โครงงานเลขที่ คพ. 2565/1

เรื่อง

ระบบจัดการร้านอาหารเดลิเวอรี่

Delivery Restaurant Manager

**โดย**

**นายธชิณ วิสุทธิมรรคกุล 6101240160**

**โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต**

**สาขาวิชาวิทยาการคอมพิวเตอร์ คณะบริหารธุรกิจ มหาวิทยาลัยพายัพ**

**พ.ศ.2565**

โครงงานเลขที่ คพ. 2565/1

เรื่อง

ระบบจัดการร้านอาหารเดลิเวอรี่

Delivery Restaurant Manager

**โดย**

**นายธชิณ วิสุทธิมรรคกุล 6101240160**

**โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต**

**สาขาวิชาวิทยาการคอมพิวเตอร์ คณะบริหารธุรกิจ มหาวิทยาลัยพายัพ**

**พ.ศ.2565**

หัวโครงงาน ระบบจัดการร้านอาหารเดลิเวอรี่

โดย นายธชิณ วิสุทธิมรรคกุล

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์พัฒน์นรี ศรีสมพันธ์

ปีการศึกษา 2565

สาขาวิทยาการคอมพิวเตอร์ คณะบริหารธุรกิจ มหาวิทยาลัยพายัพ อนุมัติให้นับโครงงานนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตร์บัณฑิต

.................................................... ประธานกรรมการ

( อาจารย์พัฒนรี ศรีสมพันธ์ )

คณะกรรมการสอบโครงงาน

อาจารย์ณัฐวุฒิ บุญโรจน์วงค์

อาจารย์นิภาภรณ์ เอื้อตรงจิตต์

อาจารนย์สุรเชษฐ์ วงศ์ชมภู

# ข้อตกลงในการใช้งานซอฟต์แวร์

โครงงานชื่อ ระบบจัดการร้านอาหารเดลิเวอรี่ (Delivery Restaurant Manager) นี้เป็นผลงานที่พัฒนาขึ้นโดย นายธชิณ วิสุทธิมรรคกุล รหัสนักศึกษา 6101240160 ภายใต้การดูแลของ อาจารย์พัฒน์นรี ศรีสมพันธ์ เป็นส่วนหนึ่งของการศึกษารายวิชา คพ.499 โครงงานคอมพิวเตอร์ 2 ตามหลักสูตร ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะบริหารธุรกิจ มหาวิทยาลัยพายัพ พ.ศ. 2565 โดยมีวัตถุประสงค์เพื่อให้นักศึกษาได้เรียนรู้และฝึกทักษะในการพัฒนาซอฟต์แวร์ ลิขสิทธิ์ของซอฟต์แวร์นี้จึงเป็นของผู้พัฒนา ซึ่งผู้พัฒนาได้อนุญาตให้ทางสาขาวิชาวิทยาการคอมพิวเตอร์ คณะบริหารธุรกิจมหาวิทยาลัยพายัพ เผยแพร่ซอฟต์แวร์นี้ตาม “ต้นฉบับ” โดยไม่มีการแก้ไขดัดแปลงใดๆ ทั้งสิ้นให้แก่บุคคลทั่วไปได้ใช้เพื่อประโยชน์ส่วนบุคคล หรือประโยชน์ทางการศึกษาที่ไม่มีวัตถุประสงค์ในเชิงพาณิชย์ โดยไม่คิดค่าตอบแทนการใช้ซอฟต์แวร์ ดังนั้นสาขาวิชาฯ จึงไม่มีหน้าที่ในการดูแล บำรุงรักษา จัดการอบรมการใช้งาน หรือพัฒนาประสิทธิภาพซอฟต์แวร์ รวมทั้งไม่รับรองความถูกต้องหรือประสิทธิภาพการทำงานของซอฟต์แวร์ ตลอดจนไม่รับประกันความเสียหายต่างๆ อันเกิดจากการใช้ซอฟต์แวร์นี้ทั้งสิ้น

# บทคัดย่อ

โครงงานระบบจัดการร้านอาหารเดลิเวอรี่ เป็นเว็บแอพพลิเคชั่นสำหรับการจัดการร้านอาหารที่เน้นไปที่การขายผ่านแพลตฟอร์ม (platform) ต่างๆเป็นหลัก เพื่อเก็บสถิติการขายและต้นทุนอาหารผ่านการขายในแพลตฟอร์มต่างๆ โดยการนำไปใช้ผ่านเว็บเบราว์เซอร์ (web brower) เป็นหลัก

โครงการนี้ ประกอบด้วยเนื้อหาสามส่วน ส่วนแรกเป็นการตั้งค่าข้อมูลต่างๆภายในร้านอาหาร ซึ่งประกอบด้วย 4 หัวข้อ คือ วัตถุดิบภายในร้าน อาหารภายในร้าน เมนูอาหารภายในร้าน และแพลตฟอร์มที่ร้านค้าเข้าร่วมรวมไปถึงราคาเมนูอาหารภายในแพลตฟอร์มต่างๆ ส่วนที่สองเป็นการลงบันทึกรายจ่ายและคำสั่งซื้อ ส่วนสุดท้ายคือรายงานผลการขาย

โปรแกรมนี้พัฒนาโดยใช้ภาษาจาวาสคริปต์ (JavaScript) เป็นหลัก โดยใช้ MERN Stack ในการพัฒนา ซึ่งประกอบด้วย ฐานข้อมูลแบบ NoSQL – MongoDB สำหรับเก็บข้อมูลอาหาร แพลตฟอร์ม รายจ่าย และคำสั่งซื้อ, ระบบ Node.js สำหรับรัน JavaScript เป็นเซิฟเวอร์, Express.js สำหรับการสร้าง Rest API เพื่อเชื่อมต่อกับฐานข้อมูล, React ซึ่งเป็น JavaScript Library สำหรับสร้างส่วนแสดงผลของเว็บแอพพลิเคชั่น

ผลที่ได้จากโครงงานนี้ ได้ช่วยให้ผู้ประกอบการร้านอาหารแบบออนไลน์จัดการรายจ่ายและคำสั่งซื้อได้สะดวกสบายมากยิ่งขึ้น ไม่ต้องขึ้นกับแอพพลิเคชั่นอื่นๆเพื่อเก็บส่วนต่างๆ, ลดค่าใช้จ่ายจากการจ่ายค่าสมาชิกรายเดือนในแอพพลิเคชั่นต่างๆ และมีสถิติการขายที่ชัดเจน, เข้าถึงง่าย

# Abstract

This ‘Delivery Restaurant Manager’ project aim to develop web applications. Which is use to manage restaurant that only serve via online delivery platform(s). To collect sales statistics and foods cost.

There are three parts of the project; setting up the store’s system, orders&expenses manager and sales report. In the first part ‘setting up’ are divided into 4 sections which are materials, food, menu and platform&pricing. The second part divided into 2 sections; expenses tracking and orders tracking. And lastly sales report which show how each platforms performs.

The project developed in all JavaScript languages with MERN Stack. Which contains MongoDB a NoSQL database, Express a backend framework, React a frontend library and Node.js for JavaScript environments.

The result of this application can help the store’s personnel manage expenses and orders easier; no more depends on the separate applications. Reduce subscription fees from which applications. And easily accessible sales report.

# กิตติกรรมประกาศ

ผู้จัดทำโครงงานขอขอบพระคุณอาจารย์พัฒน์นรี ศรีสมพันธ์ อาจารย์ที่ปรึกษาโครงงาน ที่คอยให้คำปรึกษา และคำแนะนำ ตลอดจนให้กำลังใจในการทำโครงงานนี้ด้วยดีตลอดมา รวมทั้งอาจารย์ประจำสาขาวิชาวิทยาการคอมพิวเตอร์ คณะบริหารธุรกิจมหาวิทยาลัยพายัพทุกท่าน ที่ได้ให้ความรู้ และสร้างประสบการณ์เกี่ยวกับคอมพิวเตอร์ในด้านต่างๆ นอกจากนี้ขอขอบคุณเพื่อนๆ รุ่นพี่และรุ่นน้อง สาขาวิชาวิทยาการคอมพิวเตอร์ได้คอยช่วยเหลือ และให้กำลังใจเสมอมา

โครงงานนี้คงจะสำเร็จลงมิได้ หากไม่ได้กำลังใจหลักจากคุณพ่อและคุณแม่ ที่คอยดูแลและสนับสนุนอย่างเต็มที่และสม่ำเสมอ

ธชิณ วิสุทธิมรรคกุล

# สารบัญ

[บทคัดย่อ ก](#_Toc118122011)

[Abstract ข](#_Toc118122012)

[กิตติกรรมประกาศ ค](#_Toc118122013)

[สารบัญ ง](#_Toc118122014)

[สารบัญรูปภาพ ฉ](#_Toc118122015)

[สารบัญตาราง ฌ](#_Toc118122016)

[บทที่ 1 บทนำ 1](#_Toc118122017)

[1.1. ความเป็นมาและความสำคัญของโครงงาน 1](#_Toc118122018)

[1.2. วัตถุประสงค์ของโครงงาน 1](#_Toc118122019)

[1.3. ขอบเขตโครงงาน 2](#_Toc118122020)

[1.4. ขอบเขตของโครงงาน 3](#_Toc118122021)

[1.5. กรอบแนวคิดการพัฒนาโครงงาน 5](#_Toc118122022)

[1.6. ประโยชน์ที่ได้รับ 5](#_Toc118122023)

[1.7. นิยามศัพท์เฉพาะ 6](#_Toc118122024)

[บทที่ 2 แนวคิดทฤษฎีและงานวิจัยที่เกี่ยวข้อง (Literature Review) 7](#_Toc118122025)

[2.1 แนวคิดทฤษฎี 7](#_Toc118122026)

[2.2. งานวิจัยที่เกี่ยวข้อง 16](#_Toc118122027)

[บทที่ 3 วิธีดำเนินโครงงาน (Research Methodology) 21](#_Toc118122028)

[3.1. ขั้นตอนการดำเนินงานตาม Software Process Model 21](#_Toc118122029)

[3.2. แผนภาพกระแสข้อมูล 23](#_Toc118122030)

[3.3. โครงสร้างซอฟต์แวร์ (Software Structure) 24](#_Toc118122031)

[3.4. ข้อกำหนดซอฟต์แวร์ (Input/Output Specifications) 26](#_Toc118122032)

[3.5. E-R Diagram 34](#_Toc118122033)

[3.6. ตารางฐานข้อมูล (Tables) 35](#_Toc118122034)

[3.7. ส่วนต่อประสาน (User Interface / Screen Layout) 40](#_Toc118122035)

[บทที่ 4 ผลการพัฒนาระบบ (Results) 53](#_Toc118122036)

[4.1. ผลการทดสอบ (Test report) 53](#_Toc118122037)

[4.2. ผลลัพธ์ของระบบที่พัฒนาในภาพรวม (Output) 69](#_Toc118122038)

[4.3. ผลประเมินความพึงพอใจของผู้ใช้ระบบ (Assessment) 74](#_Toc118122039)

[4.4. คู่มือการติดตั้ง (Installation) 75](#_Toc118122040)

[4.5. คู่มือการใช้งาน (User manual) 79](#_Toc118122041)

[บทที่ 5 บทสรุปและข้อเสนอแนะ (Conclusions and Recommendations) 92](#_Toc118122042)

[5.1. วัตถุประสงค์หรือประเด็นสำคัญในการพัฒนา 92](#_Toc118122043)

[5.2. ผลการพัฒนาสามารถแก้ปัญหาได้มากน้อยเพียงใด 93](#_Toc118122044)

[5.3. สิ่งที่คาดหวังที่ระบบจะสามารถช่วยในการทำงานได้มากน้อยเพียงใด 94](#_Toc118122045)

[5.4. ปัญหาและข้อเสนอแนะ 94](#_Toc118122046)

[บรรณานุกรม 97](#_Toc118122047)

[ประวัติย่อผู้พัฒนา 99](#_Toc118122048)

[ตัวอย่างเอกสาร 100](#_Toc118122049)

# สารบัญรูปภาพ

[รูปประกอบ 1.1 Waterfall model 5](#_Toc118121792)

[รูปประกอบ 1.2 Gantt Chart 5](file:///D:\Projects\DeliveryRestaurantManager\Documents\Final\CS499_Documentations.docx#_Toc118121793)

[รูปประกอบ 3.1 Dataflow Diagram 23](#_Toc118121794)

[รูปประกอบ 3.2 Software Structure 24](#_Toc118121795)

[รูปประกอบ 3.3 Entity Relationship Diagram 34](#_Toc118121796)

[รูปประกอบ 3.4 Login UI 40](#_Toc118121797)

[รูปประกอบ 3.5 UI ตางรางสถิติการขาย 41](#_Toc118121798)

[รูปประกอบ 3.6 UI กราฟสถิติการขาย 42](#_Toc118121799)

[รูปประกอบ 3.7 UI การจัดการคำสั่งซื้อ 43](#_Toc118121800)

[รูปประกอบ 3.8 UI ตารางบันทึกรายจ่าย 44](#_Toc118121801)

[รูปประกอบ 3.9 UI การสร้างรายจ่าย 45](#_Toc118121802)

[รูปประกอบ 3.10 UI การแก้ไขรายจ่าย 45](#_Toc118121803)

[รูปประกอบ 3.11 UI การสร้างและลบหมวดหมู่รายจ่าย 46](#_Toc118121804)

[รูปประกอบ 3.12 UI การจัดการวัตถุดิบ 47](#_Toc118121805)

[รูปประกอบ 3.13 UIการแก้ไขวัตถุดิบ 48](#_Toc118121806)

[รูปประกอบ 3.14 UI ตารางรายการอาหาร 48](#_Toc118121807)

[รูปประกอบ 3.15 UI การสร้างอาหาร 49](#_Toc118121808)

[รูปประกอบ 3.16 UI การแก้ไข้อาหาร 49](#_Toc118121809)

[รูปประกอบ 3.17 UI ตารางและการลบเมนูอาหาร 50](#_Toc118121810)

[รูปประกอบ 3.18 UIการสร้างเมนูอาหาร 50](#_Toc118121811)

[รูปประกอบ 3.19 UIการแก้ไขเมนูอาหาร 51](#_Toc118121812)

[รูปประกอบ 3.20 UI แพลตฟอร์มและตารางราคา 51](#_Toc118121813)

[รูปประกอบ 3.21 UI การสร้างแพลตฟอร์ม 52](#_Toc118121814)

[รูปประกอบ 4.1 การทำงานในภาพรวม 69](#_Toc118121815)

[รูปประกอบ 4.2 การเก็บสูตรอาหาร 70](#_Toc118121816)

[รูปประกอบ 4.3 การตั้งราคาแยกตามแพลตฟอร์ม 71](#_Toc118121817)

[รูปประกอบ 4.4 การสร้างคำสั่งซื้อ 72](#_Toc118121818)

[รูปประกอบ 4.5 การทำซ้ำคำสั่งซื้อ 72](#_Toc118121819)

[รูปประกอบ 4.6 การค้นหาข้อมูลในตาราง 73](#_Toc118121820)

[รูปประกอบ 4.7 การบันทึกคำสั่งซื้อ 73](#_Toc118121821)

[รูปประกอบ 4.8 กราฟแสดงสถิติการขาย 74](#_Toc118121822)

[รูปประกอบ 4.9 คำสั่งสำหรับติดตั้งโปรแกรม 76](#_Toc118121823)

[รูปประกอบ 4.10 ตัวอย่างไฟล์ .env 77](#_Toc118121824)

[รูปประกอบ 4.11 คำสั่งเริ่มต้นระบบ 77](#_Toc118121825)

[รูปประกอบ 4.12 ตัวอย่างเมื่อเริ่มต้นระบบสำเร็จ 78](#_Toc118121826)

[รูปประกอบ 4.13 ตัวอย่างหน้าเข้าสู่ระบบ 78](#_Toc118121827)

[รูปประกอบ 4.14 ตัวอย่างการกรอกข้อมูลหน้าสมัครสมาชิก 79](#_Toc118121828)

[รูปประกอบ 4.15 ลักษณะของหน้าเข้าสู่ระบบ 79](#_Toc118121829)

[รูปประกอบ 4.16 ตำแหน่งเมนู Material 80](#_Toc118121830)

[รูปประกอบ 4.17 การสร้างวัตถุดิบ 81](#_Toc118121831)

[รูปประกอบ 4.18 ตำแหน่งเมนู Item 81](#_Toc118121832)

[รูปประกอบ 4.19 ตำแหน่งปุ่มสร้างอาหาร 82](#_Toc118121833)

[รูปประกอบ 4.20 Searchable dropdown 82](#_Toc118121834)

[รูปประกอบ 4.21 ตำแหน่งเมนู Rider&Pricing 84](#_Toc118121835)

[รูปประกอบ 4.22 ตำแหน่งปุ่มสร้างแพลตฟอร์ม 84](#_Toc118121836)

[รูปประกอบ 4.23 การเลือกไรเดอร์ 85](#_Toc118121837)

[รูปประกอบ 4.24 ตำแหน่งในการแก้ไขราคา 85](#_Toc118121838)

[รูปประกอบ 4.25 ตำแหน่งเมนู Expense 86](#_Toc118121839)

[รูปประกอบ 4.26 ตำแหน่งปุ่มแก้ไขหมวดหมู่รายจ่าย 86](#_Toc118121840)

[รูปประกอบ 4.27 ตำแหน่งปุ่มสร้างคำสั่งซื้อ 87](#_Toc118121841)

