

Usecase name	รับ order จากลูกค้า	
Usecase description	บันทึกคำสั่งซื้อของลูกค้าเป็น Order + OrderItem อย่างถูกต้อง	
Actor	Staff (role = Sale)	
Precondition	มีข้อมูลลูกค้า (Customer)	
Normal Flow	Action	System
	1. กด "Create Order" 3. กดเข้า กล้อง Customer 5. เริ่มพิมพ์ keyword ของ Customer 7. เลือกลูกค้า 9. กดเข้า กล้อง Product	2. display order form. Application สร้าง object/list ใน memory เช่น {customer_id: NULL staff_id: NULL items: [] total_amount: 0 } *init 4. preload ลูกค้าทั้งหมด SELECT customer_id , customer_name FROM Customer ORDER BY customer_name *display customer 6. context search filter จากข้อมูลที่ preload มาแล้ว 8. setCustomer(customer_id) ลง object 10. preload สินค้าทั้งหมด SELECT product_id , product_name , price_per_unit FROM Product ORDER BY product_name LIMIT 50;

	<p>11. เริ่มพิมพ์ keyword ของ Product</p> <p>13. เลือกสินค้า + ใส่จำนวน (qty)</p> <p>15. เลือกสินค้าเพิ่ม (loop ขั้9)</p> <p>17. ตรวจสอบ Order</p> <p>19. กด “Confirm Order”</p>	<p>12. context search filter จากข้อมูลที่ preload มาแล้ว (ไม่ query DB ขั้)</p> <p>display product ตาม Filter</p> <p>14. Validate</p> <p>IF (product_id IS NULL): error</p> <p>IF (qty IS NULL OR qty <= 0): error</p> <p>ELSE:</p> <p>line_total = qty × price_per_unit</p> <p>items.push({ product_id : product_id, qty : qty, unit_price : price_per_unit, line_total : line_total, fulfilled_qty: 0, remaining_qty: qty })</p> <p>display สินค้าที่เลือก</p> <p>16. Application update list + update total_amount = SUM(line_total) (ยังไม่ insert DB)</p> <p>18. ระบบแสดง Order summary จาก object (Customer + Items + total_amount)</p> <p>Final validation</p> <p>IF (items IS EMPTY): error: "ยังไม่มีสินค้าใน Order"</p> <p>IF (customer_id IS NULL): error: "ยังไม่ได้เลือก customer"</p> <p>20. INSERT DB:</p> <p>① Insert Order</p> <p>INSERT INTO "Order" (</p>
--	--	---

		<pre>order_date, staff_id, customer_id, total_amount, status) VALUES (NOW(), <staff_id_from_token>, ?, ?, 'Confirmed') RETURNING order_id; ② Batch insert OrderItem FOR each item IN items[] LOOP INSERT INTO OrderItem (order_id, product_id, quantity, unit_price, line_total, fulfilled_qty, remaining_qty) VALUES (order_id, item.product_id, item.qty, item.unit_price, item.line_total, 0, item.qty); END LOOP;</pre>
--	--	---

Usecase name	ทำ request	
Usecase description	สร้างคำขอเบิกในตาราง Request ให้หัวหน้าช่างตรวจ/อนุมัติ	
Actor	Staff (role = Technician)	
Precondition	1. ผู้ใช้ล็อกอินเป็นระเบียบ Staff ที่ Active 2. ระบบทราบ role ของผู้ใช้จาก Staff.role 3. ผู้มีสิทธิ “ยื่นคำขอ” คือ Technician	
Normal Flow	Action	System
	1. เลือกเมนู Create Request 3. กดปุ่มค้นหาเลือก Order ที่ต้องการอ้างอิง 5. พิมพ์ค้นหา/กรอง Order 7. เลือก Order และดูรายการสินค้า 9. กดเลือกสินค้า + กรอก qty ที่ต้องการเบิก	2. display create request form 4. preload Order ที่สถานะยัง available (เฉพาะ Confirmed) SELECT order_id ,order_date ,customer_id ,total_amount ,STATUS FROM "Order" WHERE STATUS = 'Confirmed'; 6. Filter order ที่ preload มาแล้ว 8. ระบบแสดง OrderItem ของ order_id ที่เลือก SELECT product_id ,quantity ,unit_price ,fulfilled_qty ,remaining_qty FROM OrderItem WHERE order_id = :order_id; 10. Validate

