

รายงานเรื่อง

การออกแบบ UML จากโมเดลธุรกิจร้านขนมญี่ปุ่น

เสนอ

ผู้ช่วยศาสตราจารย์ ดร.อรรณณ เชาวลิต

จัดทำโดย

นาย วิจักขณ์สมา ห่องทองแดง 640710759

รายวิชา

OBJECT-ORIENTED SOFTWARE DEVELOPMENT (517-221)

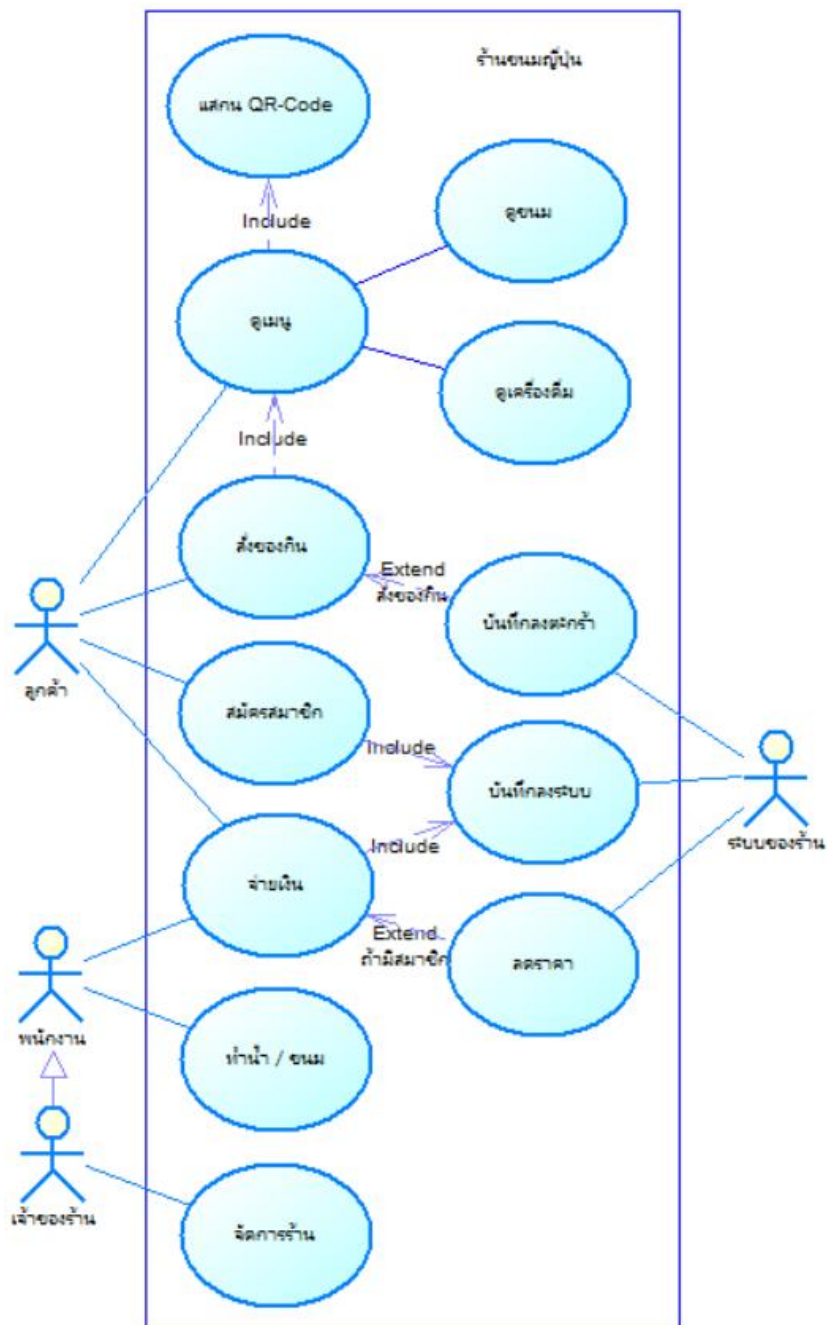
ประจำภาคการศึกษาที่ 2 ปีการศึกษา 2566

ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร

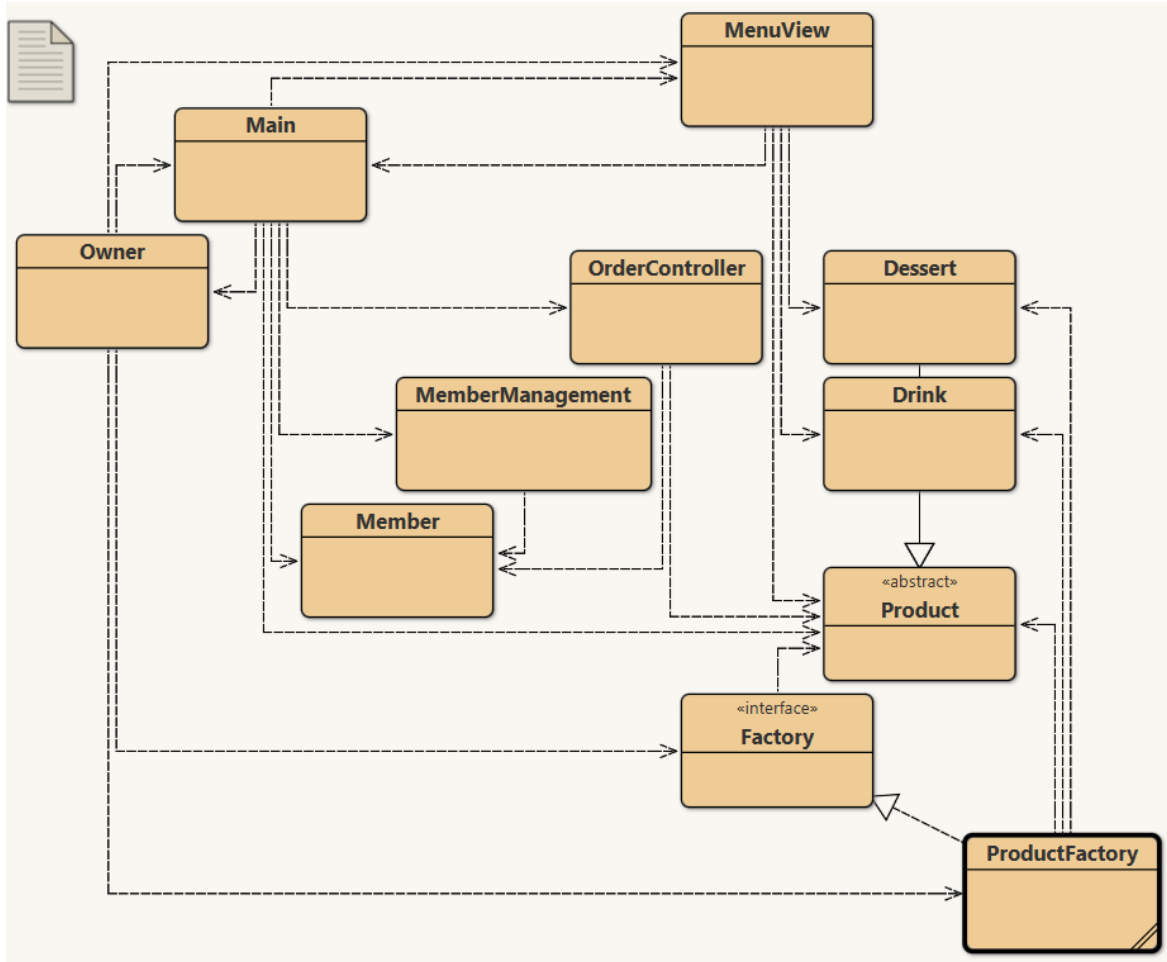
Business Model

ธุรกิจร้านขนมญี่ปุ่นโดยมีระบบร้านค้าออนไลน์ที่ช่วยจัดการร้าน ลูกค้าสามารถดูเมนู สั่งสินค้าในร้านและจ่ายเงินได้ในเว็บ โดยลูกค้าจะต้องสแกน QR-Code ก่อนเพื่อเข้าไปในหน้าร้าน เมื่ออยู่หน้าร้านแล้วลูกค้าสามารถดูเมนูได้โดยจะแยกเป็นขนมและเครื่องดื่ม ลูกค้าสามารถสั่งสินค้าในร้านได้จากหน้าเว็บ เมื่อสั่งแล้วระบบจะบันทึกรายการลงในตะกร้าสินค้าหากเลือกสินค้าเสร็จแล้วให้กดปุ่มจ่ายเงิน ระบบจะคิดราคารวมจากสินค้าในตะกร้าทั้งหมด หากลูกค้าเป็นสมาชิกจะลดราคารวมของสินค้า 10% หรือถ้าลูกค้าต้องการสมัครสมาชิกตอนจ่ายเงินเลยก็สามารถทำได้เพื่อใช้เป็นส่วนลด เมื่อจ่ายเงินแล้วระบบจะบันทึกลงในฐานข้อมูลของร้าน ส่วนของพนักงานทำหน้าที่ดูรายการของที่ต้องทำผ่านใบเสร็จที่ส่งมาเพื่อทำขนมและเครื่องดื่มตามรายการ เจ้าของร้านสามารถทำหน้าที่เดียวกับพนักงานได้และยังสามารถเพิ่มหรือลบสินค้าในร้านได้ด้วย