[รูปประกอบ 4.28 ตำแหน่งเมนู Order 88](#_Toc118121842)

[รูปประกอบ 4.29 ส่วนสำหรับเลือกไรเดอร์ 88](#_Toc118121843)

[รูปประกอบ 4.30 ส่วนสำหรับเลือกเมนูอาหาร 89](#_Toc118121844)

[รูปประกอบ 4.31 ตำแหน่งเมนู Dashboard 89](#_Toc118121845)

[รูปประกอบ 4.32 การเลือกไรเดอร์ใน Dashboard 90](#_Toc118121846)

[รูปประกอบ 4.33 การเลือกขอบเขตเวลา 90](#_Toc118121847)

[รูปประกอบ 4.34 กราฟสถิติการขาย 91](#_Toc118121848)

# สารบัญตาราง

[ตาราง 3.1 Login module 26](#_Toc118121849)

[ตาราง 3.2 Authentication module 26](#_Toc118121850)

[ตาราง 3.3 โมดูลเพิ่มวัตถุดิบ 27](#_Toc118121851)

[ตาราง 3.4 โมดูลแก้ไขวัตถุดิบ 27](#_Toc118121852)

[ตาราง 3.5 โมดูลลบวัตถุดิบ 27](#_Toc118121853)

[ตาราง 3.6 โมดูลเพิ่มอาหาร 27](#_Toc118121854)

[ตาราง 3.7 โมดูลแก้ไขอาหาร 28](#_Toc118121855)

[ตาราง 3.8 โมดูลลบอาหาร 28](#_Toc118121856)

[ตาราง 3.9 โมดูลเพิ่มเมนูอาหาร 28](#_Toc118121857)

[ตาราง 3.10 โมดูลแก้ไขเมนูอาหาร 29](#_Toc118121858)

[ตาราง 3.11 โมดูลลบเมนูอาหาร 29](#_Toc118121859)

[ตาราง 3.12 โมดูลเพิ่มไรเดอร์ 29](#_Toc118121860)

[ตาราง 3.13 โมดูลลบไรเดอร์ 30](#_Toc118121861)

[ตาราง 3.14 โมดูลแก้ไขราคาเมนูอาหาร 30](#_Toc118121862)

[ตาราง 3.15 โมดูลเพิ่มหมวดหมู่รายจ่าย 30](#_Toc118121863)

[ตาราง 3.16 โมดูลแก้ไขหมวดหมู่รายจ่าย 30](#_Toc118121864)

[ตาราง 3.17 โมดูลลบหมวดหมู่รายจ่าย 31](#_Toc118121865)

[ตาราง 3.18 โมดูลเพิ่มรายจ่าย 31](#_Toc118121866)

[ตาราง 3.19 โมดูลแก้ไขรายจ่าย 31](#_Toc118121867)

[ตาราง 3.20 โมดูลลบรายจ่าย 32](#_Toc118121868)

[ตาราง 3.21 โมดูลทำซ้ำรายจ่าย 32](#_Toc118121869)

[ตาราง 3.22 โมดูลเพิ่มคำสั่งซื้อ 32](#_Toc118121870)

[ตาราง 3.23 โมดูลลบคำสั่งซื้อ 33](#_Toc118121871)

[ตาราง 3.24 โมดูลแสดงสถิติการขาย 33](#_Toc118121872)

[ตาราง 3.25 ฐานข้อมูลผู้ใช้ 35](#_Toc118121873)

[ตาราง 3.26 ฐานข้อมูลวัตถุดิบ 35](#_Toc118121874)

[ตาราง 3.27 ฐานข้อมูลอาหาร 36](#_Toc118121875)

[ตาราง 3.28 ฐานข้อมูลเมนูอาหาร 37](#_Toc118121876)

[ตาราง 3.29 ฐานข้อมูลแพลตฟอร์มการขาย 38](#_Toc118121877)

[ตาราง 3.30 ฐานข้อมูลหมวดหมู่รายจ่าย 38](#_Toc118121878)

[ตาราง 3.31 ฐานข้อมูลรายจ่าย 39](#_Toc118121879)

[ตาราง 3.32 ฐานข้อมูลคำสั่งซื้อ 40](#_Toc118121880)

[ตาราง 4.1 ข้อมูลการทดสอบ 53](#_Toc118121881)

[ตาราง 4.2 ทดสอบ ระบบตรวจสอบตัวตน 54](#_Toc118121882)

[ตาราง 4.3 ทดสอบ การแสดงสถิติการขาย 55](#_Toc118121883)

[ตาราง 4.4 ทดสอบ เพิ่มคำสั่งซิ้อ 56](#_Toc118121884)

[ตาราง 4.5 ทดสอบ ลบคำสั่งซื้อ 57](#_Toc118121885)

[ตาราง 4.6 ทดสอบ สร้างหมวดหมู่รายจ่าย 57](#_Toc118121886)

[ตาราง 4.7 ทดสอบ ลบหมวดหมู่รายจ่าย 57](#_Toc118121887)

[ตาราง 4.8 ทดสอบ สร้างรายจ่าย 60](#_Toc118121888)

[ตาราง 4.9 ทดสอบ ลบรายจ่าย 60](#_Toc118121889)

[ตาราง 4.10 ทดสอบ ทำซ้ำรายจ่าย 61](#_Toc118121890)

[ตาราง 4.11 ทดสอบ แก้ไขรายจ่าย 61](#_Toc118121891)

[ตาราง 4.12 ทดสอบ เพิ่มวัตถุดิบ 62](#_Toc118121892)

[ตาราง 4.13 ทดสอบ แก้ไขวัตถุดิบ 62](#_Toc118121893)

[ตาราง 4.14 ทดสอบ ลบวัตถุดิบ 63](#_Toc118121894)

[ตาราง 4.15 ทดสอบ เพิ่มอาหาร 63](#_Toc118121895)

[ตาราง 4.16 ทดสอบ แก้ไขอาหาร 64](#_Toc118121896)

[ตาราง 4.17 ทดสอบ ลบอาหาร 64](#_Toc118121897)

[ตาราง 4.18 ทดสอบ เพิ่มเมนูอาหาร 65](#_Toc118121898)

[ตาราง 4.19 ทดสอบ แก้ไขเมนูอาหาร 66](#_Toc118121899)

[ตาราง 4.20 ทดสอบ ลบเมนูอาหาร 66](#_Toc118121900)

[ตาราง 4.21 ทดสอบ เพิ่มแพลตฟอร์ม 67](#_Toc118121901)

[ตาราง 4.22 ทดสอบ ลบแพลตฟอร์ม 67](#_Toc118121902)

[ตาราง 4.23 ทดสอบ ตั้งราคาเมนูอาหาร 68](#_Toc118121903)

[ตาราง 4.24 แบบประเมิณความพึงพอใจของพนักงานร้าน 74](#_Toc118121904)

[ตาราง 4.25 แบบประเมิณความพึงพอใจของเจ้าของร้าน 75](#_Toc118121905)

# บทนำ

## ความเป็นมาและความสำคัญของโครงงาน

จากเหตุการณ์การแพร่ระบาดของเชื้อโรคโควิด-19 ส่งผลให้ธุรกิจส่งอาหาร เดลิเวอรี่เติบโตมากขึ้น สะดวกสบายมากขึ้น และมีแพลตฟอร์มสำหรับรับ-ส่งอาหารเหล่านี้เกิดขึ้นมากมาย ทำให้ผู้ประกอบการจำนวนมากหันมาสนใจใน แพลตฟอร์มเหล่านี้ และเกิดเป็นธุรกิจร้านอาหารแบบใหม่ ที่ไม่มีหน้าร้าน ไม่มีโต๊ะอาหาร รับคำสั่งซื้อผ่านระบบ online delivery อย่างเดียว

ในปัจจุบันแพลตฟอร์ม การส่งอาหารออนไลน์ในประเทศไทยมีหลากหลายเป็นอย่างมาก เช่น GrabFood, LinemanWongnai, ShopeeFood, Robinhood โดยแต่ละบริษัทก็มีระบบการให้บริการที่แตกต่างกันไป ส่งผลให้การการคิดรายรับ รายจ่ายต่างๆในร้านค้าเกิดความซับซ้อนมากกว่าการขายหน้าร้านปกติทั่วไป

ผู้จัดทำโครงการจึงมีแนวคิดในการพัฒนา “ระบบจัดการร้านอาหารเดลิเวอรี่” โดยมุ่งเป้าไปที่การสร้างระบบหลังบ้านที่สามารถจัดการร้านอาหารที่ใช้ระบบส่งออนไลน์เป็นหลักได้อย่างมีประสิทธิภาพ สามารถทำบัญชีและจัดการเงินเข้าออกจากแอปพลิเคชั่น ต่างๆได้ง่าย โดยมีการเพิ่ม กฎการคิดบัญชีของแต่ละแอปพลิเคชั่นขึ้นมา ซึ่งถือเป็น feature สำคัญของระบบ เก็บบันทึกสถิติการขายเมนูต่างๆของทุกแอปพลิเคชั่น เพื่อทำแบบวิเคราะห์การขายและออกโปรโมชั่น ในเบื้องต้นของการพัฒนาโครงงานนี้จะใช้โมเดลธุรกิจของร้านขายอาหารดอง ‘สมายดอง ’ เป็นหลักในการพัฒนา ซึ่งเป็นร้านประเภทที่ไม่มีหน้าร้านไม่มีโต๊ะอาหารมีเพียงบริการเดลิเวอรี่เท่านั้น

## วัตถุประสงค์ของโครงงาน

ระบบจัดการร้านอาหารเดลิเวอรี่สามารถ

1. จัดเก็บข้อมูลสูตรอาหารและวัตถุดิบต่างๆภายในร้าน
2. กำหนดราคาอาหารและค่าธรรมเนียมของแต่ละแพลตฟอร์มแยกกันได้
3. บันทึกรายจ่ายภายในร้านโดยแบ่งประเภทของรายจ่ายได้ถามที่ผู้ใช้ต้องการ
4. บันทึกคำสั่งซื้อได้ง่ายและรวดเร็ว ผ่านการคลิ๊กหรือสัมผัสโดยไม่ต้องพิมพ์ค่าต่างๆ
5. แสดงข้อมูลสถิติคำสั่งซื้อจากแพลตฟอร์มต่างๆได้ตามระยะเวลาที่กำหนด
6. แสดงข้อมูลรายจ่ายภายในร้านเป็นตารางที่สามารถจัดเรียงหรือกรองข้อมูลได้ตามที่ต้องการ

## ขอบเขตโครงงาน

### ด้านประชากร

1.) เจ้าของร้าน ‘สมายดอง’

2.) พนักงานร้าน ‘สมายดอง’

### ด้านเทคโนโลยีที่ใช้ในการพัฒนาโครงการ

Hardware

- Desktop Computer (RAM 16GB, AMD FX-8320E, NVIDIA GeForce GTX 1050 Ti)

- Smart Phone (Xiaomi 10T pro)

2.) Software

- Microsoft Windows 10 Pro build 19043

- Visual Studio Code 1.66.2

- Notepad++ 8.3.3

- Github Desktop 3.1.2

- MongoDB Atlas

- NodeJS 16.14.2 LTS

- Express.js 4.0

- React 18.0.0

- Redux 8.0.2

- ReduxjsToolkit 1.8.3

- MaterialUI 5.10.10

- Axios 1.1.3

- JsonWebToken 8.5.1

- Mongoose 6.4.5

- dayjs 1.11.5

- Google Charts 4.0.0

3.2.3 Technology

- Model View Controller

- REST API

- Cloud Database

- Node.JS

- Framework

- HTTPS Request

- State management

- Immutable data

## ขอบเขตของโครงงาน

1. ระบบตรวจสอบสิทธิในการเข้าใช้งานซอฟต์แวร์ (Authentication)
2. การตั้งค่าร้านอาหาร
   1. ตั้งค่าสูตรอาหาร
      1. จัดการวัตถุดิบและบรรจุภัณฑ์
         1. จัดเก็บ ชื่อ และหน่วย
         2. สามารถ สร้าง แก้ไข ลบ และค้นหาได้
      2. จัดการอาหาร
         1. จัดเก็บ ชื่อ และวัตถุดิบที่ประกอบอาหาร
         2. สามารถ สร้าง แก้ไข และลบได้
      3. จัดการเมนูอาหาร
         1. จัดเก็บ ชื่อ อาหารในเมนู ค่าแรง และค่าน้ำมัน
         2. สามารถ สร้าง แก้ไข และลบได้
   2. ตั้งค่าแพลตฟอร์ม
      1. จัดเก็บ ชื่อ ค่าธรรมเนียม ภาษี ค่าน้ำมัน ราคาของเมนูต่างๆ
      2. สามารถ สร้าง และลบแพลตฟอร์มได้
      3. จัดการราคาของเมนูต่างๆของแต่ละแพลตฟอร์มได้
3. การจัดการคำสั่งซื้อและรายจ่ายภายในร้าน
   1. รายจ่าย
      1. การจัดการรายจ่าย
         1. จัดเก็บหมวดหมู่ของรายจาย คำอธิบาย จำนวน ราคา และวันที่รายจ่ายเกิด
         2. สามารถ สร้าง แก้ไข ลบ และค้นหาได้
      2. การจัดการหมวดหมู่รายจ่าย
         1. แต่ละหมวดหมู่สามารถมีหมวดหมู่รองได้
         2. สามารถ สร้าง แก้ไข และลบได้
         3. สามารถสร้างรายการซ้ำได้ง่าย
   2. การจัดการคำสั่งซื้อ
      1. จัดเก็บ แพลตฟอร์ม รายการอาหาร จำนวน และวันที่เกิดคำสั่งซื้อ
      2. สามารถ เพิ่ม ลบ และค้นหาได้
      3. การเพิ่มจะต้องสามารถเพิ่มได้โดยไม่จำเป็นต้องใช้แป้นพิมพ์
4. สามารถแสดงสถิติการขาย
   1. สามารถแสดงสถิติรวม หรือแยกแต่ละแพลตฟอร์มได้
   2. สามารถเลือกขอบเขตเวลาเป็น วัน สัปดาห์ เดือน หรือปีได้
   3. สามารถแสดงข้อมูลเป็นตารางได้
   4. สามารถแสดงข้อมูลเป็นกราฟแท่งได้

## กรอบแนวคิดการพัฒนาโครงงาน

### แบบจำลองกระบวนการ (Software Process Model)

เลือกใช้ Waterfall Model ซึ่งประกอบไปด้วย การวิเคราะห์ความต้องการ การออกแบบระบบ การพัฒนาระบบ การทดสอบระบบ การนำระบบไปใช้ และการปรับปรุงแก้ไข เนื่องจากต้องจัดทำเอกสารโครงการ ซึ่งจำเป็นต้องมีการวางแผนการพัฒนาล่วงหน้าอยู่แล้ว โมเดล Waterfall ที่ต้องการการวางแผนล่วงหน้าจึงจัดได้ว่าเหมาะสมกับรูปแบบการทำงานมาก

Diagram

Description automatically generated

รูปประกอบ 1.1 Waterfall model

### Diagram Description automatically generatedแผนภูมิแกนต์ (Gantt Chart)

รูปประกอบ 1.2 Gantt Chart

## ประโยชน์ที่ได้รับ

1.) ร้านอาหารมีระบบฐานข้อมูลสำหรับเก็บสูตรอาหาร

2.) ร้านอาหารสามารถบันทึกคำสั่งซื้อได้ง่าย

3.) ร้านอาหารสามารถบันทึกรายจ่ายโดยแบ่งเป็นประเภทได้

4.) ร้านอาหารมีสถิติการขายที่สามารถดูแยกตามแพลตฟอร์มและระยะเวลาได้

## นิยามศัพท์เฉพาะ

ผู้ใช้ หมายถึง ผู้จัดการหรือเจ้าของร้านสมายดอง

ลูกค้า หมายถึง ผู้ใช้บริการแพลตฟอร์มส่งอาหาร

ไรเดอร์ หมายถึง แพลตฟอร์มหรือบริษัทที่รับบทบาทในการรับและส่งอาหารจากร้านอาหารไปยังลูกค้า เช่น Grab, Lineman, ShopeeFood, FoodPanda

ออเดอร์ หมายถึง คำสั่งซื้อที่ร้านค้าได้รับจากไรเดอร์ โดยในคำสั่งซื้อจะระบุถึงเมนูอาหาร และจำนวนที่ลูกค้าต้องการ

# แนวคิดทฤษฎีและงานวิจัยที่เกี่ยวข้อง (Literature Review)

## แนวคิดทฤษฎี

### ภาษา HTML

**เนื้อหา :**

HTML ย่อมาจาก Hypertext Markup Language มันคือภาษาเขียนเว็บไซต์ที่ใช้เพื่อกำกับข้อมูลต่างๆ และแสดงผลคำสั่งบนหน้า Web Browser เป็นเหมือนภาษาพื้นฐานที่มีไว้ในการพัฒนาหน้าเว็บไซต์ ในปัจจุบันอยู่ภายใต้การดูแลของ องค์กร World Wide Web Consortium (W3C) และถูกพัฒนามาถึงเวอร์ชั่น 5 หรือที่เรียกว่า HTML5 โดยเริ่มจาก HTML1 ที่ทำอะไรได้ไม่มากนอกจากโบว์ชัวร์ออนไลน์

