บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้องกับระบบงาน

โครงงานวิศวกรรมนี้มุ่งเน้นการพัฒนา “ระบบจัดการคำสั่งซื้อและตรวจสอบคลังสินค้า (Order Management and Inventory Control System: SOML)” เพื่อแก้ไขปัญหาคอขวดในกระบวนการปฏิบัติงานของธุรกิจค้าปลีกวัสดุก่อสร้างขนาดกลางและขนาดเล็ก กรณีศึกษาร้านวัสดุก่อสร้างอำพรคอนกรีต ซึ่งเดิมอาศัยกระบวนการทางเอกสารกระดาษที่ขาดการเชื่อมโยงระหว่างจุดชำระเงินหน้าร้านและจุดจ่ายสินค้าที่คลังสินค้า วัตถุประสงค์ของโครงงานคือการออกแบบและพัฒนาเว็บแอปพลิเคชันที่เชื่อมโยงกระบวนการหน้าร้านและคลังสินค้าเข้าด้วยกันแบบเรียลไทม์

โดยประยุกต์ใช้สถาปัตยกรรมซอฟต์แวร์แบบ 3 ชั้น ร่วมกับการสื่อสารข้อมูลแบบขับเคลื่อนด้วยเหตุการณ์ การควบคุมสถานะคำสั่งซื้อเพื่อป้องกันการรับสินค้าซ้ำ และมาตรฐานความมั่นคงปลอดภัยของแอปพลิเคชัน เพื่อให้ได้ระบบต้นแบบที่ใช้งานได้จริงและสามารถต่อยอดสู่การใช้งานเชิงพาณิชย์ได้ต่อไป

2.1.1 สถาปัตยกรรมซอฟต์แวร์แบบหลายชั้น (N-tier / 3-Tier Architecture)

สถาปัตยกรรมซอฟต์แวร์แบบหลายชั้นเป็นแนวทางการออกแบบที่แยกความรับผิดชอบของระบบออกเป็นชั้นตามหน้าที่การทำงาน ได้แก่ ชั้นการนำเสนอข้อมูล (Presentation Layer) ชั้นตรรกะทางธุรกิจ (Business Logic Layer) และชั้นข้อมูล (Data Layer) ตามหลักการแยกความรับผิดชอบ (Separation of Concerns) [1], [2] การแยกชั้นดังกล่าวช่วยให้แต่ละส่วนของระบบสามารถพัฒนา ทดสอบ ปรับขนาด และบำรุงรักษาได้อย่างเป็นอิสระจากกัน

ซึ่งเป็นแนวทางที่ได้รับการยอมรับอย่างแพร่หลายในการออกแบบเว็บแอปพลิเคชันระดับองค์กร [3] โครงงานนี้อ้างอิงแนวคิดดังกล่าวโดยแบ่งระบบออกเป็นชั้นการแสดงผลที่พัฒนาด้วยเว็บแอปพลิเคชัน Responsive Design ชั้นประมวลผลตรรกะธุรกิจที่ควบคุมการตัดสต็อกและการควบคุมสิทธิ์ และชั้นจัดเก็บข้อมูลเชิงสัมพันธ์ เพื่อให้ระบบมีความยืดหยุ่นและรองรับการต่อยอดในอนาคต

2.1.2 การออกแบบสถาปัตยกรรมด้วย C4 Model

C4 Model เป็นกรอบแนวคิดสำหรับอธิบายสถาปัตยกรรมซอฟต์แวร์แบบเป็นลำดับชั้น พัฒนาโดย Simon Brown ประกอบด้วย 4 ระดับ ได้แก่ ระดับบริบทของระบบ (Context) ระดับ

คอนเทนเนอร์ (Container) ระดับส่วนประกอบ (Component) และระดับรหัสต้นฉบับ (Code) [4] แนวทางนี้ช่วยให้ผู้พัฒนา ผู้ใช้งาน และผู้บริหารสามารถสื่อสารภาพรวมของระบบในระดับความละเอียดที่แตกต่างกันได้อย่างตรงกัน โดยไม่จำเป็นต้องลงรายละเอียดทางเทคนิคทั้งหมดในคราวเดียว

โครงงานนี้ใช้ C4 Model ในการอธิบายโครงสร้างของระบบ SOML สามระดับ ได้แก่ ระดับบริบท (C1) เพื่อแสดงขอบเขตของระบบและผู้เกี่ยวข้องภายนอก ระดับคอนเทนเนอร์ (C2) เพื่อแสดงการทำงานร่วมกันระหว่างเว็บแอปพลิเคชันฝั่งผู้ใช้ ระบบหลังบ้าน (Backend API) ฐานข้อมูล และบริการพิมพ์ใบเสร็จ และระดับส่วนประกอบ (C3) เพื่อแสดงการแบ่งส่วนประกอบภายในระบบหลังบ้านตามหน้าที่ ส่วนระดับรหัสต้นฉบับ (C4) ไม่ได้จัดทำเป็นแผนภาพ เนื่องจากรายละเอียดระดับนั้นปรากฏอยู่ในซอร์สโค้ดของระบบโดยตรงแล้ว

2.1.3 คุณสมบัติ ACID และธุรกรรมเชิงอะตอมมิกในฐานข้อมูล

ฐานข้อมูลเชิงสัมพันธ์ที่ใช้ในระบบงานที่มีการเคลื่อนไหวของข้อมูลตลอดเวลาจำเป็นต้องมีคุณสมบัติ ACID ได้แก่ Atomicity, Consistency, Isolation และ Durability เพื่อรับประกันความถูกต้องของธุรกรรม [5] โดยเฉพาะ Atomicity ซึ่งกำหนดให้ธุรกรรมต้องดำเนินการสำเร็จครบทั้งกระบวนการหรือไม่ดำเนินการเลย ช่วยป้องกันความคลาดเคลื่อนของข้อมูลเมื่อมีการเข้าถึงพร้อมกันจากผู้ใช้งานหลายคน โครงงานนี้นำหลักการดังกล่าวมาใช้ในขั้นตอนการตัดสต็อกสินค้าเมื่อพนักงานคลังสินค้ายืนยันจ่ายสินค้า เพื่อป้องกันภาวะ Race Condition และรับประกันว่ายอดสต็อกในระบบจะสะท้อนจำนวนสินค้าจริงในคลังอยู่เสมอ