Use Case Diagram



Design Pattern



Factory Method Pattern

- ใช้สร้างและคืนค่าอ็อบเจกต์ของคลาส 'Coffee' หรือ 'Bakery' ตามประเภทที่ระบุโดยใช้ Factory Method ('createProduct') ในคลาส 'ProductFactory'.

- ในโค้ด: เมื่อต้องการสร้างสินค้าใหม่ เรียกใช้ 'ProductFactory.createProduct' โดยระบุประเภทของสินค้าและพารามิเตอร์อื่น ๆ เพื่อรับคืนอ็อบเจกต์ของสินค้าที่ต้องการ.

Iterator Pattern

- ใช้สำหรับการเข้าถึงข้อมูลในลำดับต่าง ๆ โดยไม่ต้องระบุโครงสร้างของลำดับนั้นๆ โดยตรง.

- ในโค้ด: ใช้ Iterator Pattern ในคลาส 'MenuView' เพื่อวนลูปผ่าน Main.products และแสดงรายการสินค้าในเมนูโดยไม่ต้องรู้ลักษณะของ Main.products โดยตรง.

Command Patter

- ใช้สำหรับการแยกคำสั่งจากผู้ใช้ และทำให้ผู้ใช้สามารถทำงานกับคำสั่งโดยไม่ต้องรู้ถึงรายละเอียดการทำงานของคำสั่งนั้น.

- ในโค้ด: ใช้ Command Pattern โดยใช้ switch statement ใน main method เพื่อรับคำสั่งต่าง ๆ เช่น ดูเมนู, เลือกสินค้า, ชำระเงิน, ยกเลิกการสั่งซื้อ, และสมัครสมาชิก โดยแต่ละ case จะเป็น Command ที่จะดำเนินการต่าง ๆ ตามคำสั่งที่รับ.

Singleton Pattern

- ใช้เพื่อให้มีแค่อ็อบเจกต์เดียวของคลาสที่เรียกใช้งานภายในโปรแกรม.

- ในโค้ด: ใช้ Singleton Pattern ในคลาส 'Main' โดยการประกาศตัวแปร 'scanner' เป็น static ซึ่งจะสร้างอ็อบเจกต์ Scanner เพียงอันเดียวและสามารถเข้าถึงได้ทุกที่ในโปรแกรม.

Strategy Pattern

- ใช้สำหรับการกำหนดวิธีการต่าง ๆ ในการปฏิบัติงานตามเงื่อนไขหรือสถานะที่กำหนด.

- ในโค้ด: ใช้ Strategy Pattern ในคลาส 'OrderController' เพื่อคำนวณส่วนลดในการชำระเงินตามสถานะของสมาชิก.

Template Method Pattern

- ใช้สำหรับกำหนดลำดับขั้นตอนในการทำงานโดยมีเมธอดหลัก (Template Method) ที่กำหนดลำดับขั้นตอนหลักและเรียกเมธอดย่อยเพื่อดำเนินการตามลำดับ.