ข้อมูลต่างๆ บนหน้าเว็บไซต์จะถูกเชื่อมโยงกันด้วยชุดคำสั่งต่างๆ เพื่อให้แสดงผลออกมาในรูปแบบที่นักออกแบบต้องการให้เป็น ข้อมูลเหล่านั้นถูกควบคุมโดยการเขียน HTML ทั้งที่เป็นรูปภาพ ข้อความ หรือวัตถุอื่นๆบนหน้าเว็บไซต์

โครงสร้างของ HTML จะเป็นในรูปแบบของ Tag ต่างๆ และ Web Browser จะแปลความของ Tag แต่ละ Tag ออกมาเป็นหน้าตาเว็บไซต์

**แหล่งอ้างอิง :**

Jirayut Intachai. (2564). **HTML คืออะไร? ทำไมคนเขียนเว็บไซต์ต้องใช้งาน.** จาก Terrestrial: https://goterrestrial.com/2021/05/19/what-is-html/

**สรุปความรู้ :**

HTML เป็นภาษาที่ความซับซ้อนต่ำและไม่มี Dynamic แต่เป็นรากฐานสำคัญในการเขียนเว็บแอปพลิเคชั่น เพราะเป็นส่วนที่ใช้กำหนดโครงสร้างหลักของเว็บ

### ภาษา CSS

**เนื้อหา :**

CSS คือ ภาษาที่ใช้สำหรับตกแต่งเอกสาร HTML/XHTML ให้มีหน้าตา สีสัน ระยะห่าง พื้นหลัง เส้นขอบและอื่นๆ ตามที่ต้องการ CSS ย่อมาจาก Cascading Style Sheets มีลักษณะเป็นภาษาที่มีรูปแบบในการเขียน Syntax แบบเฉพาะและได้ถูกกำหนดมาตรฐานโดย W3C เป็นภาษาหนึ่งในการตกแต่งเว็บไซต์ ได้รับความนิยมอย่างแพร่หลาย

CSS มีประโยชน์อย่างหลากหลาย ซึ่งได้แก่

1. ช่วยให้เนื้อหาภายในเอกสาร HTML มีความเข้าใจได้ง่ายขึ้นและในการแก้ไขเอกสารก็สามารถทำได้ง่ายกว่าเดิม เพราะการใช้ CSS จะช่วยลดการใช้ภาษา HTML ลงได้ในระดับหนึ่ง และแยกระหว่างเนื้อหากับรูปแบบในการแสดงผลได้อย่างชัดเจน
2. ทำให้สามารถดาวน์โหลดไฟล์ได้เร็ว เนื่องจาก code ในเอกสาร HTML ลดลง จึงทำให้ไฟล์มีขนาดเล็กลง
3. สามารถกำหนดรูปแบบการแสดผลจากคำสั่ง style sheet ชุดเดียวกัน ให้มีการแสดงผลในเอกสารแบบเดียวทั้งหน้าหรือในทุกๆ หน้าได้ ช่วยลดเวลาในการปรับปรุงและทำให้การสร้างเอกสารบนเว็บมีความรวดเร็วยิ่งขึ้น นอกจากนี้ยังสามารถควบคุมการแสดงผล ให้คล้ายหรือเหมือนกันได้ในหลาย Web Browser
4. ช่วยในการกำหนดการแสดงผลในรูปแบบที่มีความเหมาะกับสื่อต่างๆเป็นอย่างดี
5. ทำให้เว็บไซต์มีความเป็นมาตรฐานมากขึ้นและมีความทันสมัย รองรับการใช้งานในอนาคตได้ดี

**แหล่งอ้างอิง :**

Kipakaporn. (2561**). CSS คืออะไร มีประโยชน์ อย่างไร.** จาก SoGoodWeb: https://blog.sogoodweb.com/Article/Detail/79237/CSS-คืออะไร-มีประโยชน์-อย่างไร

**สรุปความรู้ :**

ภาษา CSS เป็นภาษาสำหรับตกแต่งและออกแบบ UI ของ เว็บแอปพลิเคชั่น แต่ตัวภาษาเองไม่สามารถทำงานได้หากขาดโครงสร้างเว็บจาก HTML

ภาษา CSS มีเครื่องมือให้เลือกใช้เป็นจำนวนมาก ทำให้เป็นภาษาที่เรียนรู้ขั้นพื้นฐานนั้นง่าย แต่การจะเข้าใจและใช้ได้อย่างเต็มประสิทธิภาพต้องอาศัยประสบการณ์และความรู้ความเข้าใจเป็นอย่างมาก

### ภาษา JavaScript

**เนื้อหา :**

JavaScript คือ ภาษาคอมพิวเตอร์ภาษาหนึ่งที่มีไว้เพื่อจัดการเอฟเฟต์หรือพฤติกรรมของหน้าเว็บไซต์ นอกเหนือจาก HTML ที่ใช้จัดการเรื่องของเนื้อหาของเว็บและ CSS ที่ใช้จัดการเรื่องของโครงสร้างและดีไซน์ของเว็บ

JavaScript ยังสามารถใช้งานในส่วนของโปรแกรมบนคอมพิวเตอร์และเซิฟเวอร์ ยกตัวอย่างเช่น Node.js หรือในฐานข้อมูลก็สามารถใช้งานได้เช่นกัน เช่น MongoDB และ CouchDB

JavaScript สามารถช่วยเราจัดการงานต่างๆ โดยไม่จำเป็นที่จะต้องทำการโหลดหน้าเว็บใหม่ซ้ำๆ หากเราเคยเห็นการกดปุ่ม Button แล้ว ข้อความบางข้อความแสดงผลหรือซ่อนข้อความนั้นโดยไม่จำเป็นต้องโหลดหน้านั้นใหม่อีกครั้ง หรือมีการเปลี่ยนสีของตัวอักษร เปลี่ยนแปลงเนื้อหา นั่นละครับ คือ หน้าเว็บไซต์นั้นๆ ถูกเรียกใช้งานด้วย JavaScript แล้ว

**แหล่งอ้างอิง :**

Worapong. (2563). **JavaScript คืออะไร ใช้งานอย่างไร.** จาก WebDoDee: https://www.webdodee.com/what-is-javascript/

**สรุปความรู้ :**

ภาษา JavaScript เป็นภาษาสำหรับเขียนเว็บไซต์ที่สามารถเรียกได้ว่าเป็นภาษาสำหรับเขียนโปรแกรมจริงๆ เมื่อเทียบกับ HTML ที่ไม่สามารถเรียกได้ว่าภาษาสำหรับเขียนโปรแกรมได้เต็มปาก

แรกเริ่มเดิมที JavaScript ถูกพัฒนาเพื่อช่วยให้สามารถปรับปรุง/แก้ไขหน้าเว็บได้โดยที่ไม่ต้องโหลดหน้าเว็บใหม่อีกครั้ง โดยการแก้ไขส่วนต่างๆของ HTML โดยตรง

ปัจจุบันภาษาถูกพัฒนาจนสามารถใช้ในการเขียนโปรแกรมได้โดยตรงโดยไม่ต้องผ่าน web browser ทำให้การใช้งานมีความหลากหลายมากขึ้น ใช้เขียนโปรแกรมก็ได้ ใช้เป็น Backend ก็ได้ JavaScript จึงได้รับความนิยมมากยิ่งขึ้น

### Framework

**เนื้อหา :**

เฟรมเวิร์ก (Framework) หรือ โครงร่างซอฟต์แวร์ เป็นขอบเขตของระบบงาน ที่เป็นรูปแบบที่นำกลับมาใช้ใหม่ได้สำหรับระบบซอฟต์แวร์ (หรือระบบย่อย) ซึ่งสามารถอยู่ในรูปของคลาสนามธรรม (Abstract class) และกับวิธีในการใช้ตัวตน (instance) ของคลาสร่วมกันจำเพาะสำหรับซอฟต์แวร์ชนิดใดชนิดหนึ่ง โครงร่างซอฟต์แวร์ทุกโครงร่างใช้การออกแบบเชิงวัตถุ(Object-oriented programming, OOP)

โปรแกรมของโครงร่างซอฟต์แวร์มักจะเป็นโปรแกรมเชิงวัตถุ และ PHP Framework ในปัจจุบัน ส่วนใหญ่อยู่ภายใต้แนวคิด MVC (Model-view-controller)

**แหล่งอ้างอิง :**

**Framework คืออะไร โครงร่างหรือขอบเขตของระบบงาน.** (2561). จาก MindPHP: https://www.mindphp.com/developer/25-oop-php-framework/1957-framework-คืออะไร.html

**สรุปความรู้ :**

เฟรมเวิร์กคือโครงสร้างสำหรับการเขียนโปรแกรม ช่วยให้โปรแกรมมีมาตรฐานและเป็นระเบียบ และ Framework ส่วนใหญ่ยังนำเสนอเครื่องมือและโมดูลต่างๆที่ช่วยให้การเขียนโปรแกรมง่ายขึ้นหรือดีขึ้นอีกด้วย

### MVC Model

**เนื้อหา :**

MVC เป็นตัวย่อของคำว่า Model View Controller ใช้เรียกรูปแบบการพัฒนาซอฟต์แวร์ที่มีโครงสร้างซึ่งแบ่งออกมาเป็น 3 ส่วนหลัก ตามตัวย่อของชื่อ รูปแบบการพัฒนาซอฟต์แวร์แบบ MVC ถูกนำไปใช้ในขั้นตอนการพัฒนาหลากหลายภาษา เพราะ MVC เป็นเพียงหลักการออกแบบโปรแกรม ( Design Pattern ) รูปแบบหนึ่งเท่านั้น ซึ่งเป็นที่นิยมมากในการนำมาพัฒนาแอปพลิเคชั่น ซอฟต์แวร์แต่ละแพลตฟอร์ม และประยุกต์ใช้ในอีกหลาย ๆ ด้าน

model คือส่วนของการเก็บรวบรวมข้อมูล ไม่ว่าข้อมูลนั้น ๆ จะถูกจัดเก็บในรูปแบบใดก็ตาม ในฐานข้อมูลแบบเป็น Object Class หรือที่นิยมเรียกกันว่า VO ( Value Object ) หรือเก็บเป็นไฟล์ข้อมูลเลยเมื่อข้อมูลถูกโหลดเข้ามาจากที่ต่าง ๆ และเข้ามายังส่วนของโมเดล ตัวโมเดลจะทำการจัดการตระเตรียมข้อมูลให้เป็นรูปแบบที่เหมาะสม เพื่อรอการร้องขอข้อมูลจากส่วนของ Controller

view คือส่วนของการแสดงผล หรือส่วนที่จะปฏิสัมพันธ์กับผู้ใช้งาน ( User Interface ) หน้าที่ของ view ในการเขียนโปรแกรมแบบ MVC คือคอยรับคำสั่งจากส่วนของ Controller และ End User เริ่มแรกเลยตัววิวอาจจะได้รับคำสั่งจาก Controller ให้แสดงผลหน้า Home และเมื่อผู้ใช้งานหน้าเว็บกดปุ่มสั่งซื้อ View จะส่งข้อมูลไปให้ Controller เพื่อประมวลผลและแสดงบางอย่างจาก Action นั้น

controller คือส่วนของการเริ่มทำงาน และรับคำสั่ง โดยที่คำสั่งนั้นจะเกิดขึ้นในส่วนการติดต่อกับผู้ใช้งานคือ view เมื่อผู้ใช้งานทำการ Interactive กับ UI view จะเกิดเหตุการณ์หรือข้อมูลบางอย่างขึ้น ตัววิวจะส่งข้อมูลนั้นมายัง controller ตัว controller จะทำการประมวลผลโดยบางคำสั่งอาจจะต้องไปติดต่อกับ model ก่อน เพื่อทำการประมวลผลข้อมูลอย่างถูกต้องเรียบร้อยแล้วก็จะส่งไปยัง view เพื่อแสดงผลตามคำสั่งที่ end user ร้องขอมา Controller จะทำหน้าที่เป็นตัวกลางระหว่าง Model และ View ให้ทำงานร่วมกันอย่างมีประสิทธิภาพและตรงกับความต้องการของ End User มากที่สุด

**แหล่งอ้างอิง :**

MVC คืออะไร ทำความเข้าใจรูปแบบการเขียนโปรแกรม. (2559). จาก บริษัท โค๊ดบี จำกัด: h ttps://www.codebee.co.th/labs/mvc-คืออะไร-ทำความเข้าใจรู/

**สรุปความรู้ :**

MVC คือรูปแบบการพัฒนาซอฟต์แวร์โดยการแบ่งโครงสร้างออกเป็น 3 ส่วน คือ

1. Model ส่วนของการกำหนดรูปแบบการจัดเก็บข้อมูล

2. View ส่วนของการแสดงผล

3. Controller ส่วนของการประมวลผล

### ฐานข้อมูล MongoDB

**เนื้อหา :**

MongoDB เป็น open-source document database โดยเป็นฐานข้อมูลแบบ NoSQL คือไม่มี relation (ความสัมพันธ์) ของตารางแบบ SQL ทั่วๆไป แต่จะเก็บข้อมูลเป็นแบบ JSON (JavaScript Object Notation) แทน การบันทึกข้อมูลทุกๆ record ใน MongoDB เราจะเรียกมันว่า Document ซึ่งจะเก็บค่าเป็น key และ value จะเห็นว่ามันก็คือ JSON นั่นแหละ

การเก็บข้อมูล document ใน MongoDB จะถูกเก็บไว้ใน Collections (เปรียบเทียบได้กับ Table ใน Relational Database ทั่วๆไป) แต่แตกต่างกันที่ collection ไม่จำเป็นที่จะต้องมี schema เหมือนกันก็สามารถบันทึกข้อมูลได้

Schemaless คือการไม่ต้องกำหนดโครงสร้างใดๆให้มันเหมือน SQL ปกติทั่วไป เช่น collection User มีเก็บแค่ name ต่อมาเราสามารถเพิ่มการเก็บ position เข้ามาได้เลย

**แหล่งอ้างอิง :**

Chai Phonbopit. (2558). **MongoDB คืออะไร? + สอนวิธีใช้งานเบื้องต้น.** จาก Devahoy: https://www.devahoy.com/blog/2015/08/getting-started-with-mongodb

**สรุปความรู้ :**

MongoDB คือ ระบบฐานข้อมูล opensource แบบ NoSQL ที่มีการจัดเก็บข้อมูลแบบ Document ซึ่งเก็บค่าเป็น Key และ Value คล้ายกับระบบ JSON

MongoDB เป็นฐานข้อมูลแบบ Schemaless คือ ไม่มีการกำหนดโครงสร้างข้อมูล ทำให้ฐานข้อมูลมีความยืดหยุ่นสูงและขยายได้ง่าย

### Framework Express.js

**เนื้อหา :**

Express.js เอ็กเพรส ดอทเจเอส เป็น Web Application Framework ชื่อดังที่ได้รับความนิยมมาก สำหรับทำงานบน platform ของ Node.js ซึ่งเป็น Server ตัวหนึ่ง โดยทั้ง Express.js และ Node.js ต่างก็ใช้ภาษา Javascript ในการพัฒนา ถ้าเป็น Web Application Framework ในสมัยก่อน คนที่พัฒนาจะต้องมีความรู้มากกว่า 1 ภาษา ภาษาที่ทำงานทางฝั่ง Server อย่าง PHP หรือ ASP และภาษาที่ทำงานทางฝั่ง Client อย่าง JavaScript เพื่อลดความยุ่งยากรวมถึงเวลาในการต้องเรียนรู้หลายๆ ภาษาทำให้เกิด Node.js กับ Express.js เพียงแค่มีความรู้ Javascript ก็สามารถเขียนได้ทั้ง Server และ Client นอกจากนี้ถ้าใครเคยเขียน Javascript จะรู้ว่ามันมีการตอบสนองที่รวดเร็ว แน่นอนว่า Express.js ก็ยกเอามาเป็นข้อเด่นในเรื่องความเร็ว ในเรื่องการการเรียนรู้การเขียน Express.js จะใช้รูปแบบที่ง่ายในการเรียนรู้มากที่สุด

สำหรับการ พัฒนา Express.js ในเว็บไซต์จะพูดถึงการใช้ Routing (การกำหนดเส้นทางของระบบ) และ Middleware (การรับส่งข้อมูลของระบบ) สามารถเขียนได้ในรูปแบบ MVC ส่วนการเชื่อมต่อกับฐานข้อมูลสามารถใช้ MongoDB หรือจะใช้ MySql ก็ได้ สำหรับนามสกุลของไฟล์คือ .js ขณะนี้ได้พัฒนามาถึงเวอร์ชั่นที่ 4.x แล้ว

**แหล่งอ้างอิง :**

Ae. (2559). **Express.js เอ็กเพรส ดอทเจเอส คืออะไร Express.js คือ เป็น Web Application Framework ชื่อดังที่ได้รับความนิยมมาก สำหรับทำงานบน platform ของ Node.js.** จาก MindPHP: https://www.mindphp.com/คู่มือ/73-คืออะไร/3874-what-is-express-js.html

**สรุปความรู้ :**

Express.js คือ Framework สำหรับ Web Application โดยทำงานบน Node.js ด้วยภาษา JavaScript. เหมาะสำหรับการทำระบบหลังบ้านและ API

### Framework React

**เนื้อหา :**

React เป็น JavaScript Library หรือจะเรียกว่าเป็น JavaScript Framework ที่ใช้สำหรับสร้างหน้าเว็บของเราให้ออกมาดี พร้อมด้วย action ต่างๆ ที่ทำให้เว็บดูน่าสนใจ