		<p>IF (product_id IS NULL): error</p> <p>IF (qty IS NULL OR qty <= 0): error</p> <p>IF (qty > remaining_qty): error "จำนวนที่ขอเกินกว่าที่เหลืออยู่"</p> <p>ELSE:</p> <pre> RequestItem.push({ product_id : product_id, requested_qty: qty, fulfilled_qty: 0, remaining_qty: qty }) </pre>
	11. เพิ่มสินค้าต่อ (loopข้อ)	<p>12. Application update list + คำนวณยอดรวม display สินค้าที่เลือก</p> <p>13. ระบบแสดง summary ของ RequestItem[] จาก object</p> <p>Validation</p> <p>IF (RequestItem IS EMPTY): error</p>
	14. ตรวจสอบรายการ Request	
	15. กด Submit Request	<p>16. Insert DB</p> <p>① Insert Request</p> <pre> INSERT INTO Request (request_date, status, order_id, staff_id, description) VALUES (SYSTEMDATE(), </pre>

		<pre> 'Awaiting Approval', ?, -- order_id <staff_id_from_token>, ? -- description) RETURNING request_id; ② While loop insert RequestItem FOR item IN ARRAY RequestItemArray LOOP INSERT INTO RequestItem (request_id, product_id, quantity, fulfilled_qty, remaining_qty) VALUES (request_id, item.product_id, item.requested_qty, 0, item.requested_qty); END LOOP; 17. ระบบ query Request + RequestItem (status='Pending') และแสดงผล SELECT r.request_id, r.request_date, r.status, r.order_id, ri.product_id, ri.quantity, ri.fulfilled_qty, ri.remaining_qty </pre>
--	--	--

	<p>6. กด Approve หรือ Reject</p>	<pre> ,fulfilled_qty ,remaining_qty FROM RequestItem WHERE request_id = :request_id; 7. Update Status: ① Approve UPDATE Request SET STATUS = 'Approved' ,approved_by = :staff_id ,approved_date = NOW() WHERE request_id = :request_id; ② Reject UPDATE Request SET STATUS = 'Rejected' ,approved_by = :staff_id ,approved_date = NOW() WHERE request_id = :request_id; </pre>
--	----------------------------------	--

Usecase name	เบิกของ	
Usecase description	สร้างคำขอเบิกวัสดุ/อุปกรณ์ โดยระบุรายการและจำนวนที่ต้องการ เพื่อให้หัวหน้างานตรวจสอบและอนุมัติการเบิก	
Actor	Staff (role = Warehouse)	
Precondition	1. ผู้ใช้ล็อกอินเป็นระเบียบ Staff ที่ Active 2. ระบบทราบ role ของผู้ใช้จาก Staff.role 3. มี Request ที่อยู่ในสถานะ Approved 4. มี product อยู่ใน stock 5. จำนวนที่เบิกไม่เกินจำนวนคงเหลือ	
Normal Flow	Action	System
	1. เลือกเมนู Fulfill Request	2. display form เบิกของ

		<p>3. preload Request ที่ status = 'Approved'</p> <pre>SELECT request_id ,request_date ,staff_id ,order_id ,STATUS FROM Request WHERE STATUS = 'Approved';</pre> <p>display request ที่โหลดมา</p>
	4. กดพิมพ์ค้นหา request	5. Filter จาก ข้อมูลที่ preload มา
	6. กดเลือก Request ที่ต้องการดำเนินการ	<p>7. ระบบแสดงรายละเอียด RequestItem ของ request_id นั้น</p> <pre>SELECT request_item_id ,product_id ,quantity ,fulfilled_qty ,remaining_qty FROM RequestItem WHERE request_id = :request_id;</pre>
	8. กรอกจำนวนที่จะเบิก	<p>9. Validate</p> <p>① IF (fulfill_qty IS NULL OR fulfill_qty <= 0): error "จำนวนที่เบิกไม่ถูกต้อง"</p> <p>② IF (fulfill_qty > remaining_qty): error "จำนวนที่เบิกไม่ถูกต้อง"</p> <p>③ ตรวจสอบ stock ปัจจุบันใน Product</p> <pre>SELECT quantity FROM Product WHERE product_id = :product_id;</pre> <p>IF fulfill_qty > Product.quantity: error "สินค้าไม่พอในคลัง"</p>
	10. กด Confirm Fulfillment	11. Update Fulfillment