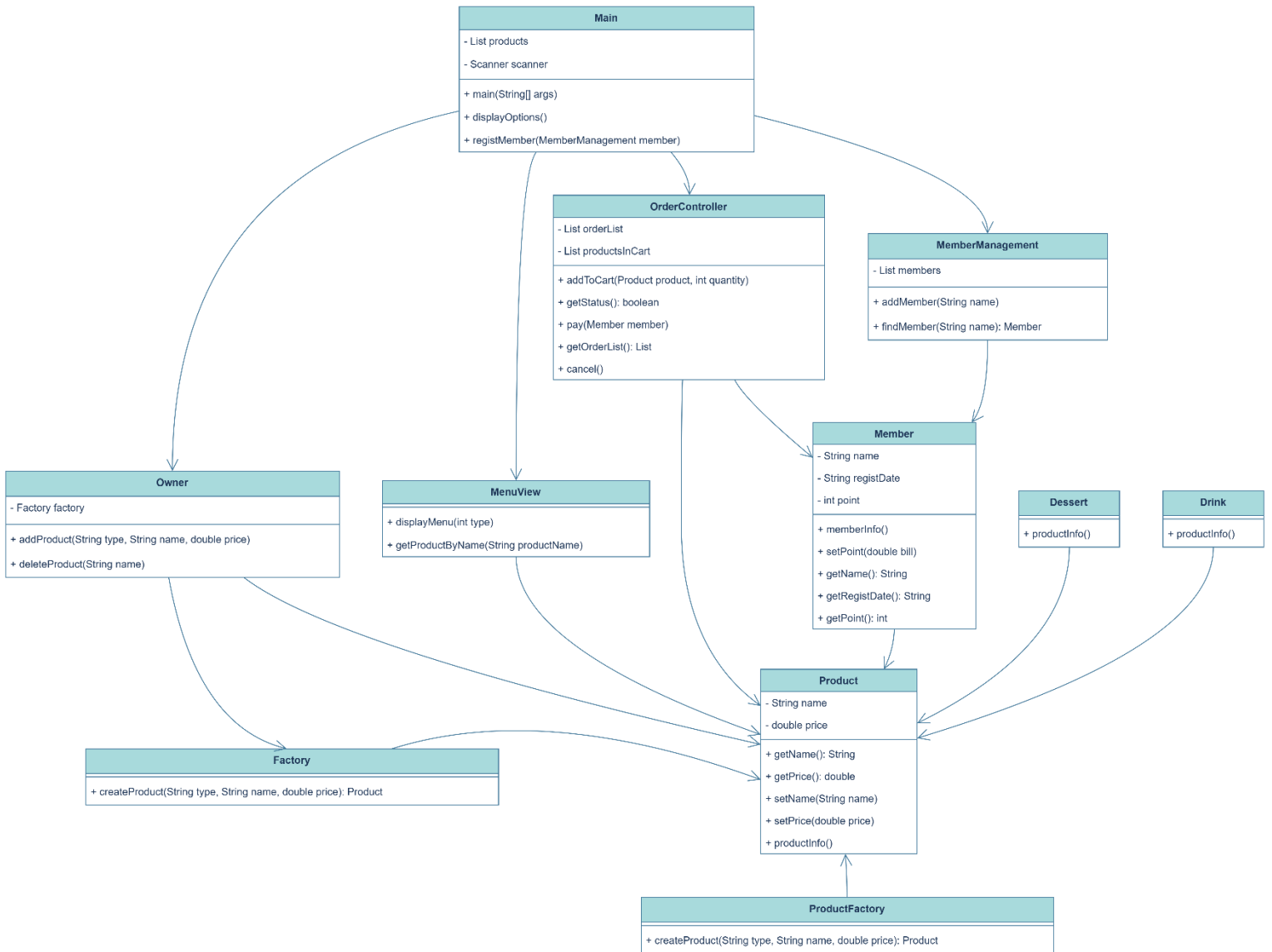
- ในโค้ด: ใช้ Template Method Pattern ในคลาส 'OrderController' โดยมีเมธอด 'pay' ที่กำหนดลำดับขั้นตอนในการชำระเงิน เช่น แสดงรายการสินค้า, คำนวณราคา, และส่วนลด.

Composite Pattern

- ใช้สำหรับสร้างโครงสร้างข้อมูลที่มีลำดับหรือโครงสร้างที่ซับซ้อน โดยทำให้อ็อบเจกต์แต่ละตัวมีลักษณะเดียวกันและเป็นส่วนหนึ่งของโครงสร้างที่ใหญ่ขึ้น.

- ในโค้ด: ใช้ Composite Pattern ในคลาส 'OrderController' โดยการเก็บรายการสินค้าในตะกร้า (productsInCart) แบบ List ซึ่งเป็นโครงสร้างแบบ Composite ซึ่งใช้เพื่อจัดการกับรายการสินค้าที่มีลักษณะเดียวกันโดยทั่วไปได้เป็นกลุ่มเดียวกันโดยอัตโนมัติโดยไม่ต้องระบุว่าเป็นแต่ละประเภทของสินค้าโดยตรง.