React นั้นเป็น JavaScript Framework ที่ทำหน้าที่ Compile Code ให้กลายเป็น HTML + CSS + JavaScript เพื่อให้สามารถใช้งานบน Browser ได้ โดยที่ตัว React นั้นอาศัย NodeJS เพื่อทำการ Compile Code

จุดเด่นของ React ที่ทำให้น่านำมาใช้งานนั้นก็คือ การที่มีระบบแคชในตัวทำให้หน้าเว็บมีการตอบสนองที่เร็ว เหมาะแก่การนำไปทำ SPA การเขียน React ยังสามารถแยกองค์ประกอบของหน้าเว็บ ออกเป็นส่วนๆ เรียกว่าเป็น component แล้วนำมาประกอบกันเป็นหน้าเว็บได้ ซึ่งทำให้สามารถ นำ component ของไปใช้ซ้ำที่อื่นได้ ไม่ต้องเสียเวลาเขียนใหม่

**แหล่งอ้างอิง :**

Mos Programmer. (2562). **มารู้จัก React กันเถอะ.** จาก Medium: https://mosprogramer.medium.com/มารู้จัก-react-กันเถอะ-eb04ab8b8ec1

**สรุปความรู้ :**

React คือ JavaScript Framework สำหรับการสร้างส่วนหน้าของ Web Application มีจุดเด่นที่การมีระบบแคชในตัว ทำให้หน้าเว็บตอบสนองอย่างรวดเร็วแล้วปรับแต่งได้ง่าย และยังแยกองค์ประกอบของเว็บออกเป็นส่วนๆ ซึ่งแต่ละส่วนสามารถนำไปใช้ซ้ำได้ ไม่ต้องเขียนใหม่

### Node.js

**เนื้อหา :**

Node.js คือสภาพแวดล้อมการทำงานของภาษา JavaScript นอกเว็บเบราว์เซอร์ที่ทำงานด้วย V8 engine นั่นหมายความว่าเราสามารถใช้ Node.js ในการพัตนาแอพพลิเคชันแบบ Command line แอพพลิเคชัน Desktop หรือแม้แต่เว็บเซิร์ฟเวอร์ได้ โดยที่ Node.js จะมี APIs ที่เราสามารถใช้สำหรับทำงานกับระบบปฏิบัติการ เช่น การรับค่าและการแสดงผล การอ่านเขียนไฟล์ และการทำงานกับเน็ตเวิร์ก เป็นต้น

Node.js ถูกพัฒนาและทำงานด้วยใช้ Chrome V8 engine สำหรับคอมไพล์ภาษา JavaScript ให้เป็นภาษาเครื่องด้วยการคอมไพล์แบบ Just-in-time (JIT) เพื่อเพิ่มประสิทธิภาพการทำงานของภาษา JavaScript จากที่แต่เดิมมันเป็นภาษาที่มีการทำงานแบบ Interpreted

Node.js เป็นโปรแกรมที่สามารถใช้ได้ทั้งบน Windows, Linux และ Mac OS X นั่นหมายความว่าคุณสามารถเขียนโปรแกรมในภาษา JavaScript และนำไปรันได้ทุกระบบปฏิบัติการที่สนับสนุนโดย Node.js นี่เป็นแนวคิดของการเขียนครั้งเดียวแต่ทำงานได้ทุกที่ (Write once, run anywhere)

**แหล่งอ้างอิง :**

**ทำความรู้จักกับ Node.js.** (2564). จาก MarcusCode: http://marcuscode.com/tutorials/nodejs/introducing-nodejs

**สรุปความรู้ :**

Node.js คือสภาพแวดล้อมการทำงานของภาษา JavaScript นอกเว็บเบราว์เซอร์ ถูกพัฒนาและทำงานด้วยการใช้ Chrome V8 engine สำหรับคอมไพล์ภาษา JavaScript ให้เป็นภาษาเครื่องด้วยการคอมไพล์แบบ Just-in-time (JIT)

**สรุปแนวคิดทฤษฎี :**

เนื่องจากผู้จัดทำต้องการให้ระบบจัดการร้านอาหารเดลิเวอรี่สามารถเข้าถึงได้ง่าย ใช้งานได้ทั้งในคอมพิวเตอร์และมือถือโดยไม่ต้องติดตั้งระบบ จึงตัดสินใจที่จะทำระบบนี้ในรูปแบบของ web application ที่มีความ responsive และสามารถขยายและพัฒนาต่อไปได้ง่ายในอนาคต

หลักจากศึกษาแนวคิดทฤษฎีการพัฒนาเว็บไซต์ จึงสรุปได้ว่า การพัฒนาเว็บไซต์จะต้องใช้ HTML เป็นพื้นฐานโครงสร้างหลัก, ใช้ CSS ในการทำให้เว็บไซต์มี UX ที่ดีมีลูกเล่นต่างๆ, ต้องมีภาษาใดอีกภาษาหนึ่งในการพัฒนาระบบหลังบ้านเพื่อประมวลผลข้อมูลต่างๆที่ผู้ใช้ส่งเข้ามา และยังต้องมีระบบฐานข้อมูลในการเก็บข้อมูลอีกด้วย

แต่การพัฒนาเว็บไซต์โดยการเขียนระบบใหม่หมดตั้งแต่ต้น ทั้งเบื้องหน้า (Frontend : HTML, CSS) และเบื้องหลัง (Backend : ระบบหลังบ้าน, ระบบฐานข้อมูล) หากผู้พัฒนาไม่มีความรู้ความสามารถในภาษาต่างๆและการจัดการระบบมากพอ ระบบจะมีความซับซ้อนและไม่เป็นระเบียบ ทำให้การขยาย การปรับปรุง หรือการส่งต่อระบบเป็นไปได้ยากหรือเชื่องช้า ผู้จัดทำจึงศึกษาทฤษฎี Framework และนำมาใช้ในการพัฒนา

หลักจากศึกษา Framework ต่างๆ ผู้จัดทำตัดสินใจจะใช้ MERN สแตคในการพัฒนาระบบนี้ MERN สแตคเป็นหนึ่งสแตคที่ได้รับความนิยมสูงในปัจจุบัน ซึ่งประกอบไปด้วย React.js Framework สำหรับพัฒนาระบบเบื้องหน้า, Express.js Framework สำหรับพัฒนาระบบเบื้องหลัง, Node.js ระบบเบื้องหลังที่ช่วยให้ภาษา JavaScript สามารถทำงานภายนอกเว็บบราวเซอร์ได้ และสุดท้ายคือระบบฐานข้อมูลแบบ No-SQL MongoDB

ผู้จัดทำตัดสินใจใช้สแตคนี้เนื่องจาก 1.) ระบบ Framework ทั้งเบื้องหน้าและเบื้องหลังเป็นภาษาเดียวกัน ทำให้การสื่อสารระหว่างสองฝากเป็นไปได้ง่าย 2.) ระบบฐานข้อมูลแบบ No-SQL มีความยืดหยุ่นสูงและสามารถปรับเปลี่ยนแก้ไขได้ง่าย 3.) MERN Stack ได้รับความนิยมสูงทั้งจากผู้ว่าจ้างและผู้พัฒนาในสังคมผู้พัฒนาเว็บแอปพลิเคชั่น ทำให้เทคโนโลยีต่างๆมีการบำรุงรักษาและสนับสนุนสูง 4.) เทคโนโลยีส่วนใหญ่ใน MERN Stack เป็นเทคโนโลยีที่ผู้จัดทำไม่เคยใช้งานแต่ต้องการเรียนรู้

## งานวิจัยที่เกี่ยวข้อง

### Restaurant Management System

**เนื้อหา :**

ภาคนิพนธ์นี้ออกแบบและพัฒนาระบบจัดการร้านอาหารให้มีประสิทธิภาพมากยิ่งขึ้น ทั้งในส่วนของการจัดเก็บข้อมูล เพื่อป้องกันการสูญหายของข้อมูลต่างๆ และช่วยให้การปปฎิบัติงานมีความสะดวกรวดเร็วมากยิ่งขึ้น การพัฒนาระบบจัดการร้านอาหารโดยใช้โปรแกรม Microsoft Visual Studio 2017 ในการออกแบบพัฒนาระบบจัดการร้านอาหาร ใช้ภาษา C# ในการเขียนโปรแกรมและใช้การจัดการฐานข้อมูลโดยใช้โปรแกรม Microsoft SQL Server 2017 ระบบจัดการร้านอาหารสามารถใช้งานด้ารการเก็บข้อมูลต่างๆมีเมนูให้เลือก เช่น การจองโต๊ะ ตัดจำหน่ายวัตถุดิบ เพิ่มสูตรวัตถุดิบ รายงานสรุปยอดแบบเลือกวันที่ ซึ่งผลที่ได้รับจากการพัฒนาระบบแสดงให้เห็นว่าระบบสามารถทำงานได้ตรงขอบเขต ลดข้อผิดพลาดในการให้บริการ ตลอดจนทาให้ลดความซับซ้อนของการสั่งซื้อ และจัดการวัตถุดิบพร้อมทั้งการตัดจำหน่ายวัตถุดิบให้มีประสิทธิภาพมากขึ้น

**แหล่งอ้างอิง :**

ปริญญา อภัยภักดิ์ และณัฐพล เสาวพงษ์. **ระบบจัดการร้านอาหาร.** (ภาคนิพนธ์ปริญญาตรี คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยสยาม, 2562)

### ระบบจัดการการสั่งอาหาร กรณีศึกษา ร้านอาหาร Zero

**เนื้อหา :**

โครงงานนี้มีวัตถุประสงค์เพื่อ พัฒนาระบบบริหารจัดการการสั่งอาหาร กรณีศึกษา ร้านอาหาร Zero ซึ่งสามารถแยกออกได้เป็น 3 ส่วนหลักๆ คือ ส่วนของการจัดการระบบเมนูอาหาร ส่วนของการทํารายการขาย และส่วนของการออกรายงานยอดการขาย โดยส่วนของการจัดการระบบเมนูอาหาร จะมีหน้าที่ในการเก็บข้อมูลเมนูอาหารทั้งหมดของร้าน ส่วนในด้านของการทํารายการขายเป็นส่วนที่สําคัญเนื่องจากเป็นการดูถึงรายละเอียดการขายภายในร้าน และในส่วนของการออก รายงานยอดการขาย จะเป็นการสรุปยอดการขายรายวันและยอดการขายรายเดือน

ซึ่งในการจัดทําโครงงานระบบจัดการการสั่งอาหารนี้ได้มีการพัฒนาและออกแบบ โดยใช้ภาษาที่ใช้ในการพัฒนาระบบ Microsoft Visual Basic 2017 และใช้ระบบฐานข้อมูลด้วยโปรแกรม SQL Server 2008

**แหล่งอ้างอิง :**

วิกรม ศรศรี และนำโชค ทองละมุล. **ระบบจัดการการสั่งอาหาร กรณีศึกษา ร้านอาหาร Zero.** (ปริญญาตรี คณะบริหารธุรกิจ มหาวิทยาลัยราชพฤกษ์, 2560)

### ระบบการจัดการร้านอาหาร

**เนื้อหา :**

ปริญญานิพนธ์ฉบับนีมีวัตถุประสงค์เพือพัฒนาระบบการจัดการร้านอาหาร โดยระบบสามารถจัดการข้อมูลพนักงาน ข้อมูลการขาย ข้อมูลการสังซือวัตถุดิบ ข้อมูลตรวจสอบวัตถุดิบคงเหลือ และการออกรายงานสําหรับผู้บริหาร เพือช่วยสนับสนุนการตัดสินใจในการวางแผนบริหารกิจการให้เกิดผลกําไรสูงสุด รวมถึงการอํานวยความสะดวกต่อผู้ใช้บริการ และการปฏิบัติงานของผู้ให้บริการได้อย่างมีประสิทธิภาพมากยิงขึน ระบบการจัดการร้านอาหาร มีลักษณะเป็นเว็บแอพพลิเคชัน(Web-Based Application) พัฒนาโดยภาษา PHP, HTML และใช้ MySQL client version: 5.0.51a เป็นระบบจัดการฐานข้อมูล

ระบบการจัดการร้านอาหาร คือ ซอฟต์แวร์ทีพัฒนาสําหรับจัดการร้านอาหาร เพือช่วยอํานวยความสะดวกในการจัดการร้านอาหาร แบ่งการใช้งานออกเป็นสองส่วน ดังนี พ๊อคเก็ตพีซี ใช้ในการรับสังอาหาร ตรวจสอบสถานะการทําอาหาร คอมพิวเตอร์ใช้สําหรับจัดเก็บข้อมูลต่าง ๆ และลงโปรแกรมการจัดการร้านอาหาร ใช้ในการบันทึกข้อมูลต่าง ๆ ภายในร้านอาหาร บันทึกข้อมูลการจัดซือวัตถุดิบบันทึกข้อมูลวัตถุดิบ บันทึกข้อมูลรายการอาหาร บันทึกข้อมูลรายละเอียดการขาย จัดทํารายงาน บันทึกการเปลี่ยนแปลงข้อมูลต่าง ๆ ซึงทําให้มีความสามารถจัดการร้านอาหารได้อย่างมีประสิทธิภาพ มีความถูกต้องถูกต้อง สะดวกรวดเร็ว และตรวจสอบความถูกต้องของข้อมูลย้อนหลังได้

**แหล่งอ้างอิง :**

ธีรพงศ์ ชูชื่น และจิรรัฐ ศรีโชค. **ระบบจัดการร้านอาหาร.** (ปริญญาตรี คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสำเด็จเจ้าพระยา, 2555)

### ระบบจัดการร้านกังฟูหม่าล่า เบียร์หิมะ

**เนื้อหา :**

โครงงานนี้มีวัตถุประสงค์เพื่อวิเคราะห์ ออกแบบ และพัฒนาระบบจัดการร้าน กังฟูหม่าล่า เบียร์หิมะ (KungFu Mhala Snow Beer) เพื่อสามารถบริหารจัดการงานที่มี ให้เหมาะสมกับการทํางาน และเพื่อช่วยตรวจสอบสินค้าคงเหลือ จัดเก็บข้อมูลให้เป็นระเบียบ ค้นหาได้สะดวกรวดเร็วความสะดวกในการทํางานของพนักงาน ลดความผิดพลาดในการคํานวณ และจัดเก็บข้อมูล

ปัจจุบันทางร้านได้ประสบกับปัญหาการเช็คสต็อกจํานวนวัตถุดิบที่ทางร้านได้นํามาจําหน่ายในแต่ละวัน เช่น จดบันทึก จึงทําให้เกิดปัญหา ข้อมูลอาจมีผิดพลาด หรือสูญหาย ลืมจดบันทึก และทําให้การคํานวณยอดขายในแต่ละวันไม่สะดวกและไม่รวดเร็ว และยังเชื่อมโยงไปถึงการออกใบเสร็จให้แก่ลูกค้าเพราะทางร้านยังรับการสั่งรายการอาหารที่ลูกค้าสั่งเป็นกระดาษ ทําให้การรับอาหารในแต่ละครั้งอาจมีข้อผิดพลาดและไม่สะดวก อีกทั้งทางร้านยังไม่มีระบบให้บริการการจองโต๊ะ ก่อให้เกิดความไม่สะดวกในการติดต่อในการจองโต๊ะและยังเสียเวลาในการมานั่งรอที่หน้าร้านอีกด้วย

ดังนั้น จากปัญหาที่กล่าวข้างต้นจึงได้ทําการวิเคราะห์ หากเรานําเทคโนโลยีที่ถูกพัฒนาขึ้นบางตัว มาประยุกต์ใช้ให้เข้ากับกิจการร้านอาหารของเรา ก็จะทําให้การจัดการในธุรกิจร้านอาหารของเราดีขึ้น ซึ่งโปรแกรมนี้ ผู้ประกอบการสมารถนําเข้าภาพที่ออกแบบรูปของสถานที่ร้านอาหารมาใช้ในโปรแกรมได้ รวมทั้งสามารถ เพิ่ม,ลด,แก้ไข ตําแหน่ง สี ขนาด จํานวนคน ทราบเมนูที่ลูกค้าสั่งและเสิร์ฟได้ถูกต้องทันตามความต้องการ ทั้งยังมีระบบการจัดการรายได้ เช่น สามารถตรวจสอบยอดเงินที่ต้องชําระของลูกค้าแต่ละโต๊ะได้อย่างรวดเร็ว และยังมีตารางข้อมูลรายได้ของยอดเงินใน แต่ละวันและรายชื่ออาหาร ซึ่งจะทําให้การรวมยอดขายในแต่ละวันง่ายขึ้น มีความสะดวกสบายมากขึ้นในการรวมยอดขายในแต่ละวัน

**แหล่งอ้างอิง :**

ปรินทร อ่อนโก้ก, นิรัชชา บุญวิเทียน และ ชัญญานุช ศรีวรรณ์. **ระบบจัดการร้านกังฟูหม่าล่า เบียร์หิมะ.** (ปริญญาตรี คณะวิทยาการจัดการ มหาวิทยาลัยราชภัฏอุดรธานี, 2563)

**สรุปความรู้ 2.1 – 2.4 :**

งานวิจัยทั้ง 4 งานเป็นงานวิจัยที่เกี่ยวข้องกับการพัฒนาระบบจัดการร้านอาหารทั้งหมด แม้จะไม่ได้ใช้เทคโนโลยีเดียวกันกับ Delivery Restaurant Manager แต่ก็มีความคล้ายคลึงกันในหลายๆส่วน ทั้งในส่วนของรูปแบบโครงสร้างของระบบ โครงสร้างฐานข้อมูล และการส่งผ่านข้อมูล

