ขั้นตอนที่ 1: Actors & System Boundary

System Boundary (ขอบเขตของระบบ)

- The System: เว็บแอปพลิเคชัน "Mini Shopping Site"
- The Boundary (ขอบเขต): การพัฒนาเว็บแอปพลิเคชันที่มีฟังก์ชันหลักสำหรับ **ลูกค้า** ได้แก่ การค้นหาสินค้า, การจัดการตะกร้าสินค้าที่ตรวจสอบสต็อกได้ทันที และการสั่งซื้อ (Checkout) รวมถึงฟังก์ชันสำหรับ **ผู้ดูแลระบบ (Admin)** ในการเพิ่ม/ลบสินค้าและอัปเดตสถานะคำสั่งซื้อ โดยระบบจะมีปฏิสัมพันธ์กับ **Payment Processor** ซึ่งเป็นระบบภายนอกเพื่อยืนยันการชำระเงินเท่านั้น และในเวอร์ชันนี้ยังไม่รองรับการสั่งซื้อแบบไม่เป็นสมาชิก (Guest Checkout)

Actors (ผู้กระทำ)

1. Customer (ลูกค้า) - Primary Actor
    - บทบาท: ผู้ใช้งานทั่วไปที่เข้ามาค้นหา เลือกสินค้า และสั่งซื้อ เป็นผู้ริเริ่มเป้าหมายหลักทางธุรกิจ (การซื้อขาย)
2. Store Administrator (ผู้ดูแลร้านค้า) - Primary Acto r
    - บทบาท: ผู้ใช้งานภายในที่ดูแลสต็อกสินค้าและสถานะคำสั่งซื้อเป็นผู้ป้อนข้อมูลดิบ (สินค้า) ที่ระบบต้องใช้ และจัดการผลลัพธ์ (Order) ที่เกิดขึ้น
3. Payment Processor (ตัวประมวลผลการชำระเงิน) - Secondary Actor / External System
    - บทบาท: ระบบภายนอก (เช่น Stripe/PayPal) ที่ทำหน้าที่ยืนยันธุรกรรมการเงินเป็นระบบ ภายนอก ที่เราควบคุมไม่ได้ ระบบของเราแค่ส่งข้อมูลไปและรอผลลัพธ์ (Success/Fail) กลับมา

ขั้นตอนที่ 2: Use Case List (ระดับ User-Goal)

Actor: Customer (ลูกค้า)