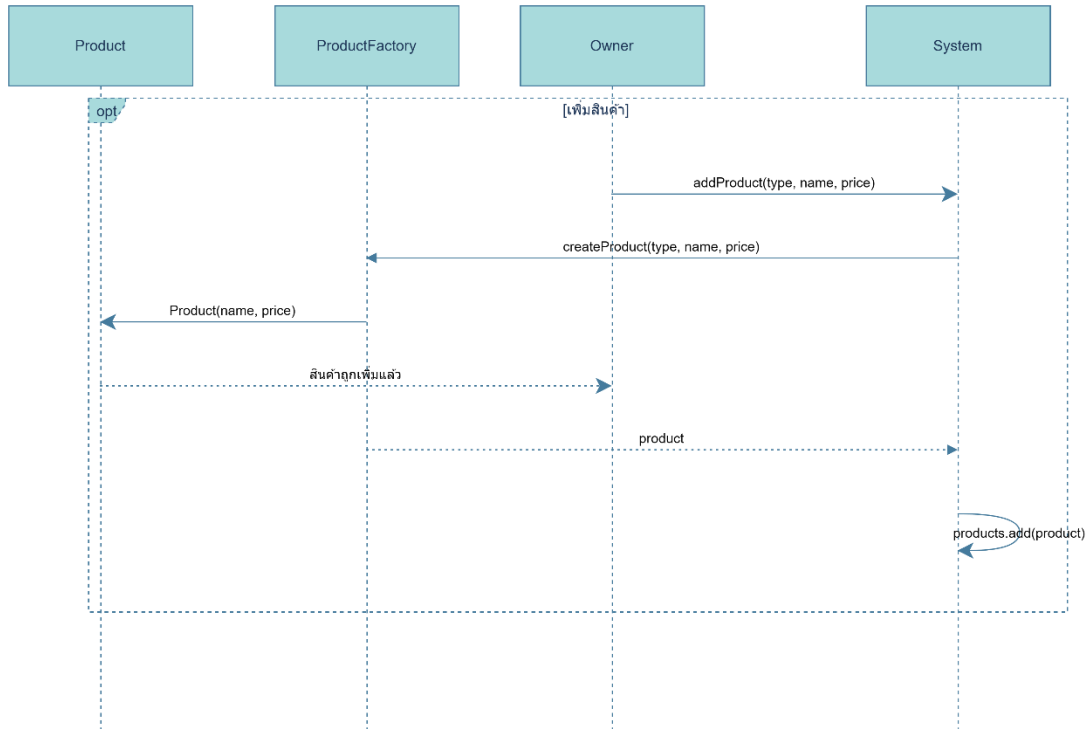
Class Diagram



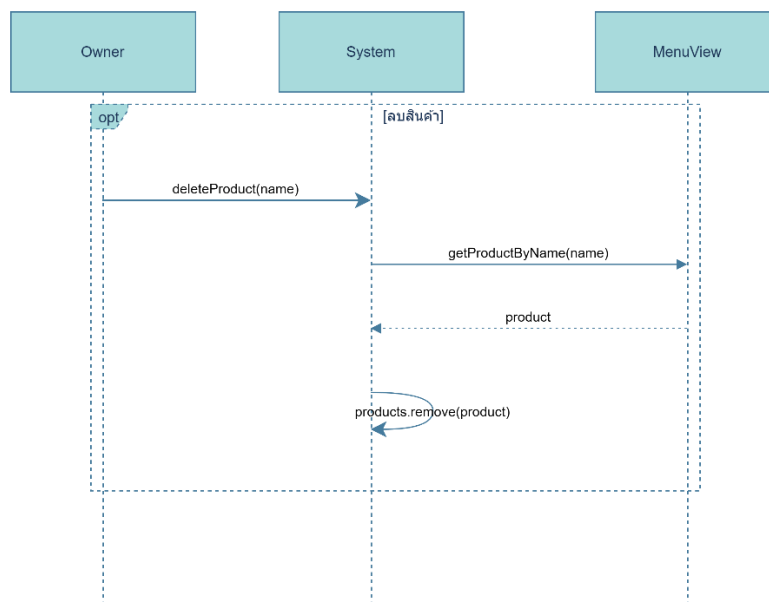
Sequence Diagram

เจ้าของร้าน

