้องการว่ามีอะไรบ้าง เพื่อจัดลำดับความสำคัญของงาน กำหนดรายละเอียดของความต้องการ และวางแผนการทดสอบต่างๆ ว่าต้องทำอะไรบ้าง

ระยะที่ 3 : การออกแบบ (Design Phase)

เป็นการตัดสินใจว่าระบบจะดำเนินการอย่างไร

- กลยุทธ์การจัดหาระบบ เพื่อได้ความชัดเจนเกี่ยวกับแนวทางการพัฒนาระบบ ว่าจะพัฒนาเอง หรือ ซื้อซอฟต์แวร์มาใช้ หรือจ้างหน่วยงานภายนอกมาพัฒนาระบบ
- ออกแบบสถาปัตยกรรมของระบบ ที่อธิบายถึง ฮาร์ดแวร์ ซอฟต์แวร์และโครงสร้าง พื้นฐาน เช่น ด้านเครือข่าย จะมีการเพิ่มเติมหรือปรับปรุงจากโครงสร้างพื้นฐานเดิมที่มีอยู่
- การออกแบบอินเตอร์เฟซ UX/UI เกี่ยวข้องกับการปฏิสัมพันธ์ระหว่างผู้ใช้กับระบบ รวมถึง องค์ประกอบและรายการต่างๆ
- ออกแบบฐานข้อมูล เพื่อให้ทราบว่าระบบต้องมีข้อมูลอะไรบ้าง ที่ต้องจัดเก็บในฐานข้อมูล
- การออกแบบโปรแกรม เพื่อนำไปสู่การเขียนโปรแกรมด้วยคอมพิวเตอร์ในระยะการนำไปใช้งานต่อไป

ระยะที่ 4 : การทำไปใช้ (Implement Phase)

เริ่มต้นพัฒนาระบบ เขียนโปรแกรมจากแผนที่ได้จากขั้นตอน Design โดยระหว่างการพัฒนาระบบในขั้นตอนนี้ของทีม Software Developer หรือ Programmer จะมีการใช้เครื่องมือต่าง ๆ เข้ามาช่วยให้ทำพัฒนาระบบซอฟต์แวร์ ไม่ว่าจะเป็น Version control หรือการทำ Code Review ที่มีส่วนช่วยให้ทีม Software Developer ทำงานร่วมกันมีประสิทธิภาพมากขึ้น การพัฒนาซอฟต์แวร์เป็นขั้นตอนที่กำลังสำคัญใน SDLC เนื่องจากมันเป็นกระบวนการสร้างซอฟต์แวร์จริง ๆ ที่ตอบสนองต่อความต้องการและการออกแบบของโครงการ การพัฒนาซอฟต์แวร์ที่มีคุณภาพและมีประสิทธิภาพมีบทบาทสำคัญในการสร้างผลลัพธ์ที่เป็นที่พึงพอใจของผู้ใช้งาน

ระยะที่ 5 : การทดสอบการรวมระบบ (Testing and Integration Phase)

เมื่อพัฒนาเสร็จก็จะทำการทดสอบระบบ เพื่อตรวจสอบความผิดพลาด หรือ ความเสียที่สามารถเกิดขึ้นได้ รวมไปถึงทดสอบว่าระบบที่พัฒนานั้นสามารถทำได้ตาม Requirements ที่ตั้งไว้หรือไม่ โดยจะมีการตั้ง Test Case เพื่อทำการทดสอบการทำงานของระบบตามรายการที่ลิสไว้ หากเจอปัญหาจะต้องวิเคราะห์ว่าควรนำไปแก้ไขที่ส่วนไหน ในปัจจุบันนิยมการทำ Automated Testing มาช่วยในการ Run test ต่าง ๆ ให้มีประสิทธิภาพ และมีความรวดเร็วมากยิ่งขึ้น หากพบเจอปัญหาเช่น Bug Error ต่าง จะได้ส่งกลับให้ทีมพัฒนาอีกรอบเสียก่อนจะนำไปให้กับผู้ใช้งาน

ระยะที่ 6 : การบำรุงรักษา (Maintenance Phase)

ขั้นตอนสุดท้ายหลังจากที่ซอฟต์แวร์ถูกติดตั้งใช้งานจริงเรียบร้อย ก็ยังจำเป็นต้องได้รับการดูแลและปรับปรุงอย่างต่อเนื่อง เพื่อให้แน่ใจว่าซอฟต์แวร์จะยังคงทำงานได้ดีไปตามการเปลี่ยนแปลงของผู้ใช้และธุรกิจ การ Maintenance ไม่ใช่เพียงการแก้ไขข้อผิดพลาดที่อาจเกิดขึ้นเท่านั้น แต่รวมถึงการอัปเดต การเพิ่มฟังก์ชันใหม่ และการพัฒนาระบบที่เข้ากับเทคโนโลยีล่าสุดอีกด้วย มักต้องเกิดขึ้นเรื่อย ๆ เพื่อคุณภาพกับความน่าเชื่อถือของซอฟต์แวร์ในระยะยาว

2.1.2 ระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูล (Database System) คือระบบที่ใช้สำหรับจัดการข้อมูลที่มีโครงสร้างอย่างเป็นระเบียบ โดยเป็นแหล่งเก็บข้อมูลที่สามารถเรียกใช้และจัดการได้อย่างมีประสิทธิภาพ ผ่านทางโปรแกรมจัดการฐานข้อมูล (Database Management System - DBMS) ซึ่งมีหน้าที่สำคัญในการจัดการข้อมูลให้ปลอดภัยและเข้าถึงได้สะดวก

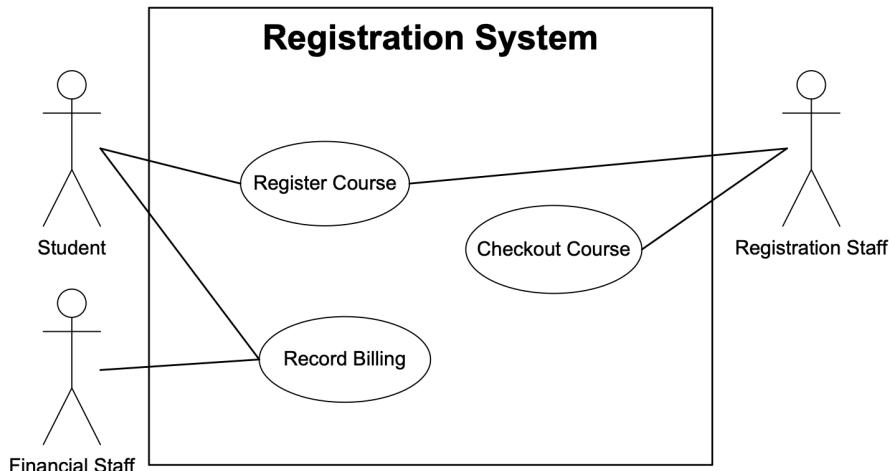
องค์ประกอบของระบบฐานข้อมูล

- ข้อมูล (Data):** ข้อมูลที่ถูกจัดเก็บในฐานข้อมูล โดยทั่วไปจะถูกจัดเก็บในรูปแบบตาราง (Table) แต่ละตารางจะมีข้อมูลที่เกี่ยวข้องกับวัตถุหรือเหตุการณ์ในโลกจริง ซึ่งแบ่งเป็นแถว (Row) และคอลัมน์ (Column) แถวจะเป็นข้อมูลแต่ละรายการ ส่วนคอลัมน์จะเป็นข้อมูลลักษณะเฉพาะของแต่ละรายการ เช่น ชื่อ, นามสกุล, ที่อยู่ เป็นต้น
- DBMS (Database Management System):** ซอฟต์แวร์ที่ใช้ในการจัดการและควบคุมฐานข้อมูล ทำหน้าที่จัดการการจัดเก็บและดึงข้อมูล รวมถึงการจัดการด้านความปลอดภัย การสำรองข้อมูล การควบคุมการเข้าถึงของผู้ใช้งาน และการทำงานร่วมกับแอปพลิเคชันต่าง ๆ ตัวอย่าง DBMS ได้แก่ MySQL, PostgreSQL, MongoDB และ Microsoft SQL Server
- ภาษา SQL (Structured Query Language):** ภาษาในการจัดการฐานข้อมูลที่ใช้สำหรับการเพิ่ม ลบ แก้ไข และค้นหาข้อมูลในฐานข้อมูล โดย SQL มีคำสั่งหลัก ๆ เช่น SELECT, INSERT, UPDATE, และ DELETE
- ผู้ใช้ฐานข้อมูล (Database Users):** ผู้ที่มีส่วนเกี่ยวข้องกับการใช้ฐานข้อมูลสามารถแบ่งได้เป็น 2 ประเภทหลักคือ ผู้ใช้ปลายทาง
 - ผู้ใช้ปลายทาง (End Users):** ผู้ใช้งานที่รับผิดชอบการเข้าถึงข้อมูลผ่านแอปพลิเคชัน
 - ผู้ดูแลระบบฐานข้อมูล (DBA - Database Administrator):** ผู้ที่รับผิดชอบในการดูแลรักษาฐานข้อมูล การกำหนดสิทธิ์การเข้าถึง และการดูแลความปลอดภัยของระบบ

2.1.3 ทฤษฎีการสร้าง Diagram

2.1.3.1 Use Case Diagram

Use Case Diagram เป็นแผนภาพที่ใช้แสดงให้ทราบว่าระบบทำงานหรือมีหน้าที่ใดบ้าง โดย มีสัญลักษณ์รูปวงรีแทน Use Case และสัญลักษณ์รูปคน (Stick Man Icon) แทน Actor สำหรับชื่อ Use Case นั้น ให้ใช้คำกริยาหรือกริยาลี (คำกริยาไม่มีกรรมมารองรับ) เช่น ลงทะเบียนเรียน, ตรวจสอบรายวิชา, บันทึกการซ าระเงิน, Generate Report, Enter Sales Data, Compute Commission เป็นต้น ส่วนการมีปฏิสัมพันธ์ระหว่าง Use Case และ Actor จะใช้เส้นตรงลาก เชื่อมต่อกัน หรือจะใช้เส้นตรงมีหัวลูกศรก็ได้ ในที่นี้เลือกใช้เส้นตรงไม่มีหัวลูกศร ส่วนเส้นแบ่งขอบเขต ระหว่าง Actor กับ Use Case จะใช้เส้นกรอบสี่เหลี่ยม เรียกว่า “System Boundary” และสิ่ง สำคัญส่วนสุดท้ายก็คือ “ชื่อของระบบ (System Name)” ให้แสดงไว้ด้านบนสุดของแผนภาพ



แสดง Use Case Diagram ของระบบลงทะเบียน

นอกจากนี้ หากกล่าวถึง Use Case Diagram ในด้านการพัฒนาระบบ นอกจากนี้จากการนำมาใช้เก็บรวบรวมความต้องการต่างๆ แล้ว Use Case Diagram ยังถูกน ำไปใช้เป็นพื้นฐานเพื่อการสร้างแผนภาพ (Diagram) ชนิดอื่นในขั้นตอนต่อๆ ไป และทีมงานยังสามารถใช้ Use Case Diagram เพื่อติดตามผลการดำเนินงานได้อีกด้วย

สัญลักษณ์และความสัมพันธ์ใน Use Case Diagram

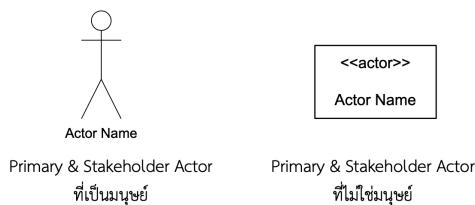
สัญลักษณ์ที่สำคัญของ Use Case Diagram มีดังต่อไปนี้

1. **Use Case** คือ คือ หน้าที่ที่ระบบต้องกระทำ ใช้สัญลักษณ์รูปวงรี พร้อมทั้งเขียนชื่อ Use Case ซึ่งต้องใช้คำกริยาหรือกริยาลีก์ได้



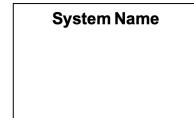
แสดงสัญลักษณ์ Use Case

2. **Actor** คือ ผู้เกี่ยวข้องกับระบบ ซึ่งรวมทั้ง Primary Actor และ Stakeholder Actor ที่เป็นมนุษย์ ในที่นี้จะใช้สัญลักษณ์รูปคน (Stick Man Icon) เมื่อกัน พร้อมทั้งเขียนชื่อ Actor ไว้ด้านล่างของสัญลักษณ์ ด้วย แต่หากเป็น Actor ที่ไม่ใช่มนุษย์ เช่น ระบบงานอื่นที่อยู่นอกเหนือระบบที่เราสนใจ จะใช้รูปสี่เหลี่ยมแล้วเขียนคำว่า “<>” ไว้ด้านบน



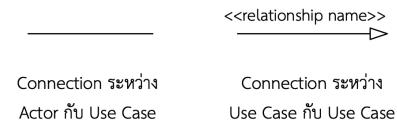
แสดงสัญลักษณ์ Actor

3. **System Boundary** คือ เส้นแบ่งขอบเขตระหว่างระบบกับผู้กระทำต่อระบบ (Use Case กับ Actor) ใช้รูป สี่เหลี่ยมเป็นสัญลักษณ์ พร้อมทั้งเขียนชื่อระบบไว้ด้าน



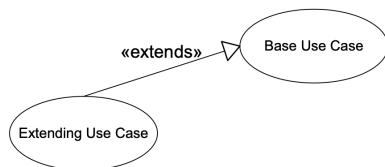
แสดงสัญลักษณ์ System Boundary

4. **Connection** คือ เส้นที่ลากเชื่อมต่อระหว่าง Actor กับ Use Case ที่มีปฏิสัมพันธ์กัน ใช้เส้นตรงไม่มีหัวลูกศรเป็นสัญลักษณ์ของ Connection ส่วน Connection ที่ใช้เชื่อมต่อระหว่าง Use Case กับ Use Case กรณีที่ Use Case นั้นมีความสัมพันธ์ซึ่งกันและกัน จะใช้ สัญลักษณ์เส้นตรงมีหัวลูกศร พร้อมทั้งเขียนชื่อความสัมพันธ์ไว้ตรงกลางเส้นด้วย โดย เขียนไว้ภายในเครื่องหมาย <<...>>



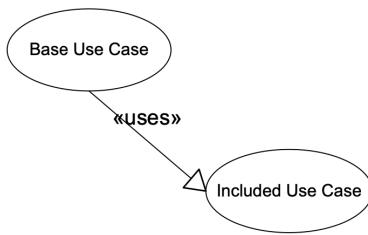
แสดงสัญลักษณ์ Connection

5. Extend Relationship คือ เป็นความสัมพันธ์แบบขยายหรือเพิ่ม เกิดขึ้นในกรณีที่บาง Use Case ดำเนินกิจกรรมของตนเองไปตามปกติ แต่อาจจะมีเงื่อนไขหรือสิ่งกระตุ้นบางอย่างที่ส่งผลให้กิจกรรมตามปกติของ Use Case นั้นถูกรบกวนจนเบี่ยงเบนไป ซึ่งความสามารถแสดงเงื่อนไขหรือสิ่งกระตุ้นเหล่านี้ได้ในรูปของ “Use Case” และเรียกว่าความสัมพันธ์ระหว่าง Use Case ในลักษณะนี้ว่า “Extend Relationship” โดยเรียก Use Case ที่ถูกรบกวนหรือ Use Case ที่ดำเนินงานตามปกติว่า “Base Use Case” และเรียก Use Case ที่ทำหน้าที่รบกวนหรือกระตุ้น Base Use Case ว่า “Extending Use Case”



แสดงการวัดเส้น Connection เชื่อมระหว่าง Extending Use Case กับ Base Use Case

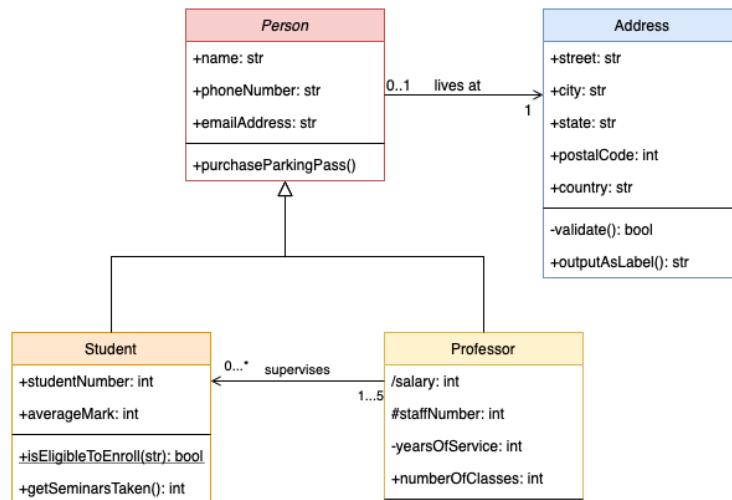
6. Include Relationship คือ ความสัมพันธ์อีกรูปแบบหนึ่งของ Use Case Diagram ก็คือ ความสัมพันธ์แบบเรียกใช้เกิดขึ้นในกรณีที่ Use Case หนึ่งไปเรียกหรือดึงกิจกรรมของอีก Use Case หนึ่งมาใช้เพื่อให้กิจกรรมนั้นเกิดขึ้นจริงใน Use Case ของตนเอง หรือกล่าวให้ง่ายกว่า นั่นคือ กิจกรรมใน Use Case หนึ่งอาจจะถูกผนวกเข้าไปรวมกับกิจกรรมของอีก Use Case หนึ่ง เราเรียกว่าความสัมพันธ์ระหว่าง Use Case ในลักษณะนี้ว่า “IncludeRelationship” 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



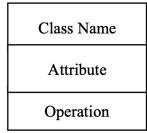
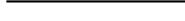
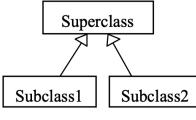
แสดงการลากเส้น Connection ระหว่าง Base Use Case กับ Included Use Case

2.1.3.2 Class Diagram

Class Diagram คือ แผนภาพที่ใช้แสดง Class และความสัมพันธ์ในแบบต่าง ๆ (Relation) ระหว่าง Class เหล่านั้น ซึ่งความสัมพันธ์ที่กล่าวถึงใน Class Diagram นี้ก็เป็นความสัมพันธ์เชิงสถิตย์ (StaticRelationship) หมายถึง ความสัมพันธ์ที่มีอยู่แล้วเป็นปกติในระหว่าง Class ต่าง ๆ ไม่ใช่ความสัมพันธ์ที่เกิดขึ้นเนื่องจากกิจกรรมต่าง ๆ ซึ่งเรียกว่า ความสัมพันธ์เชิงกิจกรรม (Dynamic Relationship) สิ่งที่ปรากฏใน Class Diagram นั้นประกอบด้วยกลุ่มของ Class และกลุ่มของ Relationship โดยสัญลักษณ์ที่ใช้ในการแสดง Class นั้นจะแทนด้วยสี่เหลี่ยมที่แบ่งออกเป็น 3 ส่วน โดยแต่ละส่วนนั้น (จากบนลงล่าง) จะใช้ในการแสดง ชื่อของ Class, Attribute, และฟังก์ชัน ต่าง ๆ ตามลำดับ



สัญลักษณ์พื้นฐาน

สัญลักษณ์	ชื่อ	คำอธิบาย
	Class	<p>Class name: ชื่อของกลุ่มข้อมูล</p> <p>Attribute: รายการข้อมูลในกลุ่มข้อมูลที่เราสนใจ</p> <p>Operation: พังก์ชั่นการทำงานที่ Class นั้นสามารถทำได้</p>
	Association	<p>แสดงความสัมพันธ์ระหว่าง Class โดยทั่วไป จะแสดงถึงความสัมพันธ์ในลักษณะไป - กลับ</p>
	Generalization	<p>แสดงความสัมพันธ์ของ Class ที่มีลักษณะ พิเศษ เพิ่มเติมจาก Class แรก โดยเรียก Class ที่มีลักษณะ พิเศษว่า Subclass และเรียก Class ที่เป็น Class หลักกว่า Super class</p>
	Aggregation	<p>แสดงความสัมพันธ์ Class ที่เป็นส่วนประกอบหนึ่ง ของส่วนประกอบทั้งหมดใน Class หลัก ซึ่งส่วนประกอบนั้นอาจถูกใช้งานจากส่วนประกอบหลักอื่นได้</p>
	Composition	<p>แสดงความสัมพันธ์ Class ที่เป็น ส่วนประกอบหนึ่ง ของส่วนประกอบทั้งหมดใน Class หลัก ซึ่งส่วนประกอบนั้นเป็นส่วนหนึ่งของส่วนประกอบหลัก</p>
	Dependency	<p>แสดงความสัมพันธ์ของ Class ที่มีอิทธิพลต่อ กัน การเปลี่ยนแปลงของ Class หนึ่งส่งผลให้อีก Class หนึ่ง เปลี่ยนแปลง</p>
	Realization	<p>แสดงความสัมพันธ์ของ Interface และ Class ซึ่ง เป็นการรับประกันว่า Class นั้นมีทุกสิ่งจาก Interface</p>

Multiplicity of Association

สัญลักษณ์	ชื่อ	คำอธิบาย
<u>_____</u>	Exactly One	แสดงความสัมพันธ์แบบ 1 ต่อ 1
<u>n</u>	Many	แสดงความสัมพันธ์แบบ 1 ต่อ มากกว่า 1
<u>0..1</u>	Optional	แสดงความสัมพันธ์แบบ 1 ต่อ 0 หรือ 1
<u>1..n</u>	One or More	แสดงความสัมพันธ์แบบ 1 หรือ มากกว่า 1
<u>0..n</u>	Zero Or More	แสดงความสัมพันธ์แบบ 1 ต่อ 0 หรือ ตั้งแต่ 1 ขึ้นไป

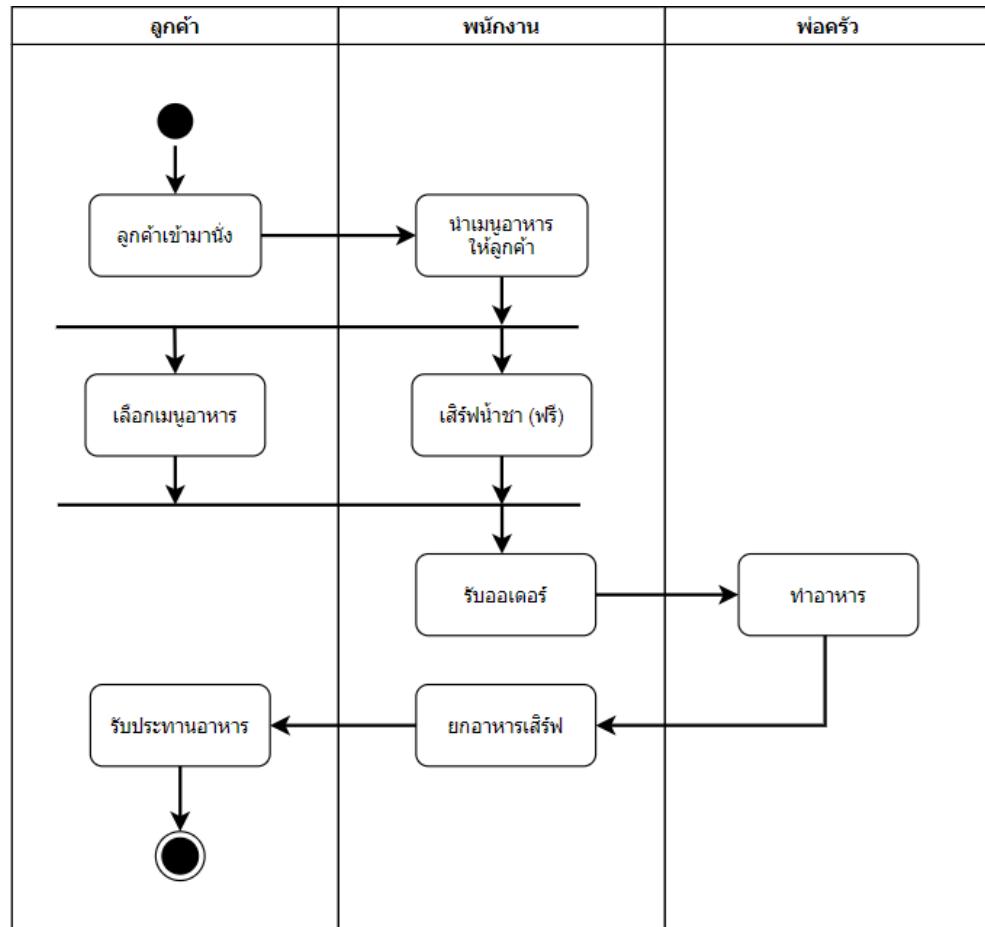
2.1.3.3 Activity diagram

Activity Diagram เป็นแผนภาพที่ใช้แสดงขั้นตอนการทำงานหรือกระบวนการ (Process Flow) ในระบบ โดยเน้นที่การทำงานแบบลำดับ (Sequential) และการตัดสินใจ (Decision) เพื่ออธิบายการไหลของ กิจกรรม (Activities) ต่าง ๆ ในระบบ ช่วยให้เห็นภาพรวมของการทำงานของระบบได้ชัดเจน

สัญลักษณ์พื้นฐาน

สัญลักษณ์	ชื่อ	คำอธิบาย
	Initial State / Start Point	จุดเริ่มต้น
	Activity or Action State	กิจกรรม
	Object Flow Guards	การไหลของกิจกรรม
	Guards	กรณีมีเงื่อนไข
	Synchronization	กรณีมีการทำกิจกรรมพร้อม ๆ กัน
	Final State / End Point	จุดสิ้นสุด

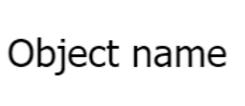
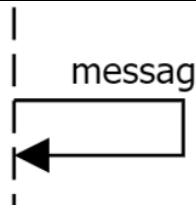
ตัวอย่าง Activity diagram



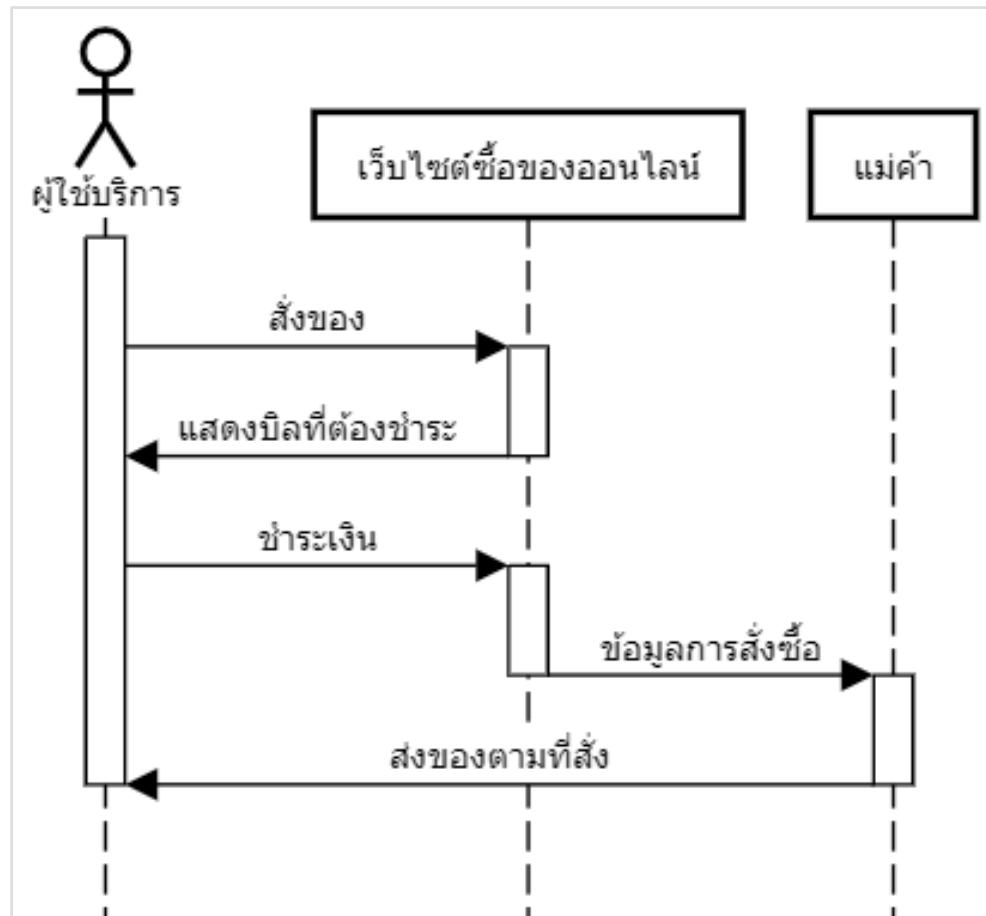
2.1.3.4 Sequence diagram

Sequence diagram ใช้แสดงถึงการปฏิสัมพันธ์กันในแต่ละส่วน ว่ามีการส่งข้อความหรือข้อมูลอะไรถึงกัน โดยแสดงตามลำดับของเวลา