2.1.4 การจัดการสถานะและวงจรชีวิตของคำสั่งซื้อ (Order Status Control / State Management)

แนวคิดการจัดการสถานะข้อมูล (State Management) เป็นการกำหนดสถานะและการเปลี่ยนผ่านสถานะของข้อมูลอย่างเป็นระบบ เพื่อควบคุมพฤติกรรมของระบบให้สอดคล้องกับวงจรชีวิตของข้อมูลนั้น ๆ ในบริบทของระบบขายและคลังสินค้า การกำหนดสถานะคำสั่งซื้อ เช่น “รอจ่ายสินค้า” และ “ส่งมอบสำเร็จ” ช่วยให้ระบบสามารถควบคุมได้ว่าคำสั่งซื้อใดยังรอการดำเนินการและคำสั่งซื้อใดเสร็จสิ้นแล้ว โครงงานนี้ประยุกต์ใช้แนวคิดดังกล่าวเพื่อป้องกันการรับสินค้าซ้ำ โดยเมื่อพนักงานคลังสินค้ายืนยันจ่ายสินค้าสำหรับคำสั่งซื้อใดแล้ว ระบบจะเปลี่ยนสถานะคำสั่งซื้อนั้นทันทีและไม่แสดงในคิวรอจ่ายอีกต่อไป โดยไม่ต้องอาศัยอุปกรณ์สแกนยืนยันสิทธิ์เพิ่มเติม

2.1.5 การสื่อสารข้อมูลแบบขับเคลื่อนด้วยเหตุการณ์ (Event-Driven Communication)

สถาปัตยกรรมแบบขับเคลื่อนด้วยเหตุการณ์เป็นแนวทางการออกแบบระบบที่ให้องค์ประกอบต่าง ๆ ของระบบสื่อสารกันผ่านการส่งและรับเหตุการณ์ (Event) แทนการร้องขอข้อมูลซ้ำเป็นระยะ (Polling) แนวคิดนี้เหมาะสมกับสถาปัตยกรรมแบบ Non-blocking I/O ของ Node.js ซึ่งถูกออกแบบมาให้จัดการการเชื่อมต่อจำนวนมากพร้อมกันได้อย่างมีประสิทธิภาพ [6] เมื่อมีเหตุการณ์เกิดขึ้นในระบบ เช่น การยืนยันคำสั่งซื้อที่หน้าร้าน ระบบสามารถผลักข้อมูล (Push) ไปยังผู้ใช้งานที่เกี่ยวข้องได้ทันทีโดยไม่ต้องให้ผู้ใช้งานกดรีเฟรชหน้าจอ โครงงานนี้ใช้แนวคิดนี้ในการส่งคำสั่งซื้อใหม่จากหน้าร้านไปยังหน้าจอคิวงานที่คลังสินค้าแบบเรียลไทม์ทันทีที่ลูกค้าชำระเงิน ทำให้พนักงานคลังสินค้าสามารถเริ่มจัดเตรียมสินค้าได้ก่อนที่ลูกค้าจะเดินถือใบเสร็จมาถึง

2.1.6 การควบคุมสิทธิ์การเข้าถึงตามบทบาทและมาตรฐานความมั่นคงปลอดภัย (RBAC & OWASP ASVS)

การควบคุมสิทธิ์การเข้าถึงตามบทบาท (Role-Based Access Control: RBAC) เป็นแนวทางการจัดการสิทธิ์ที่กำหนดขอบเขตการเข้าถึงข้อมูลและฟังก์ชันการทำงานตามบทบาทของผู้ใช้งาน ซึ่งเหมาะสมกับระบบที่มีผู้ใช้งานหลายกลุ่มที่มีหน้าที่แตกต่างกัน มาตรฐาน OWASP Application Security Verification Standard (ASVS) เป็นกรอบอ้างอิงที่ครอบคลุมการพิสูจน์ตัวตน การควบคุมสิทธิ์ และการปกป้องข้อมูลสำคัญของแอปพลิเคชันเว็บ [7] โครงงานนี้นำหลักการทั้งสองมาใช้ในการจำกัดสิทธิ์การเข้าถึงเมนูและข้อมูลของผู้ใช้งานแต่ละบทบาท ได้แก่ พนักงานขาย พนักงานคลังสินค้า และผู้จัดการ โดยพิสูจน์ตัวตนด้วยโทเคนที่ระบุบทบาทของผู้ใช้ จัดเก็บรหัสผ่านในรูปแบบที่ผ่านการแฮชแล้ว และตรวจสอบสิทธิ์ที่ฝั่งเซิร์ฟเวอร์ทุกครั้งก่อนเข้าถึงข้อมูล เพื่อป้องกันการเข้าถึงหรือแก้ไขข้อมูลที่อยู่นอกเหนือความรับผิดชอบของตนเอง

2.1.7 มาตรฐานการชำระเงินด้วย QR Code (EMVCo และพร้อมเพย์)