เพิ่มสินค้า

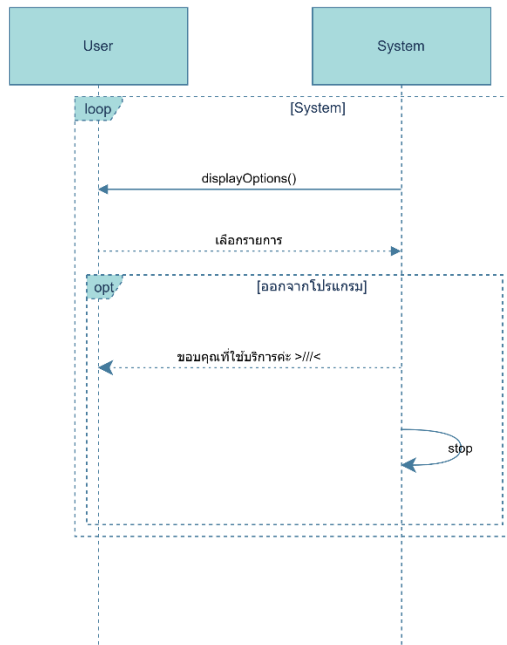


ลบสินค้า

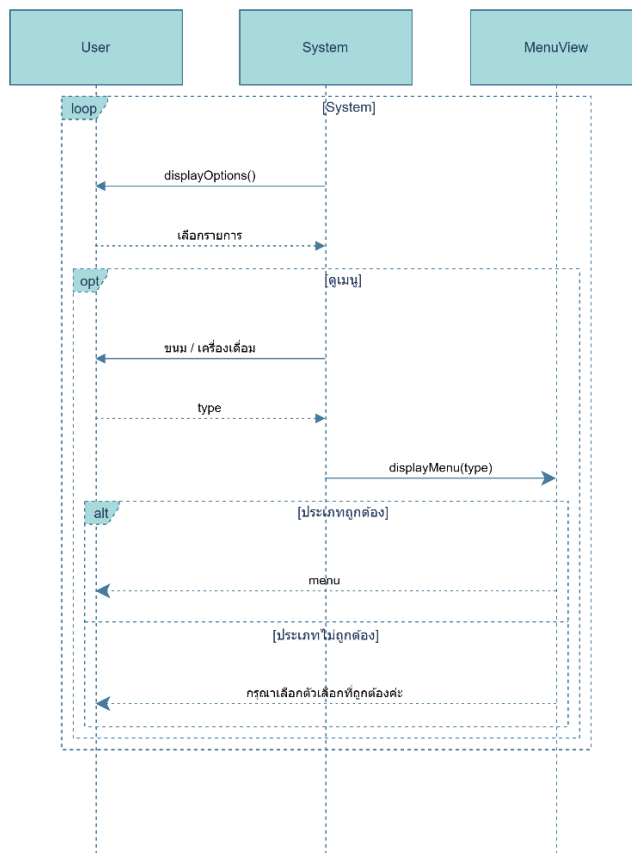


ลูกค้า