ัญลักษณ์พื้นฐาน

สัญลักษณ์	ชื่อ	คำอธิบาย
 User	Actor	ผู้กระทำการต่อระบบ
	Object	อ็อบเจกต์ที่ต้องทำกิจกรรม
	Lifeline	เส้นการมีอยู่ของอ็อบเจกต์
	Focus of Control / Activation	แสดงถึงจุดเริ่มต้นและจุดสิ้นสุดของช่วงการทำกิจกรรม
	Message	คำสั่ง หรือฟังก์ชันการทำงานที่มีการส่งไป
	Callback / Self Delegation	คำสั่ง หรือฟังก์ชันการทำงานที่เรียกวากายในตัวเอง

ตัวอย่าง sequence diagram



2.1.4 Software Development Model

แบบจำลองที่แสดงให้เห็นถึงกิจกรรมหลัก (Key Activities) ของกระบวนการพัฒนา ซอฟต์แวร์ ด้วย การกำหนดรายละเอียดหรือข้อบัญญัติไว้ในแต่ละกิจกรรมในแต่ละขั้นตอนที่มีลำดับ ขั้นตอนการพัฒนาที่ชัดเจน เพื่อต้องการให้การพัฒนาซอฟต์แวร์ดำเนินต่อไปให้เกิดปัญหาน้อยที่สุด

สาเหตุสำคัญที่จำเป็นต้องใช้โมเดลการพัฒนาซอฟต์แวร์

1. โมเดลการพัฒนาซอฟต์แวร์จะมีการแบ่งขั้นตอนของกระบวนการพัฒนาออกเป็นระยะ (Phase)
2. ซอฟต์แวร์ที่พัฒนามีความซับซ้อนมากขึ้น
3. การแบ่งเป็นกระบวนการพัฒนาเป็นเฟสหรือเป็นระยะ จะทำให้ง่ายต่อการจัดการ
4. แต่ละระยะมีแนวทางต่าง ๆ ให้เลือกปฏิบัติ

1. Big Bang Model

Big Bang model เป็นโมเดล SDLC ที่เน้นไปที่การเขียนโค้ดของซอฟต์แวร์โดยตรง ไม่เน้นการวางแผน หรือการทำงานทีละขั้นตอนอย่างเป็นระบบ การพัฒนาจะเริ่มต้นโดยไม่มีแผนที่ละเอียดและโครงสร้างไม่ชัดเจน เหมาะกับโครงการขนาดเล็กที่มีนักพัฒนา 1-2 คนทำงานร่วมกันในระยะเวลาพัฒนาที่จำกัด มักจะทดสอบ ซอฟต์แวร์เมื่อเสร็จหรือใกล้เสร็จในที่เดียว

2. Waterfall Model

Waterfall model เป็นโมเดล SDLC เก่าแก่ แต่ยังคงได้รับความนิยมมาก เพราะมีลำดับขั้นตอนพัฒนา ซอฟต์แวร์ที่ชัดเจน เรียงต่อกันเป็นลำดับ แต่ละขั้นต้องเสร็จเรียบร้อยก่อนถึงจะเริ่มขั้นถัดไปได้ โมเดลนี้มีโครงสร้าง ที่เข้าใจง่ายแต่ใช้เวลาทำงาน เหมาะกับโครงการที่มีความต้องการกับข้อกำหนดชัดเจน แน่นอน และไม่มีการ

3. V-Shaped Model

V-Shaped model เป็นโมเดล SDLC ที่เน้นการทดสอบอย่างเข้มงวดในแต่ละขั้นตอน มีลำดับขั้นตอนที่ชัดเจน สามารถย้อนกลับไปทำขั้นก่อนหน้าได้ โมเดลนี้เป็นการขยายและปรับปรุงมาจาก Waterfall model ด้วยการเพิ่ม ส่วนการทดสอบเข้ามาในทุกขั้นตอน มีจุดเด่นเรื่องการรับประกัน ว่าในแต่ละขั้นตอนการพัฒนาจะได้รับการ ทดสอบอย่างถูกต้อง ทำให้ข้อผิดพลาดหรือบคคลูกราจพบ และแก้ไขได้ตั้งแต่ระยะเริ่มต้นพัฒนาซอฟต์แวร์

4. Iterative Model

Iterative Model เป็นโมเดล SDLC ขนาดใหญ่ ที่แบ่งการทำงานออกเป็นส่วนย่อย ๆ เน้นการพัฒนาผ่านการทำทวนซ้ำหลายครั้ง ในแต่ละรอบของการวนซ้ำจะมุ่งเน้นไปที่การปรับปรุงและเพิ่มเติมฟีเจอร์เฉพาะไปเรื่อย ๆ สิ่งนี้จะช่วยให้ทีมพัฒนาตอบสนองต่อ requirement ที่เปลี่ยนแปลงได้อย่างรวดเร็ว

SDLC Model นี้ยังช่วยให้ผู้ใช้งานกับฝ่ายส่วนได้ส่วนเสียได้มีโอกาสให้ข้อเสนอแนะ และมีส่วนร่วมในกระบวนการพัฒนา ทำให้ Fenolซอฟต์แวร์ออกแบบความต้องการของผู้ใช้ได้เป็นอย่างดี

5. Spiral Model

Spiral model เป็นโมเดล SDLC ที่นำจุดแข็งของ Waterfall Model มาพัฒนาให้ดีขึ้น เพราะแต่เดิมหากขั้นตอนไหนผิดพลาดจะไม่สามารถแก้ไขได้ ต้องรอจบกระบวนการทั้งหมดก่อนถึงจะแก้ไขได้ตั้งแต่ขั้นแรกและเรียงลำดับขึ้นไปเรื่อย ๆ ทำให้มีความเสี่ยงที่การพัฒนาซอฟต์แวร์จะไม่สำเร็จ

จึงมีการปรับเปลี่ยนรูปแบบเป็น Spiral model หรือโมเดลรูปเกลียว ที่เน้นการประเมินความเสี่ยงของโครงการ ด้วยการแบ่งโครงการออกเป็นโครงการที่กลุ่มย่อย ช่วยให้การปรับเปลี่ยน requirement ในระหว่างพัฒนาทำได้ง่ายขึ้น ตลอดจนประเมินความเสี่ยงกับความต่อเนื่องของโครงการในเรื่องค่าใช้จ่าย SDLC model นี้ therefore กับโครงการที่มีความซับซ้อนสูง และมีความเสี่ยงที่ต้องควบคุมอย่างเข้มงวด

6. Agile Model

Agile model เป็น SDLC Model ที่เน้นความยืดหยุ่นและการทำงานเป็นทีม การสื่อสารที่คล่องตัวจะช่วยให้สามารถพัฒนาซอฟต์แวร์ที่ตอบสนองต่อ requirement ของลูกค้าได้เร็ว นักพัฒนาจะแบ่งโครงการเป็นส่วนย่อย ๆ ที่เรียกว่า Sprint ซึ่งแต่ละ Sprint จะใช้ระยะเวลาประมาณ 1-4 สัปดาห์ เพื่อให้ซอฟต์แวร์เสร็จภายในระยะเวลาสั้น ๆ และวนกำไรบททดสอบกับผู้ใช้ เพื่อรับฟีดแบคกลับมาปรับปรุงต่อ ใน Sprint ถัดไป

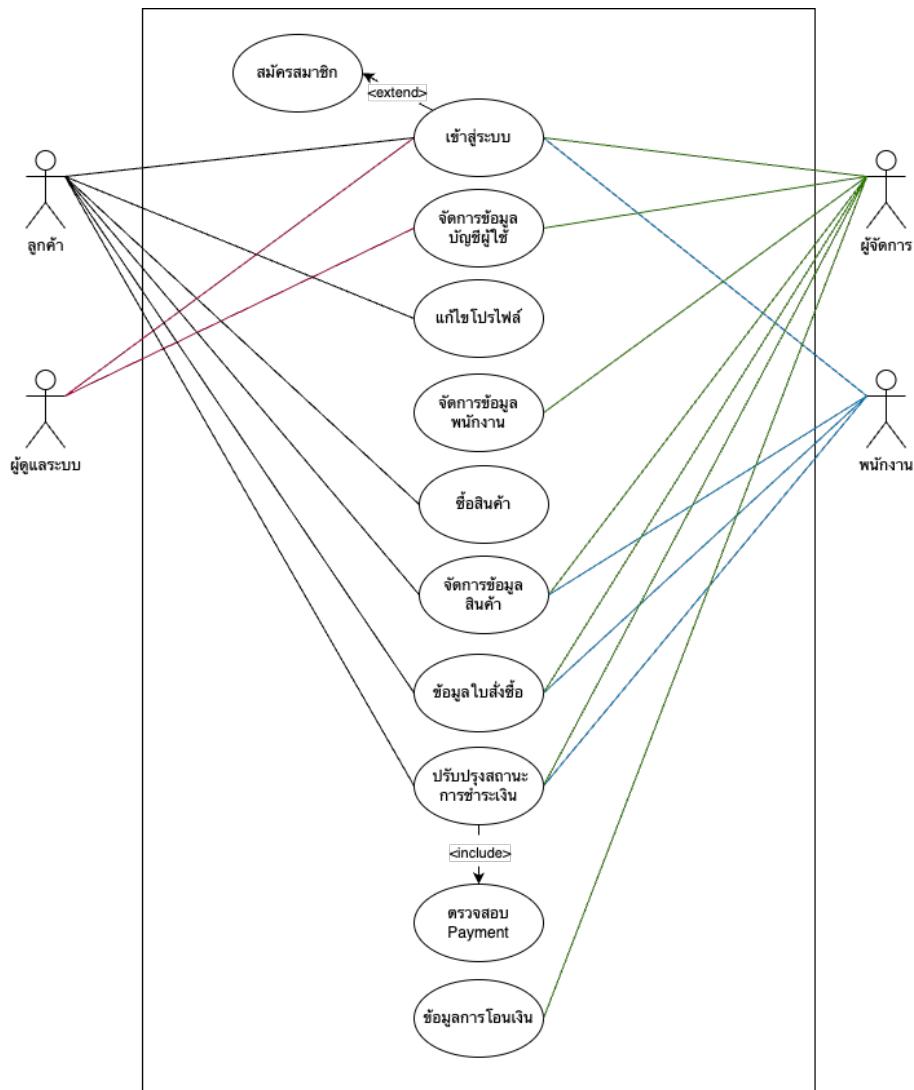
3. ขั้นตอนการดำเนินงาน

3.1 การวิเคราะห์ระบบ

3.1.1 Use Case Diagram

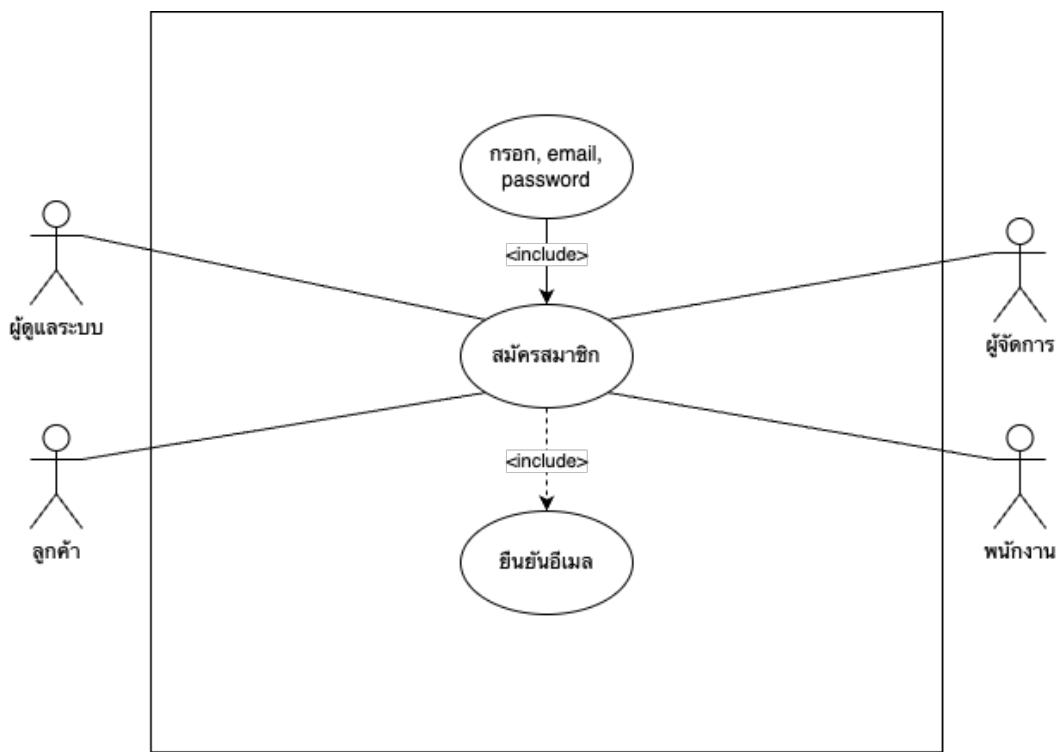
Diagram ID	Diagram Name	Diagram Description
UC-Main	UC_Main	UCD Main
UC-01	UC-001	สมัครสมาชิก
UC-02	UC-002	เข้าสู่ระบบ
UC-03	UC-003	แก้ไขบัญชีผู้ใช้
UC-04	UC-004	จัดการพนักงาน
UC-05	UC-005	ซื้อสินค้า
UC-06	UC-006	จัดการสินค้า
UC-07	UC-007	ข้อมูลใบสั่งซื้อ
UC-08	UC-008	ข้อมูลใบสั่งซื้อ
UC-09	UC-009	ปรับปรุงสถานะใบสั่งซื้อ

Use Case Diagram(UC-Main)



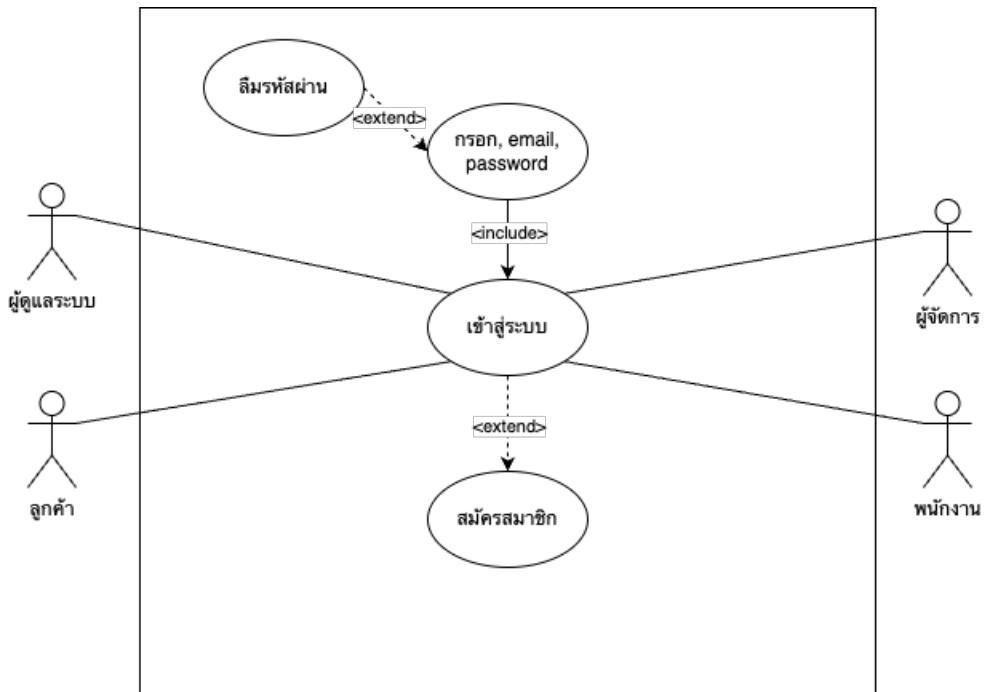
ภาพที่ 1 Use Case Diagram (Main)

Use Case Diagram (UC-01)



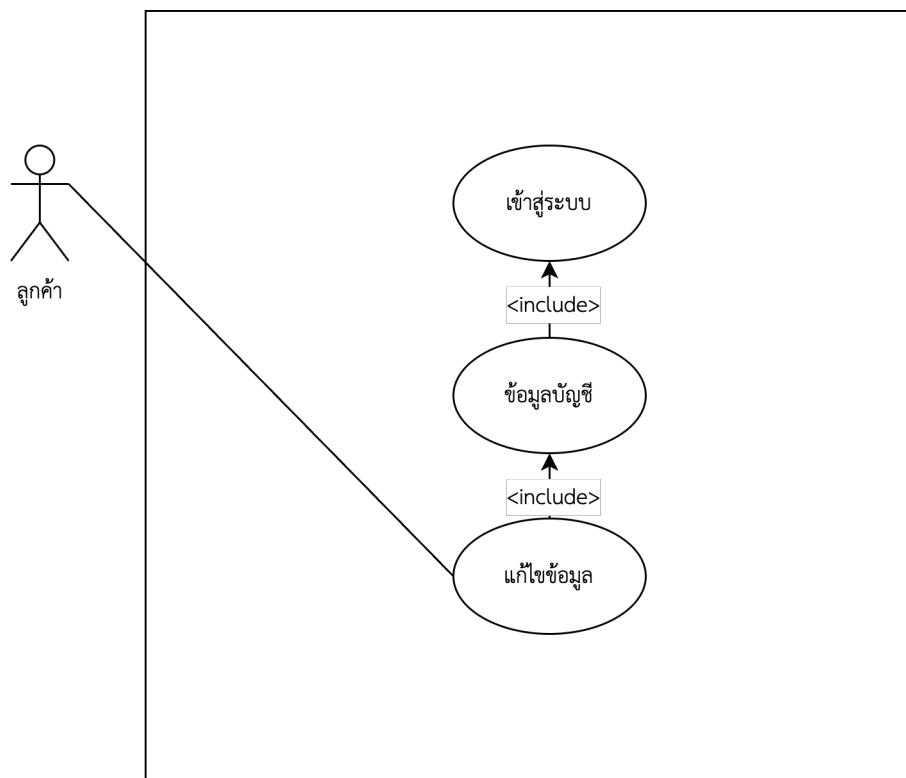
ภาพที่ 2 Use Case Diagram สมัครสมาชิก

Use Case Diagram (UC-02)



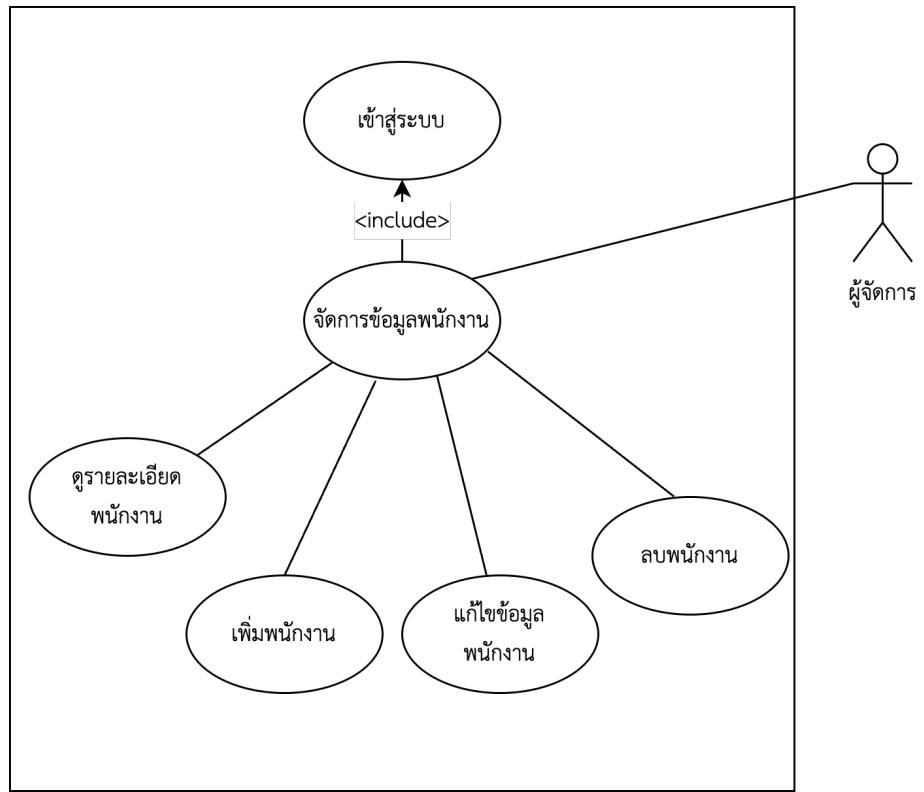
ภาพที่ 3 Use Case Diagram เข้าสู่ระบบ

Use Case Diagram (UC-03)



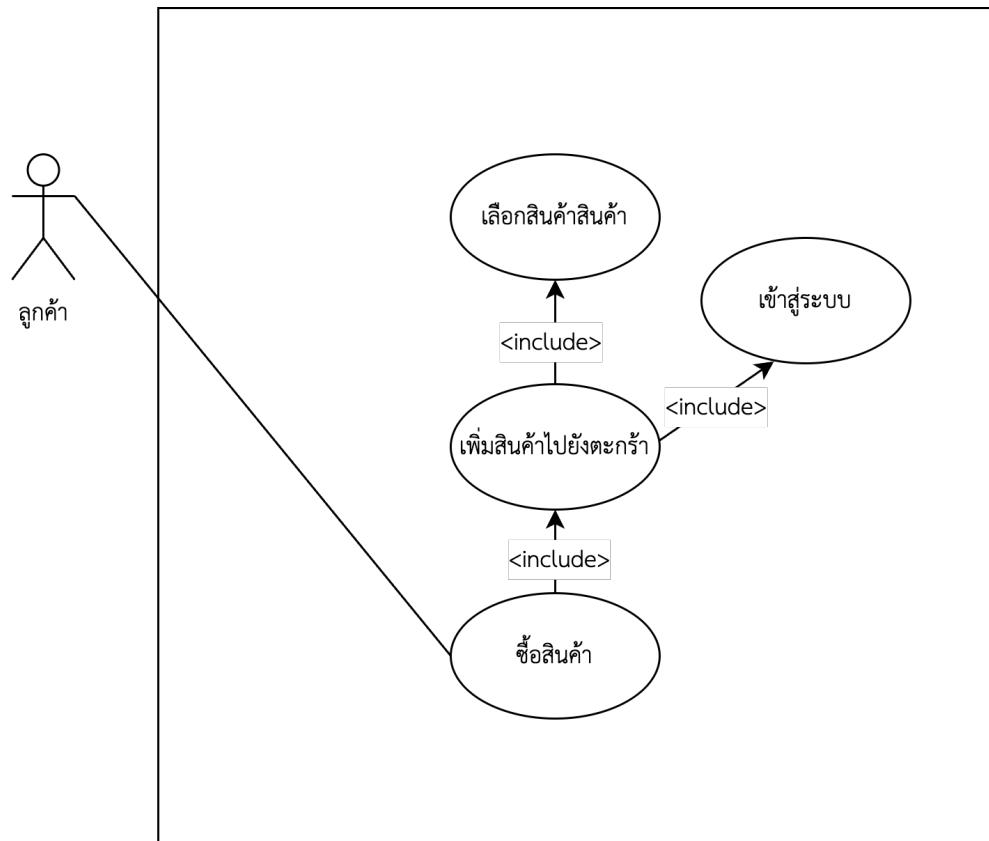
ภาพที่ 4 Use Case Diagram แก้ไขบัญชีผู้ใช้

Use Case Diagram (UC-04)



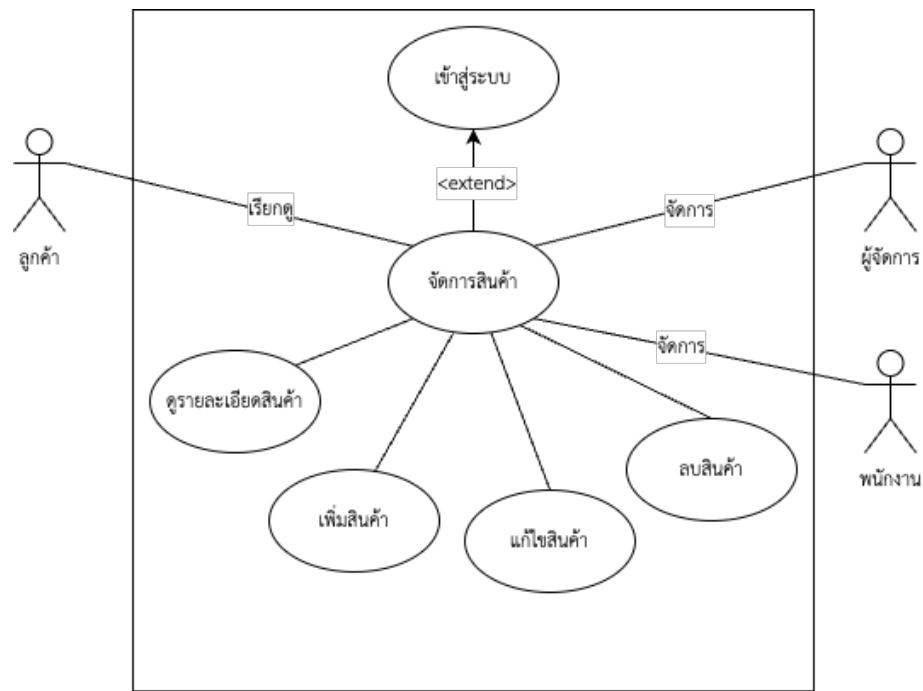
ภาพที่ 5 Use Case Diagram จัดการพนักงาน

Use Case Diagram (UC-05)



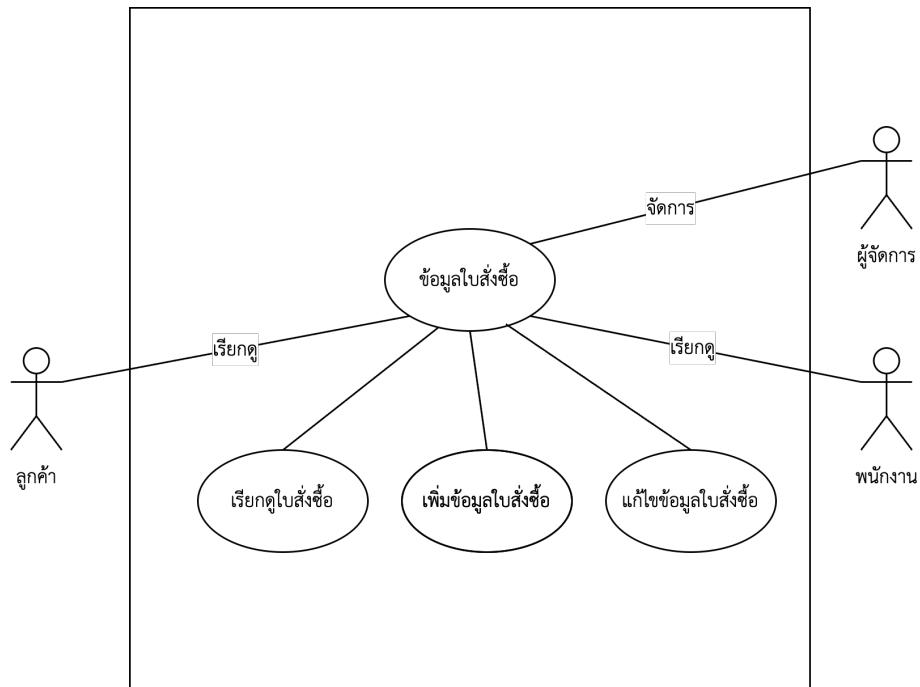
ภาพที่ 6 Use Case Diagram ซื้อสินค้า

Use Case Diagram (UC-06)