		<p>① Update RequestItem</p> <pre>UPDATE RequestItem SET fulfilled_qty = fulfilled_qty + :fulfill_qty, remaining_qty = remaining_qty - :fulfill_qty WHERE request_item_id = :request_item_id;</pre> <p>② Update Stock ใน Product</p> <pre>UPDATE Product SET quantity = quantity - :fulfill_qty WHERE product_id = :product_id;</pre> <p>③ Insert ลง StockTransaction (OUT)</p> <pre>INSERT INTO StockTransaction (transaction_date, type, product_id, quantity, staff_id, note) VALUES (NOW(), 'OUT', :product_id, :fulfill_qty, :current_staff_id, CONCAT('Fulfill Request ', :request_id));</pre>
--	--	---

Usecase name	add stock (เพิ่มของ)
Usecase description	บันทึกการเพิ่มจำนวนสินค้าหรือวัสดุเข้าคลัง โดยระบุรายการและจำนวน เพื่อปรับปรุงสต็อกในระบบให้เป็นปัจจุบัน
Actor	Staff (role = Warehouse)

Precondition	1. ผู้ใช้ล็อกอินเป็นระเบียบ Staff ที่ Active 2. ผู้ใช้ต้องมี role เป็น Warehouse 3. สินค้าที่ต้องการเพิ่มต้องมีอยู่ในตาราง Product	
Normal Flow	Action	System
	1. เลือกเมนู Stock-In 3. กดเข้าช่องเลือก supplier 5. พิมพ์ค้นหา Supplier 7. กดเลือก Supplier 9. กดช่องค้นหา Product	2. display form เพิ่ม stock 4. preload Supplier SELECT supplier_id, supplier_name FROM Supplier ORDER BY supplier_name; display supplier ที่โหลดมา 6. Context Search จากรายการ preload ที่ดึงมาแล้ว 8. Validation IF supplier_id IS NULL: เลือก supplier IF supplier_id IS NOT NULL: display supplier ที่เลือกบน form 10. preload Product SELECT product_id, product_name, quantity FROM Product ORDER BY product_name

	<p>11. พิมพ์ค้นหา Product</p> <p>13. Staff เลือกสินค้า + ใส่จำนวน qty ที่รับเข้า</p> <p>15. กด Confirm Stock-In</p>	<p>LIMIT 50;</p> <p>12. ทำ Context Search จาก Product preload</p> <p>14. Validation IF product_id IS NULL: error “ไม่พบสินค้า” IF qty IS NULL OR qty <= 0: error “กรุณาใส่จำนวนสินค้า” IF qty > 0: display product ที่เลือกและข้อมูลบน form</p> <p>16. Update Stock UPDATE Product SET quantity = quantity + :qty WHERE product_id = :product_id;</p> <p>17. บันทึกประวัติ Stock Transaction INSERT INTO StockTransaction (transaction_date, type, product_id, quantity, staff_id, note) VALUES (NOW(), 'IN', :product_id, :qty, :staff_id, CONCAT('Stock-In from Supplier ', :supplier_id)</p>
--	---	---

);
--	--	----

Usecase name	create staffs	
Usecase description	เพิ่มข้อมูลพนักงานใหม่เข้าสู่ระบบ	
Actor	Staff (role = Admin)	
Precondition	1. ผู้ใช้ล็อกอินเข้าสู่ระบบและมีสิทธิ์เป็น admin 2. ระบบตรวจสอบว่าไม่มีบัญชีผู้ใช้ที่ซ้ำกัน (เช่น username หรือ staff_id ไม่ซ้ำ) 3. ต้องมีการระบุข้อมูลพื้นฐานครบถ้วน เช่น ชื่อผู้ใช้ และบทบาท 4. ฐานข้อมูลเชื่อมต่อและพร้อมสำหรับการบันทึกข้อมูลพนักงานใหม่	
Normal Flow	Action	System
	1. เลือกเมนู Create Staff 3. กรอกข้อมูล Staff ใหม่ 5. กด Confirm Create	2. display create staff form 4. Validation IF (staff_name IS NULL): error “กรุณาใส่ชื่อ” IF (role IS NULL): error “กรุณาเลือก role” IF (email ซ้ำใน DB): error "อีเมลนี้ถูกใช้แล้ว" 6. Insert ข้อมูลลงตาราง Staff sql INSERT INTO Staff (staff_name, role, phone, email, active) VALUES (:staff_name, :role,