มาตรฐาน QR Code สำหรับการชำระเงินของประเทศไทยพัฒนาขึ้นบนพื้นฐานของ EMVCo QR Code Specification for Payment Systems: Merchant-Presented Mode ซึ่งเป็นมาตรฐานสากลที่เปิดให้นำไปประยุกต์ใช้ได้โดยไม่มีค่าลิขสิทธิ์ [8] ธนาคารแห่งประเทศไทยได้ประกาศแนวนโยบายมาตรฐาน Thai QR Code สำหรับธุรกรรมการชำระเงิน เพื่อให้ผู้ให้บริการชำระเงินทุกรายสามารถออก QR Code ที่รองรับการสแกนจ่ายข้ามผู้ให้บริการได้ [9] โดย QR Code จะบรรจุข้อมูลผู้รับเงินและยอดชำระตามรูปแบบที่กำหนด เพื่อให้แอปพลิเคชันธนาคารของผู้ซื้อสามารถอ่านและโอนเงินได้โดยอัตโนมัติ โครงงานนี้สร้างรหัส QR ตามมาตรฐานพร้อมเพย์เพื่อคำนวณและ

แสดงยอดชำระของแต่ละคำสั่งซื้อ โดยพนักงานขายเป็นผู้ตรวจสอบและยืนยันการรับเงินด้วยตนเองจากการแจ้งเตือนของแอปพลิเคชันธนาคาร

2.1.8 การรวบรวมข้อมูลเพื่อสนับสนุนการตัดสินใจ (Data Aggregation & Operational Dashboard)

การรวบรวมข้อมูล (Data Aggregation) เป็นกระบวนการดึงข้อมูลจากหลายแหล่งภายในระบบมาประมวลผลและนำเสนอในรูปแบบที่เข้าใจง่าย เพื่อสนับสนุนการตัดสินใจของผู้บริหาร แดชบอร์ดปฏิบัติการ (Operational Dashboard) เป็นเครื่องมือที่นำแนวคิดนี้มาใช้แสดงสถิติยอดขาย ระดับสต็อก และสถานะคิวงานแบบเรียลไทม์ในหน้าจอเดียว ช่วยลดเวลาที่ผู้บริหารต้องใช้ในการรวบรวมและสรุปข้อมูลด้วยตนเอง โครงงานนี้ใช้แนวคิดดังกล่าวในการพัฒนาแดชบอร์ดสำหรับผู้จัดการ พร้อมระบบแจ้งเตือนภายในแอปพลิเคชันเมื่อสต็อกสินค้าต่ำกว่าจุดสั่งซื้อเพิ่ม (Reorder Point) หรือมีคิวงานค้างนานผิดปกติ เพื่อให้ผู้บริหารสามารถตัดสินใจสั่งซื้อสินค้าจากข้อมูลจริงแทนการใช้ประสบการณ์และสัญชาตญาณ

2.1.9 การออกแบบประสบการณ์ผู้ใช้สำหรับระบบปฏิบัติการ (User Experience for Operational Systems)

ระบบที่ใช้งานหน้างานจริงในภาคปฏิบัติการ เช่น หน้าจอขายและหน้าจอคิวงานคลังสินค้า จำเป็นต้องออกแบบตามหลักปฏิสัมพันธ์ระหว่างมนุษย์กับคอมพิวเตอร์ (Human-Computer Interaction) โดยเน้นความเรียบง่าย ความชัดเจนของข้อมูล และการยืนยันก่อนดำเนินการที่ส่งผลต่อข้อมูลสำคัญ เพื่อลดความผิดพลาดจากการใช้งาน โครงงานนี้ออกแบบหน้าจอคิวงานคลังสินค้าให้แสดงรายการคำสั่งซื้อที่รอจ่ายอย่างชัดเจนเป็นลำดับ พร้อมปุ่มยืนยันจ่ายสินค้าที่ต้องมีการตรวจสอบรายการก่อนกดยืนยัน เพื่อให้พนักงานที่ไม่มีความรู้ด้านเทคโนโลยีมาก่อนสามารถใช้งานระบบได้อย่างรวดเร็วและถูกต้อง

2.1.10 การทดสอบและประกันคุณภาพซอฟต์แวร์ (Software Testing & Quality Assurance)

การประกันคุณภาพซอฟต์แวร์ครอบคลุมการทดสอบหลายระดับ ตั้งแต่การทดสอบระดับหน่วย (Unit Testing) การทดสอบการทำงานร่วมกันของส่วนประกอบ (Integration Testing) การทดสอบระดับระบบ (System Testing) และการทดสอบการยอมรับของผู้ใช้ (User Acceptance Test: UAT) [1], [2] โดยในแต่ละระดับยังครอบคลุมมิติของความถูกต้องเชิงฟังก์ชัน (Functional Testing) ประสิทธิภาพ (Performance Testing) และความสอดคล้องของข้อมูล (Data Integrity

Testing) การทดสอบเหล่านี้ช่วยยืนยันว่าระบบทำงานได้ตรงตามความต้องการ มีความเสถียรภายใต้สภาวะการใช้งานจริง และเป็นที่ยอมรับของผู้ใช้งานปลายทาง โครงงานนี้กำหนดแผนการทดสอบให้ครอบคลุมทั้งความถูกต้องของกรณีการใช้งาน ความสอดคล้องระหว่างยอดสต็อกกับสินค้าจริง ความเร็วในการส่งข้อมูลคำสั่งซื้อไปยังคิวคลังสินค้า และความพึงพอใจของพนักงานผู้ใช้งานจริงในสถานประกอบการ

2.2 เทคโนโลยีและเครื่องมือที่ใช้พัฒนาระบบ

2.2.1 วิชวลสตูดิโอโค้ด (Visual Studio Code: VS Code)

Visual Studio Code เป็นโปรแกรมแก้ไขซอร์สโค้ดที่มีความคล่องตัวสูงแต่ยังคงมีเครื่องมือสนับสนุนการเขียนโปรแกรมเทียบเท่าสภาพแวดล้อมการพัฒนาแบบครบวงจร เช่น การตรวจสอบรหัสคำสั่ง การเติมคำสั่งอัตโนมัติ และการจัดการไฟล์โครงการ [10] โปรแกรมนี้ใช้สถาปัตยกรรมแบบแยกส่วน ทำให้สามารถเพิ่มขยายความสามารถผ่านส่วนขยาย (Extension) ได้อย่างไม่จำกัด โครงงานนี้ใช้ Visual Studio Code เป็นเครื่องมือหลักในการพัฒนาโค้ดฝั่งหน้าเว็บ (React.js) ฝั่งหลังบ้าน (Node.js/Express.js) และการจัดการฐานข้อมูล (MySQL) ตลอดกระบวนการพัฒนาระบบ