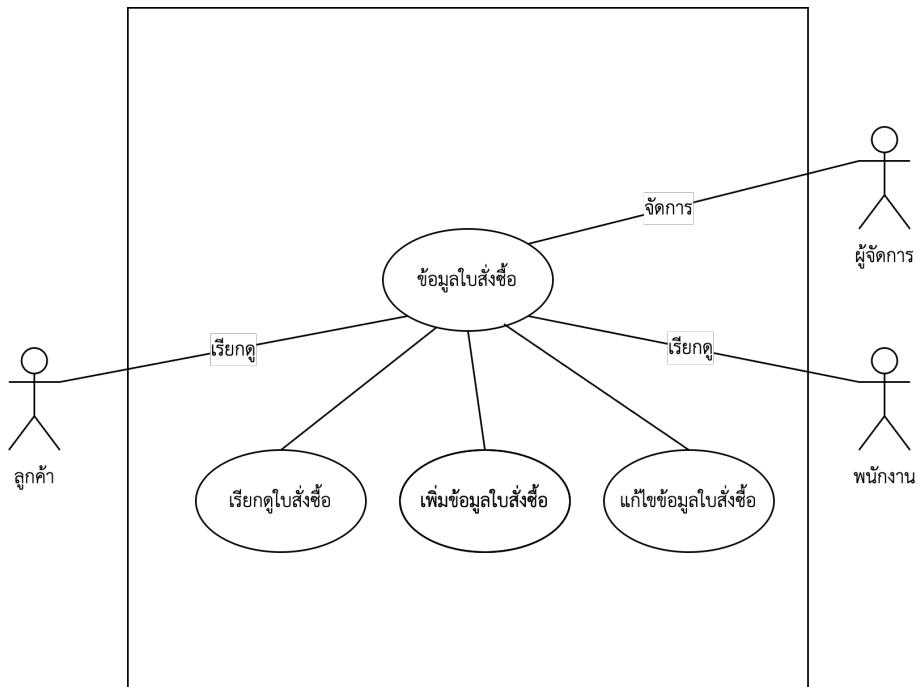
ภาพที่ 7 Use Case Diagram จัดการสินค้า

Use Case Diagram (UC-07)



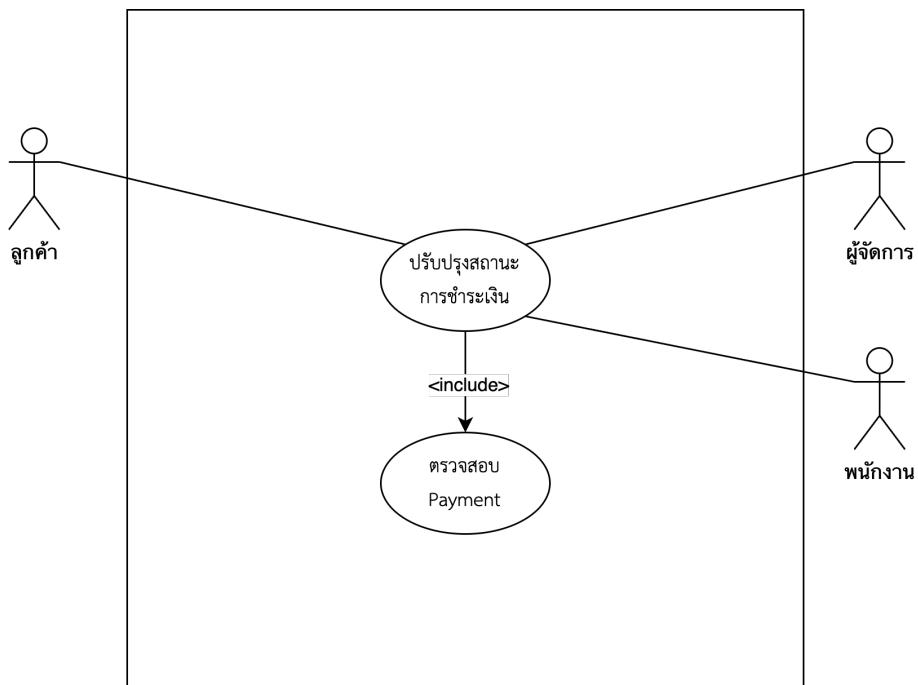
ภาพที่ 8 Use Case Diagram ข้อมูลใบสั่งซื้อ

Use Case Diagram (UC-08)



ภาพที่ 9 Use Case Diagram ข้อมูลใบสั่งซื้อ

Use Case Diagram (UC-09)



ภาพที่ 10 Use Case Diagram ปรับปรุงสถานะการชำระเงิน

3.1.2 Use Case Description

ตาราง Use Case Description (UCD - 01)

Use Case Name	สมัครสมาชิก							
Scope	สมัครสมาชิก							
Level	Primary Use Case							
Primary Actor	ลูกค้า							
Supporting Actor	-							
Offstage Actor	-							
Pre condition	ผู้ใช้ต้องกรอกข้อมูลสมัครสมาชิกครบถ้วน							
Post Condition	บัญชีผู้ใช้ถูกสร้างสำเร็จ และผู้ใช้ได้รับอีเมลยืนยันการสมัครสมาชิก							
Main success scenario	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. เลือกเมนู สมัครสมาชิก</td> <td>1. แสดงหน้าแสดงฟอร์มสมัครสมาชิก มีช่องกรอกข้อมูล เช่น อีเมล, รหัสผ่าน, ยืนยันรหัสผ่าน และ ปุ่มสมัครสมาชิก</td> </tr> <tr> <td>2. ป้อนข้อมูล 2.1 ป้อน “อีเมล” 2.2 ป้อน “รหัสผ่าน” 2.3 ป้อน “ยืนยันรหัสผ่าน”</td> <td>2. ระบบจะทำการตรวจสอบข้อมูล 2.1 ตรวจสอบข้อมูลอีเมลที่กรอกลูก ต้องตามรูปแบบที่กำหนดหรือไม่ 2.2 ตรวจสอบความปลอดภัยของรหัส ผ่าน เช่น มีความยาวขั้นต่ำ 8 ตัว อักษร ประกอบด้วยตัวพิมพ์ใหญ่ ตัว พิมพ์เล็ก และตัวเลข</td> </tr> </tbody> </table>	Actor	System	1. เลือกเมนู สมัครสมาชิก	1. แสดงหน้าแสดงฟอร์มสมัครสมาชิก มีช่องกรอกข้อมูล เช่น อีเมล, รหัสผ่าน, ยืนยันรหัสผ่าน และ ปุ่มสมัครสมาชิก	2. ป้อนข้อมูล 2.1 ป้อน “อีเมล” 2.2 ป้อน “รหัสผ่าน” 2.3 ป้อน “ยืนยันรหัสผ่าน”	2. ระบบจะทำการตรวจสอบข้อมูล 2.1 ตรวจสอบข้อมูลอีเมลที่กรอกลูก ต้องตามรูปแบบที่กำหนดหรือไม่ 2.2 ตรวจสอบความปลอดภัยของรหัส ผ่าน เช่น มีความยาวขั้นต่ำ 8 ตัว อักษร ประกอบด้วยตัวพิมพ์ใหญ่ ตัว พิมพ์เล็ก และตัวเลข	
Actor	System							
1. เลือกเมนู สมัครสมาชิก	1. แสดงหน้าแสดงฟอร์มสมัครสมาชิก มีช่องกรอกข้อมูล เช่น อีเมล, รหัสผ่าน, ยืนยันรหัสผ่าน และ ปุ่มสมัครสมาชิก							
2. ป้อนข้อมูล 2.1 ป้อน “อีเมล” 2.2 ป้อน “รหัสผ่าน” 2.3 ป้อน “ยืนยันรหัสผ่าน”	2. ระบบจะทำการตรวจสอบข้อมูล 2.1 ตรวจสอบข้อมูลอีเมลที่กรอกลูก ต้องตามรูปแบบที่กำหนดหรือไม่ 2.2 ตรวจสอบความปลอดภัยของรหัส ผ่าน เช่น มีความยาวขั้นต่ำ 8 ตัว อักษร ประกอบด้วยตัวพิมพ์ใหญ่ ตัว พิมพ์เล็ก และตัวเลข							

	<p>5. กดปุ่ม “สมัครสมาชิก”</p>	<p>3. ระบบบันทึกข้อมูล และ ระบบแสดงข้อความ “สมัครสมาชิกสำเร็จ” และ ระบบส่งยืนยันอีเมลไปทางอีเมล เพื่อให้การสมัครสมาชิกสมบูรณ์</p>
Open issues	<ul style="list-style-type: none"> - กรอกข้อมูลไม่ครบ - ป้อนรหัสผ่านไม่ปลอดภัย 	
Extensions	<ol style="list-style-type: none"> 1. กรณีขัดข้อง <ul style="list-style-type: none"> 1.1 ปิด-เปิด browser ใหม่อีกครั้ง 1.2 เข้าเมนูสมัครสมาชิก 1.3 ทำการสมัครสมาชิกใหม่อีกครั้ง 2. กรณีมีบัญชีอยู่แล้ว <ul style="list-style-type: none"> 2.1 ระบบแจ้งว่าอีเมลนี้มีบัญชีอยู่แล้ว 2.2 ไปที่เมนู เข้าสู่ระบบ หรือ ล็อกอินรหัสผ่าน 	
Special requirement	<ul style="list-style-type: none"> - มีการออกแบบ UX/UI ให้ใช้งานง่าย - มีการแสดงข้อความข้อผิดพลาดชัดเจน เมื่อป้อนข้อมูลไม่ถูกต้อง 	
Technology and Date Variation List	<ul style="list-style-type: none"> - ระบบ Regular Expression ตรวจสอบอีเมลที่ป้อนถูกต้องหรือไม่ - ระบบตรวจสอบอีเมลซ้ำกับผู้ใช้อื่น - ระบบตรวจสอบรหัสผ่านไม่ปลอดภัย 	
Frequency of Occurance	ทุกครั้งที่มีการสมัครสมาชิก	

ตาราง Use Case Description (UCD - 02)

Use Case Name	เข้าสู่ระบบ	
Scope	เข้าสู่ระบบ	
Level	Primary Use Case	
Primary Actor	ลูกค้า	
Supporting Actor	-	
Offstage Actor	-	
Pre condition	<ul style="list-style-type: none"> - ผู้ใช้ต้องมีการลงชื่อเข้าใช้ จะต้องมีบัญชีผู้ใช้ก่อน - ผู้ใช้กรอกข้อมูลครบถ้วน - ป้อนรหัสผ่านให้ต้องถูกต้อง 	
Post Condition	- ผู้ใช้เข้าสู่ระบบสำเร็จ	
Main success scenario	Actor	System
	เข้าสู่ระบบ	
	1. เลือกเมนู “เข้าสู่ระบบ”	1. แสดงหน้าแสดงฟอร์มเข้าสู่ระบบ มีช่องกรอกข้อมูล เช่น อีเมล, รหัสผ่าน, และ ปุ่มเข้าสู่ระบบ
	2. ป้อนข้อมูลที่จำเป็นเข้าสู่ระบบ 2.1 ป้อน อีเมล 2.2 ป้อน รหัสผ่าน	2. ระบบจะทำการตรวจสอบข้อมูล 2.1 ตรวจสอบข้อมูลอีเมลที่กรอกถูกต้องตามรูปแบบที่กำหนดหรือไม่

	<p>3. กดปุ่ม “เข้าสู่ระบบ”</p>	<p>3. ระบบตรวจสอบว่ามีบัญชีที่ตรงกับอีเมลและรหัสผ่านที่กรอกหรือไม่ โดยระบบจะค้นหาอีเมลที่ตรงในฐานข้อมูล</p> <p>3.1 หากเข้าสู่ระบบสำเร็จ ระบบแสดงข้อความ “เข้าสู่ระบบสำเร็จ”</p> <p>3.2 หากไม่พบบัญชีผู้ใช้ กรุณาตรวจสอบข้อมูลของคุณ”</p> <p>3.3 หากอีเมลถูกต้องแต่รหัสผ่านไม่ตรง ระบบจะแสดงข้อความ “รหัสผ่านไม่ถูกต้อง กรุณาลองใหม่อีกครั้ง”</p>
	ลืมรหัสผ่าน	
	<p>1. กดปุ่ม “ลืมรหัสผ่าน” ได้จากหน้าเข้าสู่ระบบ</p>	<p>1. แสดงหน้าแสดงฟอร์มลืมรหัสผ่าน มีช่องกรอกข้อมูล เช่น อีเมล, ปุ่มกดลืมรหัสผ่าน</p>
	<p>2. ป้อนข้อมูลที่จำเป็นในการกู้คืน</p> <p>2.1 ป้อน อีเมล</p>	<p>2. ระบบตรวจสอบข้อมูลอีเมลที่กรอกถูกต้องตามรูปแบบที่กำหนดหรือไม่</p>
	<p>3. กดปุ่ม “ลืมรหัสผ่าน”</p>	<p>3. ระบบจะส่งลิงค์กู้คืนรหัสผ่านไปยังอีเมลที่ป้อนไว้</p>
	<p>4. กดลิงก์จากอีเมล และ กรอกรหัสผ่านใหม่ พร้อมยืนยันรหัสผ่าน</p>	<p>4. ระบบบันทึกข้อมูล และ ระบบแสดงข้อความ “เปลี่ยนรหัสผ่านสำเร็จ”</p>

Open issues	<ul style="list-style-type: none"> - กรอกข้อมูลไม่ครบ - ไม่พบบัญชีผู้ใช้นี้ - รหัสผ่านไม่ถูกต้อง - การลืมรหัสผ่าน
Extensions	<ol style="list-style-type: none"> 1. กรณีขัดข้อง <ol style="list-style-type: none"> 1.1 ปิด-เปิด browser ใหม่อีกครั้ง 1.2 เข้าเมนูสมัครสมาชิก 1.3 ทำการสมัครสมาชิกใหม่อีกครั้ง 2. กรณียังไม่มีบัญชีผู้ใช้ <ol style="list-style-type: none"> 2.1 ไปที่เมนู สมัครสมาชิก 2.2 ทำการสมัครสมาชิกให้เรียบร้อย 2.3 กลับมาหน้า เข้าสู่ระบบใหม่อีกครั้ง 3. กรณีลืมรหัสผ่าน <ol style="list-style-type: none"> 3.1 ไปที่เมนู ลืมรหัสผ่าน 3.2 ทำการรีเซ็ตรหัสผ่านให้สำเร็จ 3.3 กลับมาหน้า เข้าสู่ระบบใหม่อีกครั้ง
Special requirement	<ul style="list-style-type: none"> - มีการออกแบบ UX/UI ให้ใช้งานง่าย - มีปุ่มสมัครสมาชิก - มีปุ่มลืมรหัสผ่าน
Technology and Date Variation List	<ul style="list-style-type: none"> - ระบบ regular expression ตรวจสอบอีเมลที่ป้อนถูกต้องหรือไม่
Frequency of Occurance	ทุกครั้งที่มีการเข้าสู่ระบบ

ตาราง Use Case Description (UCD - 03)

Use Case Name	แก้ไขบัญชีผู้ใช้									
Scope	ระบบแก้ไขข้อมูลบัญชีผู้ใช้									
Level	User Goal									
Primary Actor	ลูกค้า									
Supporting Actor	-									
Offstage Actor	-									
Pre condition	<ul style="list-style-type: none"> - ผู้ใช้ต้องมีการลงทะเบียนเข้าใช้ จะต้องมีบัญชีผู้ใช้ก่อน - ข้อมูล บัญชีผู้ใช้ 									
Post Condition	ข้อมูลบัญชีผู้ใช้ถูกแก้ไขสำเร็จและบันทึกเรียบร้อย									
Main success scenario	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Actor</th> <th style="text-align: center;">System</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1. ไปที่เมนู บัญชีผู้ใช้</td> <td> 1. ระบบแสดงหน้าข้อมูลบัญชีผู้ใช้ ระบบแสดงข้อมูลปัจจุบัน เช่น ชื่อ อีเมล หมายเลขโทรศัพท์ </td> </tr> <tr> <td style="text-align: center;">2. ป้อนข้อมูลใหม่ที่ต้องการแก้ไข เช่น อีเมล, ชื่อ, นามสกุล, เบอร์โทรศัพท์, รหัสผ่านใหม่</td> <td> 2. ระบบตรวจสอบข้อมูล 2.1 ระบบตรวจสอบความถูกต้องของข้อมูลที่ผู้ใช้แก้ไข เช่น ตรวจสอบรูปแบบ อีเมล, รูปแบบของเบอร์โทรศัพท์, รูปแบบรหัสผ่าน 2.2 หากข้อมูลไม่ถูกต้อง ระบบจะแจ้งเตือนและให้ผู้ใช้แก้ไข เช่น “อีเมลนี้ถูกใช้งานแล้ว” หรือ “รหัสผ่านต้องมีความยาว 8 ตัวอักษรขึ้นไป” </td> </tr> <tr> <td style="text-align: center;">3. กดปุ่ม “บันทึกการแก้ไข”</td> <td> 3. ระบบจะทำการบันทึกข้อมูลใหม่ลงในฐานข้อมูล หากสำเร็จระบบจะแสดงข้อความ “แก้ไขข้อมูลเรียบร้อยแล้ว” </td> </tr> </tbody> </table>	Actor	System	1. ไปที่เมนู บัญชีผู้ใช้	1. ระบบแสดงหน้าข้อมูลบัญชีผู้ใช้ ระบบแสดงข้อมูลปัจจุบัน เช่น ชื่อ อีเมล หมายเลขโทรศัพท์	2. ป้อนข้อมูลใหม่ที่ต้องการแก้ไข เช่น อีเมล, ชื่อ, นามสกุล, เบอร์โทรศัพท์, รหัสผ่านใหม่	2. ระบบตรวจสอบข้อมูล 2.1 ระบบตรวจสอบความถูกต้องของข้อมูลที่ผู้ใช้แก้ไข เช่น ตรวจสอบรูปแบบ อีเมล, รูปแบบของเบอร์โทรศัพท์, รูปแบบรหัสผ่าน 2.2 หากข้อมูลไม่ถูกต้อง ระบบจะแจ้งเตือนและให้ผู้ใช้แก้ไข เช่น “อีเมลนี้ถูกใช้งานแล้ว” หรือ “รหัสผ่านต้องมีความยาว 8 ตัวอักษรขึ้นไป”	3. กดปุ่ม “บันทึกการแก้ไข”	3. ระบบจะทำการบันทึกข้อมูลใหม่ลงในฐานข้อมูล หากสำเร็จระบบจะแสดงข้อความ “แก้ไขข้อมูลเรียบร้อยแล้ว”	
Actor	System									
1. ไปที่เมนู บัญชีผู้ใช้	1. ระบบแสดงหน้าข้อมูลบัญชีผู้ใช้ ระบบแสดงข้อมูลปัจจุบัน เช่น ชื่อ อีเมล หมายเลขโทรศัพท์									
2. ป้อนข้อมูลใหม่ที่ต้องการแก้ไข เช่น อีเมล, ชื่อ, นามสกุล, เบอร์โทรศัพท์, รหัสผ่านใหม่	2. ระบบตรวจสอบข้อมูล 2.1 ระบบตรวจสอบความถูกต้องของข้อมูลที่ผู้ใช้แก้ไข เช่น ตรวจสอบรูปแบบ อีเมล, รูปแบบของเบอร์โทรศัพท์, รูปแบบรหัสผ่าน 2.2 หากข้อมูลไม่ถูกต้อง ระบบจะแจ้งเตือนและให้ผู้ใช้แก้ไข เช่น “อีเมลนี้ถูกใช้งานแล้ว” หรือ “รหัสผ่านต้องมีความยาว 8 ตัวอักษรขึ้นไป”									
3. กดปุ่ม “บันทึกการแก้ไข”	3. ระบบจะทำการบันทึกข้อมูลใหม่ลงในฐานข้อมูล หากสำเร็จระบบจะแสดงข้อความ “แก้ไขข้อมูลเรียบร้อยแล้ว”									

Open issues	<ul style="list-style-type: none"> - เมื่อแก้ไขอีเมล ควรมีขั้นตอนการยืนยัน อีเมลยืนยัน - เมื่อแก้ไขเบอร์โทรศัพท์ ควรมีขั้นตอนการยืนยัน ส่ง OTP
Extensions	<ul style="list-style-type: none"> - หากข้อมูลไม่ถูกต้อง เช่น อีเมลซ้ำ ระบบจะแจ้งเตือนและให้ผู้ใช้แก้ไข - หากข้อมูลไม่ปลอดภัย ระบบจะไม่อนุญาตให้บันทึก เช่น รหัสผ่าน
Special requirement	<ul style="list-style-type: none"> - มีการออกแบบ UX/UI ให้ใช้งานง่าย - มีการแสดงข้อความข้อผิดพลาดชัดเจน เมื่อป้อนข้อมูลไม่ถูกต้อง
Technology and Date Variation List	<ul style="list-style-type: none"> - ระบบตรวจสอบความปลอดภัยรหัสผ่าน
Frequency of Occurance	เมื่อผู้ใช้ต้องการแก้ไขข้อมูลบัญชี

ตาราง Use Case Description (UCD - 04)

Use Case Name	จัดการข้อมูลพนักงาน	
Scope	ระบบจัดการข้อมูลพนักงาน	
Level	User Goal	
Primary Actor	ผู้จัดการ	
Supporting Actor	-	
Offstage Actor	-	
Pre condition	<ul style="list-style-type: none"> - ผู้ใช้ต้องมีการลงชื่อเข้าใช้ - บัญชีผู้ใช้ต้องเป็นระดับผู้จัดการ 	
Post Condition	- สามารถดำเนินการเรียกดู, เพิ่ม, แก้ไข, หรือลบข้อมูลพนักงานสำเร็จ	
Main success scenario	Actor	System
	เพิ่มพนักงาน	
	1. ไปที่เมนู พนักงาน	1. ระบบแสดงรายการพนักงานที่มีอยู่ทั้งหมด
	2. กดเมนู เพิ่มพนักงาน	2. ระบบแสดงหน้าจอฟอร์มสำหรับกรอกข้อมูลพนักงานใหม่ เช่น ชื่อ, อีเมล, หมายเลขโทรศัพท์, ตำแหน่งงาน
	3. ป้อนข้อมูลพนักงานใหม่ เช่น ชื่อ, อีเมล, รหัสผ่าน, หมายเลขโทรศัพท์, ตำแหน่งงาน	3. ระบบตรวจสอบความถูกต้องของข้อมูล เช่น รูปแบบของอีเมล, รูปแบบเบอร์โทรศัพท์

<p>4. กดปุ่ม “เพิ่มพนักงาน”</p>	<p>4. หากข้อมูลถูกต้อง ระบบจะบันทึกข้อมูลพนักงานใหม่ลงในฐานข้อมูล และ ระบบแสดงข้อความ “เพิ่มพนักงานสำเร็จ”</p>
<p>แก้ไขพนักงาน</p>	
<p>1. ไปที่เมนู พนักงาน</p>	<p>1. ระบบแสดงรายการพนักงานที่มีอยู่ทั้งหมด</p>
<p>2. เลือกพนักงานที่ต้องการแก้ไข</p>	<p>2. ระบบแสดงข้อมูลพนักงานที่เลือก</p>
<p>3. กดปุ่ม “แก้ไขพนักงาน”</p>	<p>3. ระบบแสดงหน้าจอฟอร์มสำหรับแก้ไขข้อมูลพนักงาน เช่น ชื่อ, อีเมล, หมายเลขโทรศัพท์, ตำแหน่งงาน, ปุ่มเพิ่มพนักงาน</p>
<p>4. ป้อนข้อมูลพนักงานที่ต้องการแก้ไข เช่น ชื่อ, อีเมล, รหัสผ่าน, หมายเลขโทรศัพท์, ตำแหน่งงาน</p>	<p>4. ระบบตรวจสอบความถูกต้องของข้อมูล เช่น รูปแบบของอีเมล, รูปแบบเบอร์โทรศัพท์</p>
<p>5. กดปุ่ม “บันทึก”</p>	<p>5. ระบบตรวจสอบข้อมูลที่แก้ไข หากข้อมูลถูกต้อง ระบบจะบันทึก การเปลี่ยนแปลงลงในฐานข้อมูล และ แสดงข้อความ “แก้ไขพนักงานสำเร็จ”</p>
<p>ลบพนักงาน</p>	
<p>1. ไปที่เมนู พนักงาน</p>	<p>1. ระบบแสดงรายการพนักงานที่มีอยู่ทั้งหมด</p>
<p>2. เลือกพนักงานที่ต้องการลบ</p>	<p>2. ระบบแสดงข้อมูลพนักงานที่เลือก</p>

	<p>3. กดปุ่ม “ลบพนักงาน”</p>	<p>3. ระบบดำเนินการลบ และบันทึกการลบในฐานข้อมูล ระบบแสดงข้อความ “ลบพนักงานสำเร็จ”</p>
เรียกดูข้อมูล พนักงาน		
	<p>1. ไปที่เมนู พนักงาน</p>	<p>1. ระบบแสดงรายการพนักงานที่มีอยู่ทั้งหมด</p>
	<p>2. เลือกพนักงานที่ต้องการเรียกดูข้อมูล</p>	<p>2. ระบบแสดงข้อมูลพนักงานที่เลือก</p>
Open issues	<p>เพิ่มพนักงาน</p> <ul style="list-style-type: none"> - การตรวจสอบว่าชื่อสินค้าชำหหรือไม่ <p>แก้ไขพนักงาน</p> <ul style="list-style-type: none"> - การอัปเดตข้อมูลที่แก้ไขต้องเกิดขึ้นทันทีในฐานข้อมูล <p>ลบพนักงาน</p> <ul style="list-style-type: none"> - กรณีที่มีคำสั่งซื้อค้างอยู่ในระบบ ระบบควรเตือนผู้จัดการและป้องกันการลบสินค้า <p>เรียกดูข้อมูลพนักงาน</p> <ul style="list-style-type: none"> - กรณีระบบโหลดข้อมูลช้า ต้องตรวจสอบการเชื่อมต่อหรือฐานข้อมูล 	
Extensions	<ul style="list-style-type: none"> - ในกรณีที่ข้อมูลการป้อนไม่ถูกต้อง เช่น อีเมลหรือเบอร์โทรศัพท์ ระบบจะแจ้งเตือนผู้จัดการให้แก้ไขข้อมูลก่อนบันทึก - กรณีมีข้อผิดพลาดในการเชื่อมตอกับฐานข้อมูล ระบบควรแสดงข้อความให้ผู้ใช้งานอ่านถูกต้อง 	
Special requirement	<ul style="list-style-type: none"> - ระบบต้องมีการตรวจสอบสิทธิ์ของผู้จัดการ 	
Technology and Date Variation List	<ul style="list-style-type: none"> - ระบบควรมีการตรวจสอบรูปแบบอีเมลและเบอร์โทรศัพท์ที่ถูกต้อง เช่น ผ่าน Regular Expression - ใช้การเข้ารหัสรหัสผ่านเพื่อความปลอดภัยในการจัดเก็บข้อมูลพนักงาน 	

Frequency of Occurance

ใช้งานไม่บ่อย โดยอาจใช้งานเมื่อมีการเพิ่ม, แก้ไข, หรือลบพนักงานใหม่
หรือเมื่อต้องการเรียกดูข้อมูลพนักงาน

ตาราง Use Case Description (UCD - 05)

Use Case Name	ซื้อสินค้า	
Scope	ซื้อสินค้า	
Level	User Goal	
Primary Actor	ลูกค้า	
Supporting Actor	-	
Offstage Actor	-	
Pre condition	<ul style="list-style-type: none"> - ผู้ใช้ต้องมีการลงทะเบียนเข้าใช้ จะต้องมีบัญชีไว้ก่อน - มีรายการสินค้า มากกว่า 1 รายการในตะกร้าสินค้า 	
Post Condition	- ได้รับใบสั่งซื้อสินค้า	
Main success scenario	Actor	System
	1. เลือกสินค้า	1. ระบบแสดงรายการสินค้าและรายละเอียด เช่น ชื่อสินค้า, รูปภาพ, ราคา
	2. ลูกค้ากดปุ่ม “เพิ่มสินค้าลงในตะกร้า”	2. ระบบบันทึกสินค้าลงในตะกร้าสินค้า และแสดงข้อความ “เพิ่มสินค้าไปยังตะกร้าสำเร็จ”
	3. ลูกค้าเข้าดูรายการสินค้าในตะกร้าเพื่อยืนยันการสั่งซื้อ	3. ระบบแสดงรายการสินค้าที่อยู่ในตะกร้า เช่น ชื่อสินค้า, ราคาสินค้า
	4. ลูกค้ากดปุ่ม “ดำเนินการต่อ”	4. ระบบแสดงหน้าสรุประการสั่งซื้อ รวมถึงสินค้าทั้งหมด, ราคาน้ำหนัก, ค่าจัดส่ง และยอดรวมที่ต้องชำระ