		:phone, :email, TRUE) RETURNING staff_id; 7. ระบบ query Staff ที่เพิ่งสร้างมาแสดงผล SELECT staff_id, staff_name, role, phone, email, active FROM Staff WHERE staff_id = :new_staff_id;
	8. ตรวจสอบผลลัพธ์	

Usecase name	create customer/ new customer	
Usecase description	เพิ่มข้อมูลลูกค้าใหม่เข้าสู่ระบบ	
Actor	Staff (role = Sale)	
Precondition	1. ผู้ใช้ล็อกอินเข้าสู่ระบบและมีสิทธิ์เป็น Staff ที่มีสถานะ Active 2. ผู้ใช้ต้องมี role เป็น Sale 3. ต้องกรอกข้อมูลลูกค้าให้ครบถ้วน เช่น ชื่อ เบอร์โทร หรือที่อยู่ 4. ระบบตรวจสอบว่าไม่มีข้อมูลลูกค้าซ้ำ (เช่น เบอร์โทรศัพท์ หรือชื่อ-ที่อยู่เดิมในฐานข้อมูล) 5. ฐานข้อมูลเชื่อมต่อและพร้อมสำหรับการบันทึกข้อมูลลูกค้าใหม่	
Normal Flow	Action	System
	1. เลือกเมนู Create Customer	2. display create customer form

	<p>3. กรอกข้อมูลลูกค้าใหม่</p> <p>5. กด Confirm Create</p> <p>8. ตรวจสอบผลลัพธ์</p>	<p>4. Validation</p> <p>IF (customer_name IS NULL): error “กรุณาใส่ชื่อ”</p> <p>IF (phone หรือ email ซ้ำใน DB): error "ลูกค้านี้มีอยู่แล้ว"</p> <p>6. Insert ข้อมูลลงตาราง Customer INSERT INTO Customer (customer_name, address, phone, email) VALUES (:customer_name, :address, :phone, :email) RETURNING customer_id;</p> <p>7. ระบบ query Customer ที่เพิ่งสร้างมาแสดงผล SELECT customer_id, customer_name, address, phone, email FROM Customer WHERE customer_id = :new_customer_id;</p>
--	---	---

Usecase name	new supplier/ create supplier	
Usecase description	เพิ่มข้อมูลผู้จัดจำหน่ายสินค้าเข้าสู่ระบบ	
Actor	Staff (role = Warehouse)	
Precondition	1. ผู้ใช้ล็อกอินเข้าสู่ระบบและมีสิทธิ์เป็น Staff ที่มีสถานะ Active 2. ผู้ใช้ต้องมี role เป็น Warehouse 3. ต้องกรอกข้อมูลผู้จัดจำหน่ายครบถ้วน เช่น ชื่อบริษัท ที่อยู่ เบอร์ติดต่อ และอีเมล 4. ระบบตรวจสอบว่าไม่มีข้อมูลผู้จัดจำหน่ายซ้ำในฐานข้อมูล (เช่น ชื่อบริษัท หรือเบอร์โทรเดิม) 5. ฐานข้อมูลเชื่อมต่อและพร้อมสำหรับการบันทึกข้อมูลใหม่	
Normal Flow	Action	System
	1. เลือกเมนู Create Supplier 3. กรอกข้อมูล Supplier ใหม่ 5. กด Confirm Create	2. display create supplier form 4. Validation IF (supplier_name IS NULL): error “กรุณาใส่ชื่อ” IF (email ซ้ำใน DB): error "Supplier นี้มีอยู่แล้ว" 6. Insert ข้อมูลลงตาราง Supplier INSERT INTO Supplier (supplier_name, address, phone, email) VALUES (:supplier_name, :address, :phone, :email) RETURNING supplier_id; 7. ระบบ query Supplier ที่เพิ่งสร้างมาแสดงผล

		SELECT supplier_id, supplier_name, address, phone, email FROM Supplier WHERE supplier_id = :new_supplier_id;
	8. ตรวจสอบผลลัพธ์	