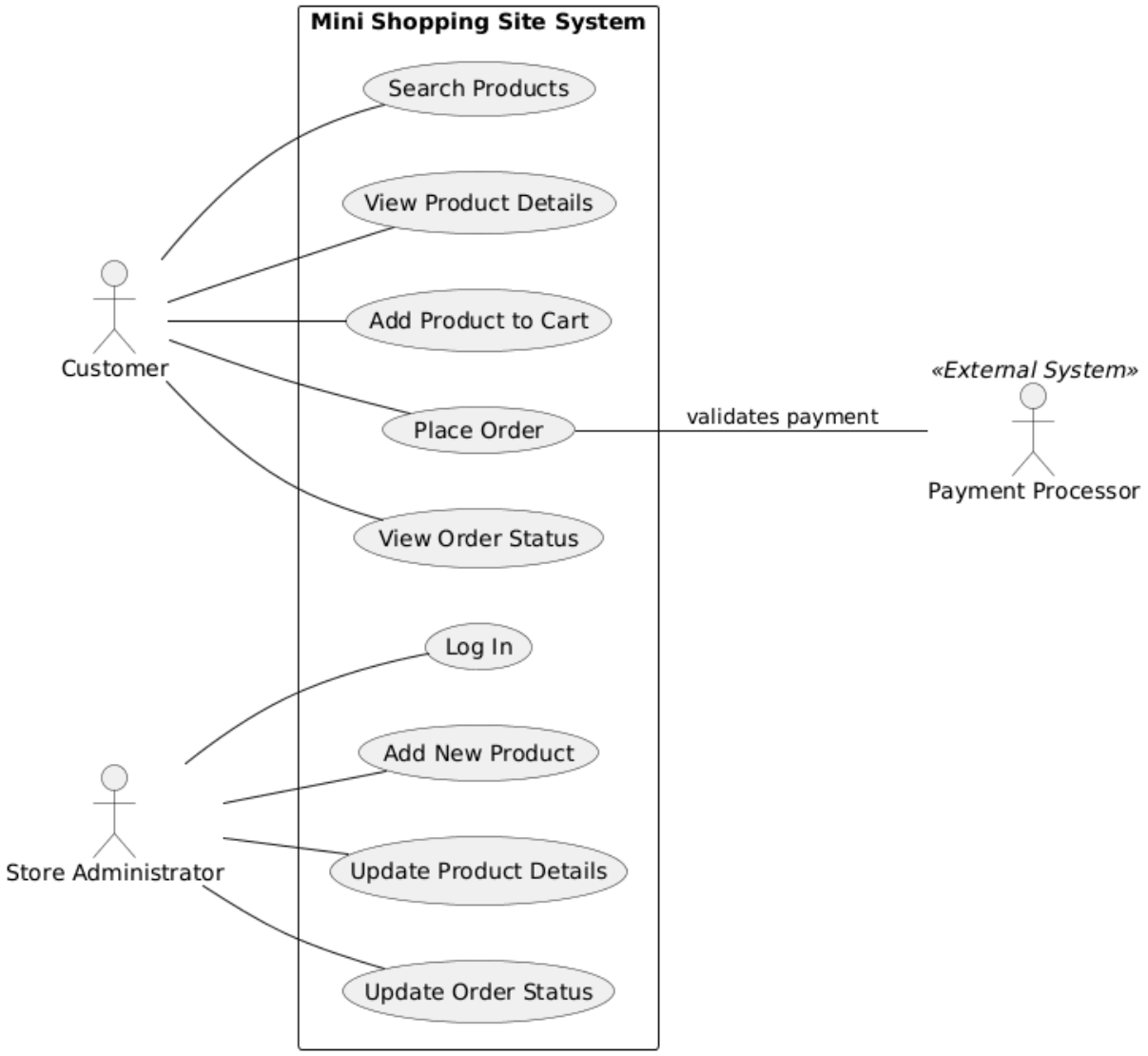
- UC-01: Search Products
    - *เป้าหมาย:* หาสินค้าที่ต้องการผ่านคำค้นหาหรือตัวกรอง
- UC-02: View Product Details
    - *เป้าหมาย:* ดูข้อมูลราคาและรายละเอียดก่อนตัดสินใจซื้อ
- UC-03: Add Product to Cart
    - *เป้าหมาย:* เลือกสินค้าและจำนวนลงตะกร้า (พร้อมเช็คสต็อก)
- UC-04: View Shopping Cart

    - *เป้าหมาย:* ตรวจสอบรายการและราคารวมก่อนยืนยัน
- UC-05: Place Order (MVP) 🌟
    - *เป้าหมาย:* ยืนยันการสั่งซื้อ ระบุที่อยู่ และชำระเงิน
- UC-06: View Order Status
    - *เป้าหมาย:* ติดตามสถานะของคำสั่งซื้อ

Actor: Store Administrator (ผู้ดูแลร้านค้า)

- UC-07: Log In
    - *เป้าหมาย:* ยืนยันตัวตนเพื่อเข้าถึงระบบหลังบ้าน
- UC-08: Add New Product (MVP) 🌟
    - *เป้าหมาย:* เพิ่มรายการสินค้าใหม่เข้าสู่ระบบ
- UC-09: Update Product Details
    - *เป้าหมาย:* แก้ไขข้อมูลสินค้าที่มีอยู่
- UC-10: Delete Product
    - *เป้าหมาย:* ลบสินค้าที่ไม่ต้องการขายแล้ว
- UC-11: Update Order Status
    - *เป้าหมาย:* เปลี่ยนสถานะคำสั่งซื้อ (เช่น เป็น "จัดส่งแล้ว")

ขั้นตอนที่ 3: UML Use Case Diagram



ขั้นตอนที่ 4: Detailed Use Case Narratives (รายละเอียด Use Case)