	5. ลูกค้าเลือกวิธีการชำระเงิน เช่น บัตรเครดิต หรือ โอนผ่านธนาคาร	5. ระบบแสดงตัวเลือกวิธีการชำระเงิน และ รอการยืนยัน
	6. ลูกค้ากรอกข้อมูลการชำระเงิน เช่น หมายเลขบัตรเครดิต หรือ ข้อมูลการโอนเงิน	6. ระบบตรวจสอบข้อมูลการชำระเงินของลูกค้า เช่น หมายเลขบัตร หรือ ข้อมูลบัญชีธนาคาร
	7. ลูกค้ากดปุ่ม “ยืนยันการชำระเงิน”	7. ระบบดำเนินการชำระเงิน และ ตรวจสอบสถานะการทำธุรกรรม
		8. หากการชำระเงินสำเร็จ ระบบสร้างคำสั่งซื้อ และ แสดงใบสั่งซื้อ และ ยอดชำระรวม
		9. ระบบแสดงข้อความ “การสั่งซื้อ และชำระเงินสำเร็จ”
Open issues	<ul style="list-style-type: none"> - ปัญหาการชำระเงินหรือสินค้าหมดสต็อกขณะดำเนินการ ระบบควรแจ้งเตือนลูกค้า และยกเลิกการสั่งซื้อสำหรับสินค้าที่หมดสต็อก - ปัญหาการชำระเงิน หากการชำระเงินไม่สำเร็จ เช่น บัตรถูกปฏิเสธหรือข้อมูลบัญชีธนาคารไม่ถูกต้อง 	
Extensions	<ul style="list-style-type: none"> - หากชำระเงินไม่สำเร็จ ระบบจะให้ผู้ใช้เลือกวิธีการชำระเงินอื่น - หากสินค้าหมดสต็อกระหว่างการสั่งซื้อ ระบบจะแจ้งเตือนและยกเลิกการสั่งซื้อ 	
Special requirement	<ul style="list-style-type: none"> - ระบบต้องแสดงรายการสรุปการสั่งซื้อก่อนยืนยัน 	
Technology and Date Variation List	<ul style="list-style-type: none"> - 	
Frequency of Occurance	ทุกครั้งที่มีการสั่งซื้อสินค้า	

ตาราง Use case description (UCD - 06)

Use Case Name	จัดการสินค้า	
Scope	จัดการสินค้า	
Level	User Goal	
Primary Actor	ผู้จัดการ	
Supporting Actor	-	
Offstage Actor	-	
Pre condition	<ul style="list-style-type: none"> - บัญชีผู้ใช้ต้องเป็นระดับผู้จัดการ - ลงชื่อเข้าใช้ระบบเรียบร้อย 	
Post Condition	<ul style="list-style-type: none"> - ข้อมูลสินค้าที่เพิ่ม, แก้ไข, ลบ, หรือเรียกดูถูกจัดการสำเร็จและบันทึกในฐานข้อมูล 	
Main success scenario	Actor	System
เพิ่มสินค้า		
	1. ไปที่เมนู สินค้า	1. ระบบแสดงรายการสินค้าที่มีอยู่ทั้งหมด
	2. กดเมนู เพิ่มสินค้า	2. ระบบแสดงหน้าจอฟอร์มสำหรับกรอกข้อมูลสินค้าใหม่ เช่น ชื่อสินค้า, ราคา, และรายละเอียดสินค้า
	3. ป้อนข้อมูลสินค้าใหม่ เช่น ชื่อสินค้า, ราคา, และรายละเอียดสินค้า	3. ระบบตรวจสอบความถูกต้องของข้อมูล เช่น ชื่อสินค้าไม่ซ้ำ, ราคาที่ถูกต้อง

4. กดปุ่ม “เพิ่มสินค้า”	4. หากข้อมูลถูกต้อง ระบบจะบันทึกข้อมูลสินค้าใหม่มีลงในฐานข้อมูล และ ระบบแสดงข้อความ “เพิ่มสินค้าสำเร็จ”
แก้ไขสินค้า	
1. ไปที่เมนู สินค้า	1. ระบบแสดงรายการสินค้าที่มีอยู่ทั้งหมด
2. เลือกสินค้าที่ต้องการแก้ไข	2. ระบบแสดงข้อมูลสินค้าที่เลือก
3. กดปุ่ม “แก้ไขสินค้า”	3. ระบบแสดงแบบฟอร์มแก้ไขข้อมูลของสินค้าที่เลือก พร้อมข้อมูลปัจจุบัน เช่น ชื่อสินค้า, ราคา, ..
4. ป้อนข้อมูลสินค้าที่ต้องการแก้ไข เช่น ชื่อ, ชื่อสินค้า, ราคา, อื่นๆ	4. ระบบตรวจสอบข้อมูลใหม่ที่กรอกเพื่อให้แน่ใจว่าข้อมูลถูกต้อง
5. กดปุ่ม “บันทึก”	5. ระบบบันทึกข้อมูลที่แก้ไขลงในฐานข้อมูล และ แสดงข้อความ “แก้ไขข้อมูลสินค้าเรียบร้อย”
ลบสินค้า	
1. ไปที่เมนู สินค้า	1. ระบบแสดงรายการสินค้าที่มีอยู่ทั้งหมด
2. เลือกสินค้าที่ต้องการลบ	2. ระบบแสดงข้อมูลสินค้าที่เลือก
3. กดปุ่ม “ลบสินค้า”	3. ระบบดำเนินการลบ และ บันทึกการลบในฐานข้อมูล ระบบแสดงข้อความ “ลบสินค้าสำเร็จ”
เรียกดูข้อมูลสินค้า	

	<p>1. ไปที่เมนู สินค้า</p> <p>2. เลือกสินค้าที่ต้องการเรียกดูข้อมูล</p>	<p>1. ระบบแสดงรายการสินค้าที่มีอยู่ทั้งหมด</p> <p>2. ระบบแสดงข้อมูลสินค้าที่เลือก</p>
Open issues	<p>เพิ่มสินค้า</p> <ul style="list-style-type: none"> - การตรวจสอบว่าชื่อสินค้าซ้ำหรือไม่ <p>แก้ไขสินค้า</p> <ul style="list-style-type: none"> - การอัปเดตข้อมูลที่แก้ไขต้องเกิดขึ้นทันทีในฐานข้อมูล 	
	<p>ลบสินค้า</p> <ul style="list-style-type: none"> - กรณีที่มีคำสั่งซื้อค้างอยู่ในระบบ ระบบควรเตือนผู้จัดการและป้องกันการลบสินค้า 	
	<p>เรียกดูข้อมูลสินค้า</p> <ul style="list-style-type: none"> - กรณีระบบโหลดข้อมูลช้า ต้องตรวจสอบการเชื่อมต่อหรือฐานข้อมูล 	
Extensions	<ul style="list-style-type: none"> - หากลบสินค้า ระบบจะตรวจสอบว่ามีรายการสั่งซื้อค้างหรือไม่ หากมีระบบจะป้องกันการลบ 	
Special requirement	<ul style="list-style-type: none"> - ระบบต้องสามารถจัดการข้อมูลสินค้าได้ง่าย และปลอดภัย 	
Technology and Date Variation List	<ul style="list-style-type: none"> - 	
Frequency of Occurance	<ul style="list-style-type: none"> - เมื่อมีการเพิ่ม แก้ไข ลบ หรือเรียกดูสินค้า 	

ตาราง Use Case Description (UCD - 07)

Use Case Name	ข้อมูลใบสั่งซื้อ	
Scope	ข้อมูลใบสั่งซื้อ	
Level	User Goal	
Primary Actor	ผู้จัดการ, พนักงาน, ลูกค้า	
Supporting Actor	-	
Offstage Actor	-	
Pre condition	<ul style="list-style-type: none"> - ผู้ใช้ต้องมีการลงชื่อเข้าใช้ จะต้องมีบัญชีผู้ใช้ก่อน - มีข้อมูลใบสั่งซื้อ 	
Post Condition	- ข้อมูลใบสั่งซื้อถูกแสดงสำเร็จ	
Main success scenario	Actor	System
	ลูกค้า	
	1. ลูกค้าเข้าสู่ระบบและไปที่เมนู “ประวัติการสั่งซื้อ”	1. แสดงรายการใบสั่งซื้อทั้งหมดของลูกค้า
	2. ลูกค้าเลือกใบสั่งซื้อที่ต้องการดูรายละเอียด	2. ระบบแสดงรายละเอียดของใบสั่งซื้อที่ลูกค้าเลือก เช่น วันที่สั่งซื้อ, รายการสินค้า, ราคาสุทธิ
	ผู้จัดการ/พนักงาน	
	1. ไปที่เมนู “ใบสั่งซื้อ”	1. แสดงรายการใบสั่งซื้อทั้งหมดที่สามารถเข้าถึงได้

	2. เลือกใบสั่งซื้อที่ต้องการดูรายละเอียด	2. ระบบแสดงข้อมูลใบสั่งซื้อที่ผู้จัดการหรือพนักงานเลือก รวมถึงรายละเอียดสินค้า, ข้อมูลลูกค้า, และสถานะการจัดส่ง
Open issues	- ไม่พบข้อมูลใบสั่งซื้อ	
Extensions	- หากไม่พบข้อมูลใบสั่งซื้อ ระบบจะแจ้งเตือนผู้ใช้และแนะนำวิธีแก้ไข	
Special requirement	- ระบบต้องแสดงรายละเอียดใบสั่งซื้ออุปกรณ์ชั้ดเจน	
Technology and Date Variation List	- ตรวจสอบการเข้าถึงใบสั่งซื้อเฉพาะผู้ใช้ที่มีสิทธิ์	
Frequency of Occurance	- ทุกครั้งที่ผู้ใช้ต้องการตรวจสอบใบสั่งซื้อ	

ตาราง Use Case Description (UCD - 08)

Use Case Name	ปรับปรุงสถานะการชำระเงิน	
Scope	ปรับปรุงสถานะการชำระเงิน	
Level	ปรับปรุงสถานะการชำระเงิน, ใบคำสั่งซื้อ	
Primary Actor	ผู้จัดการ	
Supporting Actor	ลูกค้า	
Offstage Actor	- Payment Gateway	
Pre condition	<ul style="list-style-type: none"> - ผู้ใช้ต้องมีการลงชื่อเข้าใช้ จะต้องมีบัญชีผู้ใช้ก่อน - บัญชีผู้ใช้ต้องเป็นระดับ “ผู้จัดการ” - เลือกใบสั่งซื้อที่ต้องการปรับปรุงสถานะ 	
Post Condition	- คำสั่งซื้อที่เลือกได้รับการอปเดตสถานะสำเร็จ	
Main success scenario	Actor	System
	1. ไปที่เมนู “ใบสั่งซื้อ”	1. แสดงรายการใบสั่งซื้อทั้งหมด
	2. ผู้จัดการเลือกใบสั่งซื้อที่ต้องการปรับปรุงสถานะ	2. ระบบแสดงรายละเอียดของใบสั่งซื้อที่เลือก เช่น ข้อมูลสินค้า, ราคาสินค้า, และสถานะการชำระเงินปัจจุบัน
	3. ผู้จัดการเลือกสถานะใหม่ของใบสั่งซื้อ เช่น “ชำระเงินแล้ว” หรือ “รอการชำระเงิน”	3. ระบบให้ผู้จัดการเลือกสถานะใหม่จากการสถานะที่มี เช่น “ชำระเงินแล้ว”, “รอการชำระเงิน” หรือสถานะอื่น ๆ

	4. ผู้จัดการกดปุ่ม “บันทึก” เพื่อ ยืนยันการเปลี่ยนสถานะ	4. ระบบตรวจสอบข้อมูลการ เปลี่ยนแปลงสถานะและบันทึก ข้อมูลการเปลี่ยนแปลงในฐานข้อมูล
		5. ระบบอัปเดตสถานะใบสั่งซื้อ
		6. ระบบแสดงข้อความ “การ เปลี่ยนแปลงสถานะสำเร็จ” และ แสดงสถานะใบสั่งซื้อที่ปรับปรุง ล่าสุด
Open issues	- ความล่าช้าในการซิงค์กับระบบภายนอก	
Extensions	- การอัปเดตสถานะแบบอัตโนมัติ	
Special requirement	- ระบบต้องแสดงผลการเปลี่ยนแปลงแบบ Real-time	
Technology and Date Variation List	- การเชื่อมต่อกับ Payment Gateway ผ่าน API	
Frequency of Occurance	ทุกครั้งที่มีการชำระเงิน	

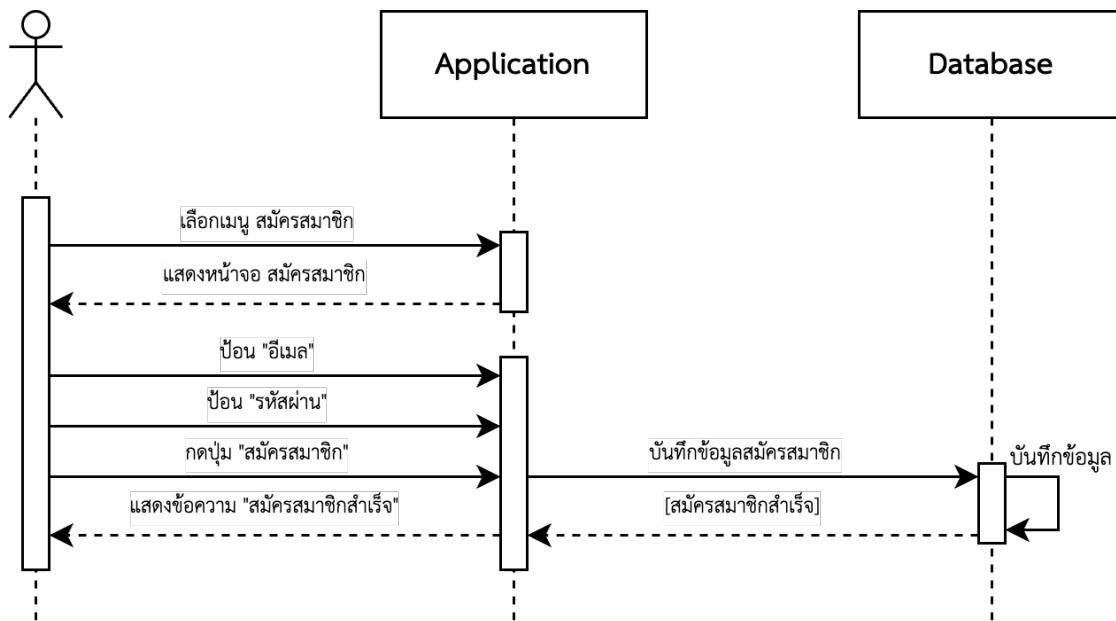
ตาราง Use Case Description (UCD - 09)

Use Case Name	ข้อมูลการโอนเงิน	
Scope	ข้อมูลการโอนเงิน, ใบสั่งซื้อ	
Level	User Goal	
Primary Actor	ผู้จัดการ	
Supporting Actor	ระบบชำระเงิน	
Offstage Actor	-	
Pre condition	<ul style="list-style-type: none"> - ผู้ใช้ต้องมีการลงชื่อเข้าใช้ จะต้องมีบัญชีผู้ใช้ก่อน - บัญชีผู้ใช้ต้องเป็นระดับผู้จัดการ 	
Post Condition	- ข้อมูลการโอนเงินถูกแสดงสำเร็จ	
Main success scenario	Actor	System
	1. ไปที่เมนู “รายการโอนเงิน”	1. ระบบแสดงรายการการโอนเงิน ทั้งหมด รวมถึงวันที่และจำนวนเงิน ที่โอน
	2. เลือกข้อมูลการโอนเงินที่ต้องการ ตรวจสอบ	2. ระบบแสดงรายละเอียดการโอน เงินที่เลือก เช่น เลขที่อ้างอิงการ โอน, วันที่, เวลา, ธนาคารที่ทำ รายการ, และสถานะการโอน
Open issues	- ไม่เพ็บข้อมูลการโอนเงิน	
Extensions	- หากไม่เพ็บข้อมูลการโอนเงิน ระบบจะแจ้งเตือนผู้ใช้และแนะนำให้ตรวจสอบใหม่	
Special requirement	- ระบบต้องแสดงรายละเอียดการโอนเงินที่ถูกต้องและครบถ้วน	

Technology and Date Variation List	- ตรวจสอบข้อมูลการโอนเงินก่อนแสดงผลเพื่อป้องกันข้อมูลที่ผิดพลาด
Frequency of Occurance	ทุกครั้งที่มีการตรวจสอบการโอนเงิน

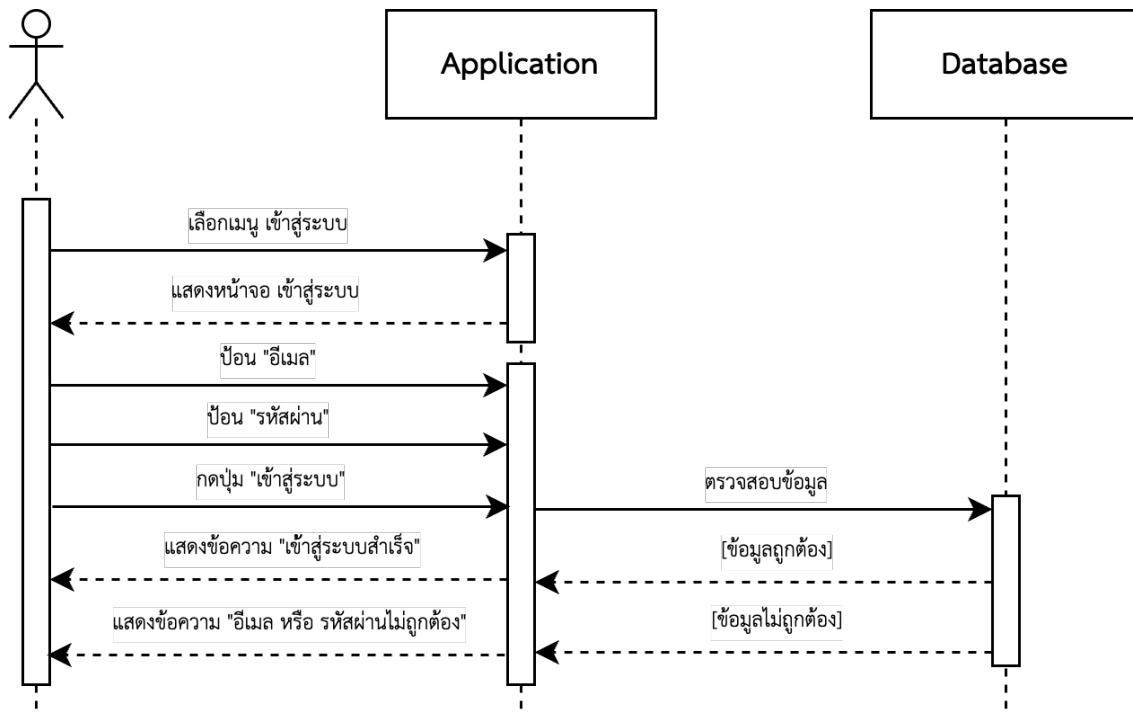
3.1.3 Sequence Diagram

3.1.3.1 Sequence diagram สมัครสมาชิก



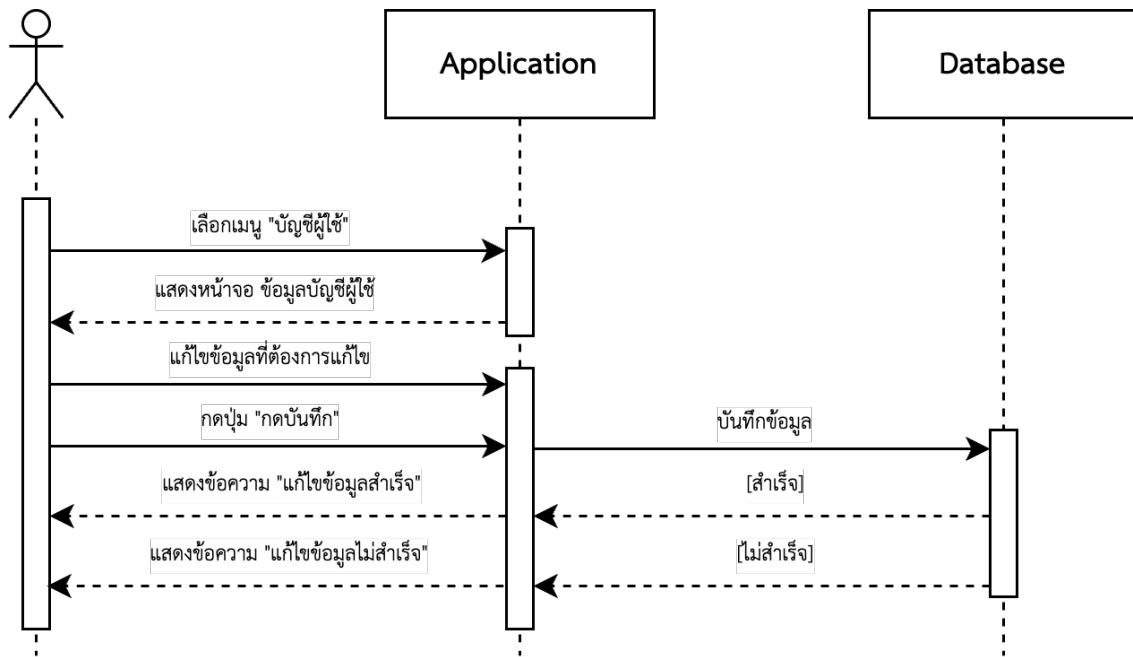
ภาพที่ 11 Sequence Diagram สมัครสมาชิก

3.1.3.2 Sequence diagram เข้าสู่ระบบ



ภาพที่ 12 Sequence Diagram เข้าสู่ระบบ

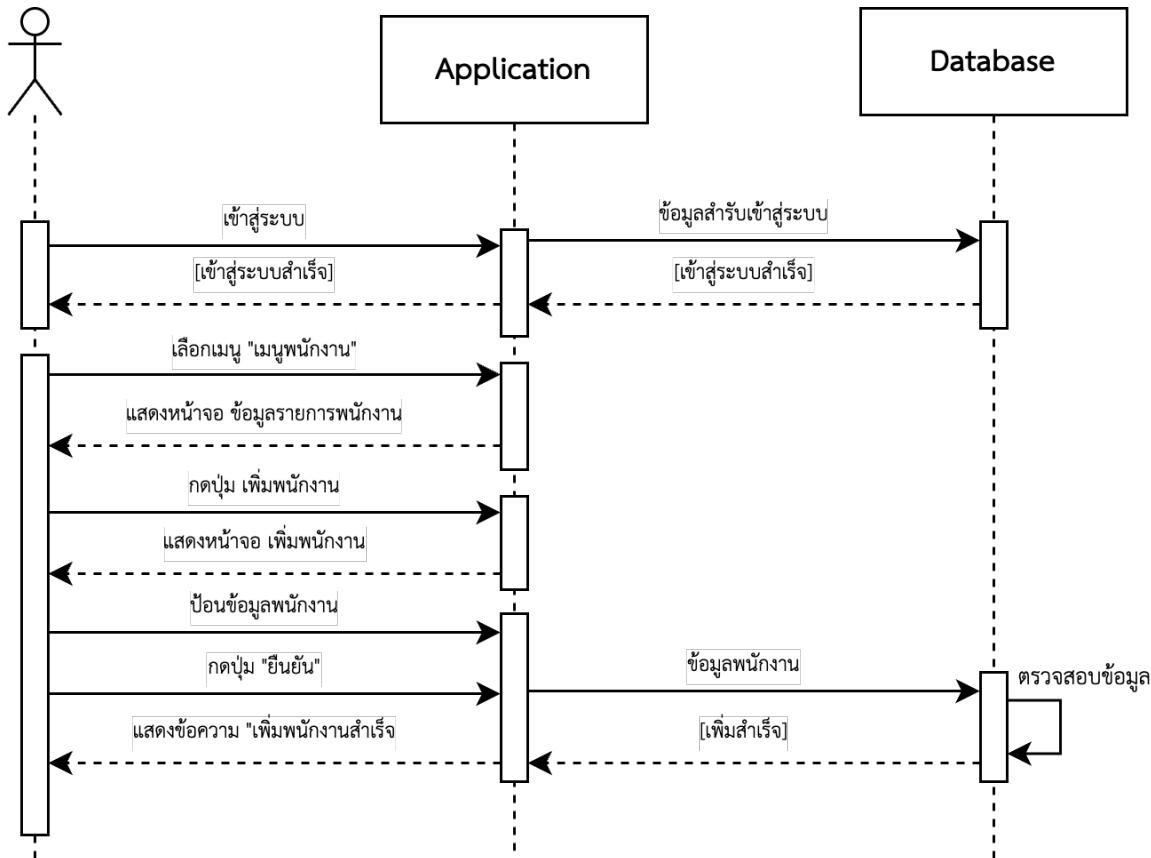
3.1.3.3 Sequence diagram แก้ไขข้อมูลบัญชีผู้ใช้