Usecase name	new product	
Usecase description	เพิ่มข้อมูลสินค้าหรือวัสดุใหม่ที่ยังไม่มีอยู่ในระบบ	
Actor	Staff = (role = Warehouse)	
Precondition	1. ผู้ใช้ล็อกอินเข้าสู่ระบบและมีสิทธิ์เป็น Staff ที่มีสถานะ Active 2. ผู้ใช้ต้องมี role เป็น Warehouse 3. ต้องกรอกข้อมูลสินค้าครบถ้วน เช่น ชื่อสินค้า ราคา หนวยนับ และหมวดหมู่ 4. ระบบตรวจสอบว่าไม่มีชื่อสินค้าซ้ำในฐานข้อมูล 5. ฐานข้อมูลพร้อมสำหรับการเพิ่มข้อมูลสินค้าใหม่	
Normal Flow	Action	System
	1. เลือกเมนู Create Product 3. กดทดลองเลือก Supplier	2. display create product form 4. preload Supplier SELECT supplier_id, supplier_name FROM Supplier ORDER BY

	<p>5. พิมพ์ค้นหา supplier</p> <p>7. Staff กรอกข้อมูลสินค้าใหม่</p> <p>9. Staff กด Confirm Create</p>	<p>supplier_name;</p> <p>6. Filter จากที่ preload มา</p> <p>8. Validation IF (product_name IS NULL): error “กรุณาใส่ชื่อ” IF (price_per_unit <= 0): error “กรุณาใส่ราคาให้ถูกต้อง” IF (supplier_id IS NULL): error "กรุณาเลือก Supplier</p> <p>10. Insert ข้อมูลลงตาราง Product INSERT INTO Product (product_name, description, unit, price_per_unit, supplier_id, quantity) VALUES (:product_name, :description, :unit, :price_per_unit, :supplier_id, 0) RETURNING product_id;</p> <p>11. ระบบ query Product ที่เพิ่งสร้างมาแสดงผล SELECT product_id, product_name, description,</p>
--	--	---

		<div>unit, price_per_unit, supplier_id, quantity</div> <div>FROM</div> <div>Product</div> <div>WHERE</div> <div>product_id = :new_product_id;</div>
--	--	---

Usecase name	ปิด Order	
Usecase description	เปลี่ยนสถานะของคำสั่งซื้อ (Order) ที่ดำเนินการจัดส่งหรือรับของเรียบร้อยแล้ว ให้เป็นสถานะ “Closed” เพื่อระบุว่าการทำการรายการเสร็จสิ้น และไม่สามารถแก้ไขหรือเพิ่มรายการในคำสั่งซื้อนั้นได้อีก	
Actor	Staff (role = Sale)	
Precondition	<div>1. ผู้ใช้ล็อกอินเข้าสู่ระบบและมีสิทธิ์เป็น Staff ที่มีสถานะ Active</div> <div>2. ผู้ใช้ต้องมี role เป็น Sale</div> <div>3. คำสั่งซื้อต้องอยู่ในสถานะที่สามารถปิดได้ เช่น 'Confirmed'</div> <div>4. ระบบตรวจสอบว่าไม่มีคำขอเบิก (Request) หรือรายการสินค้าที่ยังค้างอยู่ในคำสั่งซื้อนั้น</div> <div>5. ฐานข้อมูลพร้อมสำหรับการอัปเดตสถานะของคำสั่งซื้อ</div> <div>6. การปิดคำสั่งซื้อจะถูกบันทึกเวลาและผู้ดำเนินการลงในระบบ</div>	
Normal Flow	Action	System
	<div>1. เลือกเมนู Close Order</div>	<div>2. preload เฉพาะ Order ที่เหลือสินค้า remaining_qty = 0 และสถานะ Pending</div> <div>SELECT</div> <div>o.order_id, o.order_date, o.status</div> <div>FROM</div> <div>"Order" AS o</div> <div>WHERE</div> <div>o.status = 'Pending Close'</div> <div>AND NOT EXISTS (</div>

		<pre> SELECT 1 FROM OrderItem AS oi WHERE oi.order_id = o.order_id AND oi.remaining_qty > 0); </pre>
	4. เลือก Order ที่ต้องการปิด	<p>3. แสดงรายการ Order ที่สามารถปิดได้</p> <p>5. แสดงรายละเอียดสินค้าใน Order</p> <pre> SELECT product_id, quantity, fulfilled_qty, remaining_qty FROM OrderItem WHERE order_id = :order_id; </pre>
	6. ตรวจสอบ Request ของ Order นี้	<p>7. ตรวจสอบว่ายังมี Request ค้างอยู่หรือไม่</p> <pre> SELECT COUNT(*) FROM Request WHERE order_id = :order_id AND status != 'Closed'; </pre> <p>ถ้า count > 0 → error "ยังมีคำขอเบิกสินค้าที่ยังค้างอยู่"</p>
	8. กด Confirm Close Order	<p>9. ปิดคำสั่งซื้อ</p> <pre> UPDATE "Order" SET status = 'Closed', closed_date = NOW(), </pre>