UC-05: สั่งซื้อสินค้า (Place Order)

- **Goal:** เพื่อทำการซื้อสินค้าให้เสร็จสมบูรณ์ ชำระเงิน และสร้างรายการคำสั่งซื้อเข้าระบบ
- **Primary Actor:** Customer (ลูกค้า)
- **Preconditions (เงื่อนไขเริ่มต้น):**
    - ลูกค้ามีสินค้าอยู่ในตะกร้าอย่างน้อย 1 ชิ้น
    - ลูกค้าได้ทำการเข้าสู่ระบบ (Log in) เรียบร้อยแล้ว (ตามกฎ Business Rule: Identity)
- **Trigger (จุดกระตุ้น):** ลูกค้ากดปุ่ม "Checkout" หรือ "ชำระเงิน" จากหน้าตะกร้าสินค้า
- **Main Success Scenario (เหตุการณ์ปกติ):**
    - ระบบแสดงสรุปรายการสินค้าและราคารวมในตะกร้า
    - ลูกค้ากรอกหรือยืนยัน **ที่อยู่สำหรับการจัดส่ง**
    - ลูกค้ากรอก **ข้อมูลการชำระเงิน** (เช่น เลขบัตรจำลอง)
    - ลูกค้ายืนยันการสั่งซื้อ
    - ระบบส่งข้อมูลการชำระเงินไปยัง **Payment Processor** (Actor ภายนอก) เพื่อตรวจสอบความถูกต้อง
    - **Payment Processor** ส่งสถานะ "ยืนยันการชำระเงินสำเร็จ" กลับมายังระบบ
    - ระบบสร้างรายการคำสั่งซื้อ (Order) ใหม่ในฐานข้อมูล โดยมีสถานะเริ่มต้นเป็น "Pending" หรือ "รอตรวจสอบ"
    - ระบบตัดจำนวนสินค้าออกจากคลังสินค้าทันที (ตามกฎ Business Rule: Stock Reservation)
    - ระบบล้างรายการสินค้าในตะกร้าของผู้ใช้ให้ว่างเปล่า
    - ระบบแสดงหน้ายืนยันความสำเร็จพร้อม **เลขที่คำสั่งซื้อ (Order ID)** ให้ลูกค้าทราบ
- **Alternative / Exception Flows (เหตุการณ์ทางเลือก/ข้อยกเว้น):**
    - **6a. Payment Declined (การชำระเงินถูกปฏิเสธ):**
        1. Payment Processor แจ้งว่ายอดเงินไม่เพียงพอหรือข้อมูลบัตรไม่ถูกต้อง
        2. ระบบแสดงข้อความแจ้งเตือนข้อผิดพลาดให้ลูกค้าทราบ
        3. ระบบ **ไม่** สร้างรายการคำสั่งซื้อ และ **ไม่** ตัดสต็อกสินค้า
        4. ระบบคงสถานะอยู่ที่หน้าชำระเงินเพื่อให้ลูกค้าแก้ไขข้อมูลบัตรเครดิต
    - 8a. Concurrent Stock Conflict (สินค้าหมดระหว่างทำรายการ):
        1. ในขณะที่กำลังประมวลผลคำสั่งซื้อ ระบบตรวจสอบพบว่าสินค้าคงเหลือไม่เพียงพอ (ถูกแย่งซื้อตัดหน้า)
        2. ระบบแสดงข้อความแจ้งเตือนว่า "สินค้าบางรายการหมด"

        3. ระบบยกเลิกกระบวนการสร้างคำสั่งซื้อนั้น
        4. ระบบนำลูกค้ากลับไปยังหน้าตะกร้าสินค้าเพื่อปรับเปลี่ยนรายการ
- **Postconditions (เงื่อนไขหลังจบงาน):**
    - **Success:** บันทึก Order ลงฐานข้อมูล, จำนวนสต็อกสินค้าลดลง, ลูกค้าได้รับ Order ID
    - **Failure:** ไม่มีการสร้าง Order, สต็อกสินค้าคงเดิม