เมื่อทำการศึกษารายละเอียดงานวิจัยทั้ง 4 แล้วพบว่า สามารถนำข้อมูล โครงสร้าง และประสบการณ์การทำงานของทั้ง 4 งานวิจัยมาปรับใช้ในระบบ Delivery Restaurant Manager ได้ เพราะมีโมดูลที่เหมือนกันหรือใกล้เคียงกันอยู่มาก เช่น การบันทึกคำสั่งซื้อ การเก็บสถิติการขาย การออกรายงาน ฯลฯ

ข้อมูลเหล่านี้จะเป็นประโยชน์อย่างมากในการเก็บความต้องการของระบบ ในส่วนที่ผู้พัฒนาหรือผู้ใช้งานเองก็คาดไม่ถึง หรือในด้านการออกแบบระบบก็สามารถนำงานวิจัยเหล่านี้มาปรับปรุงแก้ไขได้เป็นอย่างดี

### เทคโนโลยีเภสัชสารสนเทศแสดงข้อมูลฉลากยาเอกสารกํากับยาแบบอัตโนมัติสําหรับบริหารจัดการคลังยาปฎิชีวนะเพื่อความปลอดภัย

**เนื้อหา :**

จากสถานการณ์การใช้ยาในปัจจุบันนี้ ผู้บริโภคขาดความรู้ความเข้าใจถึงอันตรายที่เกี่ยวกับยา อีกทั้งยังไม่มีแหล่งข้อมูลที่เข้าถึงได้ง่าย ข้อมูลที่ได้รับจากเภสัชกรที่เป็นเพียงฉลากยาจากคลินิกหรือโรงพยาบาลจึงไม่เพียงพอ จากปัญหาดังกล่าวข้างต้นจึงมีแนวคิดในการสร้างระบบแสดงข้อมูลฉลากยาเอกสารกำกับยาแบบอัตโนมัติโดยการส่งข้อมูลดิจิตอลไปเก็บไว้ที่ฐานข้อมูลในเครื่องแม่ข่าย เพื่อเป็นเครื่องมือที่ช่วยในการลดความคลาดเคลื่อนทางยา (Medication Error) ที่เกิดในกระบวนการใช้ยา ตั้งแต่การสั่งใช้ยา การคัดลอกคำสั่งใช้ยา การจ่ายยา และ สุดท้ายเป็นการบริหารยา โดยการนำเทคโนโลยี Optical Character Recognition (OCR) มาทำการแปลงไฟล์ภาพเอกสารที่ได้รับการสแกนให้กลายเป็นไฟล์ข้อความตัวอักษรที่สามารถสืบค้นได้

โดยระบบที่พัฒนาขึ้นนี้ถูกแบ่งออกเป็น 4 ส่วน โดยส่วนแรกคือส่วนของการเก็บข้อมูล ซึ่งข้อมูลยาที่ใช้ในการเก็บมาจากเอกสารกำกับยา โดยใช้เทคโนโลยี OCR (Optical Character Recognition) เป็นการแปลงรูปภาพเป็นตัวอักษร ส่วนที่สองการจัดการข้อมูลยา ซึ่งเป็นส่วนที่ใช้ในการสร้าง แก้ไข ลบ และแสดงข้อมูลยาทั้งหมดภายในระบบ ส่วนที่สามการค้นหาข้อมูลยา จะเป็นการค้นหาข้อมูลยาได้จากชื่อ สรรพคุณของยารวมถึงอาการของผู้ป่วย ส่วนที่สี่ส่วนการจัดการผู้ใช้จะควบคุมการเข้าถึงระบบในส่วนต่างๆ ของผู้ใช้แต่ละประเภท

ระบบเทคโนโลยีเภสัชสารสนเทศแสดงข้อมูลฉลากยาเอกสารกากับยาแบบอัตโนมัตินี้มีประสิทธิภาพในการแปลงรูปภาพเป็นตัวอักษรด้วยเทคโนโลยี OCR ที่ความแม่นยาได้ถึงร้อยละ 96.61 และสามารถช่วยลดเวลาของการเก็บข้อมูลลงฐานข้อมูลยา และช่วยให้ผู้บริโภคมีความเข้าใจในการใช้ยา มีความรู้ความถูกต้องเกี่ยวกับข้อมูล รู้วิธีของการเก็บรักษาของยาเพื่อให้ได้รับประโยชน์ ช่วยให้ปลอดภัยและลดปัญหาของการใช้ยาที่ผิดกับโรค

**แหล่งอ้างอิง :**

วิรุฬห์ ศรีบริรักษ์. **เทคโนโลยีเภสัชสารสนเทศแสดงข้อมูลฉลากยาเอกสารกํากับยาแบบอัตโนมัติสําหรับบริหารจัดการคลังยาปฎิชีวนะเพื่อความปลอดภัย** (คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา, 2562)

**สรุปความรู้ :**

จากเอกสารงานวิจัยข้างต้น ผู้วิจัยได้ทำการพัฒนาซอฟต์แวร์ด้วยเทคโนโลยีที่ใกล้เคียงกันกับระบบจัดการร้านอาหารเดลิเวอรี่ คือ MongoDB, NodeJS, ทฤษฎี REST API และ Angular Framework ซึ่งเป็น Frontend framework อีกตัวหนึ่งที่มีลักษณะการใช้งานใกล้เคียงกันกับ React Framework

ทำให้งานวิจัยนี้มีประโยชน์ในส่วนของการที่ทำให้ได้เห็นตัวอย่างของการนำเทคโนโลยีเหล่านี้ไปใช้งานจริง

# วิธีดำเนินโครงงาน (Research Methodology)

## ขั้นตอนการดำเนินงานตาม Software Process Model

แบบจำลองน้ำตก (Waterfall Model) เป็นแบบจำลองกระบวนการในการพัฒนาระบบรูปแบบหนึ่ง โดยในอดีตแบบจำลองน้ำตกเป็นที่นิยมใช้กันเป็นอย่างมาก ในระบบหรือซอฟต์แวร์ที่มีขนาดเล็กและไม่ซับซ้อน เนื่องจากสามารถพัฒนาทีละขั้นตอนได้ สามารถตรวจสอบทีละขั้นตอนและกาหนดทรัพยากรต่างๆ ในแต่ละขั้นตอนได้เป็นอย่างดี ทั้งนี้แบบจำลองน้ำตกมีหลักการคล้ายกับการไหลของน้ำตกที่ไหลจากด้านบนสู่ด้านล่าง ซึ่งในแต่ละขั้นตอนจะไม่สามารถย้อนกลับไปเพื่อแก้ไขหรือปรับปรุงขั้นตอนก่อนหน้านี้ได้ โดยแบบจำลองน้ำตกจึงเหมาะสมกับระบบหรือซอฟต์แวร์ที่มีขอบเขตและข้อกาหนดที่ชัดเจน ทั้งนี้แบบจำลองน้ำตกได้ถูกกล่าวถึงครั้งแรกจากงานวิจัยของ วินซ์ตัน ดับเบิลยู รอยซ์ (Winston W. Royce’s) เกี่ยวกับเรื่องการจัดการซอฟต์แวร์ที่มีขนาดใหญ่และมีความสลับซับซ้อน โดยถูกมองว่าการพัฒนาระบบหรือซอฟต์แวร์ที่มีขนาดใหญ่และมีความซับซ้อนเกินกว่าที่จะออกแบบมาด้วยวิธีการที่ไม่เป็นระบบได้ โดยได้มีการนำเสนอรูปแบบหรือวิธีการทำงานแบบใหม่ขึ้นมาที่มีขั้นตอนการทำงานต่าง ๆ เพิ่มมากขึ้น รวมไปถึงในแต่ละช่วงการพัฒนาและออกแบบจะมีการเก็บบันทึกข้อมูลลงในเอกสารเอาไว้ เพื่อช่วยในการพัฒนาออกแบบ และรวมไปถึงการปรับปรุงระบบขึ้นใหม่ ทั้งนี้แบบจำลองดังกล่าวเป็นแบบจำลองที่เหมาะสาหรับการพัฒนาระบบและซอฟต์แวร์ที่เน้นด้านคุณภาพมากกว่าต้นทุนหรือระยะเวลาของงานที่มีขอบเขตและข้อกาหนดที่ชัดเจน โดยแบบจำลองน้ำตกจะมีรูปแบบหรือขั้นตอนทั้งหมด 5 ขั้นตอนหลักด้วยกัน

1.) ขั้นตอนการวางแผนและวิเคราะห์ระบบ (Planning and Requirement Definition) เป็นขั้นตอนในการวางแผนการดำเนินงานพัฒนาระบบหรือซอฟต์แวร์ โดยจะทำการรวบรวมข้อมูลความต้องการต่าง ๆ จากผู้ใช้งาน และทำการวิเคราะห์ข้อมูลเพื่อหาข้อสรุปของข้อมูลและจัดทำบันทึกลงในเอกสาร (Requirement Document) ทั้งในส่วนของความต้องการ แผนผังการทำงานระหว่างข้อมูลและระบบที่จะพัฒนา รวมไปถึงขั้นตอนและกระบวนการทำงานต่าง ๆ

2.) ขั้นตอนการออกแบบระบบหรือซอฟต์แวร์ (System and Software Design) เป็นขั้นตอนในการออกแบบระบบและซอฟต์แวร์ให้เป็นไปตามวัตถุประสงค์หรือความต้องการของผู้ใช้งาน ไม่ว่าจะเป็นทั้งในส่วนของระบบหน้าบ้านและหลังบ้าน (User Interface) กระบวนการทำงานของระบบ (System Workflow) การจัดเก็บข้อมูลบนระบบ (Database System) รายงานเอกสารบนระบบ (Report System) เป็นต้น

3.) ขั้นตอนการพัฒนาและทดสอบระบบในระดับหน่วย (Implementation and Unit Testing) เป็นขั้นตอนที่นำข้อมูลจากขั้นตอนก่อนหน้านี้มาดำเนินการพัฒนาระบบให้สามารถใช้งานได้จริง โดยเป็นการปรับเปลี่ยนข้อมูลจากที่อยู่บนเอกสารให้กลายเป็นระบบหรือซอฟต์แวร์ที่สามารถให้ผู้ใช้งานนำไปใช้งานได้จริง โดยในขั้นตอนนี้มักจะมีการทดสอบระบบในระดับหน่วยเล็ก ๆ (Unit Test) เพื่อเป็นการตรวจสอบว่าในแต่ละส่วนของระบบหรือซอฟต์แวร์ที่พัฒนาขึ้นมานั้นสามารถนำไปใช้ได้จริงและทำงานได้ถูกต้อง

4.) ขั้นตอนการเชื่อมต่อและทดสอบระบบ (Integration and System Testing) เป็นขั้นตอนในการทดสอบการเชื่อมต่อกับระบบอื่น ๆ ที่เกี่ยวข้องว่าในแต่ละส่วนงานที่มีการเชื่อมต่อกันสามารถทำงานร่วมกันได้อย่างถูกต้องหรือไม่ นอกจากนี้ยังรวมไปถึงการทดสอบระบบทั้งหมดก่อนที่จะนำไปใช้งานจริงว่ามีความสมบูรณ์และถูกต้องที่ตรงตามความต้องการของผู้ใช้งานหรือไม่หากมีการทดสอบในขั้นตอนนี้แล้วพบข้อผิดพลาดหรือปัญหาเกิดขึ้นจะต้องดำเนินการพัฒนา แก้ไข และปรับปรุงระบบหรือซอฟต์แวร์นั้นใหม่ก่อนที่จะนาระบบขึ้นไปใช้งานจริง

5.) ขั้นตอนการนำระบบไปใช้งานจริงและการบำรุงรักษา (Operation and Maintenance System) เป็นขั้นตอนที่เมื่อระบบถูกนำไปใช้งานจริง ซึ่งจะต้องมีการติดตาม ตรวจสอบ เพื่อดำเนินการพัฒนา ปรับปรุงและหรือแก้ไขปัญหาหรือข้อผิดพลาดที่อาจเกิดขึ้นได้โดยหากพบปัญหาหรือข้อผิดพลาดดังกล่าว จะต้องทำการพัฒนา ปรับปรุงและแก้ไขเพื่อให้ระบบสามารถตอบสนองความต้องการของผู้ใช้งานได้เพิ่มเติม

## แผนภาพกระแสข้อมูล

Diagram

Description automatically generated

รูปประกอบ 3.1 Dataflow Diagram

2.1 การยืนยันตัวตน : ผู้ใช้ส่งผ่านข้อมูลในการเข้าสู่ระบบ => ระบบดึงข้อมูลผู้ใช้จากฐานข้อมูล => ระบบส่งผลการล็อกอินกลับไปยังผู้ใช้

2.2 การจัดการข้อมูลสินค้า : ผู้ใช้ส่งผ่านข้อมูลสินค้าเข้าสู่ระบบ => ระบบบันทึกข้อมูลสินค้าลงในฐานข้อมูลตามข้อมูลที่ได้รับ

2.3 การจัดการไรเดอร์ : ผู้ใช้ส่งผ่านข้อมูลไรเดอร์เข้าสู่ระบบ => ระบบบันทึกข้อมูลลงในฐานข้อมูลไรเดอร์

2.4 การจัดการราคาสินค้า : ผู้ใช้ส่งผ่านข้อมูลราคาเข้าสู่ระบบ => ระบบบันทึกข้อมูลลงในฐานข้อมูลไรเดอร์

2.5 การจัดการหมวดหมู่รายจ่าย : ผู้ใช้ส่งผ่านข้อมูลหมวดหมู่เข้าสู่ระบบ => ระบบบันทึกข้อมูลลงในฐานข้อมูลหมวดหมู่รายจ่าย

2.6 การบันทึกรายจ่าย ; ระบบดึงรายการหมวดหมู่รายจ่ายเพื่อนำไปแสดงผล => ผู้ใช้ส่งผ่านข้อมูลรายจ่ายเข้าสู่ระบบ => ระบบบันทึกข้อมูลลงในฐานข้อมูลรายจ่าย

2.7 การบันทึกคำสั่งซื้อ : ผู้ใช้ส่งผ่านข้อมูลคำสั่งซื้อเข้าสู่ระบบ => ระบบดึงข้อมูลราคาจากฐานข้อมูลไรเดอร์ => ระบบบันทึกข้อมูลเข้าสู่ฐานข้อมูลออเดอร์

## โครงสร้างซอฟต์แวร์ (Software Structure)

Diagram, timeline

Description automatically generated

รูปประกอบ 3.2 Software Structure

1. เมื่อเข้ามาในระบบ ผู้ใช้จะต้องผ่านการตรวจสอบตัวตน เมื่อผ่านการตรวจสอบตัวตนแล้ว จะสามารถเข้าถึงส่วนต่างๆของระบบได้ดังนี้
2. ส่วนการตั้งค่าร้านค้าครั้งแรก ประกอบด้วย
   1. ส่วนการจัดการอาหาร ประกอบด้วย
      1. หน้าจัดการวัตถุดิบ ภายในจะมีโมดูลต่างๆดังนี้
         1. ตารางแสดงรายการที่สามารถ ลบ ค้นหาและจัดเรียงวัตถุดิบได้
         2. ส่วนสำหรับสร้างวัตถุดิบ
         3. ส่วนสำหรับแก้ไขวัตถุดิบ
      2. หน้าจัดการอาหาร ภายในจะมีโมดูลต่างๆดังนี้
         1. ตารางแสดงรายการที่สามารถลบอาหารได้
         2. ส่วนสำหรับสร้างอาหาร
         3. ส่วนสำหรับแก้ไขอาหาร
      3. หน้าจัดการเมนูอาหาร ภายในจะมีโมดูลต่างๆดังนี้
         1. ตารางแสดงรายการที่สามารถลบเมนูอาหารได้
         2. ส่วนสำหรับสร้างเมนูอาหาร
         3. ส่วนสำหรับแก้ไขเมนูอาหาร
   2. หน้าจัดการไรเดอร์ ภายในจะโมเดลต่างๆดังนี้
      1. ส่วนสำหรับเลือกและแสดงรายการไรเดอร์
      2. ตารางแสดงรายการราคาของเมนูอาหาร ที่สามารถค้น จัดเรียง และแก้ไขราคาได้โดยตรง
      3. ส่วนสำหรับการสร้างไรเดอร์
      4. ส่วนสำหรับลบไรเดอร์
3. ส่วนสำหรับจัดการบัญชีร้าน ประกอบด้วย
   1. หน้าจัดการรายจ่าย ภายในจะมีโมดูลต่างๆดังนี้
      1. หน้าจัดการหมวดหมู่ ซึ่งมีโมดูลสำหรับ สร้าง แก้ไข ลบ แสดงรายการหมวดหมู่อยู่ภายใน
      2. ส่วนสำหรับการสร้างรายจ่าย
      3. ส่วนสำหรับแก้ไขรายจ่าย
      4. ตารางแสดงรายจ่ายที่สามารถค้นหา และจัดเรียงรายจ่ายได้
   2. หน้าจัดการคำสั่งซื้อ ภายในจะมีโมดูลต่างๆดังนี้
      1. ตารางแสดงรายการคำสั่งซื้อที่สามารถค้นหา จัดเรียง และลบคำสั่งซื้อได้
      2. ส่วนสำหรับการสร้างคำสั่งซื้อโดยไม่จำเป็นต้องใช้แป้นพิมพ์
4. ส่วนแสดงสถิติการขาย ประกอบด้วย
   * 1. หน้าแดชบอร์ด (Dashboard) ภายในจะมีโมดูลต่างๆดังนี้
        1. ส่วนแสดงผล
           1. แสดงผลเป็นตาราง
           2. แสดงผลเป็นกราฟ
        2. ส่วนปุ่มคำสั่ง ซึ่งจะประกอบด้วย
           1. ปุ่มกรองข้อมูลตามไรเดอร์
           2. ปุ่มกรองข้อมูลตามระยะเวลา
           3. ปุ่มเปลี่ยนการแสดงผลระหวางตารางและกราฟ