ภาพที่ 13 Sequence Diagram แก้ไขข้อมูลบัญชีผู้ใช้

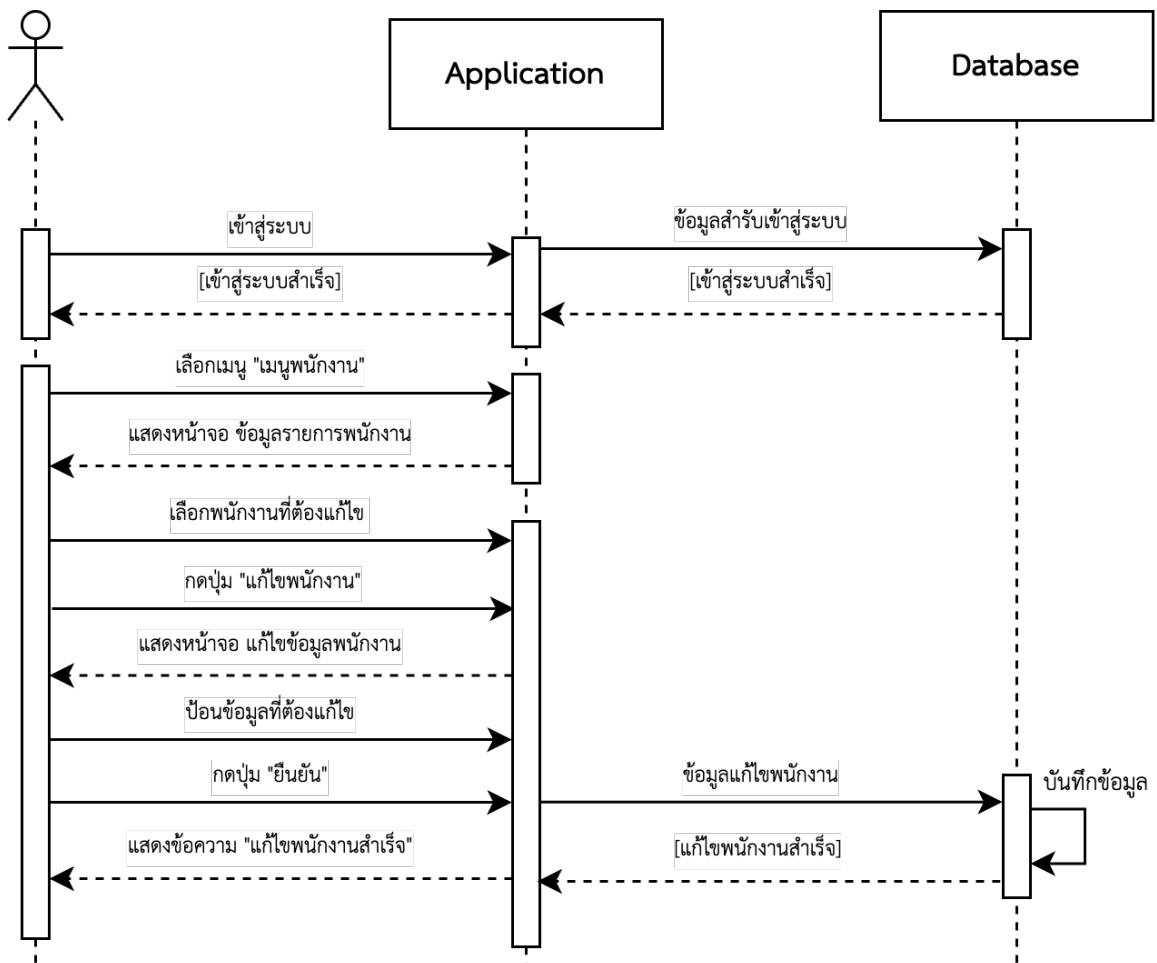
3.1.3.4 Sequence diagram จัดการพนักงาน

3.1.3.4.1 เพิ่มพนักงาน



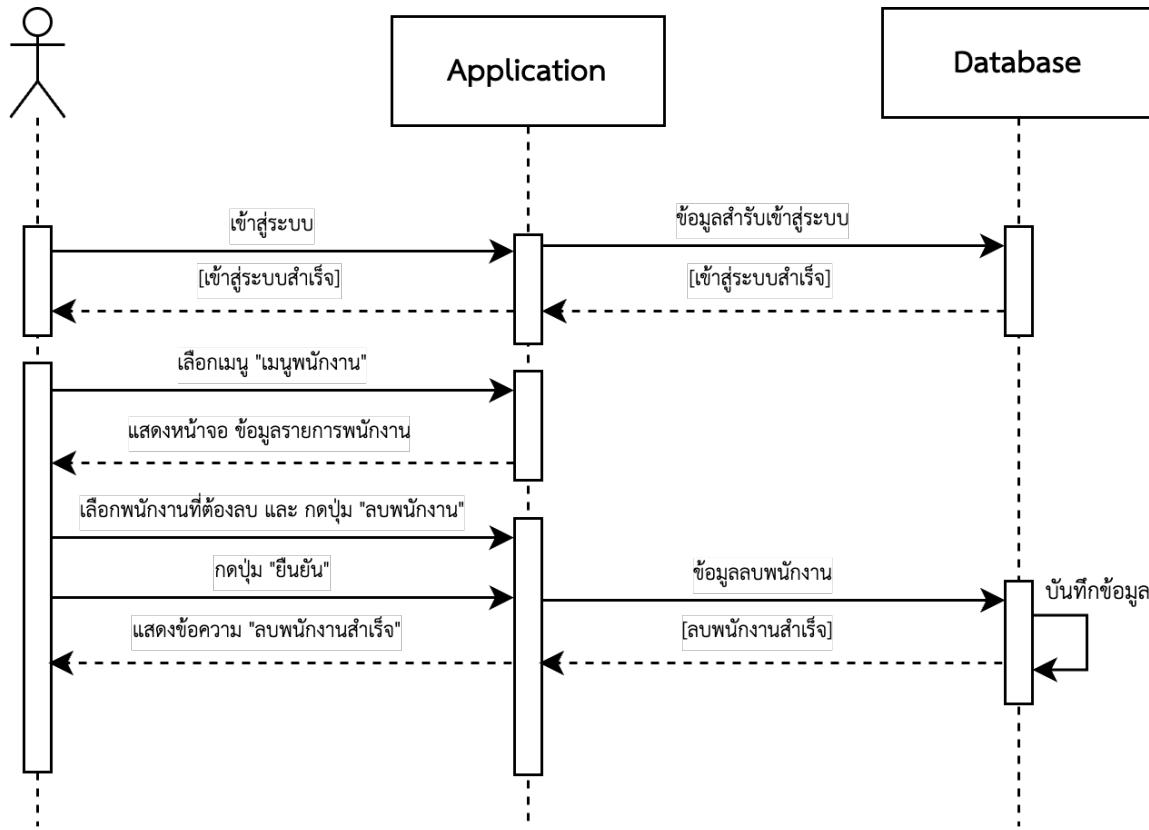
ภาพที่ 14 Sequence Diagram เพิ่มพนักงาน

3.1.3.4.1 แก้ไขพนักงาน



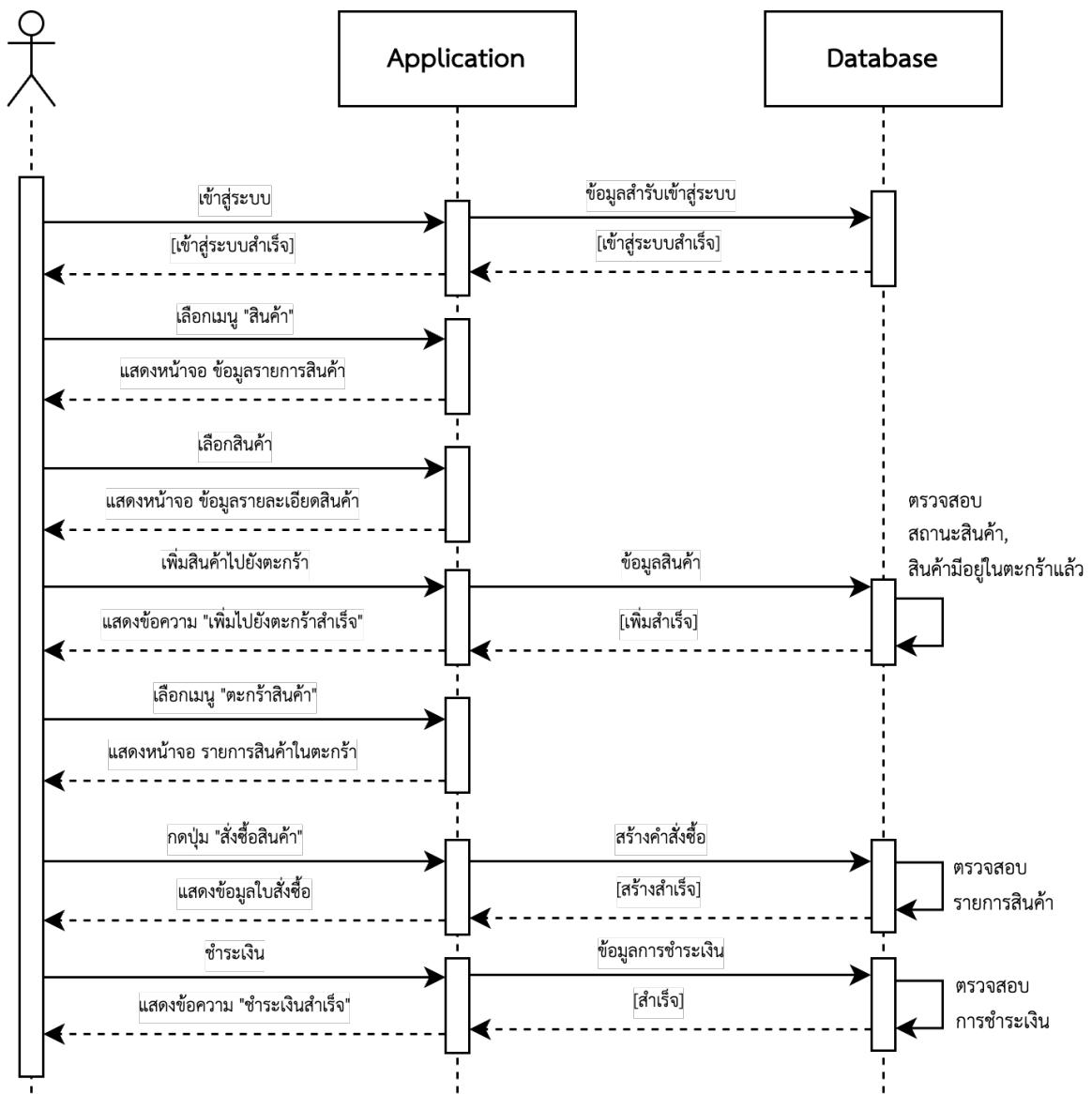
ภาพที่ 15 Sequence Diagram แก้ไขพนักงาน

3.1.3.4.1 ลบพนักงาน



ภาพที่ 16 Sequence Diagram ลบพนักงาน

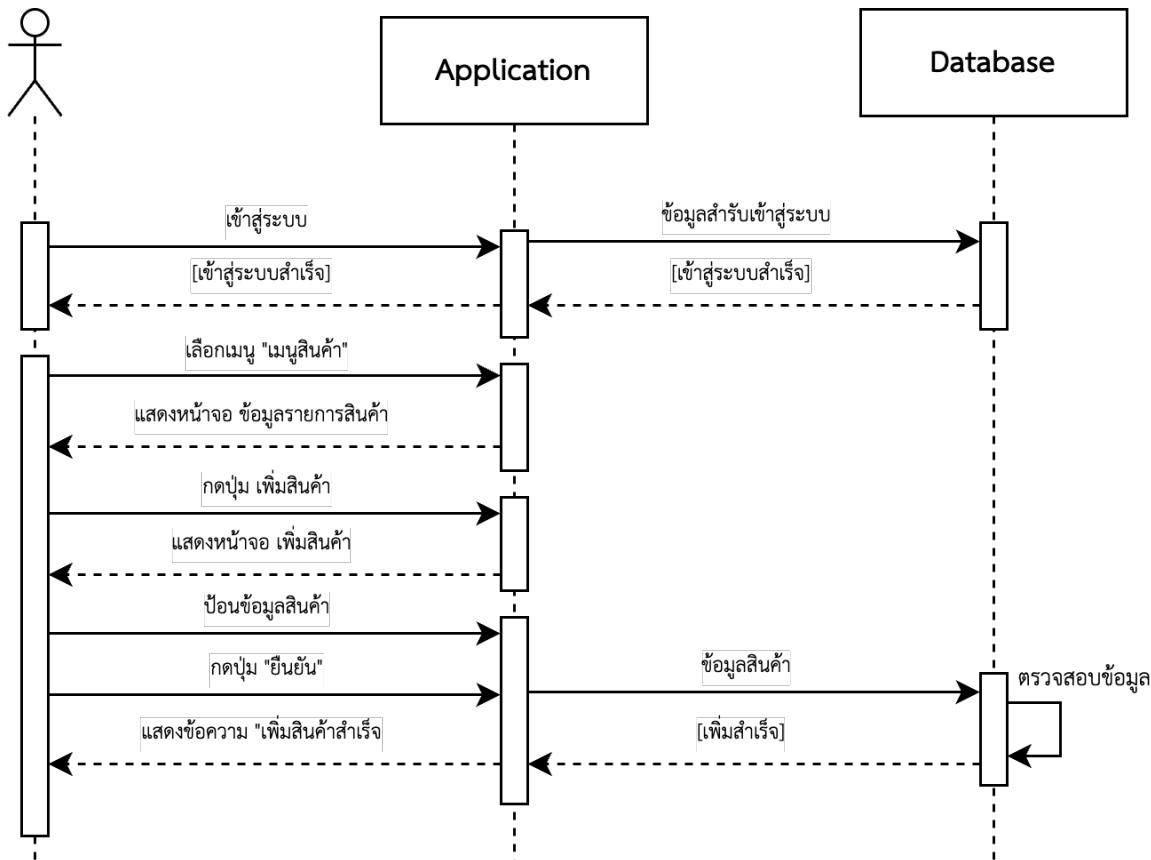
3.1.3.5 Sequence diagram ชื่อสินค้า



ภาพที่ 17 Sequence Diagram ชื่อสินค้า

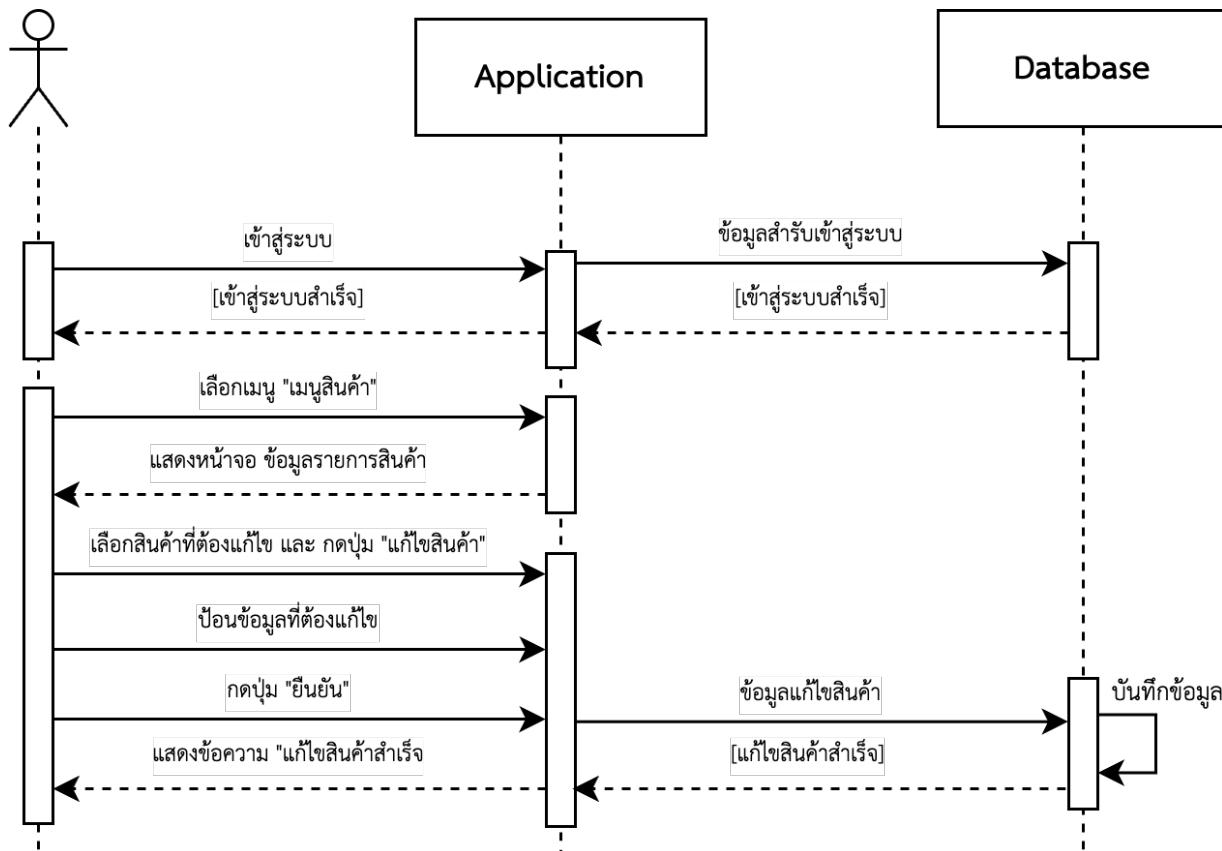
3.1.3.6 Sequence diagram จัดการสินค้า

3.1.3.6.1 เพิ่มสินค้า



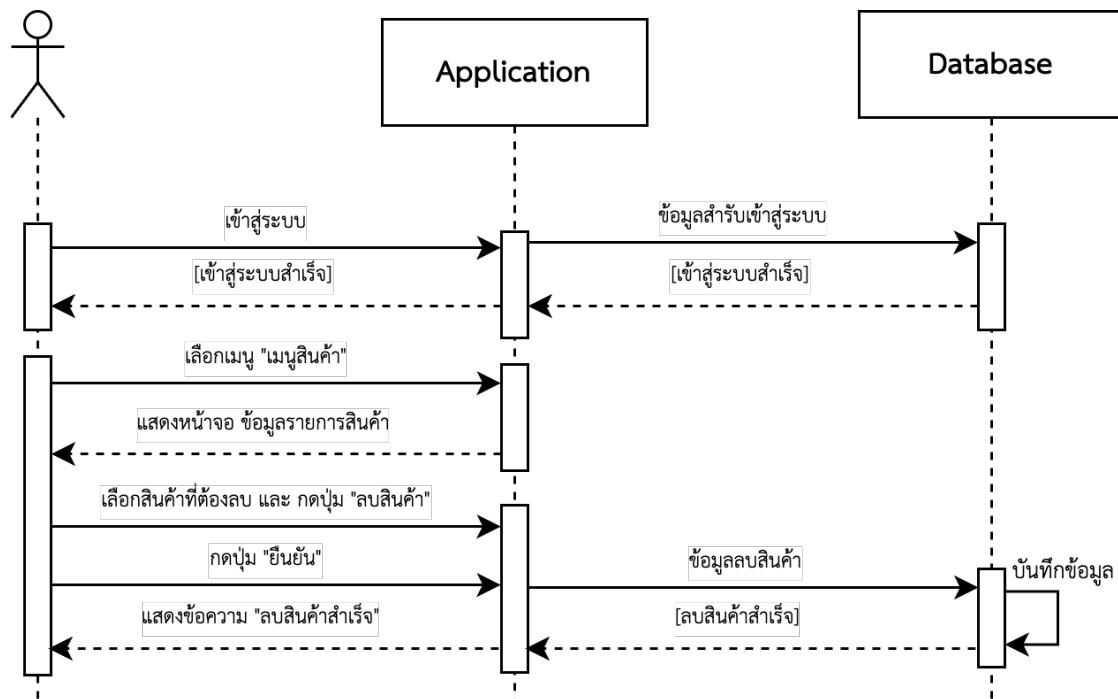
ภาพที่ 18 Sequence Diagram เพิ่มสินค้า

3.1.3.6.2 แก้ไขสินค้า



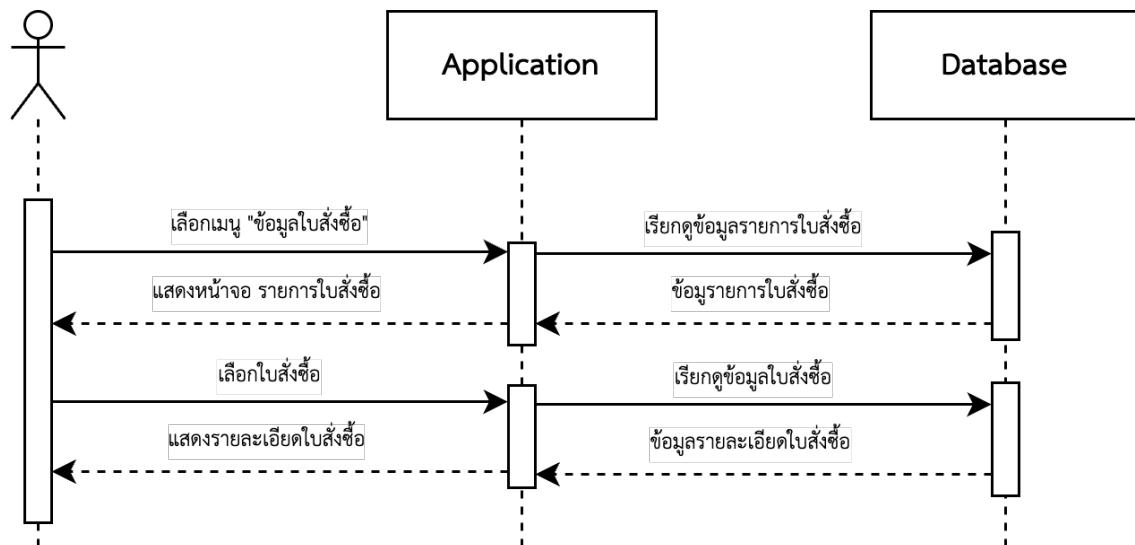
ภาพที่ 19 Sequence Diagram แก้ไขสินค้า

3.1.3.6.3 ลบสินค้า



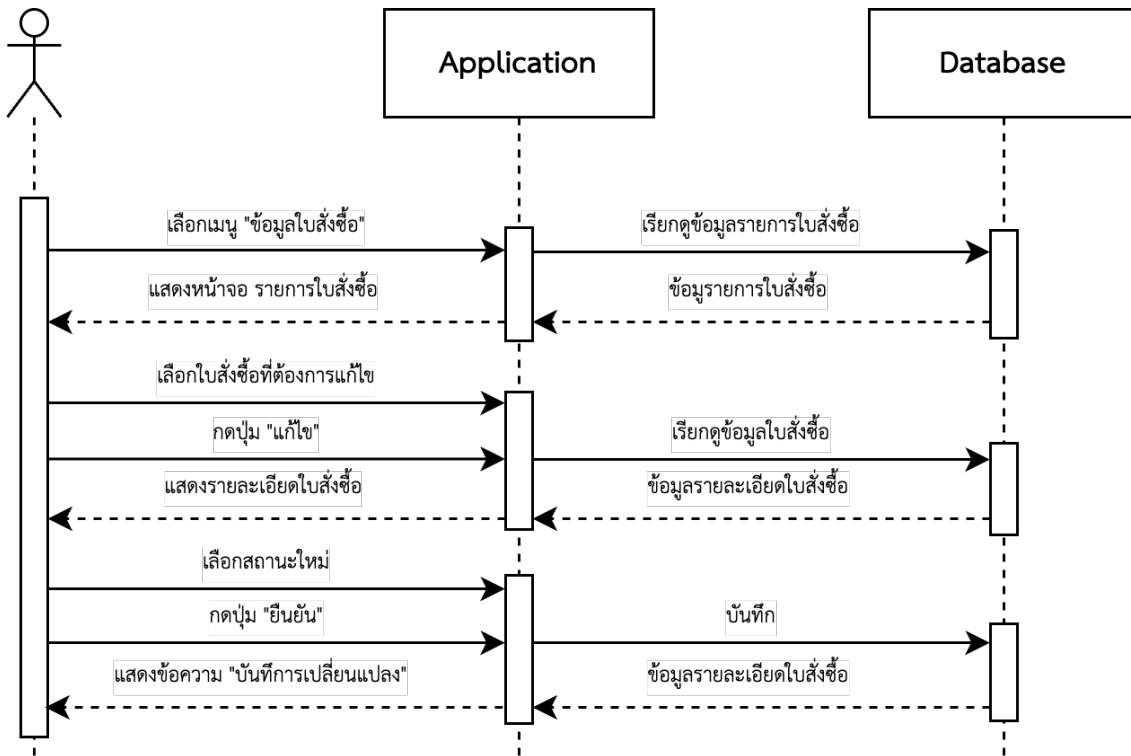
ภาพที่ 20 Sequence Diagram ลบสินค้า

3.1.3.7 Sequence diagram ข้อมูลใบสั่งซื้อ



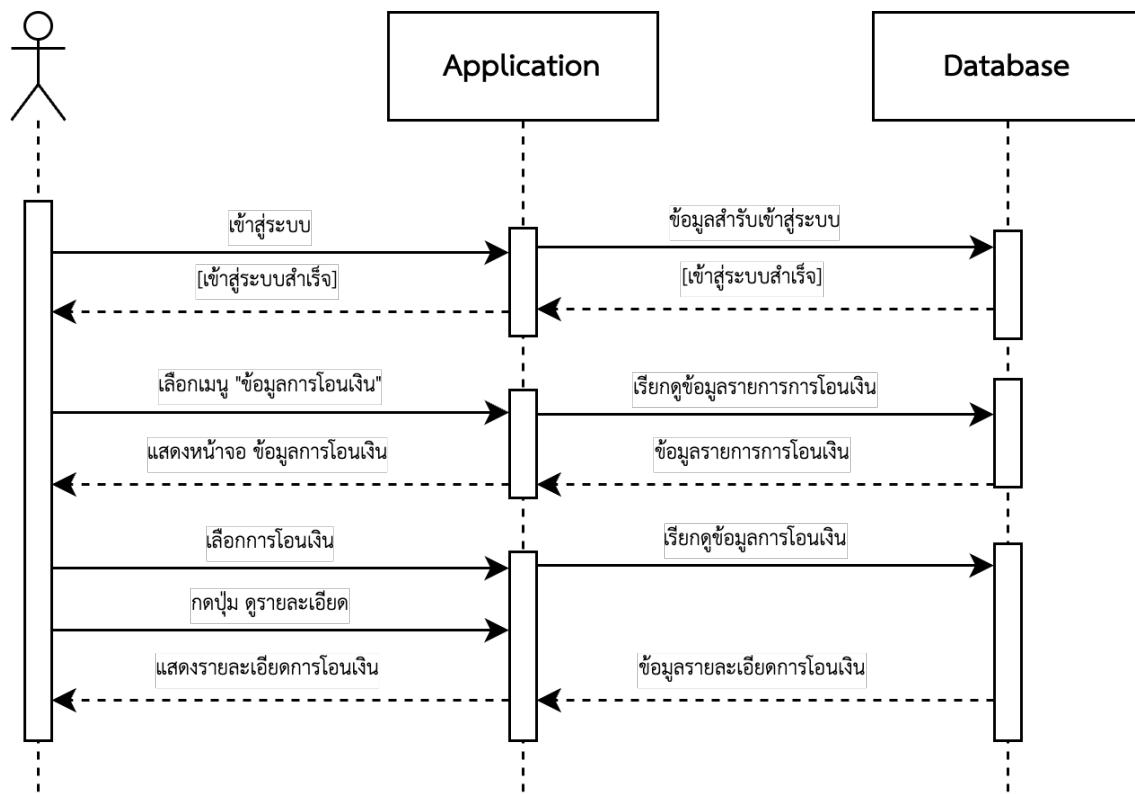
ภาพที่ 21 Sequence Diagram ข้อมูลใบสั่งซื้อ