UC-08: เพิ่มสินค้าใหม่ (Add New Product)

- **Goal:** เพื่อเพิ่มรายการสินค้าใหม่เข้าสู่ระบบ เพื่อให้ลูกค้าสามารถค้นหาและสั่งซื้อได้
- **Primary Actor:** Store Administrator (ผู้ดูแลร้านค้า)
- **Preconditions (เงื่อนไขเริ่มต้น):**
    - Admin ได้ทำการเข้าสู่ระบบ (Log in) แล้ว
    - บัญชีผู้ใช้มีสิทธิ์ระดับผู้ดูแลระบบ (Admin Privileges)
- **Trigger (จุดกระตุ้น):** Admin เลือกเมนูหรือปุ่ม "เพิ่มสินค้าใหม่" ในระบบหลังบ้าน
- **Main Success Scenario (เหตุการณ์ปกติ):**
    - ระบบแสดงแบบฟอร์มสำหรับเพิ่มสินค้า
    - Admin กรอกรายละเอียดสินค้า ได้แก่ **ชื่อสินค้า, ราคา, หมวดหมู่, ลิงก์รูปภาพ, และจำนวนสินค้าในสต็อก**
    - Admin สั่งบันทึกข้อมูล
    - ระบบตรวจสอบความครบถ้วนของข้อมูลที่จำเป็น
    - ระบบตรวจสอบความถูกต้องของราคา (Price Integrity) ว่าต้องไม่ติดลบ
    - ระบบบันทึกข้อมูลสินค้าใหม่ลงในฐานข้อมูล
    - ระบบแสดงข้อความยืนยันความสำเร็จ
    - สินค้าใหม่ปรากฏขึ้นบนหน้าร้าน (Homepage/Product Grid) ให้ลูกค้าเห็นทันที
- **Alternative / Exception Flows (เหตุการณ์ทางเลือก/ข้อยกเว้น):**
    - 5a. Invalid Price (ราคาสินค้าไม่ถูกต้อง):
        1. Admin ระบุราคาสินค้าเป็นค่าติดลบ (เช่น -100)
        2. ระบบตรวจพบว่าข้อมูลขัดแย้งกับกฎทางธุรกิจ
        3. ระบบปฏิเสธการบันทึกข้อมูล
        4. ระบบแสดงข้อความแจ้งเตือน "ราคาสินค้าต้องมากกว่า 0" ที่ช่องระบุราคา
        5. Admin แก้ไขราคาให้ถูกต้องและบันทึกใหม่อีกครั้ง

- **Postconditions (เงื่อนไขหลังจบงาน):**
    - **Success:** มี Record สินค้าใหม่ในฐานข้อมูล และแสดงผลบนเว็บไซต์
    - **Failure:** ไม่มีการเปลี่ยนแปลงข้อมูลในฐานข้อมูล

ขั้นตอนที่ 5: Open questions/Assumptions + AI log entry

Assumptions (สมมติฐานที่กำหนดเองเพื่อทำโปรเจกต์ต่อ):

- **A-01 Shipping Cost:** เนื่องจาก Brief ไม่ได้ระบุวิธีคำนวณค่าส่ง จึงขอสมมติว่าระบบใช้ **Flat Rate (ราคาเหมาจ่าย)** หรือ **ส่งฟรี** สำหรับ MVP นี้ เพื่อลดความซับซ้อนของ Use Case "Place Order"
- **A-02 Image Hosting:** จาก FR-10 ที่ระบุว่า "แปะลิงก์รูปภาพ" จึงสมมติว่าระบบ **ไม่มีฟังก์ชันอัปโหลดไฟล์** ไปยัง Server ของเราเอง แต่จะใช้การเก็บ URL จากเว็บฝากรูปภายนอกเท่านั้น
- **A-03 Notification Channel:** แม้จะมีการแจ้งเตือนสถานะคำสั่งซื้อ แต่สมมติว่าใน MVP นี้ การแจ้งเตือนจะเกิดขึ้น **ภายในหน้าเว็บ (On-site notification)** หรืออีเมลจำลองเท่านั้น ไม่มีการเชื่อมต่อ SMS Gateway จริง