## ข้อกำหนดซอฟต์แวร์ (Input/Output Specifications)

### โมดูลการเข้าสู่ระบบ

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. ชื่อผู้ใช้ (Username) 2. รหัสผ่าน (Password) | 1. นำชื่อผู้ใช้ เปรียบเทียบกับข้อมูลผู้ใช้ในฐานข้อมูล 2. นำรหัสผ่านไปเข้ารหัส จากนั้นนำไปเปรียบเทียบกับข้อมูลรหัสผ่านที่เข้ารหัสแล้วในฐานข้อมูล | 1. ข้อมูลถูกต้อง ผู้ใช้สามารถเข้าสู่ระบบได้ 2. ข้อมูลไม่ถูกต้อง ปฏิเสธการเข้าสู่ระบบ และให้ผู้ใช้กรอกข้อมูลใหม่อีกครั้ง |

ตาราง 3.1 Login module

### โมดูลตรวจสอบการเข้าสู่ระบบ

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. โทเคนการเข้าสู่ระบบ (Login Token) 2. หน้าที่ต้องการเข้า | 1. ตรวจสอบความถูกต้องของโทเคน 2. ตรวจสอบวันหมดอายุของโทเคน | 1. ข้อมูลถูกต้อง ผู้ใช้สามารถเข้าสู่หน้านั้นๆได้ 2. ข้อมูลไม่ถูกต้อง ปฎิเสธการเข้าถึง ส่งผู้ใช้ไปยังหน้าเข้าสู่ระบบ |

ตาราง 3.2 Authentication module

### โมดูลเพิ่มวัตถุดิบ

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. ชื่อ 2. ประเภท 3. หน่วย | 1. ตรวจสอบ Input 2. เพิ่มข้อมูลเข้าสู่ฐานข้อมูล | 1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ในตารางข้อมูล 2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.3 โมดูลเพิ่มวัตถุดิบ

### โมดูลแก้ไขวัตถุดิบ

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. วัตถุดิบที่ต้องการแก้ไข 2. ชื่อ 3. ประเภท 4. หน่วย | 1. ตรวจสอบ Input 2. ค้นหาข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลในฐานข้อมูล | 1. แก้ไขข้อมูลสำเร็จ แสดงข้อมูลที่แก้ไขในตารางด้วยข้อมูลใหม่ 2. ไม่พบข้อมูลที่ต้องการแก้ไข แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.4 โมดูลแก้ไขวัตถุดิบ

### โมดูลลบวัตถุดิบ

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. วัตถุดิบที่ต้องการลบ | 1. ค้นหาข้อมูลที่ต้องการลบ 2. ลบข้อมูลออกจากฐานข้อมูล | 1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล 2. ไม่พบข้อมูลที่ต้องการลบ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการลบ 3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.5 โมดูลลบวัตถุดิบ

### โมดูลเพิ่มอาหาร

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. ชื่อ 2. รายการวัตถุดิบ ประกอบด้วย    1. วัตถุดิบที่ใช้    2. จำนวนที่ใช้ | 1. ตรวจสอบ Input 2. เพิ่มข้อมูลเข้าสู่ฐานข้อมูล | 1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ในตารางข้อมูล 2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.6 โมดูลเพิ่มอาหาร

### โมดูลแก้ไขอาหาร

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. อาหารที่ต้องการแก้ไข 2. ชื่อ 3. รายการวัตถุดิบ ประกอบด้วย    1. วัตถุดิบที่ใช้    2. จำนวนที่ใช้ | 1. ตรวจสอบ Input 2. ค้นหาข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลในฐานข้อมูล | 1. แก้ไขข้อมูลสำเร็จ แสดงข้อมูลที่แก้ไขในตารางด้วยข้อมูลใหม่ 2. ไม่พบข้อมูลที่ต้องการแก้ไข แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.7 โมดูลแก้ไขอาหาร

### โมดูลลบอาหาร

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. อาหารที่ต้องการลบ | 1. ค้นหาข้อมูลที่ต้องการลบ 2. ลบข้อมูลออกจากฐานข้อมูล | 1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล 2. ไม่พบข้อมูลที่ต้องการลบ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการลบ 3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.8 โมดูลลบอาหาร

### โมดูลเพิ่มเมนูอาหาร

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. ชื่อ 2. รายการวัตถุดิบ ประกอบด้วย    1. วัตถุดิบที่ใช้    2. จำนวนที่ใช้ 3. ค่าแรง 4. ค่าน้ำมัน | 1. ตรวจสอบ Input 2. เพิ่มข้อมูลเข้าสู่ฐานข้อมูล | 1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ในตารางข้อมูล 2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.9 โมดูลเพิ่มเมนูอาหาร

### โมดูลแก้ไขเมนูอาหาร

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. เมนูอาหารที่ต้องการแก้ไข 2. ชื่อ 3. รายการวัตถุดิบ ประกอบด้วย    1. วัตถุดิบที่ใช้    2. จำนวนที่ใช้ 4. ค่าแรง 5. ค่าน้ำมัน | 1. ตรวจสอบ Input 2. ค้นหาข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลในฐานข้อมูล | 1. แก้ไขข้อมูลสำเร็จ แสดงข้อมูลที่แก้ไขในตารางด้วยข้อมูลใหม่ 2. ไม่พบข้อมูลที่ต้องการแก้ไข แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.10 โมดูลแก้ไขเมนูอาหาร

### โมดูลลบเมนูอาหาร

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. เมนูอาหารที่ต้องการลบ | 1. ค้นหาข้อมูลที่ต้องการลบ 2. ลบข้อมูลออกจากฐานข้อมูล | 1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล 2. ไม่พบข้อมูลที่ต้องการลบ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการลบ 3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.11 โมดูลลบเมนูอาหาร

### โมดูลเพิ่มไรเดอร์

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. ชื่อ 2. ค่าธรรมเนียม 3. ค่าภาษี 4. ค่าน้ำมัน | 1. ตรวจสอบ Input 2. เพิ่มข้อมูลเข้าสู่ฐานข้อมูล | 1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ในตารางข้อมูล 2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.12 โมดูลเพิ่มไรเดอร์

### โมดูลลบไรเดอร์

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. ไรเดอร์ที่ต้องการลบ | 1. ค้นหาข้อมูลที่ต้องการลบ 2. ลบข้อมูลออกจากฐานข้อมูล | 1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล 2. ไม่พบข้อมูลที่ต้องการลบ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการลบ 3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.13 โมดูลลบไรเดอร์

### โมดูลแก้ไขราคาเมนูอาหาร

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. ไรเดอร์ที่ต้องการแก้ไข 2. เมนูที่ต้องการแก้ไข 3. ราคา | 1. ตรวจสอบ Input 2. ค้นหาข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลในฐานข้อมูล | 1. แก้ไขข้อมูลสำเร็จ แสดงข้อมูลที่แก้ไขในตารางด้วยข้อมูลใหม่ 2. ไม่พบข้อมูลที่ต้องการแก้ไข แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.14 โมดูลแก้ไขราคาเมนูอาหาร

### โมดูลเพิ่มหมวดหมู่รายจ่าย

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. ชื่อ 2. (ถ้าต้องการ) หมวดหมู่หลัก | 1. ตรวจสอบ Input 2. (ถ้ามี) ค้นหาหมวดหมู่ที่อ้างถึง 3. เพิ่มข้อมูลเข้าสู่ฐานข้อมูล | 1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ในตารางข้อมูล 2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.15 โมดูลเพิ่มหมวดหมู่รายจ่าย

### โมดูลแก้ไขหมวดหมู่รายจ่าย

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. หมวดหมู่ที่ต้องการแก้ไข 2. ชื่อ 3. (ถ้าต้องการ) หมวดหมู่หลัก | 1. ตรวจสอบ Input 2. ค้นหาข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลในฐานข้อมูล | 1. แก้ไขข้อมูลสำเร็จ แสดงข้อมูลที่แก้ไขในตารางด้วยข้อมูลใหม่ 2. ไม่พบข้อมูลที่ต้องการแก้ไข แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.16 โมดูลแก้ไขหมวดหมู่รายจ่าย

### โมดูลลบหมวดหมู่รายจ่าย

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. หมวดหมู่ที่ต้องการลบ | 1. ค้นหาข้อมูลที่ต้องการลบ 2. ลบข้อมูลออกจากฐานข้อมูล | 1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล 2. ไม่พบข้อมูลที่ต้องการลบ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการลบ 3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.17 โมดูลลบหมวดหมู่รายจ่าย

### โมดูลเพิ่มรายจ่าย

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. หมวดหมู่ 2. คำอธิบายรายการ 3. จำนวนที่ซื้อ 4. ราคาที่จ่าย 5. วันเวลาที่เกิด | 1. ตรวจสอบ Input 2. เพิ่มข้อมูลเข้าสู่ฐานข้อมูล | 1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ในตารางข้อมูล 2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.18 โมดูลเพิ่มรายจ่าย

### โมดูลแก้ไขรายจ่าย

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. รายจ่ายที่ต้องการแก้ไข 2. หมวดหมู่ 3. คำอธิบายรายการ 4. จำนวนที่ซื้อ 5. ราคาที่จ่าย 6. วันเวลาที่เกิด | 1. ตรวจสอบ Input 2. ค้นหาข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลในฐานข้อมูล | 1. แก้ไขข้อมูลสำเร็จ แสดงข้อมูลที่แก้ไขในตารางด้วยข้อมูลใหม่ 2. ไม่พบข้อมูลที่ต้องการแก้ไข แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.19 โมดูลแก้ไขรายจ่าย

### โมดูลลบรายจ่าย

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. รายจ่ายที่ต้องการลบ | 1. ค้นหาข้อมูลที่ต้องการลบ 2. ลบข้อมูลออกจากฐานข้อมูล | 1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล 2. ไม่พบข้อมูลที่ต้องการลบ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการลบ 3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.20 โมดูลลบรายจ่าย

### โมดูลทำซ้ำรายจ่าย

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. รายจ่ายที่ต้องการทำซ้ำ | 1. ค้นหาข้อมูลที่ต้องการทำซ้ำ 2. เพิ่มข้อมูลในฐานข้อมูลโดยใช้ข้อมูลของรายจ่ายที่เลือก วันเวลาที่เกิดใช้เป็นเวลาปัจจุบัน | 1. ทำซ้ำข้อมูลสำเร็จ แสดงข้อมูลใหม่ในตารางด้วยข้อมูล 2. ไม่พบข้อมูลที่ต้องการทำซ้ำ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการทำซ้ำ 3. ทำซ้ำข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.21 โมดูลทำซ้ำรายจ่าย

### โมดูลเพิ่มคำสั่งซื้อ

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. ไรเดอร์ 2. วันเวลาที่เกิด 3. รายการอาหาร    * 1. เมนูอาหาร      2. จำนวนอาหาร 4. วันเวลาที่เกิด | 1. ตรวจสอบ Input 2. ค้นหาข้อมูลราคาจากฐานข้อมูลไรเดอร์ 3. เพิ่มข้อมูลเข้าสู่ฐานข้อมูลพร้อมราคาของแต่ละเมนูและยอดรวมทั้งหมด | 1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ในตารางข้อมูล 2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.22 โมดูลเพิ่มคำสั่งซื้อ

### โมดูลลบคำสั่งซื้อ

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1. คำสั่งซื้อที่ต้องการลบ | 1. ค้นหาข้อมูลที่ต้องการลบ 2. ลบข้อมูลออกจากฐานข้อมูล | 1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล 2. ไม่พบข้อมูลที่ต้องการลบ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการลบ 3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ |

ตาราง 3.23 โมดูลลบคำสั่งซื้อ

### โมดูลแสดงสถิติการขาย

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. ไรเดอร์ 2. ช่วงเวลา 3. ประเภทการแสดงผ | 1. ดึงข้อมูลจากฐานข้อมูลออเดอร์ 2. คัดเอาข้อมูลที่อยู่ในช่วงเวลา 3. คัดเอาข้อมูลจากไรเดอร์ที่เลือก 4. แปลงข้อมูลตามประเภทการแสดงผลที่เลือก | 1. แสดงสถิติการขายตามการแสดงผลที่เลือก |

ตาราง 3.24 โมดูลแสดงสถิติการขาย

## E-R Diagram

A picture containing text, sign, sale, scoreboard

Description automatically generated

รูปประกอบ 3.3 Entity Relationship Diagram

## ตารางฐานข้อมูล (Tables)

### ตารางผู้ใช้ (users)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ลำดับ | ชื่อฟิลด์ | ชนิดข้อมูล | คำอธิบาย | ข้อมูลตัวอย่าง |
| 1 | \_id (PK) | ObjectID | สร้างอัตโนมัติโดย MongoDB | ObjectId(“62861e401904f7babfb13d”) |
| 2 | email (unique) | String | Email ของผู้ใช้ | nameofemail@domain.notebook |
| 3 | password | String | Password ของผู้ใช้ที่ผ่านการเข้ารหัสแล้ว | $2y$10$/uO.IASKMRL.RQSOrgWZ5.Lx  u6elu1WET0.w9zWznlLwPfQgfwUEW |
| 4 | name | String | ชื่อของผู้ใช้ | Nick |
| 5 | token | String | Login token session ปัจจุบันที่ generate โดย Json Web Token | eyJhbGciOiJIUzI1NiIsInR5cCI6  IkpXVCJ9.eyJpZCI6IjYyZDhkZTg  5NWE1NmZkZjVkNDMyMWQ0MiIsIml  hdCI6MTY2NjU0MTQ3NiwiZXhwIjo  xNjY5MTMzNDc2fQ.fM\_B\_w2-dgSK  cq9HvW78nN1YT-0vsRWHhQ2hAcIZ  d\_4 |

ตาราง 3.25 ฐานข้อมูลผู้ใช้

### ตารางวัตถุดิบ (materials)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ลำดับ | ชื่อฟิลด์ | ชนิดข้อมูล | คำอธิบาย | ข้อมูลตัวอย่าง |
| 1 | \_id (PK) | ObjectID | สร้างอัตโนมัติโดย MongoDB | ObjectId(“62861e401904f7babfb13d”) |
| 2 | name (unique) | String | ชื่อวัตถุดิบ | เกลือ |
| 3 | type | String | ประเภทวัตถุดิบ | material |
| 4 | unit | String | หน่วยของวัตถุดิบ | กรัม |

ตาราง 3.26 ฐานข้อมูลวัตถุดิบ

### ตารางอาหาร (items)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ลำดับ | ชื่อฟิลด์ | ชนิดข้อมูล | คำอธิบาย | ข้อมูลตัวอย่าง |
| 1 | \_id (PK) | ObjectID | สร้างอัตโนมัติโดย MongoDB | ObjectId(“62861e401904f7babfb13d”) |
| 2 | name (unique) | String | ชื่ออาหาร | น้ำจิ้มซีฟู๊ด |
| 3 | materials | Array of objects | รายการวัตถุดิบที่ใช้ภายในอาหาร | [  { \_id: value, amount: value },  { \_id: value, amount: value },  { \_id: value, amount: value },  { \_id: value, amount: value },  ] |
| 3.1 | materials.\_id  (FK: materials) | ObjectID | ObjectID ของวัตถุดิบที่ใช้ | ObjectId(“62861e401904f7babfb13d”) |
| 3.2 | materials.amount | Number | จำนวนวัตถุดิบที่ใช้ต่ออาหาร 1 ชุด | 100 |

ตาราง 3.27 ฐานข้อมูลอาหาร

### ตารางเมนูอาหาร (itemsets)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ลำดับ | ชื่อฟิลด์ | ชนิดข้อมูล | คำอธิบาย | ข้อมูลตัวอย่าง |
| 1 | \_id (PK) | ObjectID | สร้างอัตโนมัติโดย MongoDB | ObjectId(“62861e401904f7babfb13d”) |
| 2 | name (unique) | String | ชื่อเมนูอาหาร | ชุดกุ้งดองซีอิ๊ว |
| 3 | items | Object | รายการอาหารที่ใช้ภายในเมนู | [  { \_id: value, amount: value },  { \_id: value, amount: value },  { \_id: value, amount: value },  { \_id: value, amount: value },  ] |
| 3.1 | items.\_id  (FK: items) | ObjectID | ObjectID ของวัตถุดิบที่ใช้ | ObjectId(“62861e401904f7babfb13d”) |
| 3.2 | items.amount | Number | จำนวนอาหารที่ใช้ต่อเมนูอาหาร 1 เมนู | 1 |
| 4 | workCost | Number | ค่าแรงที่ใช้ในการทำเมนูนี้ | 35 |
| 5 | gasCost | Number | ค่าน้ำมันที่ใช้ในการทำเมนูนี้ | 7 |