2.2.2 รีแอ็กต์ (React.js)

React.js เป็นไลบรารีจาวาสคริปต์สำหรับพัฒนาส่วนติดต่อผู้ใช้ (User Interface) ที่ใช้แนวคิดการแบ่งส่วนประกอบของหน้าเว็บออกเป็นส่วนย่อยที่เรียกว่าคอมโพเนนต์ (Component) ซึ่งสามารถนำกลับมาใช้ซ้ำได้ [11] React.js ใช้แนวคิด Virtual DOM ในการอัปเดตส่วนติดต่อผู้ใช้อย่างมีประสิทธิภาพ ทำให้เหมาะสมกับการพัฒนาเว็บแอปพลิเคชันที่ต้องอัปเดตข้อมูลบ่อยครั้ง เช่น หน้าจอแสดงคิวงานคลังสินค้าแบบเรียลไทม์ โครงงานนี้ใช้ React.js ในการพัฒนาส่วนติดต่อผู้ใช้ทั้งสามบทบาท ได้แก่ หน้าจอขายหน้าร้าน หน้าจอคิวงานคลังสินค้า และแดชบอร์ดปฏิบัติการสำหรับผู้จัดการ ในรูปแบบ Responsive Web Application ที่รองรับทั้งคอมพิวเตอร์และอุปกรณ์เคลื่อนที่

2.2.3 โหนดเจเอส (Node.js)

Node.js เป็นสภาพแวดล้อมสำหรับการรันโค้ดจาวาสคริปต์ฝั่งเซิร์ฟเวอร์ที่ใช้สถาปัตยกรรมแบบ Non-blocking I/O และขับเคลื่อนด้วยเหตุการณ์ (Event-Driven) ทำให้สามารถจัดการการเชื่อมต่อจากผู้ใช้งานจำนวนมากพร้อมกันได้อย่างมีประสิทธิภาพโดยไม่ต้องสร้างเธรดใหม่สำหรับทุกการเชื่อมต่อ [6] คุณสมบัตินี้เหมาะสมอย่างยิ่งกับระบบที่ต้องผลักข้อมูลแบบเรียลไทม์ไปยังผู้ใช้งานหลายคนพร้อมกัน โครงงานนี้ใช้ Node.js เป็นสภาพแวดล้อมหลักในการพัฒนาระบบหลังบ้าน

(Backend) ที่ทำหน้าที่ประมวลผลตรรกะธุรกิจ ควบคุมสิทธิ์การเข้าถึง และผลักข้อมูลคำสั่งซื้อใหม่ไปยังหน้าจอคิวงานคลังสินค้าแบบเรียลไทม์

2.2.4 เอ็กซ์เพรสเจเอส (Express.js)

Express.js เป็นเฟรมเวิร์กสำหรับพัฒนาเว็บแอปพลิเคชันบน Node.js ที่ช่วยจัดการเส้นทางการเข้าถึง (Routing) มิดเดิลแวร์ (Middleware) และการตอบสนองต่อคำขอ HTTP ได้อย่างเป็นระบบ [12] เฟรมเวิร์กนี้มีโครงสร้างที่เรียบง่ายแต่มีความยืดหยุ่นสูง ทำให้ผู้พัฒนาสามารถออกแบบสถาปัตยกรรม API ได้ตามความต้องการของระบบ โครงงานนี้ใช้ Express.js ในการสร้าง RESTful API สำหรับการจัดการคำสั่งซื้อ ข้อมูลสินค้า และการควบคุมสิทธิ์การเข้าถึงตามบทบาทของผู้ใช้งานร่วมกับมิดเดิลแวร์สำหรับตรวจสอบสิทธิ์ก่อนเข้าถึงข้อมูลของแต่ละบทบาท

2.2.5 มายเอสคิวแอล (MySQL)

MySQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์แบบโอเพนซอร์สที่รองรับคุณสมบัติ ACID และภาษาสอบถามเชิงโครงสร้าง (SQL) สำหรับการจัดเก็บและเรียกใช้ข้อมูลอย่างมีโครงสร้าง [5] ระบบนี้ได้รับความนิยมอย่างแพร่หลายในการพัฒนาเว็บแอปพลิเคชันเนื่องจากมีเสถียรภาพสูงและสามารถทำงานร่วมกับเฟรมเวิร์กฝั่งเซิร์ฟเวอร์ได้หลากหลาย โครงงานนี้ใช้ MySQL ในการจัดเก็บข้อมูลสินค้า คำสั่งซื้อ ผู้ใช้งาน และประวัติการทำรายการ โดยควบคุมการตัดสต็อกด้วยธุรกรรมเชิงอะตอมมิกบนเอนจิน InnoDB ซึ่งรองรับการล็อกระดับแถวและการย้อนกลับธุรกรรม เพื่อป้องกันความคลาดเคลื่อนของข้อมูลเมื่อมีการยืนยันจ่ายสินค้าพร้อมกันหลายรายการ

2.2.6 การสื่อสารแบบเรียลไทม์ (Real-time Communication)