Sequence diagram ปรับปรุงสถานะใบสั่งซื้อ



ภาพที่ 22 Sequence Diagram ปรับปรุงสถานะใบสั่งซื้อ

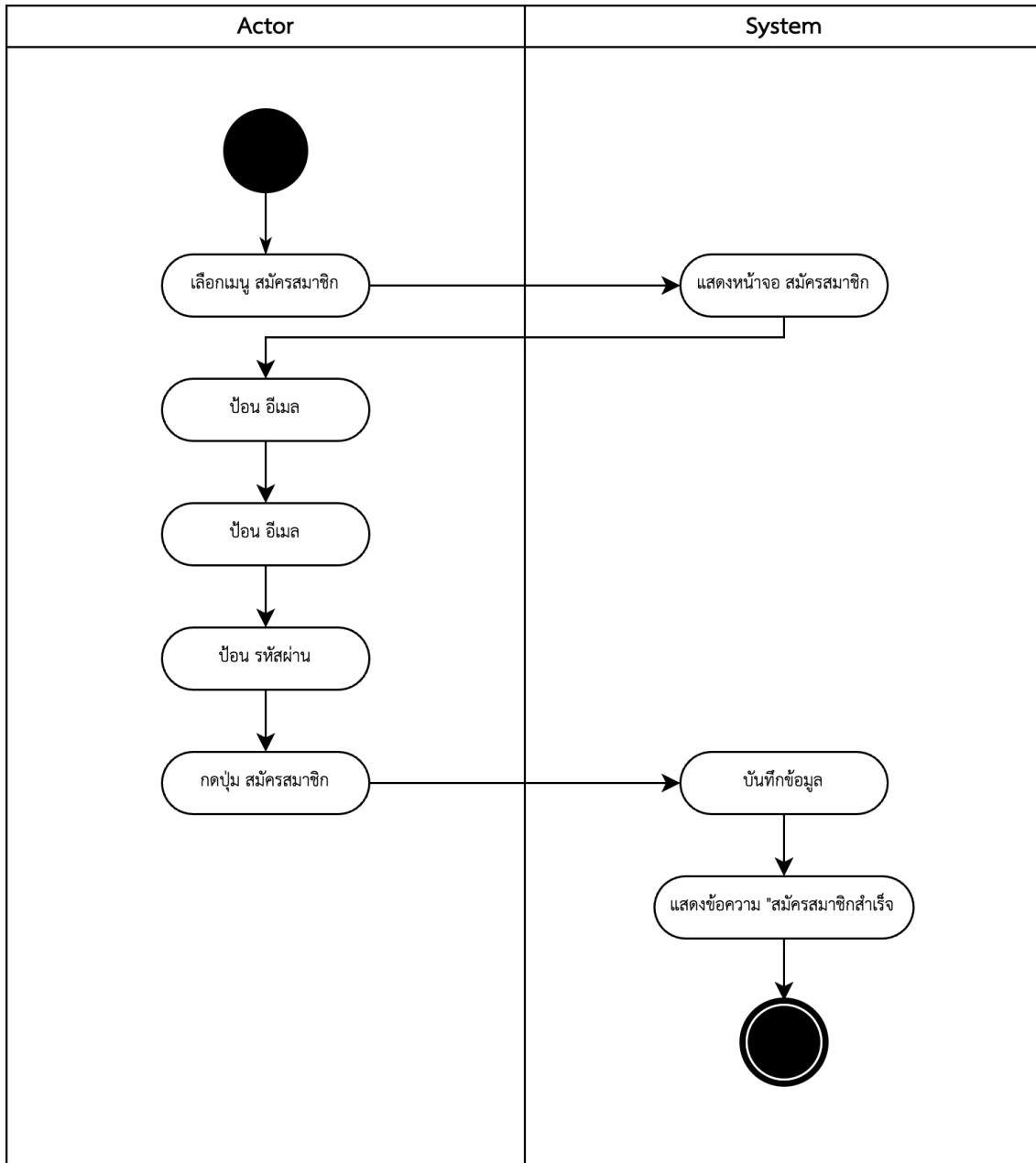
Sequence diagram ข้อมูลการโอนเงิน



ภาพที่ 23 Sequence Diagram ข้อมูลการโอนเงิน

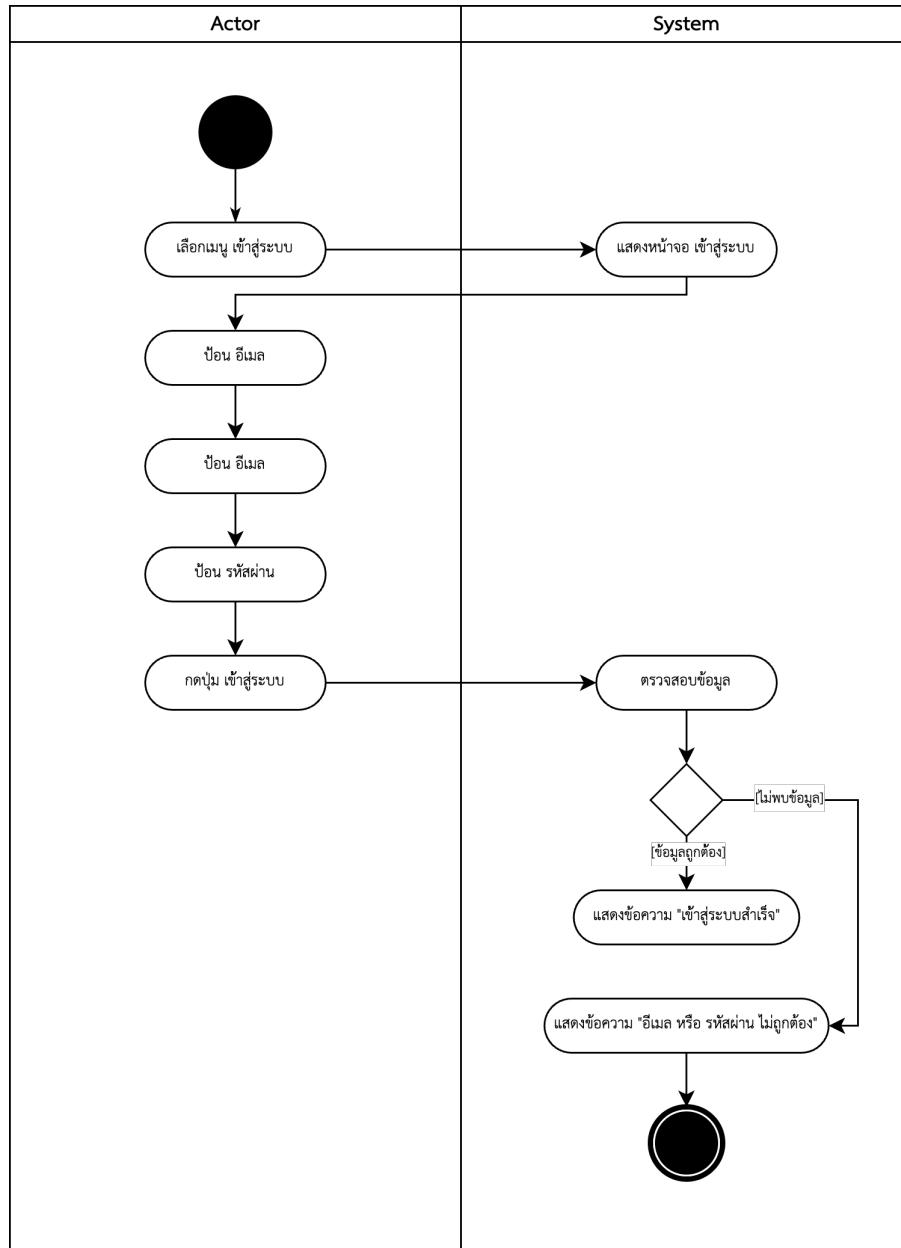
3.1.4 Activity Diagram

3.1.4.1 Activity Diagram สมัครสมาชิก



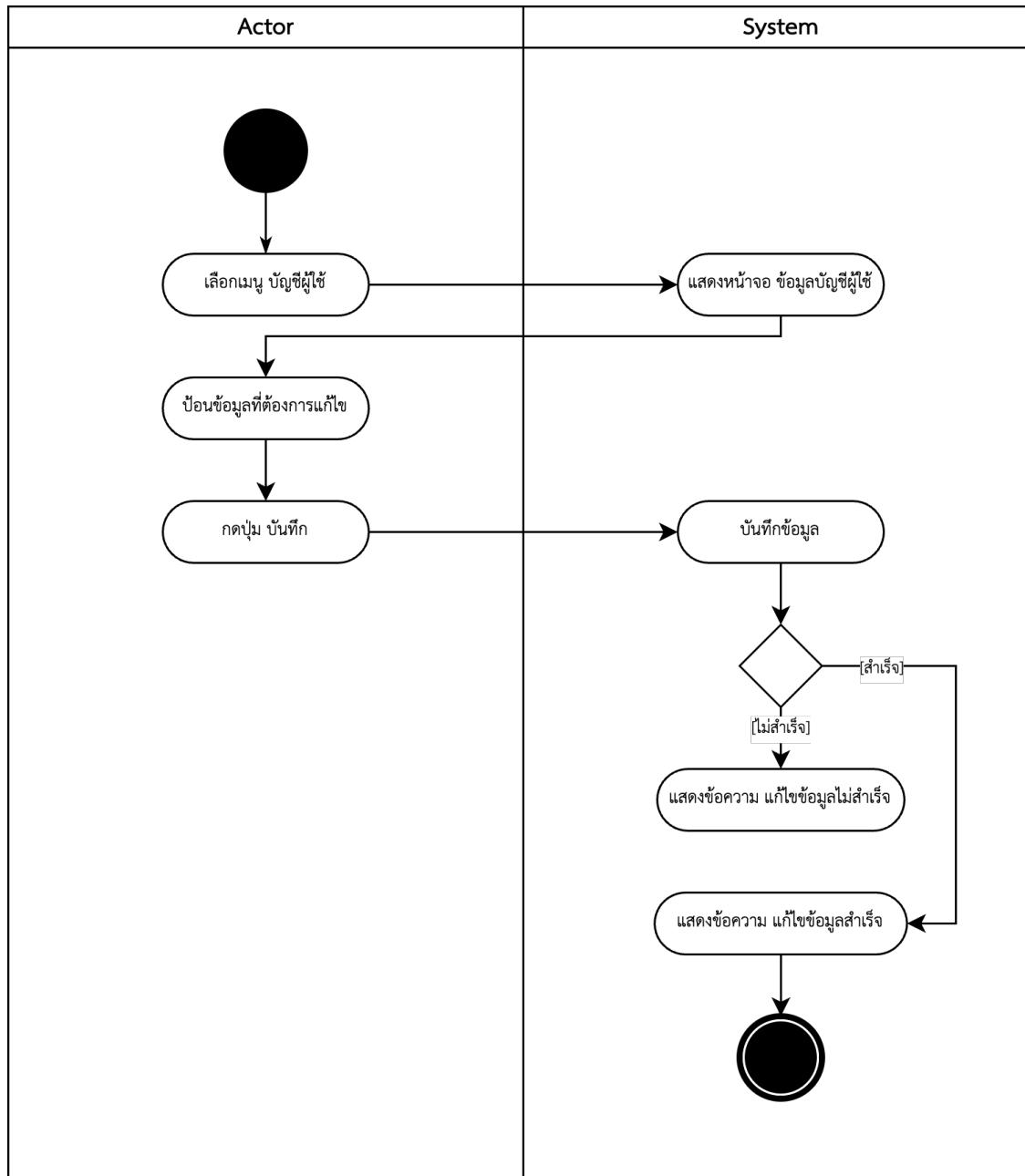
ภาพที่ 24 Activity Diagram สมัครสมาชิก

3.1.4.2 Activity Diagram เข้าสู่ระบบ



ภาพที่ 25 Activity Diagram เข้าสู่ระบบ

3.1.4.3 Activity Diagram แก้ไขบัญชีผู้ใช้

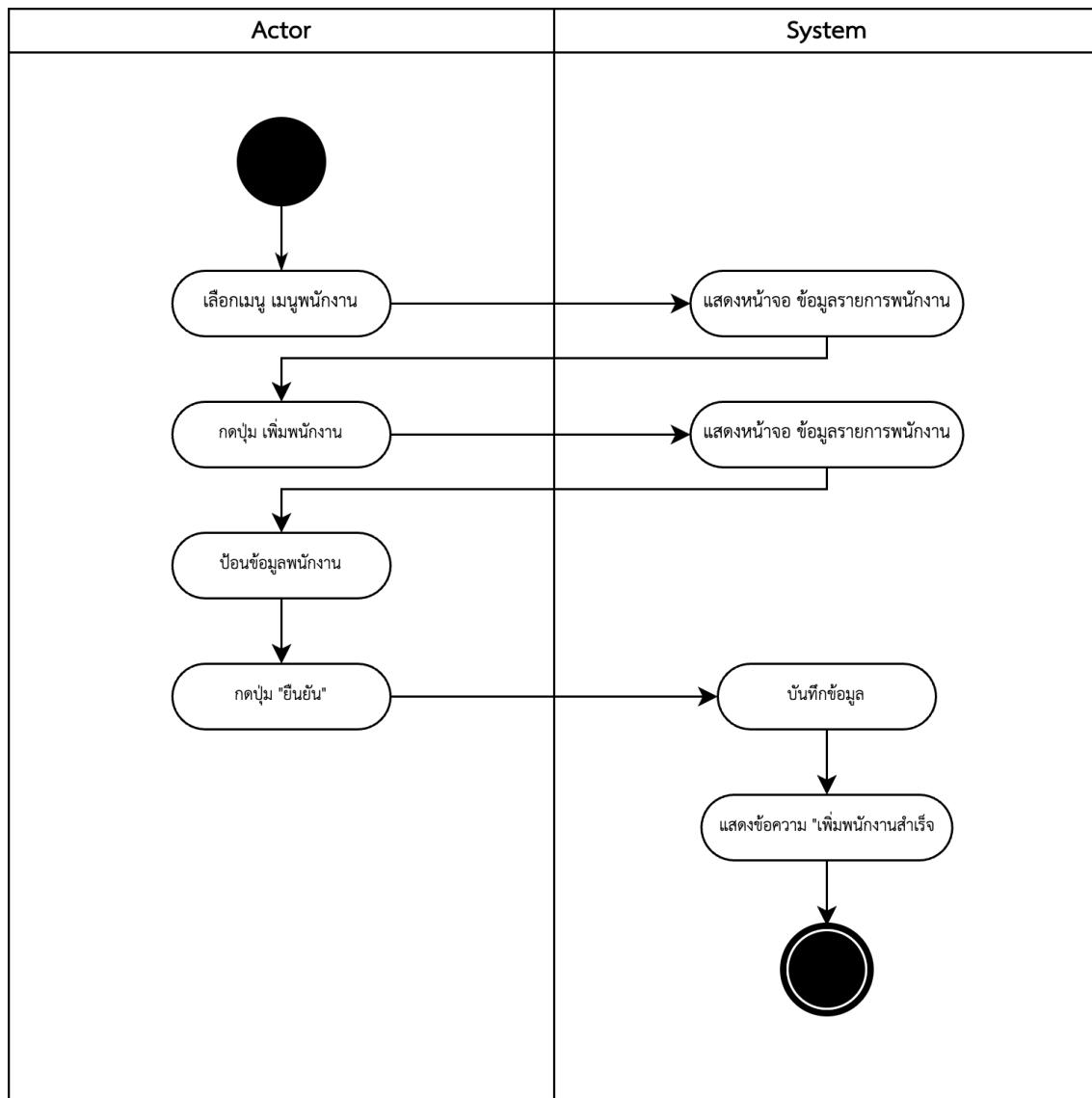


ภาพ

ภาพที่ 26 Activity Diagram แก้ไขบัญชีผู้ใช้

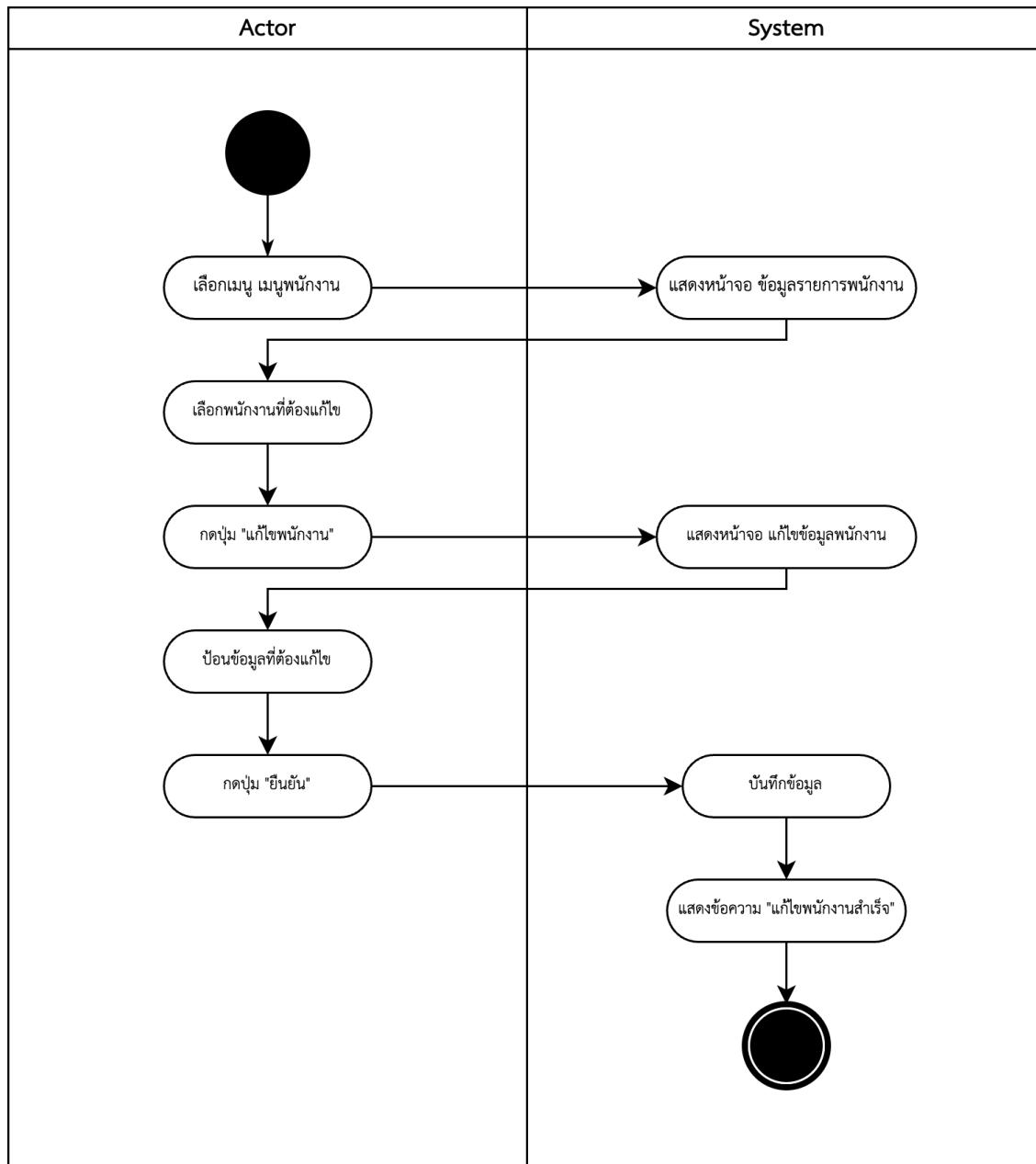
3.1.4.4 Activity Diagram จัดการพนักงาน

3.1.4.4.1 Activity Diagram เพิ่มพนักงาน



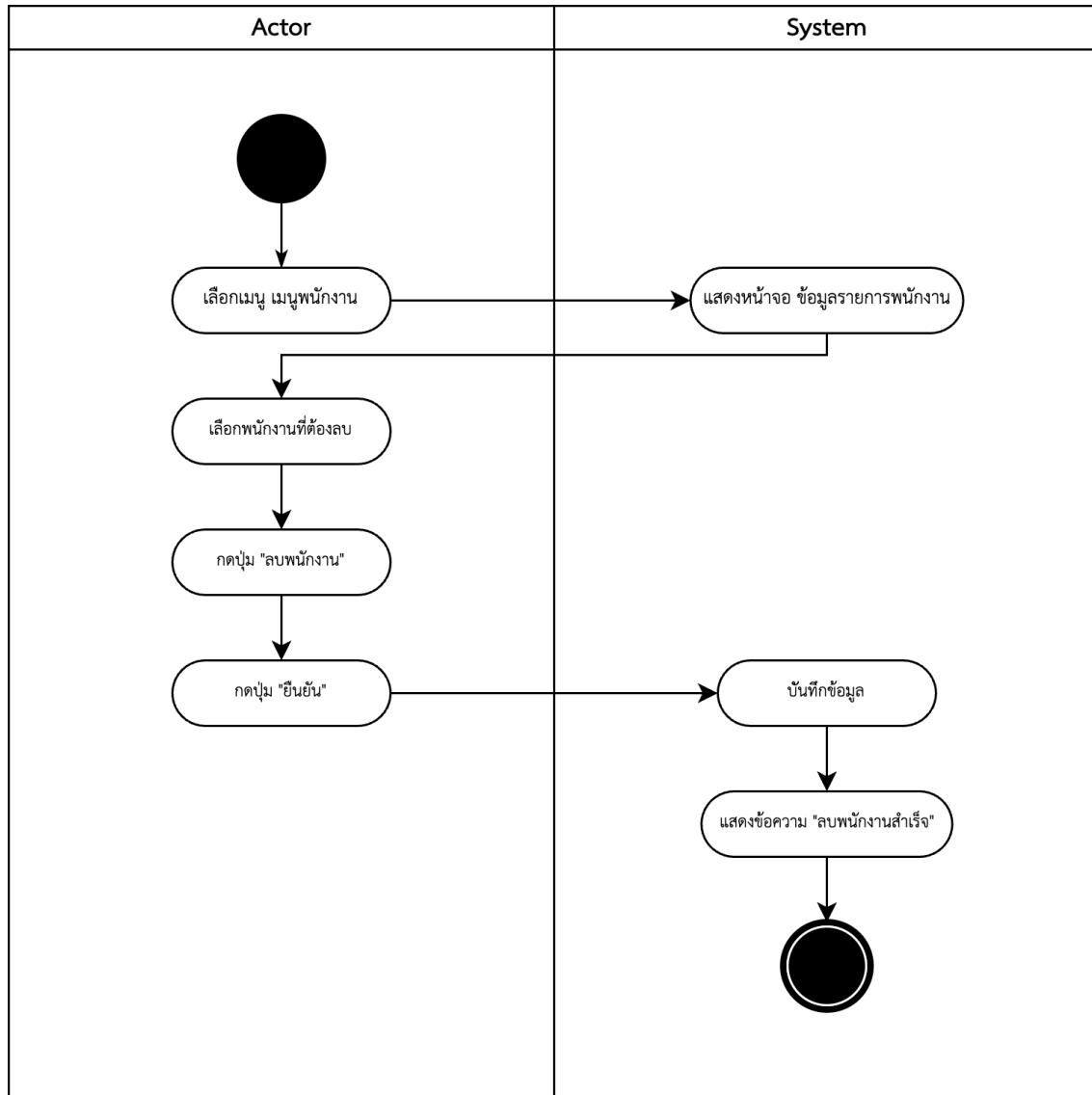
ภาพที่ 27 Activity Diagram เพิ่มพนักงาน

3.1.4.4.2 Activity Diagram แก้ไขพนักงาน



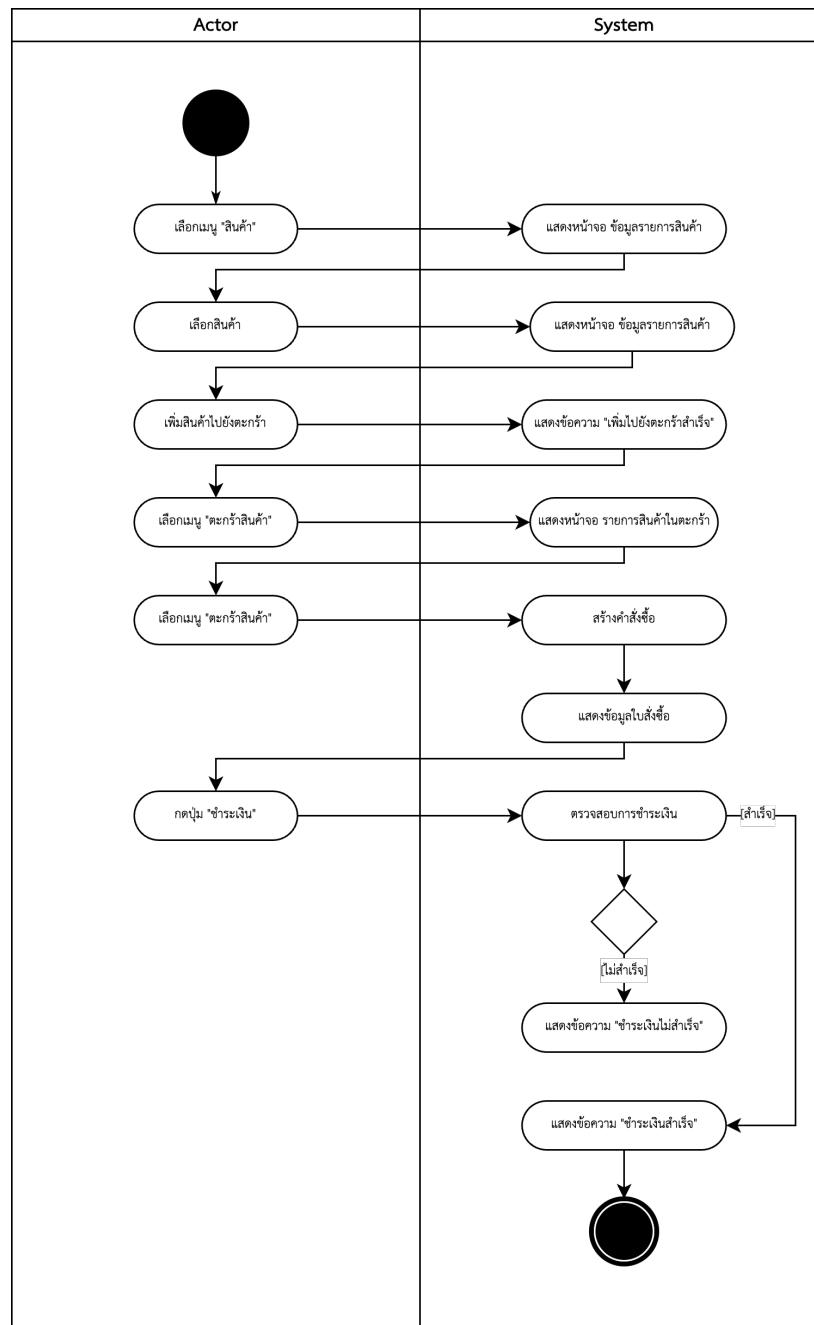
ภาพที่ 28 Activity Diagram แก้ไขพนักงาน

3.1.4.4.3 Activity Diagram ลบพนักงาน



ภาพที่ 29 Activity Diagram ลบพนักงาน

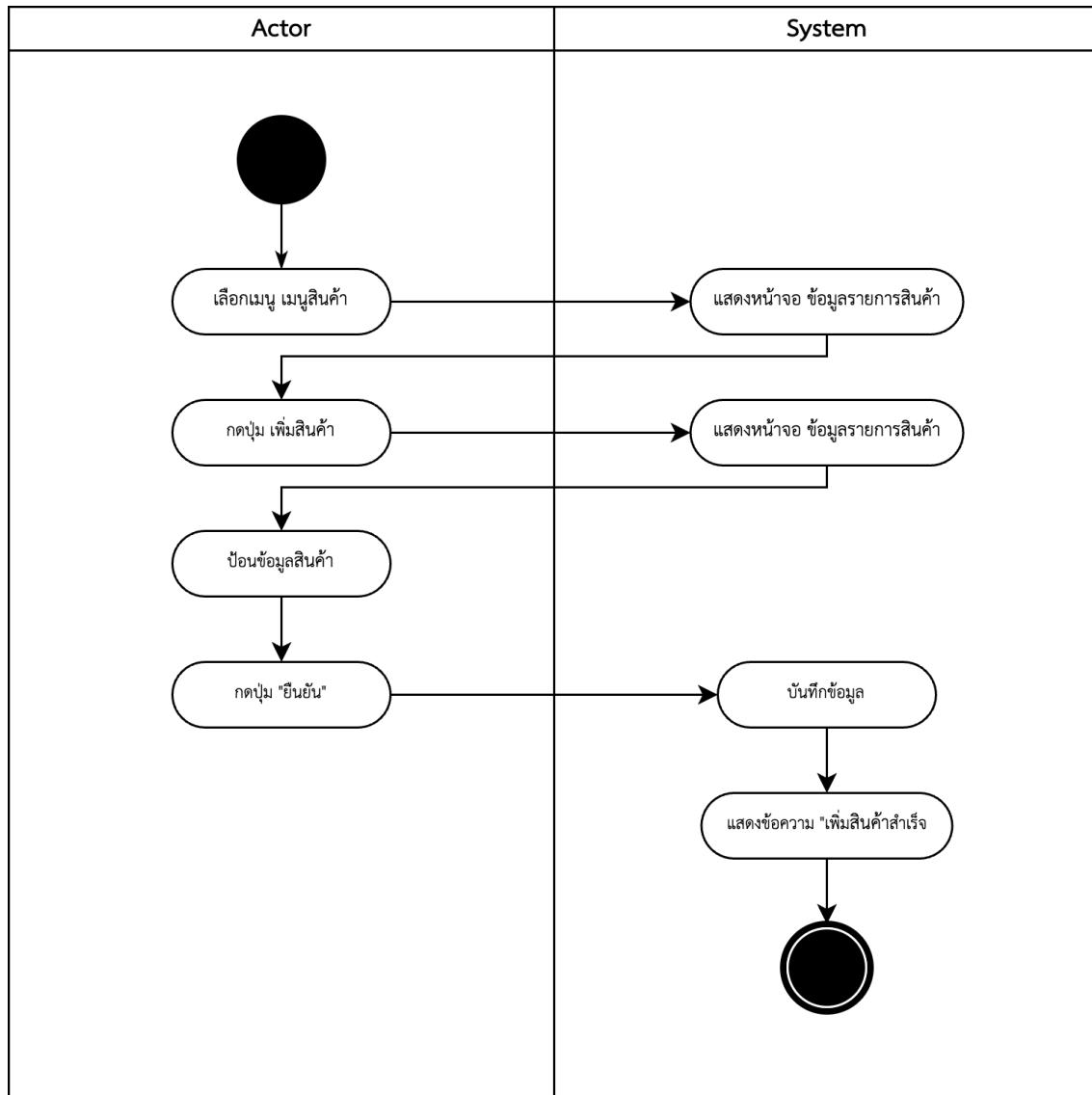
3.1.4.5 Activity Diagram ชื่อสินค้า



ภาพที่ 30 Activity Diagram ชื่อสินค้า

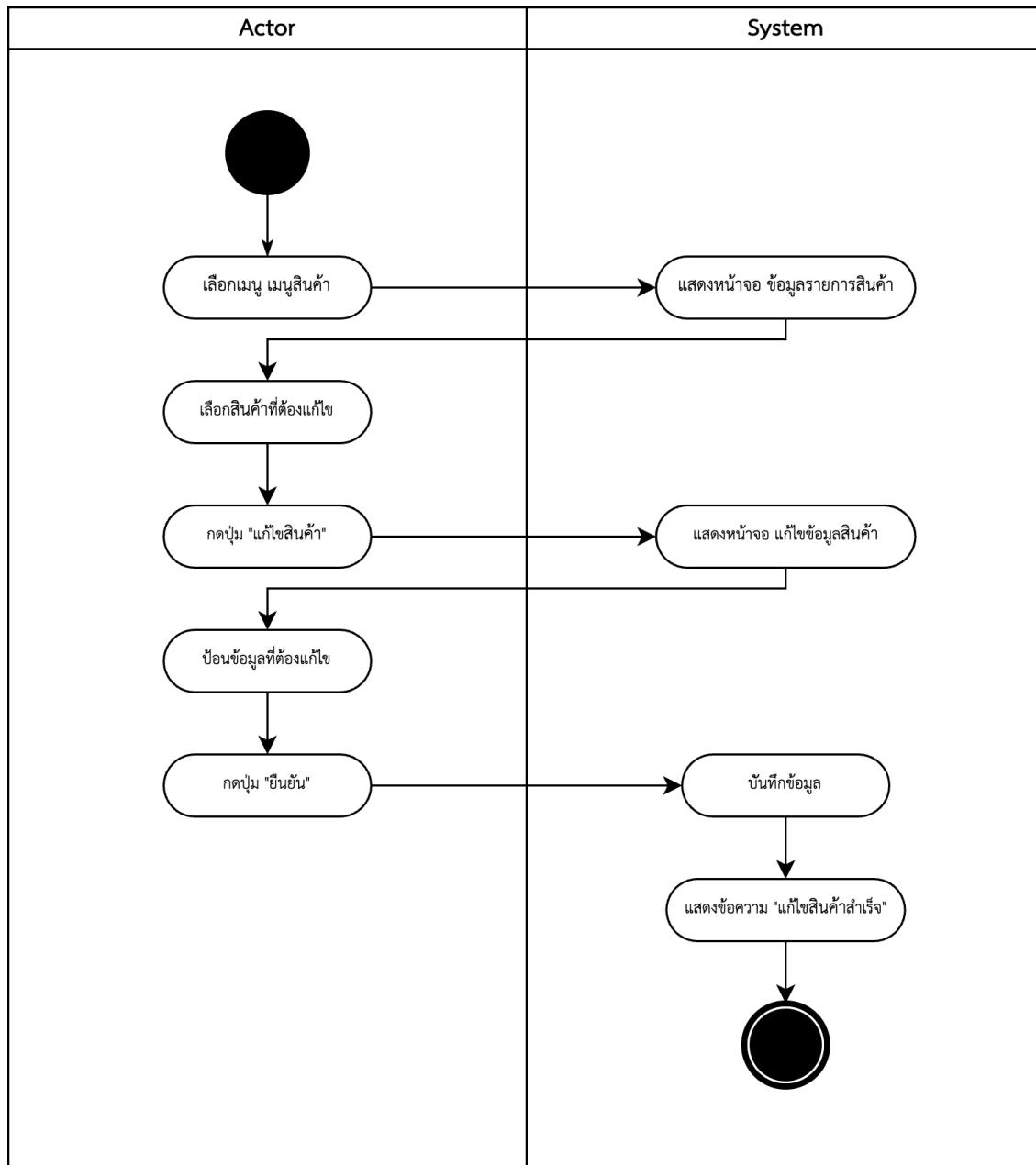
3.1.4.6 Activity Diagram จัดการสินค้า

3.1.4.6.1 Activity Diagram เพิ่มสินค้า



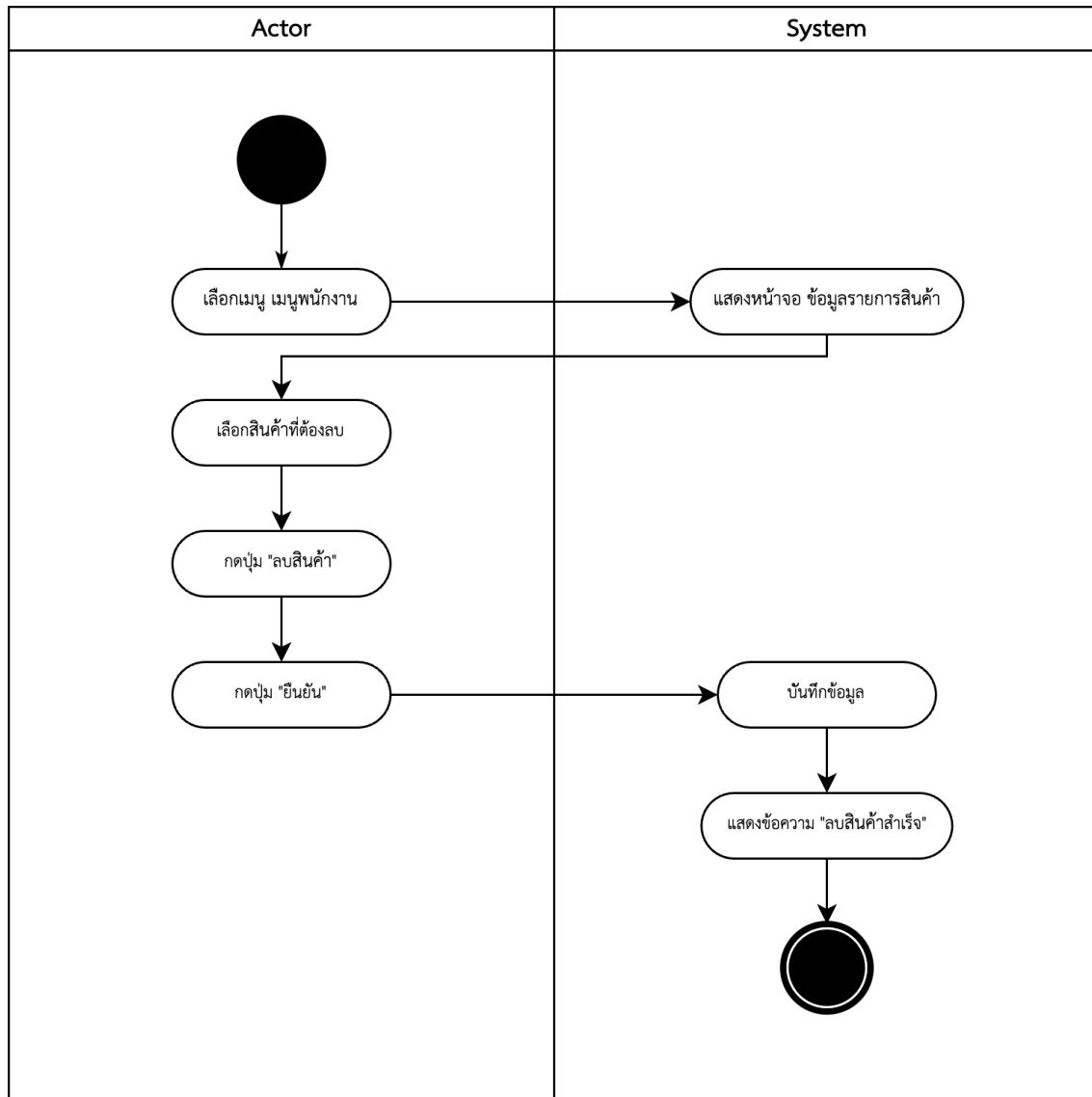
ภาพที่ 31 Activity Diagram เพิ่มสินค้า

3.1.4.6.2 Activity Diagram แก้ไขสินค้า