ตาราง 3.28 ฐานข้อมูลเมนูอาหาร

### ตารางแพลตฟอร์มการขาย (riders)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ลำดับ | ชื่อฟิลด์ | ชนิดข้อมูล | คำอธิบาย | ข้อมูลตัวอย่าง |
| 1 | \_id (PK) | ObjectID | สร้างอัตโนมัติโดย MongoDB | ObjectId(“62861e401904f7babfb13d”) |
| 2 | name (unique) | String | ชื่อแพลตฟอร์มที่ใช้ | Grab |
| 3 | fees | Number | ค่าธรรมเนียมเป็นเปอร์เซ็นต์ | 30 |
| 4 | tax | Number | ภาษีค่าธรรมเนียมเป็นเปอร์เซ็นต์ | 7 |
| 5 | gasCost | Number | ต้นทุนค่าน้ำมันของทุกเมนูในแพลตฟอร์มนี้ ในกรณีที่ไม่ต้องการใช้ต้นทุนค่าน้ำมันของแต่ละเมนู | 1 |
| 6 | prices | Array of objects | รายการอาหารที่ใช้ภายในเมนู | [  { \_id: value, price: value },  { \_id: value, price: value },  { \_id: value, price: value },  { \_id: value, price: value },  ] |
| 6.1 | prices.\_id  (FK: itemsets) | ObjectID | ObjectID ของเมนูอาหารที่จะกำหนดราคา | ObjectId(“62861e401904f7babfb13d”) |
| 6.2 | prices.price | Number | ราคาของเมนูข้างต้นในแพลตฟอร์มนี้ | 259 |

ตาราง 3.29 ฐานข้อมูลแพลตฟอร์มการขาย

### ตารางหมวดหมู่รายจ่าย (expenseCatagories)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ลำดับ | ชื่อฟิลด์ | ชนิดข้อมูล | คำอธิบาย | ข้อมูลตัวอย่าง |
| 1 | \_id (PK) | ObjectID | สร้างอัตโนมัติโดย MongoDB | ObjectId(“62861e401904f7babfb13d”) |
| 2 | name (unique) | String | ชื่อหมวดหมู่ | Business |
| 3 | Parent  (FK: expenseCategories) | ObjectID | ObjectID ของหมวดหมู่อื่น ในกรณีที่ต้องการกำหนดให้หมวดหมู่นี้เป็นหมวดหมู่รองของหมวดหมู่นั้นๆ | ObjectId(“62861e401904f7babfb13d”) |

ตาราง 3.30 ฐานข้อมูลหมวดหมู่รายจ่าย

### ตารางรายจ่าย (expenses)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ลำดับ | ชื่อฟิลด์ | ชนิดข้อมูล | คำอธิบาย | ข้อมูลตัวอย่าง |
| 1 | \_id (PK) | ObjectID | สร้างอัตโนมัติโดย MongoDB | ObjectId(“62861e401904f7babfb13d”) |
| 2 | name | String | คำอธิบายรายการ | Business |
| 3 | cateogry | ObjectID | ObjectID หมวดหมู่ของรายจ่ายนี้ | ObjectId(“62861e401904f7babfb13d”) |
| 4 | Material | ObjectID | ObjectID วัตถุดิบ ในกรณีที่เลือกหมวดหมู่ Ingredient หรือ Packaging | ObjectId(“62861e401904f7babfb13d”) |
| 5 | amount | Number | จำนวนที่ซื้อ | 100 |
| 6 | total | Number | ค่าใช้จ่ายทั้งหมด | 354 |
| 7 | date | Date | วันเวลาที่เกิดรายจ่าย | ISODATE(“06/16/2022”) |

ตาราง 3.31 ฐานข้อมูลรายจ่าย

### ตารางคำสั่งซื้อ (orders)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ลำดับ | ชื่อฟิลด์ | ชนิดข้อมูล | คำอธิบาย | ข้อมูลตัวอย่าง |
| 1 | \_id (PK) | ObjectID | สร้างอัตโนมัติโดย MongoDB | ObjectId(“62861e401904f7babfb13d”) |
| 2 | rider | ObjectID | ObjectID แพลตฟอร์มที่คำสั่งซื้อนี้เกิดขึ้น | ObjectId(“62861e401904f7babfb13d”) |
| 3 | itemsets | Array of objects | รายการเมนูอาหารภายในคำสั่งซื้อ | [  {  \_id: value,  amount: value,  price: value  },  {  \_id: value,  amount: value,  price: value  }  ] |
| 3.1 | itemset.\_id | ObjectID | ObjectID ของเมนูอาหาร | ObjectId(“62861e401904f7babfb13d”) |
| 3.2 | itemset.amount | Number | จำนวนของเมนูอาหารข้างต้น | 2 |
| 3.3 | itemset.price | Number | ราคาปัจจุบันจากไรเดอร์ที่เลือกของเมนูอาหารข้างต้น | 159 |
| 4 | total | Number | ยอดรวมทั้งหมดของคำสั่งซื้อนี้ที่ระบบคำนวณออกมา | 318 |
| 5 | date | Date | วันเวลาที่เกิดคำสั่งซื้อ | ISODATE(“06/16/2022”) |

ตาราง 3.32 ฐานข้อมูลคำสั่งซื้อ

## ส่วนต่อประสาน (User Interface / Screen Layout)

### เข้าสู่ระบบ

Graphical user interface, application

Description automatically generated

รูปประกอบ 3.4 Login UI

### Dashboard – สถิติการขาย

#### การแสดงแบบตาราง

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 3.5 UI ตางรางสถิติการขาย

#### การแสดงแบบกราฟ

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 3.6 UI กราฟสถิติการขาย

### Order – บันทึกคำสั่งซื้อ ส่วนเพิ่มคำสั่งซื้อ, ตารางและการลบคำสั่งซื้อ

Graphical user interface

Description automatically generated with low confidence

รูปประกอบ 3.7 UI การจัดการคำสั่งซื้อ

### Expense - บันทึกรายจ่าย

#### ตารางรายจ่าย การลบ และการทำซ้ำ

A picture containing graphical user interface

Description automatically generated

รูปประกอบ 3.8 UI ตารางบันทึกรายจ่าย

#### ส่วนสร้างรายจ่าย

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 3.9 UI การสร้างรายจ่าย

#### ส่วนแก้ไขรายจ่าย

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 3.10 UI การแก้ไขรายจ่าย

### Category - การสร้างและลบหมวดหมู่รายจ่าย

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 3.11 UI การสร้างและลบหมวดหมู่รายจ่าย

### Material – สร้าง ลบ และตารางข้อมูลวัตถุดิบ

Graphical user interface

Description automatically generated with medium confidence

รูปประกอบ 3.12 UI การจัดการวัตถุดิบ

#### ส่วนแก้ไขวัตถุดิบ

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 3.13 UIการแก้ไขวัตถุดิบ

### Item – ตาราง และการลบอาหาร

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 3.14 UI ตารางรายการอาหาร

#### ส่วนสร้างอาหาร

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 3.15 UI การสร้างอาหาร

#### ส่วนแก้ไขอาหาร

Graphical user interface

Description automatically generated

รูปประกอบ 3.16 UI การแก้ไข้อาหาร

### Itemset – ตาราง และการลบเมนูอาหาร

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 3.17 UI ตารางและการลบเมนูอาหาร

#### ส่วนสร้างเมนูอาหาร

Graphical user interface, application

Description automatically generated

รูปประกอบ 3.18 UIการสร้างเมนูอาหาร

#### ส่วนแก้ไขเมนูอาหาร

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 3.19 UIการแก้ไขเมนูอาหาร

### Rider&Pricing - รายชื่อไรเดอร์ การลบไรเดอร์ และตารางแก้ไขราคา

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 3.20 UI แพลตฟอร์มและตารางราคา

#### ส่วนสร้างไรเดอร์

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 3.21 UI การสร้างแพลตฟอร์ม

# ผลการพัฒนาระบบ (Results)

โปรเจคพัฒนาโปรแกรมจัดการร้านอาหารเดลิเวอรี่ถือว่าสำเร็จลุล่วงได้ดี โดยตัวโปรแกรมสามารถใช้ได้อย่างถูกต้องและตรงกับเป้าหมายที่ผู้ใช้ต้องการ เช่น การจัดเก็บสูตรอาหารได้อย่างแม่นยำ การจัดเก็บราคาอาหารแยกตามแพลตฟอร์ม การบันทึกรายจ่ายได้อย่างละเอียด และมีระบบการค้นหาหลายชั้น การบันทึกคำสั่งซื้อได้สะดวกรวดเร็ว ไม่จำเป็นต้องป้อนค่าจากแป้นพิมพ์ การคำนวณและแสดงผลสถิติที่เจ้าของร้านต้องการ โดยรวมแล้วผู้ใช้นั้นพึงพอใจอย่างมาก ทั้งยังมีความต้องการที่จะนำโปรแกรมนี้ไปใช้งานและต่อยอดพัฒนาร้านค้าต่อไป

## ผลการทดสอบ (Test report)

วิธีการทดสอบที่เลือกใช้คือการทดสอบแบบ Black-Box Testing โดยให้ผู้ที่ทดสอบเป็นที่ไม่มีความรู้ความเข้าใจในเบื้องหลังการทำงานของระบบภายใน (เจ้าของร้าน) เพื่อทดสอบทุกส่วนที่เป็นจุดประสงค์ของการพัฒนาโปรแกรม จากผู้ที่จะนำไปใช้จริง โดยได้ผลการทดสอบดังนี้

|  |  |  |  |
| --- | --- | --- | --- |
| **PROJECT :** | Delivery Restaurant Manager | | |
| **TESTER :** | Thanyaporn Anekawieng | **TEST DATE :** | 25/10/2022 |
| **PREPARED BY :** | Thachin Wisutimarkul | **DOCUMENTED BY :** | Thachin Wisutimarkul |

ตาราง 4.1 ข้อมูลการทดสอบ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Login | | **TEST OBJECTIVE :** Authentication test | | |
| **TEST DATA :** email = admin@smiledong.com, password = 00000000 | | | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Empty input |  | Toast message “Invalid credentaisl” | Pass |
| 2 | Empty email | password = 00000000 | Toast message “Invalid credentaisl” | Pass |
| 3 | Empty password | email = admin@smiledong.com | Toast message “Invalid credentaisl” | Pass |
| 4 | Wrong email | email = ad@smiledong.com, password = 00000000 | Toast message “Invalid credentaisl” | Pass |
| 5 | Wrong password | email = admin@smiledong.com, password = 0000 | Toast message “Invalid credentaisl” | Pass |
| 6 | Correct email and password | email = admin@smiledong.com, password = 00000000 | Redirected to dashboard | Pass |

ตาราง 4.2 ทดสอบ ระบบตรวจสอบตัวตน

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Dashboard | | **TEST OBJECTIVE :** Sales report display test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Default display | Display = default  Rider = default  Time = default | Show tables with sales of every rider in span of 1 month | Pass |
| 2 | Select rider | Display = default  Rider = GRAB  Time = default | Show tables with sales of rider grab in span of 1 month | Pass |
| 3 | Select time range | Display = default  Rider = default  Time = Y | Show tables with sales of every rider in span of 1 year | Pass |
| 4 | Toggle graph view | Display = graph  Rider = default  Time = default | Show bar chart with sales of every rider in span of 1 month | Pass |
| 5 | Graph select rider | Display = graph  Rider = GRAB  Time = default | Show bar chart with sales of rider grab in span of 1 month | Pass |
| 6 | Graph select time range | Display = graph  Rider = default  Time = Y | Show graph with sales of every rider in span of 1 year | Pass |

ตาราง 4.3 ทดสอบ การแสดงสถิติการขาย

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Order | | **TEST OBJECTIVE :** Add new order test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Empty input |  | Nothing happened | Pass |
| 2 | Select item without select rider |  | Unable to click item therefore nothing happened | Pass |
| 3 | Only send time data | Time = 25/10/2022 | Nothing happened | Pass |
| 4 | Only select rider | Rider = Grab | Nothing happened | Fail :  Saved with empty items |
| 5 | Select rider then select item | Item = pickled shrimp  Rider = Grab | Saved with Grab as rider and 1 pickled shrimp as items | Pass |
| 6 | Select rider then select item 3 times | Item = pickled shrimp  Rider = Grab | Saved with Grab as rider and 3 pickled shrimp as items | Pass |
| 7 | Select rider then select several items | Item = [pickled shrimp, rice, seaweed]  Rider = Lineman | Saved with Lineman as rider and 1 pickled shrimp, 1 rice, 1 seaweed as items | Pass |
| 8 | 1. Select rider 2. Select several items 3. Select first item 2 more times | Item = [pickled shrimp, rice, seaweed]  Rider = Lineman | Saved with Lineman as rider and 2 pickled shrimp, 1 rice, 1 seaweed as items | Pass |
| 9 | 1. Select rider 2. Select several items 3. Reduce first item by 1 | Item = [pickled shrimp, rice, seaweed]  Rider = Lineman | Saved with Lineman as rider and 1 rice, 1 seaweed as items | Pass |
| 10 | 1. Select rider 2. Select several items 3. Select first item 4 more times 4. Reduce first item by 2 | Item = [pickled shrimp, rice, seaweed]  Rider = Lineman | Saved with Lineman as rider and 3 pickled shrimp, 1 rice, 1 seaweed as items | Pass |

ตาราง 4.4 ทดสอบ เพิ่มคำสั่งซิ้อ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Order | | **TEST OBJECTIVE :** Delete order test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click delete icon on order |  | Order removed from table | Pass |

ตาราง 4.5 ทดสอบ ลบคำสั่งซื้อ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Expense Category | | **TEST OBJECTIVE :** Add new category test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Empty input |  | Nothing happened | Pass |
| 2 | Empty name | Parent = Business | Nothing happened | Pass |
| 3 | Empty parent | Name = testing | Add new main category name ‘testing’ | Pass |
| 4 | Duplicate name | Name = Business | Nothing happened | Pass |
| 5 | Select Business as parent | Name = testing  Parent = Business | Add new sub category name ‘testing’ under ‘Business’ category | Pass |

ตาราง 4.6 ทดสอบ สร้างหมวดหมู่รายจ่าย

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Expense Category | | **TEST OBJECTIVE :** Delete category test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Delete sub-category |  | Category removed from the list | Pass |
| 2 | Delete main category |  | All sub-category under main category and the main category itself is removed from the list | Pass |

ตาราง 4.7 ทดสอบ ลบหมวดหมู่รายจ่าย

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Expense | | **TEST OBJECTIVE :** Add new expense test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Empty input  Save & Go Back |  | Toast message with error that occurred | Pass |
| 2 | Only send category input  Save & Go Back | Category = ingredient | Redirect to expense table  Toast message with error that occurred | Pass |
| 3 | Only send description input  Save & Go Back | Description = testing | Redirect to expense table  Toast message with error that occurred | Pass |
| 4 | Only send amount input  Save & Go Back | Amount = 5 | Redirect to expense table  Toast message with error that occurred | Pass |
| 5 | Only send total input  Save & Go Back | Total = 200 | Redirect to expense table  Toast message with error that occurred | Pass |
| 6 | Only send date&time input  Save & Go Back | Date&Time =10/04/2022 00:00 | Redirect to expense table  Toast message with error that occurred | Pass |
| 7 | Select category ingredient  Save & Go Back | Category = ingredient | Redirect to expense table  Description input form turn into searchable drop-down input with ingredient list | Pass |
| 8 | Select category packaging  Save & Go Back | Category = packaging | Redirect to expense table  Description input form turn into searchable drop-down input with packaging list | Pass |
| 9 | Select category with sub-categogy | Category = Business | New searchable drop-down input appear beside category input call Sub-category input with Business’s sub category list | Pass |
| 10 | Select ingredient category  Select ingredient  Empty amount, total and date  Save & Go Back | Category = ingredient  Ingredient = Shrimp | Redirect to expense table  Toast message with error that occurred | Pass |
| 11 | Select ingredient category  Select ingredient  Enter amount  Empty total and date  Save & Go Back | Category = ingredient  Ingredient = Shrimp  Amount = 100 | Redirect to expense table  Toast message with error that occurred | Pass |
| 12 | Select ingredient category  Select ingredient  Enter amount  Enter total  Empty date  Save & Go Back | Category = ingredient  Ingredient = Shrimp  Amount = 100  Total = 345 | Redirect to expense table  Saved as Ingredient category, Descript as Shrimp, Amount 100, Total 345 and Date&Time is current time. | Pass |
| 13 | Fill every input  Save & Go Back | Category = ingredient  Ingredient = Shrimp  Amount = 100  Total = 345  Date&Time = 05/10/2022, 00:00 | Redirect to expense table  Saved as Ingredient category, Descript as Shrimp, Amount 100, Total 345 and Date&Time is 05/10/2022, 00:00 | Pass |
| 14 | Fill every input  Save & Add new expense | Category = ingredient  Ingredient = Shrimp  Amount = 100  Total = 345  Date&Time = 05/10/2022, 00:00 | Saved as details  Clear every input except Category and Date&Time | Pass |

ตาราง 4.8 ทดสอบ สร้างรายจ่าย

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Expense | | **TEST OBJECTIVE :** Delete expense test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click delete icon on expense |  | Expense removed from table | Pass |

ตาราง 4.9 ทดสอบ ลบรายจ่าย

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Expense | | **TEST OBJECTIVE :** Repeat expense test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click duplicate icon on expense |  | New expense add to the table with same details as select order except Date; which save as current time  Toast message “Create successfully” | Pass |