การสื่อสารแบบเรียลไทม์ระหว่างเว็บเบราว์เซอร์และเซิร์ฟเวอร์สามารถทำได้ผ่านการเชื่อมต่อแบบต่อเนื่อง ซึ่งอาศัยความสามารถของ Node.js ในการจัดการการเชื่อมต่อจำนวนมากพร้อมกันแบบไม่บล็อกการทำงาน [6] ช่องทางที่ใช้กันแพร่หลายคือเว็บซ็อกเก็ต (WebSocket) ซึ่งเปิดช่องทางสื่อสารสองทางค้างไว้ระหว่างเบราว์เซอร์กับเซิร์ฟเวอร์ และเหตุการณ์ที่ส่งจากเซิร์ฟเวอร์ (Server-Sent Events) ซึ่งเป็นช่องทางทางเดียวที่ใช้เป็นทางสำรองได้เมื่อเว็บซ็อกเก็ตเชื่อมต่อไม่สำเร็จ เมื่อมีเหตุการณ์เกิดขึ้นที่ฝั่งเซิร์ฟเวอร์ ข้อมูลจะถูกผลักไปยังฝั่งไคลเอนต์ที่เชื่อมต่ออยู่ได้ทันทีโดยไม่ต้องให้ฝั่งไคลเอนต์ส่งคำขอซ้ำเป็นระยะ โครงงานนี้ใช้เว็บซ็อกเก็ตเป็นช่องทางหลักและมีเหตุการณ์ที่ส่งจากเซิร์ฟเวอร์เป็นช่องทางสำรอง ในการส่งข้อมูลคำสั่งซื้อใหม่จากระบบหน้าร้านไปยังหน้าจอคิวงานคลังสินค้าทันทีที่ลูกค้าชำระเงินสำเร็จ ทำให้พนักงานคลังสินค้าเห็นคำสั่งซื้อในคิวได้โดยไม่ต้องกดรีเฟรชหน้าจอ

2.2.7 มาตรฐาน QR Code พร้อมเพย์ (PromptPay QR Generation)

การสร้างรหัส QR สำหรับรับชำระเงินตามมาตรฐานพร้อมเพย์อาศัยรูปแบบข้อมูลตามข้อกำหนด EMVCo QR Code Specification for Payment Systems: Merchant-Presented Mode ซึ่งประกอบด้วยข้อมูลผู้รับเงินและยอดชำระที่เข้ารหัสตามโครงสร้างมาตรฐาน [8] ข้อมูลถูกจัดเรียงเป็นชุดข้อมูลย่อยที่ระบุรหัสเขตข้อมูล ความยาว และค่าของข้อมูล และปิดท้ายด้วยค่าตรวจสอบความถูกต้องของข้อมูล (CRC) เมื่อลูกค้าสแกน QR Code ด้วยแอปพลิเคชันธนาคารของตนเอง แอปพลิเคชันจะอ่านและแสดงยอดเงินที่ต้องชำระโดยอัตโนมัติ โครงงานนี้สร้างรหัส QR ตามมาตรฐานดังกล่าวให้ตรงกับยอดชำระของแต่ละคำสั่งซื้อ โดยไม่มีการเชื่อมต่อ Payment Gateway หรือ API ธนาคารเพื่อตรวจสอบสถานะการชำระเงินอัตโนมัติ

2.2.8 เครื่องพิมพ์ใบเสร็จความร้อน (Thermal Receipt Printer)

เครื่องพิมพ์ใบเสร็จความร้อนเป็นอุปกรณ์ต่อพ่วงที่ใช้เทคโนโลยีการพิมพ์ด้วยความร้อนบนกระดาษเคลือบสารเคมีพิเศษ โดยไม่ต้องใช้หมึกพิมพ์ ทำให้พิมพ์ได้รวดเร็วและมีต้นทุนการใช้งานต่อแผ่นต่ำ อุปกรณ์นี้ใช้กันอย่างแพร่หลายในระบบขายหน้าร้านของร้านค้าปลีกและร้านสะดวกซื้อทั่วไป การสั่งงานเครื่องพิมพ์กลุ่มนี้ใช้ชุดคำสั่งมาตรฐาน ESC/POS ซึ่งเป็นการส่งลำดับไบต์ควบคุมการจัดรูปแบบ การเลือกตารางรหัสอักขระ และการตัดกระดาษ โดยข้อความภาษาไทยต้องแปลงเป็นตารางรหัสอักขระไทย (TIS-620) ก่อนส่งไปยังเครื่องพิมพ์ โครงงานนี้เชื่อมต่อเครื่องพิมพ์ใบเสร็จความร้อนเข้ากับระบบ เพื่อสั่งพิมพ์ใบเสร็จกระดาษระบุหมายเลขคำสั่งซื้อและรายการสินค้าให้ลูกค้าทันทีที่พนักงานยืนยันรับชำระเงิน ในลักษณะเดียวกับใบเสร็จของร้านสะดวกซื้อทั่วไป

2.3 งานวิจัยและระบบที่เกี่ยวข้อง (Related Works)

ในการดำเนินโครงงาน คณะผู้จัดทำได้สืบค้นงานวิจัยและระบบที่มีลักษณะใกล้เคียงกับระบบ SOML เพื่อนำมาเป็นแนวทางในการออกแบบและเปรียบเทียบผลลัพธ์ โดยเน้นระบบที่เกี่ยวข้องกับการจัดการสินค้าคงคลังของธุรกิจขนาดกลางและขนาดเล็ก (SME) ระบบเว็บแอปพลิเคชันสำหรับควบคุมสต็อก และระบบคิวงานที่ป้องกันการรับหรือจ่ายซ้ำ ซึ่งมีรายละเอียดดังนี้

Chin, Fang และ Abd Majid (2009) ได้พัฒนาระบบจัดการคลังสินค้าบนเว็บสำหรับธุรกิจขนาดกลางและขนาดเล็กในอุตสาหกรรมเครื่องมือก่อสร้างและเฟอร์นิเจอร์ของประเทศมาเลเซีย โดยประยุกต์ใช้แบบจำลองการทบทวนสต็อกเป็นระยะ (Periodic Review System) ร่วมกับฐานข้อมูล MySQL เพื่อคำนวณปริมาณสต็อกที่เหมาะสมและลดต้นทุนการถือครองสินค้า [13] งานวิจัยนี้มีความใกล้เคียงกับโครงงานเป็นอย่างมาก เนื่องจากเป็นกรณีศึกษาธุรกิจขนาดกลางและขนาดเล็กใน