Open Questions (คำถามที่ต้องการคำตอบเพิ่ม):

- **Q-01 Admin Creation:** ในระบบไม่มี Use Case สำหรับ "ลงทะเบียน Admin" อยากทราบว่า Admin คนแรกเกิดขึ้นได้อย่างไร? (เช่น ต้อง Insert ลง Database โดยตรง หรือมี Super Admin?)
- **Q-02 Abandoned Cart:** หากลูกค้าเพิ่มของใส่ตะกร้าทิ้งไว้แล้ว Session หมดอายุ เมื่อกลับมาล็อกอินใหม่ ของในตะกร้าควรจะหายไปเลย หรือควรดึงกลับมาแสดง (Persist Cart)?
- **Q-03 Payment Failure:** หาก Payment Processor แจ้งว่าจ่ายเงินล้มเหลว ระบบควรเก็บ Order ไว้ในสถานะ "รอชำระเงิน" หรือไม่สร้าง Order เลย? (ปัจจุบันเลือกสมมติฐานว่าไม่สร้าง Order เลยตาม UC-01

AI Usage Log Entry (บันทึกการใช้งาน AI)

ส่วนนี้เป็นการบันทึกว่าเราใช้ AI ช่วยทำงานตรงไหนบ้าง ตามรูปแบบในหน้า 6 ของ Project Brief

Prompt Summary:

"Act as an OOAD Teaching Assistant. Analyze the 'Mini Shopping Site' project brief and the 'CPE362 Modeling Use Cases' lecture slides.

1. Identify Actors and System Boundary according to the lecture principles.
2. Create a Use Case List (User-Goal Level) and select Top 2 MVP use cases.
3. Write detailed Use Case Narratives for 'Place Order' and 'Add New Product' including Happy Path and Exceptions.
4. Generate PlantUML code for the Use Case Diagram."

Accepted (ส่วนที่ยอมรับมาใช้):

- **Actor Identification:** ยอมรับการแยก **Payment Processor** ออกเป็น External Actor ตามหลักการในสไลด์หน้า 7-8
- **Use Case Names:** ยอมรับชื่อ Use Case ที่ขึ้นต้นด้วยคำกริยา (Verb Phrase) เช่น "Place Order", "Add New Product" ซึ่งตรงกับมาตรฐาน
- **Exception Flows:** ยอมรับกรณีตรวจสอบราคาติดลบ (Price Integrity) และสินค้าหมด (Out of Stock) ที่ AI แนะนำเข้ามาใน Use Case Narrative

Refined/Modified (ส่วนที่ปรับแก้):

- **Scope Adjustment:** ตัด Use Case "Register" ของลูกค้าออกชั่วคราว เพื่อโฟกัสที่ Flow การซื้อขาย (Buying) ตามเป้าหมาย MVP
- **Flow Correction:** ใน UC-01 ปรับแก้ลำดับขั้นตอนให้ "ตัดสต็อก" (Deduct Stock) เกิดขึ้นทันทีหลังยืนยันคำสั่งซื้อ เพื่อให้สอดคล้องกับ Business Rule ข้อ BR1 (Stock Reservation) ในโจทย์ จากเดิมที่ AI เสนอให้ตัดตอนชำระเงินเสร็จ

AI SUMMARY

ภาพรวมของ Use Case Model ที่ออกแบบมานี้ **มีความสมบูรณ์และพร้อมสำหรับการพัฒนาในระยะ MVP (Minimum Viable Product)** การออกแบบได้แยกขอบเขตระหว่าง "สิ่งที่ระบบทำ" และ "สิ่งที่คน/ระบบอื่นทำ" ไว้อย่างชัดเจน ทำให้ทีมพัฒนาสามารถโฟกัสงาน Coding ได้ตรงจุด ลดความเสี่ยงในการทำเกินขอบเขต (Scope Creep)

✅ จุดแข็งของการออกแบบ (Key Strengths)

1. การกำหนดขอบเขตระบบที่ชัดเจน (Clear System Boundary)