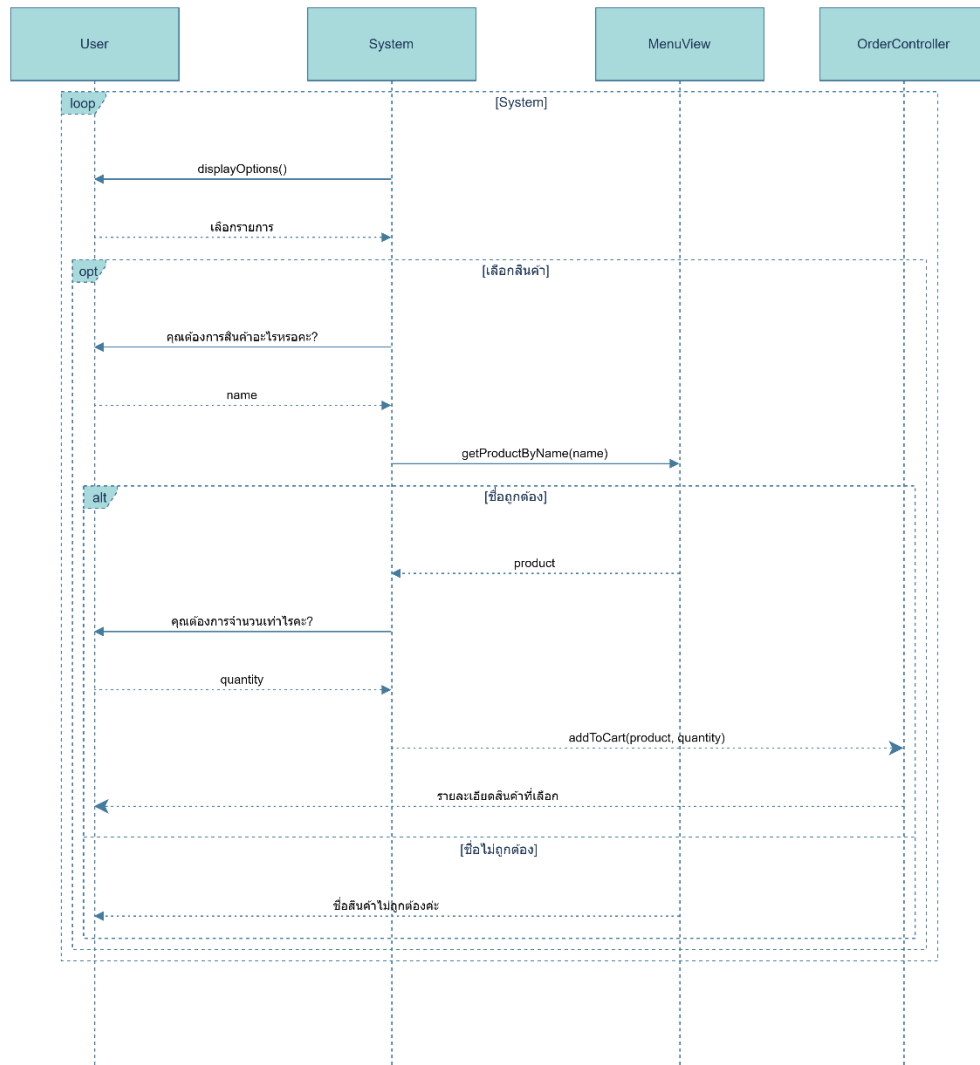
ออกจากโปรแกรม



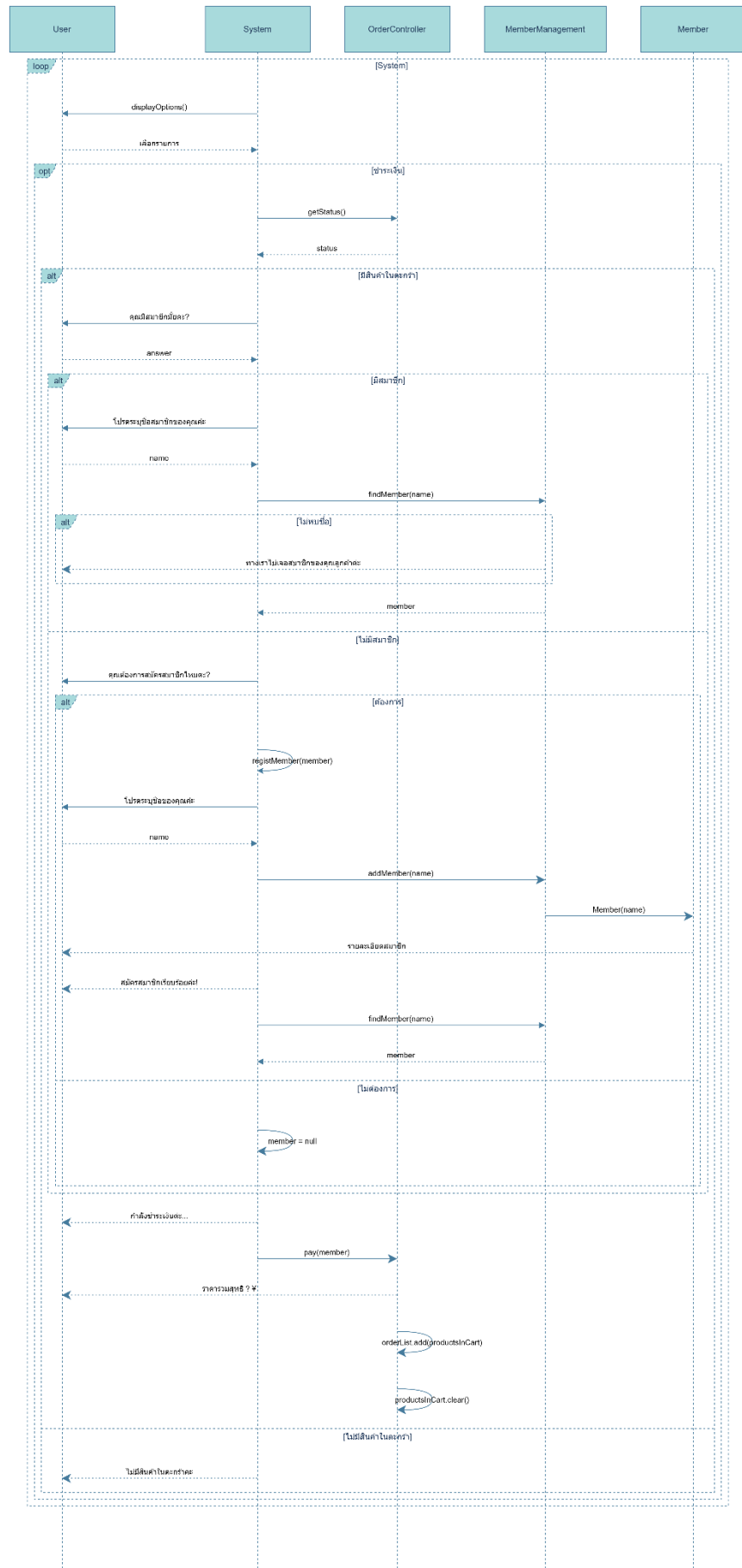
ดูเมนู



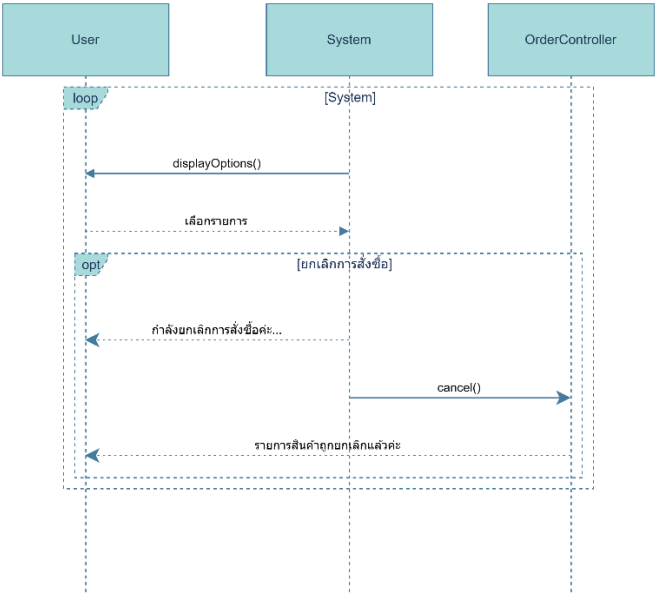