อุตสาหกรรมวัสดุก่อสร้างเช่นเดียวกับร้านวัสดุก่อสร้างอำพรคอนกรีต และยืนยันว่าการนำระบบเว็บมาช่วยจัดการสต็อกสามารถลดปัญหาข้อมูลสินค้าคงคลังคลาดเคลื่อนได้อย่างมีนัยสำคัญ ซึ่งสนับสนุนแนวทางการพัฒนาระบบ SOML ที่ใช้สถาปัตยกรรมเว็บและฐานข้อมูลเชิงสัมพันธ์ในการควบคุมสต็อกแบบเรียลไทม์

Bai และ Zhong (2008) ได้ทำวิทยานิพนธ์ระดับปริญญาโทเกี่ยวกับการปรับปรุงการจัดการสินค้าคงคลังของธุรกิจขนาดเล็กในรูปแบบกรณีศึกษา โดยพบว่าธุรกิจขนาดเล็กส่วนใหญ่ยังพึ่งพาการบันทึกข้อมูลด้วยมือ ทำให้เกิดความล่าช้าในการตรวจสอบสต็อกและขาดข้อมูลที่แม่นยำสำหรับการตัดสินใจสั่งซื้อสินค้า [14] ผลการศึกษาชี้ให้เห็นว่าการนำระบบสารสนเทศมาใช้ในการติดตามสถานะสินค้าคงคลังแบบต่อเนื่องช่วยลดความคลาดเคลื่อนของข้อมูลและเพิ่มความสามารถในการวางแผนสั่งซื้อ ซึ่งสอดคล้องกับปัญหาที่ร้านวัสดุก่อสร้างอำพรคอนกรีตเผชิญอยู่ก่อนการพัฒนาระบบ SOML และยืนยันความจำเป็นของการเปลี่ยนจากกระบวนการกระดาษสู่ระบบดิจิทัลที่คณะผู้จัดทำได้ดำเนินการในโครงงานนี้

Tuan Mat และคณะ (2018) ได้ศึกษาอิทธิพลของเทคโนโลยีสารสนเทศ ทักษะความรู้ และทรัพยากรทางการเงินที่มีต่อแนวปฏิบัติในการจัดการสินค้าคงคลังของผู้ค้าปลีกขนาดกลางและขนาดเล็กในประเทศมาเลเซีย โดยพบว่าข้อจำกัดด้านทรัพยากรทางการเงินและทักษะดิจิทัลเป็นอุปสรรคสำคัญที่ทำให้ผู้ค้าปลีกขนาดเล็กจำนวนมากยังไม่สามารถนำระบบคอมพิวเตอร์มาใช้จัดการสต็อกได้อย่างเต็มรูปแบบ [15] ข้อค้นพบนี้มีความเกี่ยวข้องโดยตรงกับแนวทางการพัฒนาระบบ SOML ที่มุ่งเน้นการใช้เทคโนโลยีที่มีต้นทุนต่ำ เช่น ฐานข้อมูลโอเพนซอร์สและเว็บแอปพลิเคชันที่พัฒนาขึ้นเอง เพื่อให้ธุรกิจขนาดกลางและขนาดเล็กสามารถเข้าถึงระบบจัดการสต็อกที่มีประสิทธิภาพได้โดยไม่ต้องลงทุนสูง

Erameh และ Odoh (2021) ได้ออกแบบและพัฒนาระบบควบคุมสินค้าคงคลังบนเว็บโดยใช้ธุรกิจขนาดกลางและขนาดเล็กเป็นกรณีศึกษา ซึ่งครอบคลุมการบันทึกข้อมูลสินค้า การติดตามการเคลื่อนไหวของสต็อก และการออกรายงานสรุปยอดสำหรับผู้บริหาร [16] ผลการทดสอบระบบพบว่าสามารถลดระยะเวลาในการตรวจสอบสต็อกและลดความผิดพลาดจากการบันทึกข้อมูลด้วยมือได้อย่างชัดเจน ซึ่งมีแนวทางการพัฒนาที่สอดคล้องกับระบบ SOML ในด้านการใช้เว็บแอปพลิเคชันเป็นเครื่องมือหลักในการจัดการสต็อกสำหรับธุรกิจขนาดกลางและขนาดเล็ก แม้จะยังไม่มีการเชื่อมโยงข้อมูลแบบเรียลไทม์ไปยังจุดจ่ายสินค้าเช่นเดียวกับระบบที่พัฒนาในโครงงานนี้

วุฒิชัย ปลื้มมาลี (2563) ได้พัฒนาระบบบริหารจัดการคลังสินค้า กรณีศึกษาร้านฤทธิ์วิศวภัณฑ์ ซึ่งเป็นร้านจำหน่ายอุปกรณ์เครื่องปรับอากาศ โดยระบบแบ่งการทำงานออกเป็นสองส่วนหลัก คือ ส่วนที่เจ้าของร้านจัดการข้อมูลเครื่องปรับอากาศ ข้อมูลพนักงาน ข้อมูลรายการขาย และการออกรายงาน และส่วนที่พนักงานเพิ่ม แก้ไข ลบ และค้นหารายละเอียดสินค้า [17] ระบบนี้มีลักษณะใกล้

เคียงกับระบบ SOML ในแง่ที่เป็นโครงงานของนักศึกษาไทยที่พัฒนาระบบจัดการคลังสินค้าให้กับร้านค้าปลีกขนาดเล็กในลักษณะเดียวกัน แต่ยังเป็นระบบจัดการข้อมูลสต็อกและรายงานเป็นหลัก โดยไม่มีการเชื่อมโยงกระบวนการชำระเงินหน้าร้านกับคิวงานคลังสินค้าแบบเรียลไทม์ ซึ่งเป็นส่วนที่ระบบ SOML พัฒนาเพิ่มเติมเพื่อลดปัญหาคอขวดระหว่างจุดขายและจุดจ่ายสินค้า