- **การวิเคราะห์:** มีการแยก **Payment Processor** ออกเป็น External Actor อย่างถูกต้อง ไม่ได้พยายามเขียนโค้ดจัดการธุรกรรมการเงินเอง ซึ่งลดความซับซ้อนและความเสี่ยงด้านความปลอดภัย
- **ผลลัพธ์:** ลดภาระงานที่ไม่จำเป็น และทำให้ Use Case "Place Order" (UC-05) กระชับขึ้น เพราะระบุแค่การ "ส่งข้อมูล" และ "รับผลลัพธ์" เท่านั้น

2. ระดับความละเอียดที่เหมาะสม (Correct Granularity)

- **การวิเคราะห์:** Use Case ทั้งหมดอยู่ในระดับ **User-Goal Level** (เช่น "Place Order", "Search Products") ตามที่เลคเชอร์แนะนำ ไม่แตกละเอียดเกินไปจนเป็นระดับ UI (เช่น "Click Button") และไม่หยาบเกินไปจนเป็นระดับ Summary (เช่น "Manage System")
- **ผลลัพธ์:** ทีมงานเข้าใจเป้าหมายทางธุรกิจของแต่ละฟังก์ชันได้ง่าย ไม่หลงทางไปกับรายละเอียดหน้าจอ

3. การครอบคลุมกรณีผิดปกติ (Robust Exception Handling)

- **การวิเคราะห์:** ใน Narrative มีการระบุ **Alternative Flows** ที่สำคัญครบถ้วน เช่น กรณี "ตัดบัตรไม่ผ่าน" หรือ "สินค้าหมดระหว่างทำรายการ" (Concurrent Stock Conflict)
- **ผลลัพธ์:** ช่วยป้องกัน Bug ที่มักเกิดจาก Edge Cases เหล่านี้ ตามหลักการ "Most bugs hide in exception flows"

4. การผสานกฎทางธุรกิจ (Business Rules Integration)

- **การวิเคราะห์:** กฎสำคัญอย่าง **Stock Reservation (BR1)** ถูกนำไปใส่ในขั้นตอนของ Use Case ชัดเจน (ตัดสต็อกทันทีที่ยืนยันออเดอร์) แทนที่จะรอไปตัดตอนจัดส่ง

- **ผลลัพธ์:** ลดปัญหาการขายสินค้าเกินจำนวน (Overselling) ซึ่งเป็นเป้าหมายหลักทางธุรกิจของระบบนี้

⚠️ จุดที่ต้องพิจารณาและข้อจำกัด (Limitations & Risks)

จากการประเมินเทียบกับความต้องการทางธุรกิจจริง ยังมีจุดที่ต้องตระหนักไว้สำหรับการพัฒนาต่อ:

1. **ขาดฟีเจอร์ Guest Checkout (ตามขอบเขต MVP):**
    - การบังคับ Login ก่อนซื้อ (Precondition: User is logged in) เป็นไปตามกฎ BR3 เพื่อลดความซับซ้อน แต่ในมุมมอง UX ของ E-commerce จริง อาจทำให้ยอดขาย (Conversion Rate) ลดลงได้
    - *ข้อเสนอแนะ:* ควรพิจารณาทำ Guest Checkout ในเฟสถัดไป
2. **การจัดการสินค้าคงคลังแบบพื้นฐาน:**
    - ปัจจุบัน UC-08 (Add Product) และ UC-05 (Place Order) จัดการสต็อกแบบง่ายๆ ไม่มีการจองสต็อกชั่วคราว (Temporary Hold) ขณะอยู่ในหน้า Payment Gateway หากลูกค้ากดจ่ายเงินช้า อาจโดนแย่งของได้ (แม้จะมี Flow ดักจับแล้ว แต่ UX อาจไม่ลื่นไหล)
3. **การพึ่งพา Manual Process ของ Admin:**
    - การอัปเดตสถานะการจัดส่ง (UC-11) ยังต้องให้ Admin มากดเองทีละรายการ ไม่มีการเชื่อมต่อกับระบบขนส่ง (Logistics API) ซึ่งอาจเป็นคอขวดถ้ามีออเดอร์จำนวนมาก