		<div>closed_by = :staff_id</div> <div>WHERE</div> <div>order_id = :order_id;</div>
--	--	--

Usecase name	login	
Usecase description	ผู้ใช้กรอกชื่อผู้ใช้ (username) และรหัสผ่าน (password) เพื่อเข้าสู่ระบบ	
Actor	All	
Precondition	<div>1. ต้องกรอกชื่อผู้ใช้และรหัสผ่านครบถ้วน</div> <div>2. ระบบตรวจสอบความถูกต้องของรหัสผ่านด้วยการเข้ารหัส (hash)</div> <div>3. หากล็อกอินสำเร็จ ระบบสร้าง session/token เพื่อระบุตัวตนในแต่ละคำขอ</div> <div>4. หากรหัสผ่านไม่ถูกต้องหรือตัวบัญชีถูกปิดใช้งาน ระบบแจ้งข้อผิดพลาดและไม่อนุญาตให้เข้าใช้งาน</div>	
Normal Flow	Action	System
	<div>1. กรอก username/email และ password</div> <div>3. กดปุ่ม Login</div>	<div>2. ตรวจสอบว่าช่อง username และ password ไม่ว่าง</div> <div>4. ระบบตรวจสอบผู้ใช้ในฐานข้อมูล</div> <div>SELECT</div> <div>staff_id,</div> <div>staff_name,</div> <div>role,</div> <div>active,</div> <div>password_hash</div> <div>FROM</div> <div>Staff</div> <div>WHERE</div> <div>email = :username</div> <div>OR staff_id = :username;</div> <div>5. IF (Username IS NULL):</div> <div>error "บัญชีผู้ใช้ไม่ถูกต้อง"</div> <div>6. ตรวจสอบ active</div>

	<p>9. เข้าสู่ระบบสำเร็จ</p>	<p>IF (active = FALSE): error "บัญชีถูกระงับ"</p> <p>7. ตรวจสอบ password IF hash(password_input) != password_hash: error "รหัสผ่านไม่ถูกต้อง"</p> <p>8. สร้าง Session/Token ให้ผู้ใช้เข้าระบบ</p> <p>10. Redirect ไปหน้าเมนูหลักตาม role</p>
--	-----------------------------	---

Usecase name	logout	
Usecase description	ผู้ใช้ทำการออกจากระบบเพื่อยุติการใช้งาน	
Actor	All	
Precondition	1. ผู้ใช้ต้องอยู่ในสถานะที่เข้าสู่ระบบแล้ว	
Normal Flow	Action	System
	1. กดปุ่ม Logout 4. ระบบ redirect ไปหน้า Login	2. ตรวจสอบว่ามี session/token อยู่หรือไม่ 3. ถ้าใช่ Session-Based DELETE FROM ActiveSession WHERE staff_id = :staff_id; 5. แสดงข้อความ "ออกจากระบบสำเร็จ"

Usecase name	ส่งของให้ช่าง (ปิด Request)
Usecase description	เมื่อสินค้าที่เบิกถูกรับเรียบร้อยแล้ว ระบบจะอัปเดตสถานะของคำขอเบิก (Request) เป็น “Closed” และบันทึกข้อมูลผู้ที่ดำเนินการจ่ายของ เพื่อระบุว่าการเบิกสินค้าสำเร็จและสิ้นสุดกระบวนการ

Usecase description	เมื่อสินค้าที่เบิกถูกรับเรียบร้อยแล้ว ระบบจะอัปเดตสถานะของคำขอเบิก (Request) เป็น “Closed” และบันทึกข้อมูลผู้ที่ดำเนินการจ่ายของ เพื่อระบุว่าการเบิกสินค้าสำเร็จและสิ้นสุดกระบวนการ
---------------------	---

[illegible]

	<p>6. ดราวละเอียดสินค้าใน Request</p> <p>7. กด Confirm ส่งของให้ช่าง</p>	<p>WHERE</p> <p>request_id = :request_id;</p> <p>8. อัปเดตสถานะ Request</p> <p>UPDATE Request</p> <p>SET</p> <p>status = 'Closed',</p> <p>closed_date = NOW(),</p> <p>closed_by = :staff_id</p> <p>WHERE</p> <p>request_id = :request_id;</p>
--	--	---