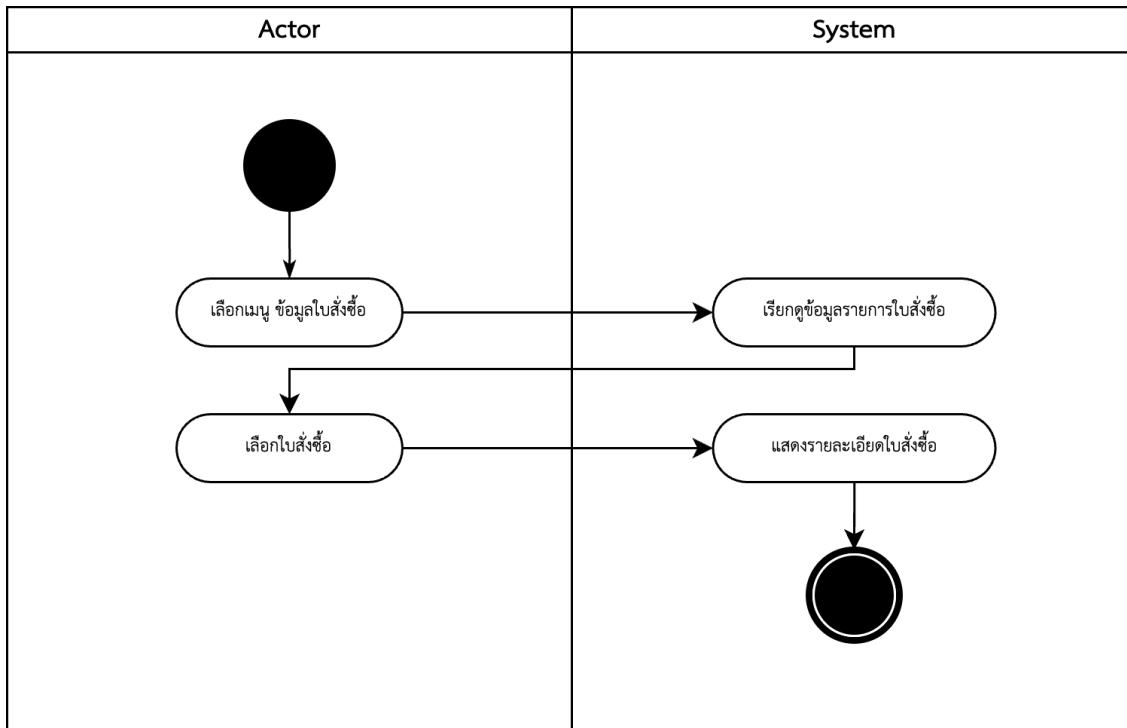
ภาพที่ 32 Activity Diagram แก้ไขสินค้า

3.1.4.6.3 Activity Diagram ลบสินค้า



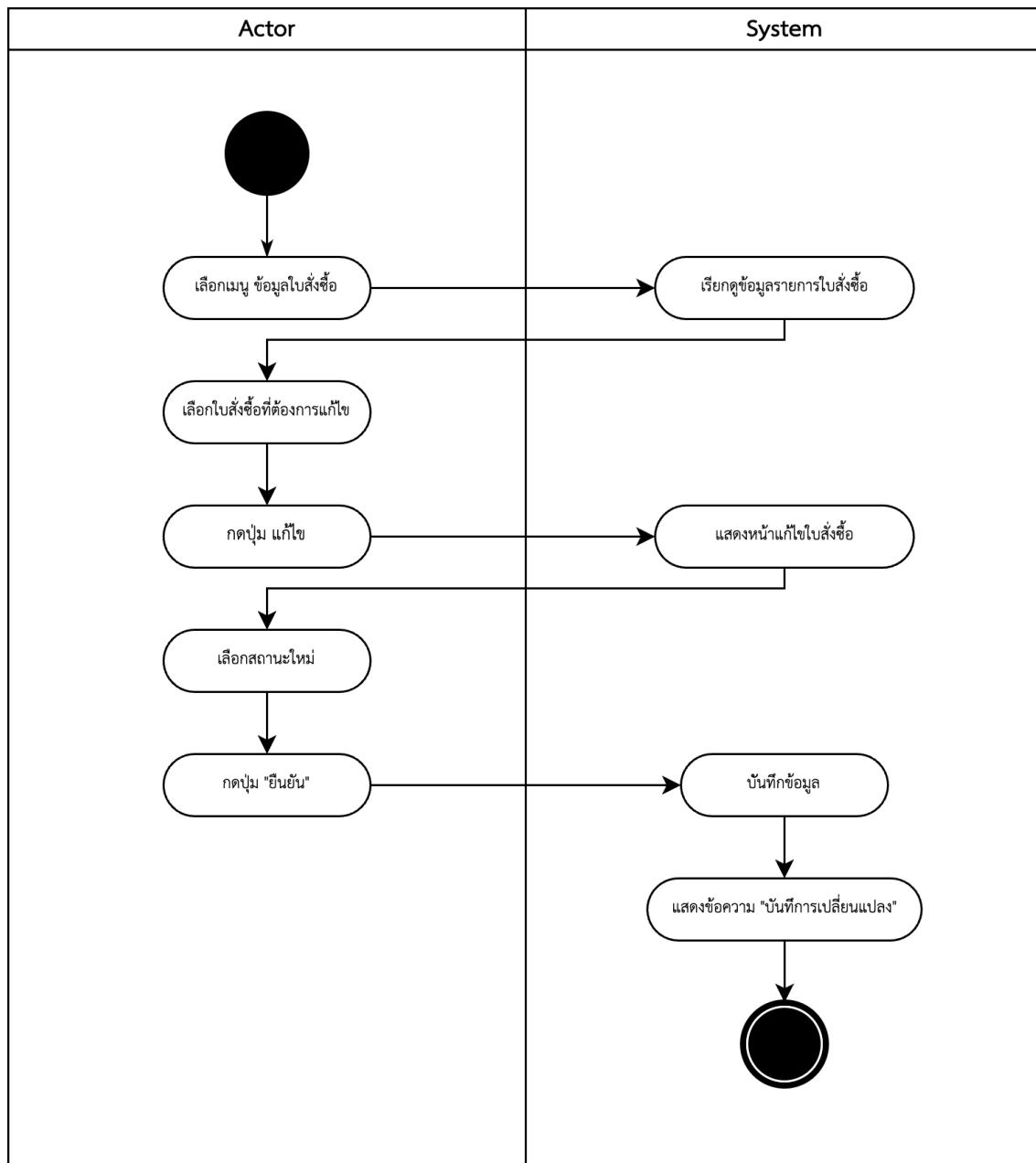
ภาพที่ 33 Activity Diagram ลบสินค้า

3.1.4.7 Activity Diagram ข้อมูลใบสั่งซื้อ



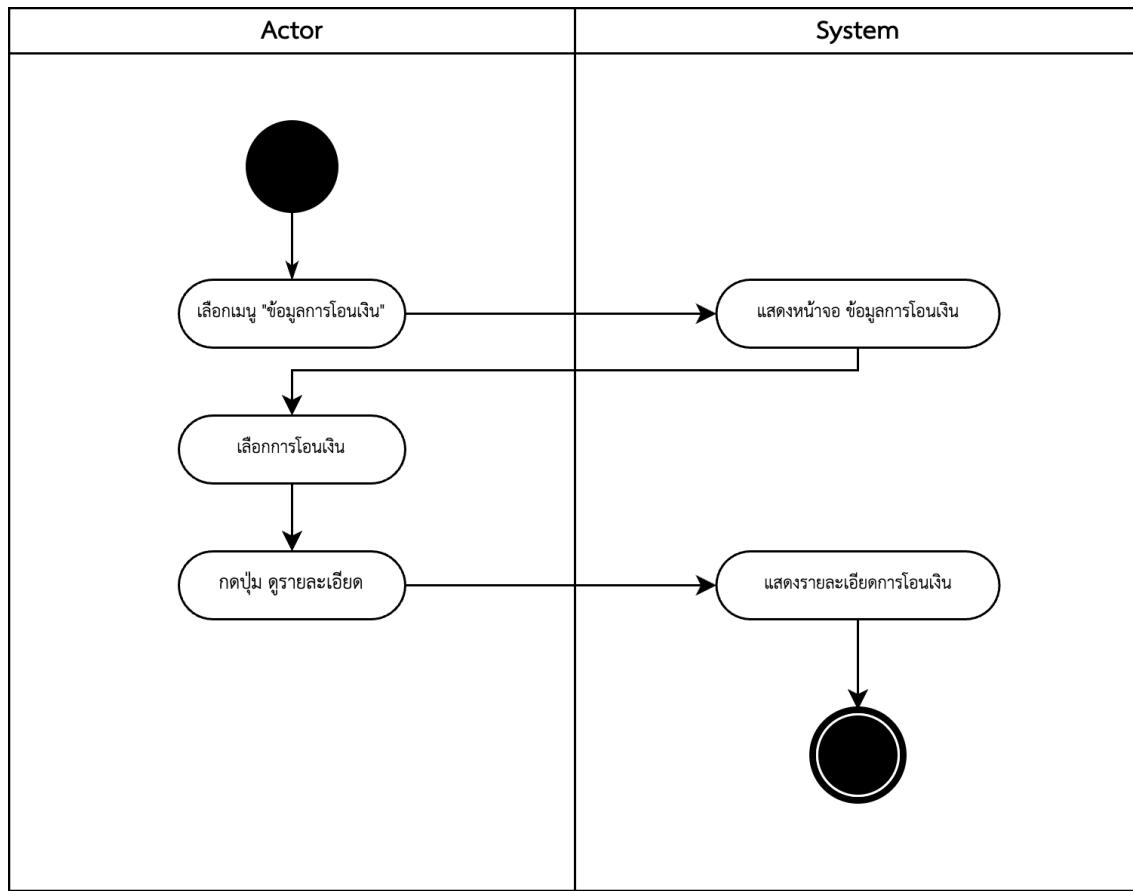
ภาพที่ 34 Activity Diagram ข้อมูลใบสั่งซื้อ

3.1.4.8 Activity Diagram ปรับปรุงสถานะใบสั่งซื้อ



ภาพที่ 35 Activity Diagram ปรับปรุงสถานะใบสั่งซื้อ

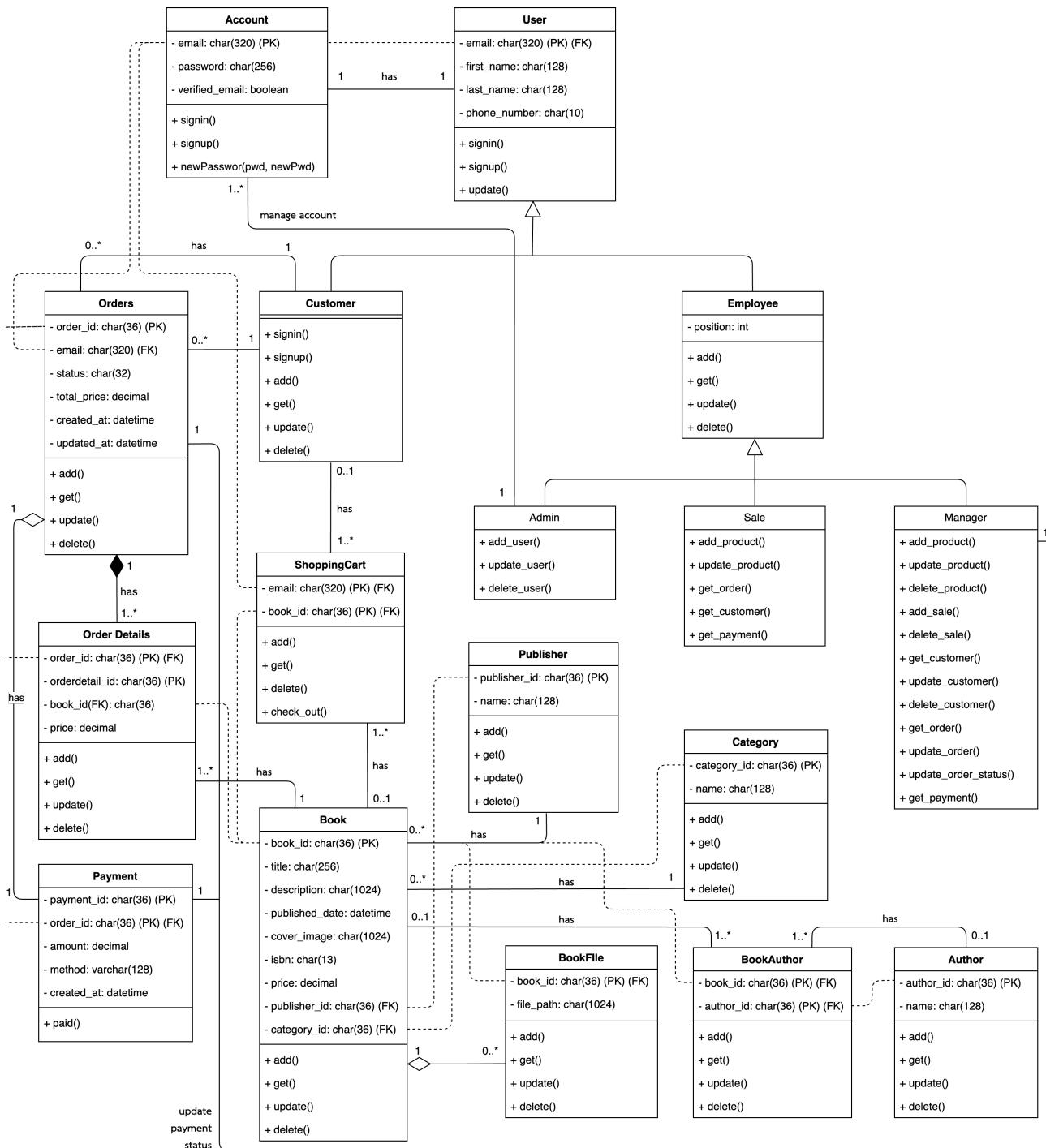
3.1.4.9 Activity Diagram ข้อมูลการโอนเงิน



ภาพที่ 36 Activity Diagram ข้อมูลการโอนเงิน

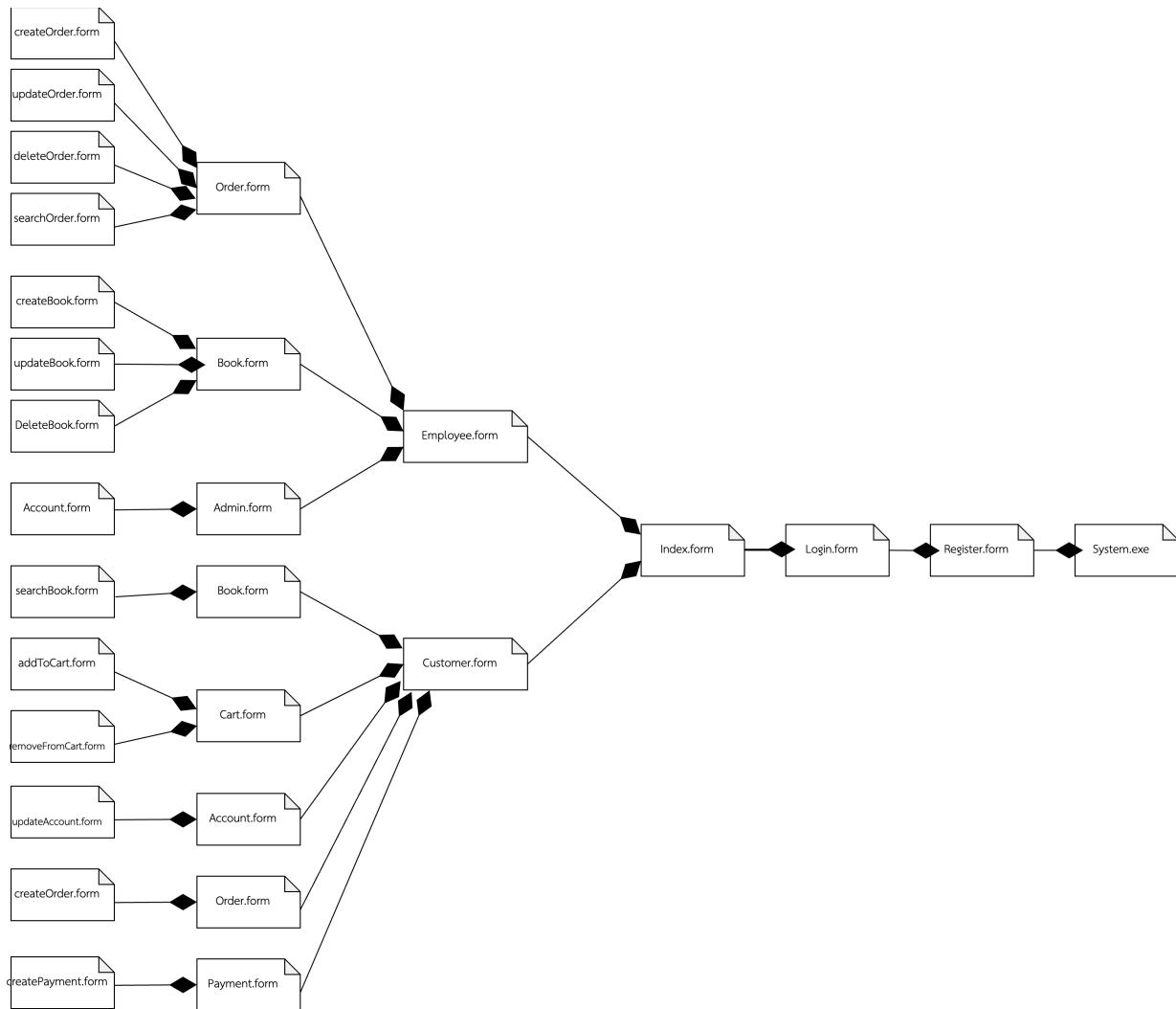
3.2 การออกแบบระบบ

3.2.1 Class Diagram



ภาพที่ 37 Class Diagram

3.2.2 Presentation Logic Subsystem



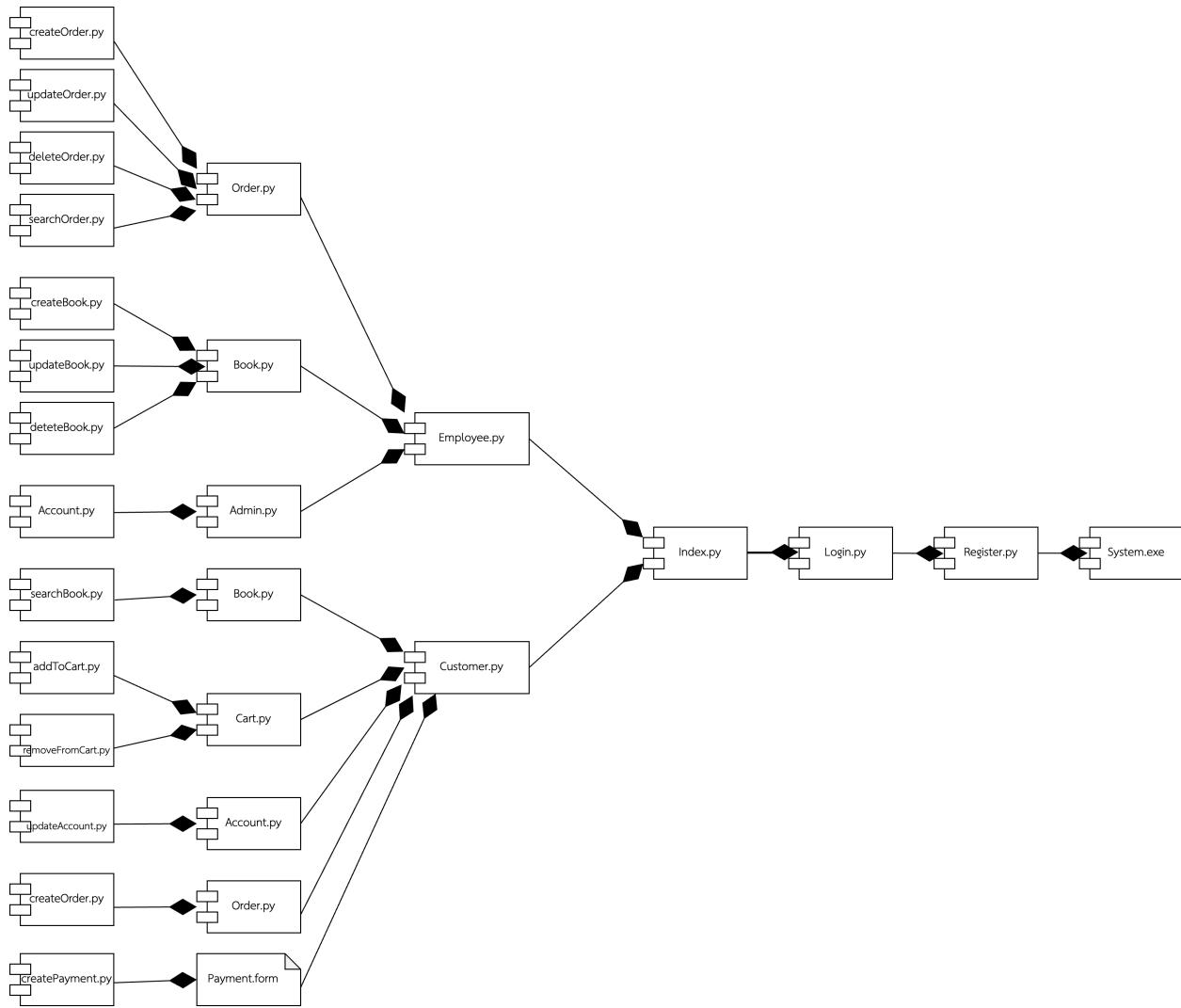
ภาพที่ 38 Presentation Logic Subsystem

แสดง User Interface เป็น Form ย่อยจำนวน 27 Form ทั้งหมดอยู่ใน Index.form

1. Customer.form เป็น User Interface ในส่วนของการจัดการลูกค้าซึ่งมี Form ย่อยได้แก่
 - 1.1 Cart.form เป็น User Interface จัดการสินค้าในตะกร้า มีระบบย่อ丫头ดังนี้
 - 1.1.1 addToCart.form เป็น User Interface เพิ่มสินค้าไปยังตะกร้า
 - 1.1.2 removeFromCart.form เป็น User Interface ลบสินค้าไปยังตะกร้า
 - 1.2 Account.form เป็น User Interface จัดการบัญชีผู้ใช้ มีระบบย่อ丫头ดังนี้
 - 1.2.1 updateAccount.form เป็น User Interface อัพเดทบัญชีผู้ใช้
 - 1.3 Order.form เป็น User Interface จัดการใบสั่งซื้อ