เลือกสินค้า



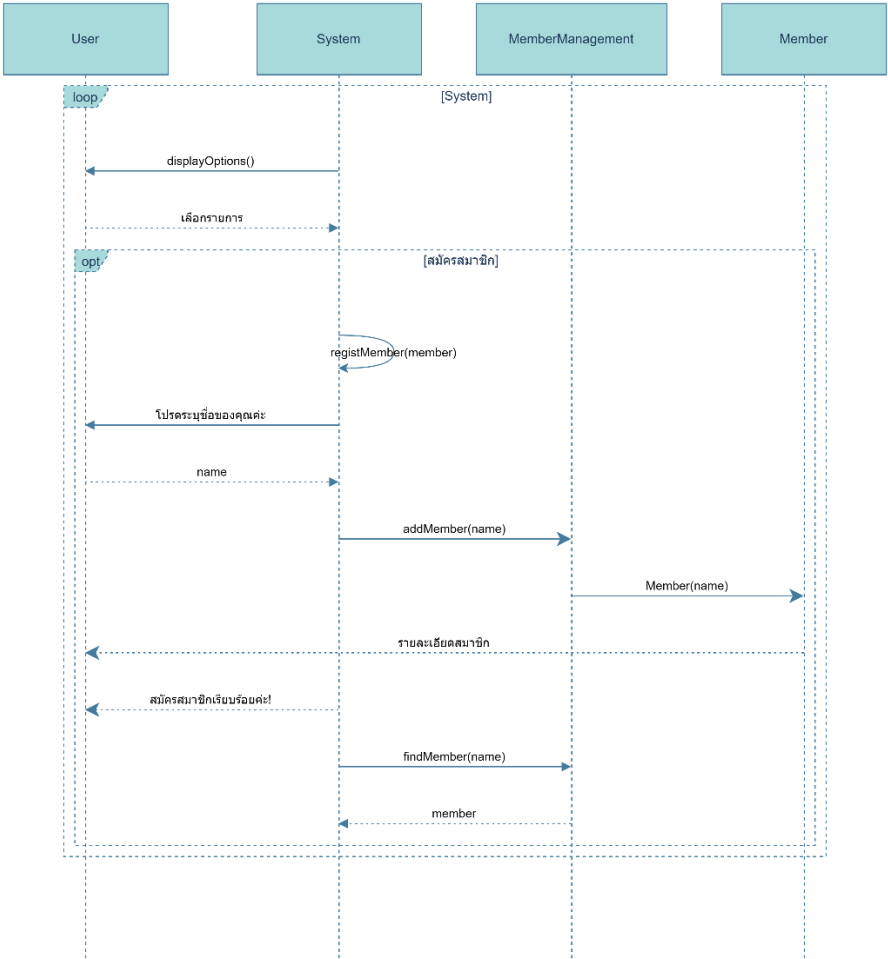
ชำระเงิน



ยกเลิกการสั่งซื้อ



สมัครสมาชิก



รายการผิด