งานวิจัยการพัฒนาระบบการส่งมอบยาที่มีคุณภาพสำหรับผู้ป่วยโรคเรื้อรังที่ส่งต่อรักษาต่อเนื่องในโรงพยาบาลส่งเสริมสุขภาพตำบล (2558) ได้ปรับปรุงกระบวนการจ่ายยาจากการพิมพ์ข้อมูลด้วยมือ มาเป็นระบบที่พิมพ์ฉลากยาได้โดยอัตโนมัติ พร้อมสรุปรายชื่อผู้ป่วยและรายการยาที่ต้องได้รับ [18] ผลการดำเนินงานพบว่าสามารถลดระยะเวลาในการจัดเตรียมยาลงจากเฉลี่ยหนึ่งชั่วโมงสามสิบนาทีเหลือเพียงสามสิบห้านาทีต่อแห่ง อีกทั้งยังลดค่าใช้จ่ายที่เกิดจากความซ้ำซ้อนของการจ่ายยาและไม่พบความคลาดเคลื่อนในการส่งมอบเกิดขึ้น แม้เป็นระบบในบริบทของงานบริการทางการแพทย์ แต่มีแนวคิดเชิงกระบวนการที่ใกล้เคียงกับระบบ SOML อย่างมาก โดยเฉพาะการเปลี่ยนจากกระบวนการเตรียมและส่งมอบด้วยมือ มาเป็นระบบที่มีการควบคุมสถานะรายการอย่างเป็นระบบเพื่อป้องกันการส่งมอบซ้ำ ซึ่งสอดคล้องกับหลักการควบคุมสถานะคำสั่งซื้อ (Order Status Control) ที่ระบบ SOML นำมาใช้ป้องกันการรับสินค้าซ้ำที่จุดจ่ายสินค้าของคลังสินค้า

Chaudhari (2024) ได้นำเสนอสถาปัตยกรรมระบบติดตามสถานะคำสั่งซื้อแบบเรียลไทม์สำหรับอีคอมเมิร์ซ โดยใช้ Node.js เป็นแกนหลักในการจัดการการเชื่อมต่อพร้อมกันจำนวนมาก ร่วมกับการสื่อสารผ่านเว็บซ็อกเก็ตเพื่อผลักสถานะคำสั่งซื้อไปยังหน้าจอผู้ใช้งานทันทีที่มีการเปลี่ยนแปลง [19] แนวทางสถาปัตยกรรมดังกล่าวมีความสอดคล้องโดยตรงกับการออกแบบระบบ SOML ในส่วนของการใช้ Node.js เป็นสภาพแวดล้อมหลักฝั่งเซิร์ฟเวอร์ และการผลักข้อมูลคำสั่งซื้อแบบเรียลไทม์ไปยังหน้าจอคิวงานคลังสินค้า จึงถูกนำมาใช้เป็นแนวทางอ้างอิงเชิงเทคนิคในการออกแบบสถาปัตยกรรมของระบบ SOML ทั้งนี้แหล่งดังกล่าวเป็นบทความเผยแพร่ทางเว็บไซต์ ไม่ใช่บทความวิจัยที่ผ่านการพิจารณาจากผู้ทรงคุณวุฒิ จึงใช้เป็นข้อมูลประกอบเชิงเทคนิคเท่านั้น

การเปรียบเทียบงานวิจัยและระบบที่เกี่ยวข้องกับระบบ SOML สรุปได้ดังตารางที่ 2.1

ตารางที่ 2.1 เปรียบเทียบงานวิจัยและระบบที่เกี่ยวข้องกับระบบ SOML

งานที่ศึกษา	บริบท	จัดการสต็อก	เชื่อมการชำระเงินหน้าร้าน	คิวงานเรียลไทม์	ป้องกันการรับของซ้ำ
Chin และคณะ (2009) [13]	ร้านวัสดุก่อสร้างและเฟอร์นิเจอร์ SME มาเลเซีย	มี	ไม่มี	ไม่มี	ไม่มี
Bai และ Zhong (2008) [14]	ธุรกิจขนาดเล็ก กรณีศึกษา	มี	ไม่มี	ไม่มี	ไม่มี

งานที่ศึกษา	บริบท	จัดการสต็อก	เชื่อมการชำระเงินหน้าร้าน	คิวงานเรียลไทม์	ป้องกันการรับของซ้ำ
Tuan Mat และคณะ (2018) [15]	ผู้ค้าปลีก SME มาเลเซีย (เชิงสำรวจ)	เชิงนโยบาย	ไม่มี	ไม่มี	ไม่มี
Erameh และ Odoh (2021) [16]	SME ทั่วไป	มี	ไม่มี	ไม่มี	ไม่มี
วุฒิชัย ปลื้มมาลี (2563) [17]	ร้านอุปกรณ์เครื่องปรับอากาศ	มี	ไม่มี	ไม่มี	ไม่มี
ระบบส่งมอบยา รพ.สต. (2558) [18]	บริการทางการแพทย์	ไม่มี	ไม่มี	ไม่มี	มี
Chaudhari (2024) [19]	อีคอมเมิร์ซ	ไม่มี	ไม่มี	มี	ไม่มี
ระบบ SOML (โครงงานนี้)	**ร้านวัสดุก่อสร้าง SME ไทย**	**มี**	**มี**	**มี**	**มี**

จากการทบทวนงานวิจัยและระบบที่เกี่ยวข้องข้างต้น สามารถสรุปได้ว่าระบบ SOML มีความแตกต่างจากงานวิจัยและระบบก่อนหน้าในประเด็นสำคัญ คือ การผสานรวมกระบวนการชำระเงินหน้าร้านด้วย QR พร้อมเพย์ การส่งคำสั่งซื้อเข้าคิวงานคลังสินค้าแบบเรียลไทม์ทันทีที่ชำระเงินสำเร็จ และการควบคุมสถานะคำสั่งซื้อเพื่อป้องกันการรับสินค้าซ้ำไว้ในระบบเดียวกันสำหรับธุรกิจค้าปลีกวัสดุก่อสร้างขนาดกลางและขนาดเล็กโดยเฉพาะ ดังที่แสดงในตารางที่ 2.1 ซึ่งจากการสืบค้นยังไม่พบงานวิจัยหรือระบบที่ครอบคลุมกระบวนการทั้งหมดนี้ไว้ในระบบเดียวในบริบทของธุรกิจขนาดกลางและขนาดเล็กของไทย

เอกสารอ้างอิง

[1] Sommerville I. Software engineering. 10th ed. Boston, MA: Pearson; 2015.