- 1.3.1 createOrder.form เป็น User Interface สร้างคำสั่งซื้อ
- 1.4 Payment.form เป็น User Interface จัดการชำระเงิน
 - 1.4.1 createOrder.form เป็น User Interface สร้างคำสั่งซื้อ
- 1.5 Book.form เป็น User Interface จัดการหนังสือ
 - 1.5.1 searchBook.form เป็น User Interface ค้นหาหนังสือ
- 2. Employee.form เป็น User Interface ในส่วนของการจัดการพนักงานซึ่งมี Form ย่อยได้แก่
 - 2.1 Admin.form เป็น User Interface จัดการผู้ใช้ระบบ มีระบบย่อตั้งนี้
 - 2.1.1 Account.form เป็น User Interface จัดการบัญชีผู้ใช้
 - 2.2 Book.form เป็น User Interface จัดการหนังสือ มีระบบย่อตั้งนี้
 - 2.2.1 createBook.form เป็น User Interface เพิ่มหนังสือ
 - 2.2.2 updateBook.form เป็น User Interface อัปเดทหนังสือ
 - 2.2.3 deleteBook.form เป็น User Interface ลบหนังสือ
 - 2.3 Order.form เป็น User Interface จัดการคำสั่งซื้อ มีระบบย่อตั้งนี้
 - 2.3.1 createOrder.form เป็น User Interface เพิ่มคำสั่งซื้อ
 - 2.3.2 updateOrder.form เป็น User Interface อัปเดตคำสั่งซื้อ
 - 2.3.3 deleteOrder.form เป็น User Interface ลบคำสั่งซื้อ
 - 2.3.4 searchOrder.form เป็น User Interface ค้นหาคำสั่งซื้อ

3.2.3 Working Logic Subsystem



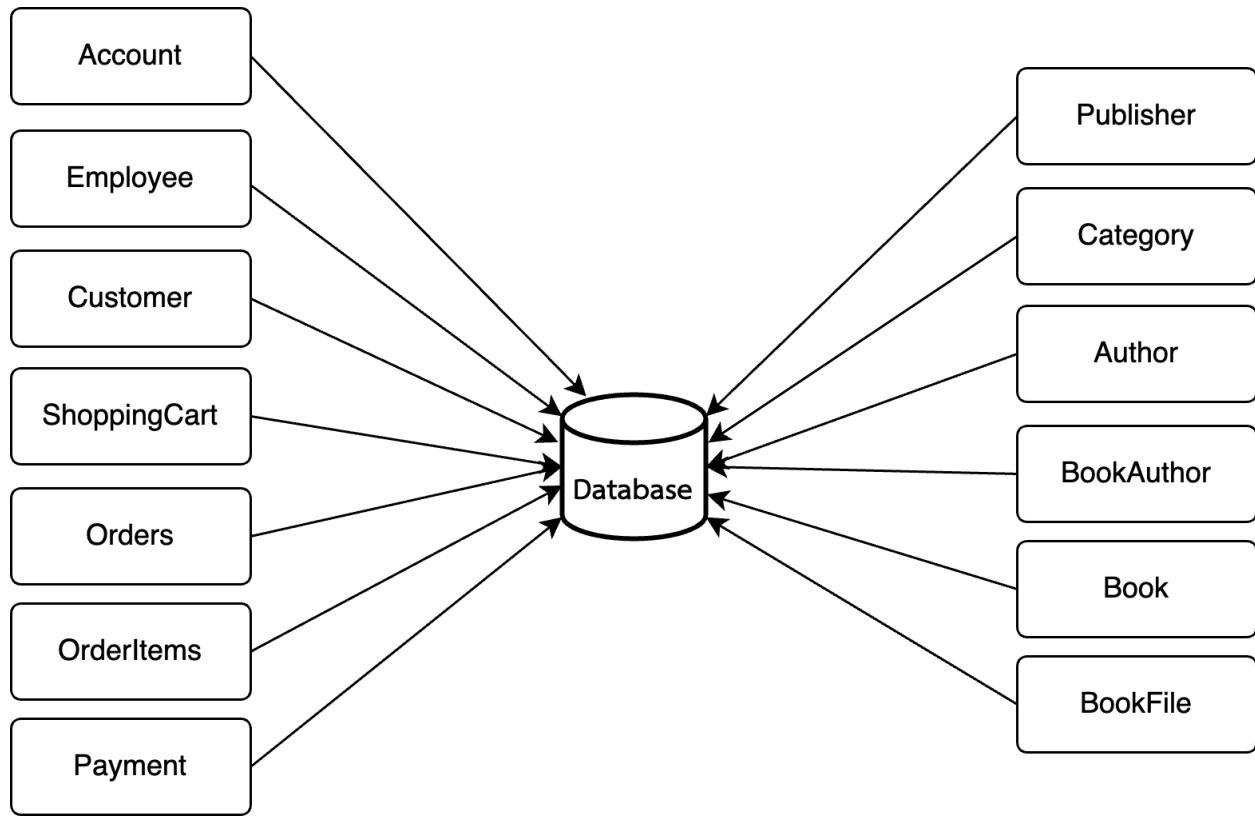
ภาพที่ 39 Working Logic Subsystem

Working Logic Subsystem แสดง โครงสร้างการทำงานของโปรแกรมย่อยต่างๆ เมื่อถูก Index เรียกใช้งาน

1. Customer.py เป็นระบบในส่วนของการจัดการลูกค้า มีระบบบัญชีแยกต่างหาก
 - 1.1 Cart.py เป็นระบบในส่วนของการจัดการสินค้าในตะกร้า มีระบบบัญชีแยกต่างนี้
 - 1.1.1 addToCart.py เป็นระบบในส่วนของการเพิ่มสินค้าไปยังตะกร้า
 - 1.1.2 removeToCart.py เป็นระบบในส่วนของการลบสินค้าไปยังตะกร้า
 - 1.2 Account.py เป็นระบบในส่วนของการจัดการบัญชีผู้ใช้ มีระบบบัญชีแยกต่างนี้
 - 1.2.1 updateAccount.py เป็นระบบในส่วนของการอัพเดทบัญชีผู้ใช้

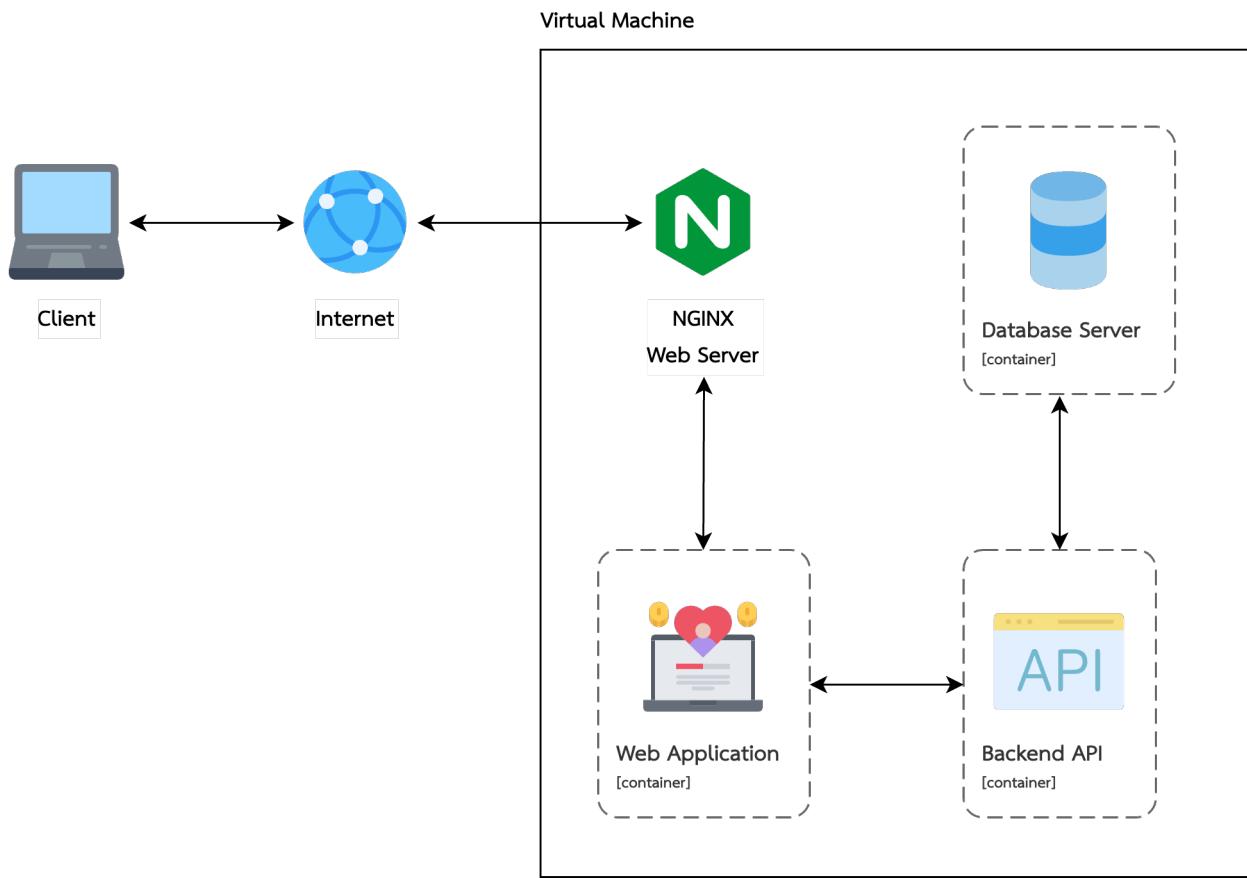
- 1.3 Order.py เป็นระบบในส่วนของการจัดการใบสั่งซื้อ
 - 1.3.1 createOrder.py เป็นระบบในส่วนของการสร้างคำสั่งซื้อ
 - 1.4 Payment.py เป็นระบบในส่วนของการจัดการชำระเงิน
 - 1.4.1 createOrder.py เป็น User Interface สร้างคำสั่งซื้อ
 - 1.5 Book.py เป็นระบบในส่วนของการจัดการหนังสือ
 - 1.5.1 searchBook.py เป็นระบบในส่วนของการค้นหาหนังสือ
2. Employee.py เป็นระบบในส่วนของการในส่วนของการจัดการพนักงานซึ่งมี Form ย่อยได้แก่
- 2.1 Admin.py เป็น User Interface จัดการผู้ใช้ระบบ มีระบบย่อยดังนี้
 - 2.1.1 Account.py เป็นระบบในส่วนของการจัดการบัญชีผู้ใช้
 - 2.2 Book.py เป็น User Interface จัดการหนังสือ มีระบบย่อยดังนี้
 - 2.2.1 createBook.py เป็นระบบในส่วนของการเพิ่มหนังสือ
 - 2.2.2 updateBook.py เป็นระบบในส่วนของการอัปเดทหนังสือ
 - 2.2.3 deleteBook.py เป็นระบบในส่วนของการลบหนังสือ
 - 2.3 Order.py เป็น User Interface จัดการคำสั่งซื้อ มีระบบย่อยดังนี้
 - 2.3.1 createOrder.py เป็นระบบในส่วนของการเพิ่มคำสั่งซื้อ
 - 2.3.2 updateOrder.py เป็นระบบในส่วนของการอัปเดตคำสั่งซื้อ
 - 2.3.3 deleteOrder.py เป็นระบบในส่วนของการลบคำสั่งซื้อ
 - 2.3.4 searchOrder.py เป็นระบบในส่วนของการค้นหาคำสั่งซื้อ

3.2.4 Database Logic Subsystem



ภาพที่ 40 Database Logic System

3.2.5 System Architecture Design



ภาพที่ 41 System Architecture Design

3.2.6 Persistent Data Design

1. ตาราง Account

Key	Attribute	Type	Size	Description
PK	email	char	320	อีเมล
	password	char	256	รหัสผ่าน
	verified_email	boolean		ยืนยันอีเมล

2. ตาราง Customer

Key	Attribute	Type	Size	Description
PK, FK	email	char	320	อีเมล
	password	char	256	รหัสผ่าน
	first_name	char	128	ชื่อ
	last_name	char	128	นามสกุล
	phone_number	char	10	เบอร์โทรศัพท์

3. ตาราง Employee

Key	Attribute	Type	Size	Description
PK, FK	email	char	320	อีเมล
	password	char	256	รหัสผ่าน
	first_name	char	128	ชื่อ
	last_name	char	128	นามสกุล
	phone_number	char	10	เบอร์โทรศัพท์
	position	Int		บทบาท เช่น 1.admin, 2.manager

4. ตาราง Book

Key	Attribute	Type	Size	Description
PK	book_id	char	36	รหัสหนังสือ
	title	char	256	ชื่อหนังสือ
	description	char	1024	คำอธิบาย
	published_date	datetime		วันที่เผยแพร่
	cover_image	char	1024	รูปภาพหน้าปกหนังสือ
	isbn	char	13	บาร์โค้ด
	price	decimal	10, 2	ราคา
FK	publisher_id	char	36	รหัสสำนักพิมพ์
FK	category_id	char	36	รหัสประเภท

5. ตาราง Book File

Key	Attribute	Type	Size	Description
PK, FK	book_id	char	36	รหัสหนังสือ
	file_path	char	1024	ที่อยู่ไฟล์

6. ตาราง Publisher

Key	Attribute	Type	Size	Description
PK	publisher_id	char	36	รหัสสำนักพิมพ์
	name	char	128	ชื่อสำนักพิมพ์

7. ตาราง Category

Key	Attribute	Type	Size	Description
PK	category_id	char	36	รหัสประเภท
	name	char	128	ชื่อประเภท

8. ตาราง Author

Key	Attribute	Type	Size	Description
PK	author_id	char	36	รหัสผู้แต่ง
	name	char	128	ชื่อผู้แต่ง

9. ตาราง BookAuthor

Key	Attribute	Type	Size	Description
PK, FK	book_id	char	36	รหัสหนังสือ
PK, FK	author_id	char	36	รหัสผู้แต่ง

10. ตาราง Shopping Cart

Key	Attribute	Type	Size	Description
PK, FK	email	char	320	อีเมล
PK, FK	book_id	char	36	รหัสหนังสือ

11. ตาราง Order

Key	Attribute	Type		Description
PK	order_id	char	36	รหัสใบสั่งซื้อ
FK	email	char	320	อีเมล
	status	char	32	สถานะใบสั่งซื้อ
	total_price	decimal	10, 2	จำนวนเงิน
	created_at	datetime		วันเวลาที่สั่งซื้อ
	updated_at	datetime		วันเวลาที่ปรับปรุง

12. ตาราง Order Details

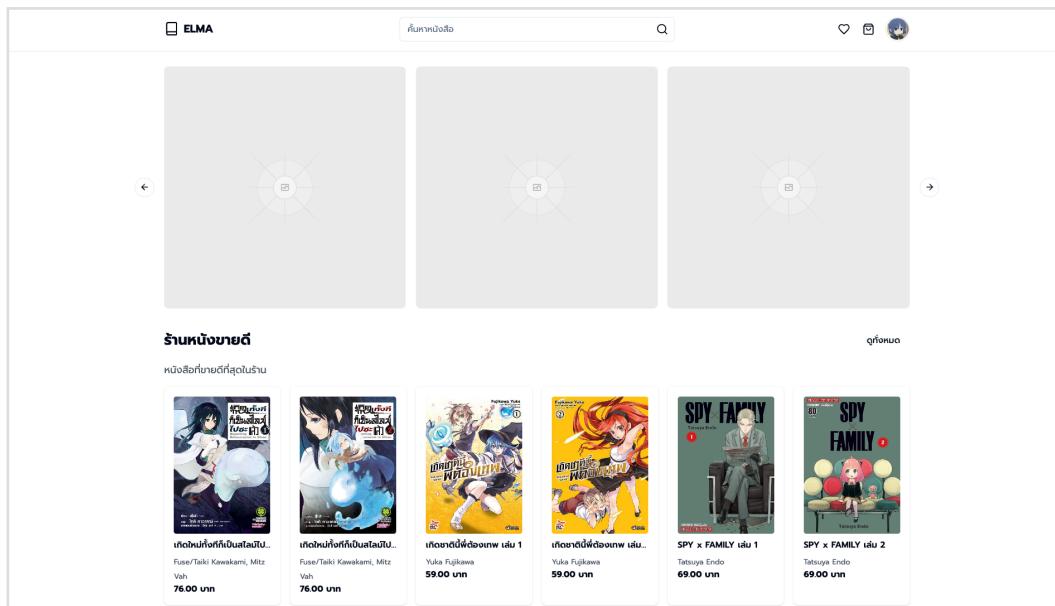
Key	Attribute	Type	Size	Description
PK	orderitem_id	char	36	รหัสใบสั่งซื้อย่อย
PK	order_id	char	36	รหัสใบสั่งซื้อ
FK	book_id	char	36	รหัสหนังสือ
	price	decimal	10, 2	ราคาที่ซื้อ

13. ตาราง Payment

Key	Attribute	Type	Size	Description
PK	payment_id	char	36	รหัสการชำระเงิน
FK	order_id	char	36	รหัสใบสั่งซื้อ
	amount	decimal	10, 2	จำนวนเงินที่ชำระ
	method	char		ช่องทางการชำระเงิน
	created_at	datetime		วันเวลาที่ชำระเงิน

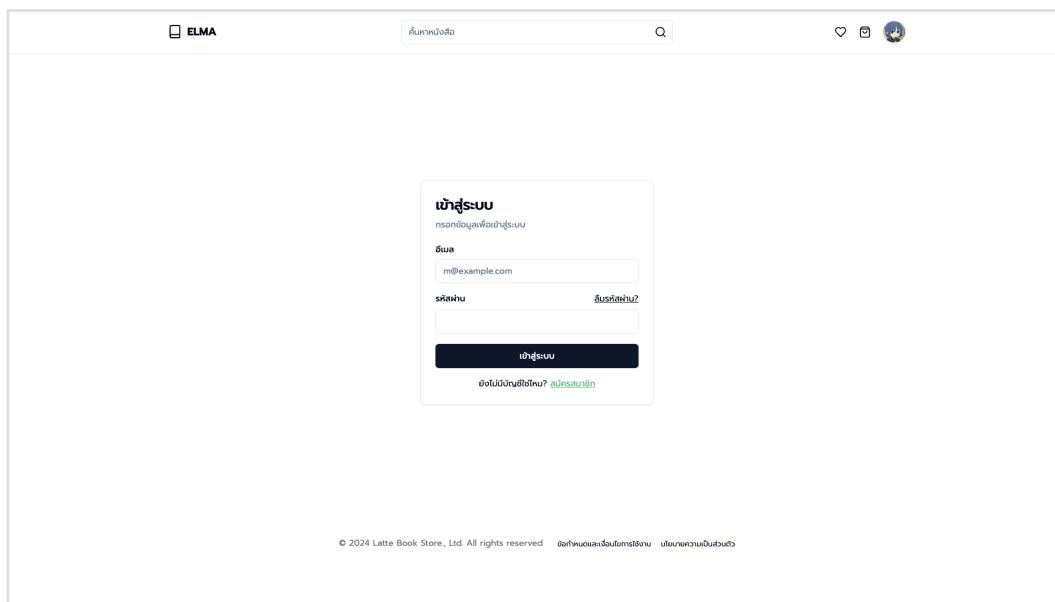
3.2.7 Screen Form Design

3.2.7.1 หน้าแรกเว็บไซต์



ภาพที่ 42 Screen Form หน้าแรกเว็บไซต์

3.2.7.2 หน้าเข้าสู่ระบบ



ภาพที่ 43 Screen Form เข้าสู่ระบบ

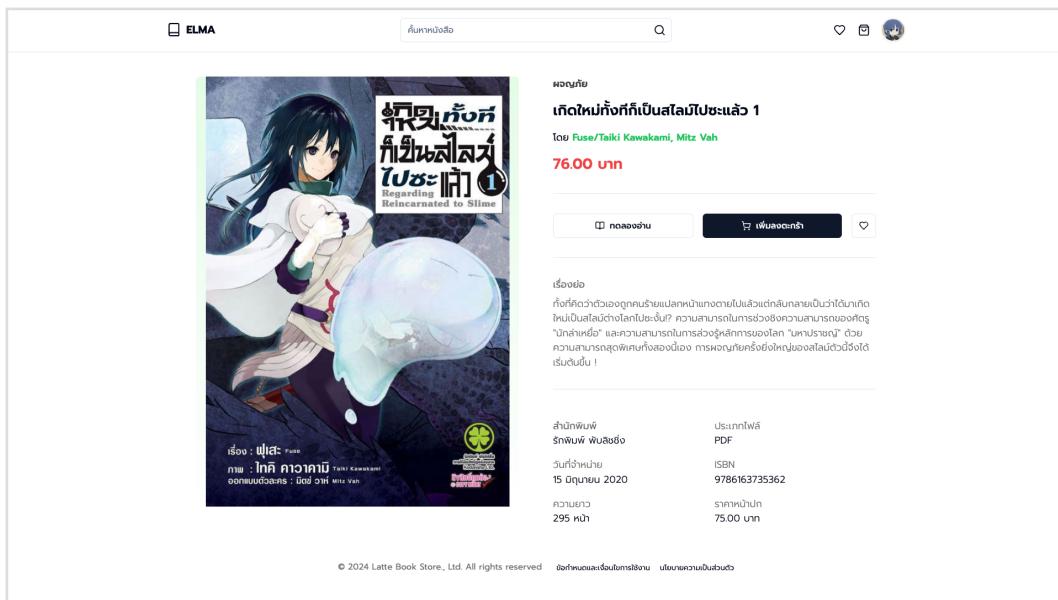
3.2.7.3 หน้าสมัครสมาชิก

The screenshot shows a registration form titled "สมัครสมาชิก" (Register Member) on the ELMA website. The form includes fields for First Name, Last Name, Email, Password, and Confirm Password. There is also a checkbox for accepting terms and conditions.

© 2024 Latte Book Store, Ltd. All rights reserved. ลิขสิทธิ์ของเว็บไซต์นี้เป็นของฉัน นโยบายความเป็นส่วนตัว

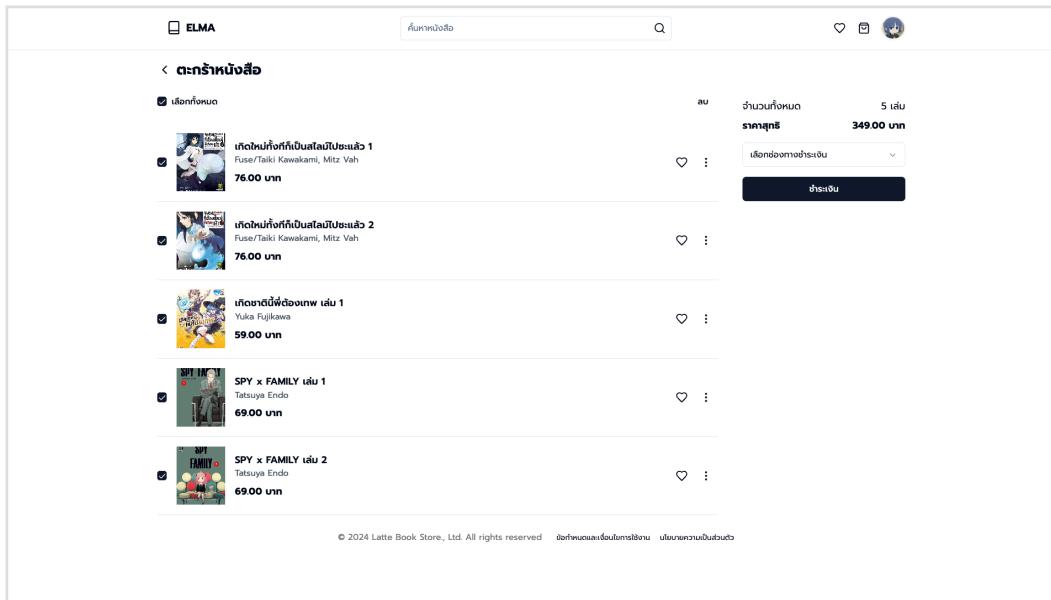
ภาพที่ 44 Screen Form สมัครสมาชิก

3.2.7.4 หน้ารายละเอียดสินค้า



ภาพที่ 45 Screen Form รายละเอียดสินค้า

3.2.7.5 หน้าตัชกร้าสินค้า



ภาพที่ 46 Screen Form ตะกร้าสินค้า

ข้อสรุปและข้อเสนอแนะ

การพัฒนาระบบระบบแอปพลิเคชันสำหรับการขายหนังสืออิเล็กทรอนิกส์ออนไลน์ใน Project นี้พัฒนาถึงขั้นตอนออกแบบ (Design) ซึ่งขั้นตอนที่จะพัฒนา ต่อไปคือ Implementation และ Operation จึงจะสมบูรณ์ด้วยงบประมาณ 300,000 บาท ด้วยงบประมาณที่กำหนดไว้สำหรับการพัฒนาระบบใหม่ ทำให้ระบบสามารถตอบสนองความต้องการของผู้ใช้ได้ดีขึ้น เช่น การแก้ไขปัญหาที่เกี่ยวข้องกับการจัดจำหน่ายหนังสือแบบรูปเล่มจริงที่องค์กรเคยประสบ การเปลี่ยนแปลงรูปแบบจากการจำหน่ายหนังสือแบบรูปเล่มมาเป็น หนังสืออิเล็กทรอนิกส์ (E-book) ได้รับความนิยมมากขึ้น และสุดท้ายระบบต้องได้การปรับปรุงและพัฒนาระบบท่อเนื่อง

บรรณานุกรม

<https://www.drawio.com/>

<https://bdi.or.th/big-data-101/sequence-diagram-and-activity-diagram/>

<https://www.foxbith.com/blog/what-is-sdlc>

http://202.44.33.44/ea-doh/backend/administrator/moduledownload/file_pdf/1480680853_75.pdf

<https://www.rama.mahidol.ac.th/ramapharmacy/sites/default/files/public/pdf/Use-Case-Diagram.pdf>

[http://old-book.ru.ac.th/e-book/c/CT478\(48\)/CT478\(48\)-5.pdf](http://old-book.ru.ac.th/e-book/c/CT478(48)/CT478(48)-5.pdf)

<https://aws.amazon.com/th/what-is/sdlc/>

<https://dol.dip.go.th/th/category/2019-02-08-08-57-30/2019-03-15-11-06-29>

<https://bluebik.com/th/blogs/5673>

<https://www.rmonlineservices.com/article/13/sdlc-model-คืออะไร-มีขั้นตอนอย่างไรบ้าง>

<https://devhub.in.th/blog/sdlc-software-development-life-cycle>