[2] Pressman RS, Maxim BR. Software engineering: a practitioner's approach. 8th ed. New York, NY: McGraw-Hill Education; 2014.

[3] Microsoft. N-tier architecture style. [online] 2024. สืบค้นจาก URL: https://learn.microsoft.com/azure/architecture/guide/architecture-styles/n-tier (สืบค้นเมื่อ 8 กันยายน 2569).

[4] Brown S. The C4 model for visualising software architecture. [online] 2020. สืบค้นจาก URL: https://c4model.com (สืบค้นเมื่อ 8 กันยายน 2569).

[5] Oracle Corporation. MySQL 8.0 reference manual. [online] 2024. สืบค้นจาก URL: https://dev.mysql.com/doc/refman/8.0/en/ (สืบค้นเมื่อ 8 กันยายน 2569).

[6] OpenJS Foundation. Node.js documentation. [online] 2024. สืบค้นจาก URL: https://nodejs.org/docs/ (สืบค้นเมื่อ 8 กันยายน 2569).

[7] OWASP Foundation. OWASP application security verification standard (ASVS). [online] 2024. สืบค้นจาก URL: https://owasp.org/www-project-application-security-verification-standard/ (สืบค้นเมื่อ 8 กันยายน 2569).

[8] EMVCo. EMV QR code specification for payment systems: merchant-presented mode. [online] 2020. สืบค้นจาก URL: https://www.emvco.com/specifications/ (สืบค้นเมื่อ 8 กันยายน 2569).

[9] ธนาคารแห่งประเทศไทย. แนวนโยบายการใช้มาตรฐาน Thai QR Code ในธุรกรรมการชำระเงิน. [online] 2562. สืบค้นจาก URL: https://www.bot.or.th/content/dam/bot/documents/th/our-roles/payment-systems/payment/payment-all-hearing/policy-guideline-thai-qr-code.pdf (สืบค้นเมื่อ 8 กันยายน 2569).

[10] Microsoft Corporation. Visual Studio Code documentation. [online] 2024. สืบค้นจาก URL: https://code.visualstudio.com/docs (สืบค้นเมื่อ 8 กันยายน 2569).

[11] Meta Open Source. React documentation: the library for web and native user interfaces. [online] 2024. สืบค้นจาก URL: https://react.dev (สืบค้นเมื่อ 8 กันยายน 2569).

[12] Express.js. Express — Node.js web application framework. [online] 2024. สืบค้นจาก URL: https://expressjs.com (สืบค้นเมื่อ 8 กันยายน 2569).

[13] Chin AJ, Fang CL, Abd Majid H. The application of web-based inventory management system for small and medium enterprise (SME/SMI): a case study for hardware and furniture industry in Malaysia. ใน: 3rd International Conference on Operations and Supply Chain Management. Malaysia; 2009.
[14] Bai L, Zhong Y. Improving inventory management in small business: a case study. วิทยานิพนธ์ปริญญามหาบัณฑิต. Jönköping International Business School; 2008. สืบค้นจาก URL: https://www.diva-portal.org/smash/get/diva2:3575/FULLTEXT01.pdf (สืบค้นเมื่อ 8 กันยายน 2569).
[15] Tuan Mat TZ, Md Johari NR, Abdul Azis MA, Hashim MR. Influence of information technology, skills and knowledge, and financial resources on inventory management practices among small and medium retailers in Malaysia. Asia-Pacific Management Accounting Journal. 2018. สืบค้นจาก URL: https://ir.uitm.edu.my/id/eprint/29581/ (สืบค้นเมื่อ 8 กันยายน 2569).
[16] Erameh KB, Odoh BI. Design and implementation of a web-based inventory control system using a small medium enterprise (SME) as a case study. NIPES Journal of Science and Technology Research. 2021;3(3):211-9. doi:10.37933/nipes/3.3.2021.21
[17] วุฒิชัย ปลื้มมาลี. ระบบบริหารจัดการคลังสินค้า (กรณีศึกษา ร้านฤทธิ์วิศวภัณฑ์). โครงงานคณะวิทยาศาสตร์ มหาวิทยาลัยสยาม; 2563. สืบค้นจาก URL: https://e-research.siam.edu/kb/warehouse-management-system/ (สืบค้นเมื่อ 8 กันยายน 2569).
[18] การพัฒนาระบบการส่งมอบยาที่มีคุณภาพในผู้ป่วยโรคเรื้อรังที่ส่งต่อรักษาต่อเนื่องในโรงพยาบาลส่งเสริมสุขภาพตำบล. ฐานข้อมูลงานวิจัยทางการแพทย์ Mednacea คณะเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยมหิดล. [online] 2558. สืบค้นจาก URL: https://mednacea.ict.mahidol.ac.th/project/2558000630 (สืบค้นเมื่อ 8 กันยายน 2569).
[19] Chaudhari R. Real-time order tracking system architecture. Medium. [online] 2024. สืบค้นจาก URL: https://medium.com/@myjob.rajesh/real-time-order-tracking-system-architecture-70a98e660906 (สืบค้นเมื่อ 8 กันยายน 2569).