ตาราง 4.10 ทดสอบ ทำซ้ำรายจ่าย

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Expense | | **TEST OBJECTIVE :** Edit expense test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click edit icon on expense | Category = Equipment  Description = Glove  Amount = 50  Total = 49  Date&Time = 10/27/2022 16:36 | Open edit form with current expense data | Pass |
| 2 | Modify input  Save & Go back | Description = Scissors  Amount = 2  Total = 69 | Redirect to expenses table  Glove expense at 10/27/2022 16:36 change to Scissors with amount of 2 and total of 69 | Pass |

ตาราง 4.11 ทดสอบ แก้ไขรายจ่าย

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Materials | | **TEST OBJECTIVE :** Add new material test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Empty input  Click ADD |  | Toast message with error that occurred | Pass |
| 2 | Empty unit  Click ADD | Name = test | Toast message with error that occurred | Pass |
| 3 | Empty name  Click ADD | unit = testUnit | Toast message with error that occurred | Pass |
| 4 | Fill all input  Click ADD | Name = test  Unit = testing  Type = materials | Add new material name test, type materials, unit testing into materials table | Pass |
| 5 | Duplicate name  Click ADD | Name = shrimp  Unit = testing  Type = materials | Toast message with error that occurred | Pass |

ตาราง 4.12 ทดสอบ เพิ่มวัตถุดิบ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Materials | | **TEST OBJECTIVE :** Edit material test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click edit icon on material | Name = test  Type = Packaging  Unit = testing | Open edit form with current material data | Pass |
| 2 | Modify input  Save | Name = test2  Type = Ingredient  Unit = testing2 | test material change to test2 with type Ingredient and unit testint2 | Pass |
| 3 | Modify to duplicate name and Save | Name = Shrimp | Toast message with error that occurred | Pass |

ตาราง 4.13 ทดสอบ แก้ไขวัตถุดิบ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Materials | | **TEST OBJECTIVE :** Delete material | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click delete icon on material |  | Material removed from table | Pass |

ตาราง 4.14 ทดสอบ ลบวัตถุดิบ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Items | | **TEST OBJECTIVE :** Add new item | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Save with empty input |  | Nothing happen | Pass |
| 2 | Save with only name | Name = test | Redirect to tables  New Item name test appear in the table | Pass |
| 3 | Enter name  Add new material  Empty material input  Save | Name = test | Nothing happen | Pass |
| 4 | Enter name  Add material with default amount  Save | Name = test  Material = Shrimp | Save with name test  and Shrimp amount 1 | Pass |
| 5 | Add several materials and save | Name = test  Material = [Shrimp, Salmon]  Amount = [1,10] | Save with name test  1 Shrimp, 10 Salmon, | Pass |

ตาราง 4.15 ทดสอบ เพิ่มอาหาร

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Items | | **TEST OBJECTIVE :** Edit item test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click edit icon on material | Name = test  Material = [Shrimp, Salmon]  Amount = [1,10] | Open edit form with current item data | Pass |
| 2 | Modify input  Save | Name = DoubleTrouble  Material = [Shrimp, Salmon]  Amount = [5,45] | test item name change to DoubleTrouble  Shrimp amount change to 5  Salemon amount change to 45 | Pass |

ตาราง 4.16 ทดสอบ แก้ไขอาหาร

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Items | | **TEST OBJECTIVE :** Delete item | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click delete icon on item |  | item removed from table | Pass |

ตาราง 4.17 ทดสอบ ลบอาหาร

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Itemsets | | **TEST OBJECTIVE :** Add new itemsets | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Save with empty input |  | Nothing happen | Pass |
| 2 | Save with only name | Name = test | Redirect to tables  New Item name test appear in the table | Pass |
| 3 | Enter name  Add new item  Empty item input  Save | Name = test | Nothing happen | Pass |
| 4 | Enter name  Add item with default amount  Save | Name = test  Material = Pickled Shrimp | Save with name test Pickled Shrimp amount 1 workcost 0 and gascost 0 | Pass |
| 5 | Add several items and save | Name = test  Material = [PickledShrimp, PickledSalmon]  Amount = [1,10] | Save with name test  1 PickledShrimp, 10 PickledSalmon, workcost 0 and gascost 0 | Pass |
| 6 | Add several items, fill every inputs and save | Name = test  Material = [PickledShrimp, PickledSalmon]  Amount = [1,10]  workCost = 20  gasCost = 6 | Save with name test  1 PickledShrimp, 10 PickledSalmon, workCost 20 and gasCost 6 |  |

ตาราง 4.18 ทดสอบ เพิ่มเมนูอาหาร

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Itemsets | | **TEST OBJECTIVE :** Edit itemset test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click edit icon on material | Name = test  Material = [PickledShrimp, PickledSalmon]  Amount = [1,10]  workCost = 20  gasCost = 6 | Open edit form with current item data | Pass |
| 2 | Modify input  Save | Name = MeatMeat  Amount = [2,2] | test item name change to MeatMeat  PickledShrimp amount change to 2  PickledSalmon amount change to 2 | Pass |

ตาราง 4.19 ทดสอบ แก้ไขเมนูอาหาร

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Itemsets | | **TEST OBJECTIVE :** Delete itemset test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click delete icon on itemset |  | itemset removed from table | Pass |

ตาราง 4.20 ทดสอบ ลบเมนูอาหาร

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Riders&Pricing | | **TEST OBJECTIVE :** Add new rider test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click + icon |  | Rider create form open | Pass |
| 2 | Click - icon |  | Rider create form close without saving | Pass |
| 3 | Save with empty input |  | Toast message with error that occurred | Pass |
| 4 | Save with only name input | Name = test | Toast message with error that occurred | Pass |
| 5 | Save with only fees input | Fees = 30 | Toast message with error that occurred | Pass |
| 6 | Save with only tax input | Tax = 7 | Toast message with error that occurred | Pass |
| 7 | Save with only gas cost input | GasCost = 1 | Toast message with error that occurred | Pass |
| 8 | Save without gas cost input | Name = test  Fees = 30  Tax = 7 | Save test to rider database with fees 30, tax 7 and gasCost 0  test appear on rider list.  When access test; prices table show every menu with 0 in price | Pass |
| 9 | Save with every input filled | Name = test  Fees = 30  Tax = 7  GasCost = 1 | Save test to rider database with fees 30, tax 7 and gasCost 1  test appear on rider list.  When access test; prices table show every menu with 0 in price | Pass |

ตาราง 4.21 ทดสอบ เพิ่มแพลตฟอร์ม

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Riders&Pricing | | **TEST OBJECTIVE :** Delete rider test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Click + icon |  | Rider create form open | Pass |

ตาราง 4.22 ทดสอบ ลบแพลตฟอร์ม

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULE :** Riders&Pricing | | **TEST OBJECTIVE :** Set menu price test | | |
| **Case no.** | **Detail** | **Data** | **Expected Results** | **Status [Fail/Pass]** |
| 1 | Double click on price number |  | Enter price editing mode | Pass |
| 2 | Enter price in string | Price = abcd | Nothing happened | Pass |
| 3 | Enter price in number | Price = 100 | Price save as 100 | Pass |

ตาราง 4.23 ทดสอบ ตั้งราคาเมนูอาหาร

## ผลลัพธ์ของระบบที่พัฒนาในภาพรวม (Output)

Diagram, schematic

Description automatically generated

รูปประกอบ 4.1 การทำงานในภาพรวม

โดยรวมแล้วระบบสามารถรวบรวมกระบวนการทำงานทั้งหมดไว้ในแอพพลิเคชั่นเดียวได้สำเร็จ ไม่มีความจำเป็นต้องแยกการทำงานไปไว้ในหลายระบบหลายอุปกรณ์เหมือนระบบเก่าที่ทางร้านเคยใช้งาน

ผู้ใช้สามารถเก็บข้อมูลสูตรอาหารและวัตถุดิบต่างๆที่ใช้ภายในร้านได้อย่างสมบูรญ์

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 4.2 การเก็บสูตรอาหาร

แต่ละแพลตฟอร์มจะมีราคาของทุกๆเมนูที่แตกต่างกัน สามารถปรับแต่งได้ตามที่ผู้ใช้ต้องการ แบบที่แอพพลิเคชั่นจัดการคำสั่งซื้อทั่วไปไม่สามารถทำได้ง่ายๆ

A screenshot of a computer

Description automatically generated with low confidence

รูปประกอบ 4.3 การตั้งราคาแยกตามแพลตฟอร์ม

ส่วนการลงรายจ่ายภายในร้านมีระบบช่วยเหลือให้เลือกวัตถุดิบได้ง่ายขึ้น สามารถแบ่งหมวดหมู่ได้อย่างชัดเจน สามารถสร้างหมวดหมู่ย่อยเพื่อความแม่นยำในการบันทึกรายจ่าย ระบบยังมีคำสั่งในการทำซ้ำคำสั่งซื้อเพื่อความรวดเร็วในการเพิ่มรายจ่ายที่มีการเพิ่มบ่อยๆอีกด้วย

Graphical user interface

Description automatically generated

รูปประกอบ 4.4 การสร้างคำสั่งซื้อ

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 4.5 การทำซ้ำคำสั่งซื้อ

ในส่วนของการแสดงรายจ่าย ระบบสามารถคัดกรองข้อมูลได้อย่างละเอียด ค้นหาตามชื่อ ค้นหาตามประเภท ค้นหาตามจำนวน ฯลฯ รวมถึงยังสามารถจัดเรียงข้อมูลได้ตามต้องการอีกด้วย

A screenshot of a computer screen

Description automatically generated with medium confidence

รูปประกอบ 4.6 การค้นหาข้อมูลในตาราง

ในการบันทึกคำสั่งซื้อ ผู้ใช้สามารถบันทึกคำสั่งซื้อได้โดยโดยง่ายเพียงเลือกแพลตฟอร์ม เลือกเมนูอาหารและกดบันทึก ทำให้กระบวนการในการบันทึกคำสั่งซื้อเป็นไปอย่างรวดเร็วและง่ายดายกว่าระบบเก่าของทางร้านมาก

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 4.7 การบันทึกคำสั่งซื้อ

ระบบมีการแสดงสถิติการขายที่รวบรวมผลการขายของทุกแพลตฟอร์มไว้ด้วยกัน และยังสามารถกำหนดช่วงระยะเวลาที่ต้องการดูสถิติได้อีกด้วย

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 4.8 กราฟแสดงสถิติการขาย

## ผลประเมินความพึงพอใจของผู้ใช้ระบบ (Assessment)

พนักงานร้าน

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ข้อคำถาม | ระดับความพึ่งพอใจ | | | | |
| น้อยที่สุด | น้อย | ปานกลาง | มาก | มากที่สุด |
| 1. ความง่ายในการใช้งาน |  |  |  | / |  |
| 1. กระบวนการทำงานของระบบ |  |  | / |  |  |
| 1. ความถูกต้องของระบบ |  |  |  |  | / |
| 1. ตรงตามวัตถุประสงค์ที่ต้องการ |  |  |  |  | / |
| 1. การออกแบบเข้าใจง่าย ไม่ซับซ้อน |  |  |  | / |  |
| 1. ความสะดวกในการใช้งาน |  |  |  | / |  |
| 1. ความเหมาะสมในการใช้งาน |  |  |  | / |  |
| 1. ความพึงพอใจในการใช้งาน |  |  |  | / |  |
| 1. ประโยชน์ในการใช้งาน |  |  |  | / |  |

ตาราง 4.24 แบบประเมิณความพึงพอใจของพนักงานร้าน

เจ้าของร้าน

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ข้อคำถาม | ระดับความพึ่งพอใจ | | | | |
| น้อยที่สุด | น้อย | ปานกลาง | มาก | มากที่สุด |
| 1. ความง่ายในการใช้งาน |  |  |  |  | / |
| 1. กระบวนการทำงานของระบบ |  |  |  | / |  |
| 1. ความถูกต้องของระบบ |  |  |  |  | / |
| 1. ตรงตามวัตถุประสงค์ที่ต้องการ |  |  |  |  | / |
| 1. การออกแบบเข้าใจง่าย ไม่ซับซ้อน |  |  |  | / |  |
| 1. ความสะดวกในการใช้งาน |  |  |  | / |  |
| 1. ความเหมาะสมในการใช้งาน |  |  |  | / |  |
| 1. ความพึงพอใจในการใช้งาน |  |  |  | / |  |
| 1. ประโยชน์ในการใช้งาน |  |  |  |  | / |

ตาราง 4.25 แบบประเมิณความพึงพอใจของเจ้าของร้าน

## คู่มือการติดตั้ง (Installation)

### คุณสมบัติของเครื่องคอมพิวเตอร์ที่สามารถติดตั้งโปรแกรมนี้ได้ (Hardware และ Software requirement)

**Hardware**

1. Server
   1. CPU 3GHz or faster with 4 or more cores
   2. At least 4GB Memory
   3. Hard disk free space at least 1.2GB
2. Client
   1. Pentium 4 processor or higher
   2. At least 1GB Memory
   3. Hard disk free space at least 100MB

**Software**

1. Server
   1. Linux
   2. SmartOS
   3. macOS 10.7 or higher
   4. Windows Server 2008 or higher
   5. Node.js 16.0 or higher
   6. NPM 8.0 or higher
   7. Git 2.37 or higher
2. Client
   1. Windows 7, Windows 8, Windows 8.1, Windows 10 or later
   2. macOS High Sierra 10.13 or later
   3. 64-bit Ubuntu 18.04+, Debian 10+, openSUSE 15.2+, or Fedora Linux 32+
   4. Android 7.0 Nougat or later

### รายละเอียดการติดตั้ง

0. จัดเตรียมฐานข้อมูล MongoDB ผู้ใช้สามารถเลือกที่จะใช้ local server ผ่าน community server หรือใช้บริการ MongoDB Atlas ก็ได้ โดยสามารถศึกษาเพิ่มเติมได้ที่ https://www.mongodb.com/docs/atlas/getting-started/

1. ตรวจสอบให้แน่ใจว่า server สามารถใช้ Node.js และ git ได้

2. เปิด command prompt หรือ terminal ในตำแหน่งที่ต้องการติดตั้ง

3. คัดลอกคำสั่งต่อไปนี้ลงใน command prompt หรือ terminal ที่ใช้แล้วรันคำสั่ง

git clone https://github.com/nickethachin/DeliveryRestaurantManager.git --branch CS499 && cd DeliveryRestaurantManager && npm install --force && npm install –prefix frontend –force && npm run build –prefix frontend



รูปประกอบ 4.9 คำสั่งสำหรับติดตั้งโปรแกรม

4. รอจนการติดตั้งโปรแกรมเสร็จสมบูรญ์

5. สร้างไฟล์ใหม่ภายในโฟลเดอร์ของโปรแกรม ตั้งชื่อว่า “.env”

6. แก้ไขไฟล์ .env ด้วยเนื้อหาดังนี้

NODE\_ENV = production

PORT = พอร์ตที่ต้องการให้โปรแกรมรัน

MONGO\_URI = URI สำหรับเชื่อมต่อกับ MongoDB

JWT\_SECRET = รหัสสำหรับสร้างและตรวจสอบ login Token

6.1 ตัวอย่างไฟล์ .env

Text

Description automatically generated

รูปประกอบ 4.10 ตัวอย่างไฟล์ .env

7. จากนั้นสามารถรันคำสั่ง npm start ได้ เมื่อต้องการรันเซิฟเวอร์



รูปประกอบ 4.11 คำสั่งเริ่มต้นระบบ

8. รอจนโปรแกรมทำงานจนเสร็จสมบูรญ์

Graphical user interface, text

Description automatically generated

รูปประกอบ 4.12 ตัวอย่างเมื่อเริ่มต้นระบบสำเร็จ

9. เมื่อเซิฟเวอร์ทำงานแล้ว เราสามารถเข้าถึงโปรแกรมได้โดยการเข้าผ่าน Web browser และใช้ url ของเครื่อง และ port ที่กำหนด ในกรณีที่เครื่องที่รันเซิฟเวอร์กับเครื่องที่จะเข้าใช้โปรแกรมเป็นเครื่องเดียวกัน และ port ที่กำหนดคือ 5000 ผู้ใช้สามารถเข้าถึงโปรแกรมได้ที่ http://localhost:5000

Graphical user interface, application

Description automatically generated

รูปประกอบ 4.13 ตัวอย่างหน้าเข้าสู่ระบบ

10. เมื่อเข้ามาแล้วระบบจะต้องการให้เราล็อกอิน ให้เราไปที http://localhost:5000/register เพื่อสมัครสมาชิกก่อน โดยช่องแรกให้กรอกชื่อที่ต้องการ ช่องที่สองกรอกอีเมล์ ช่องสามและสี่กรอกพาสเวิร์ดที่ต้องการ จากนั้นกดปุ่ม Submit ด้านล่าง

Graphical user interface, application

Description automatically generated

รูปประกอบ 4.14 ตัวอย่างการกรอกข้อมูลหน้าสมัครสมาชิก

11. เมื่อสมัครสมาชิกสำเร็จและเข้าสู่หน้า Dashboard ได้ ก็ถือได้ว่าติดตั้งเสร็จสมบูรญ์แล้ว

## คู่มือการใช้งาน (User manual)

1. เมื่อเข้าใช้งานครั้งแรก ผู้ใช้จะต้องเข้าสู่ระบบก่อน ช่องแรกใส่ email และช่องที่สองใส่ password ที่ได้สมัครไว้ในตอนติดตั้งโปรแกรม

Graphical user interface, application

Description automatically generated

รูปประกอบ 4.15 ลักษณะของหน้าเข้าสู่ระบบ

2. จากนั้น ผู้ใช้จะต้องตั้งค่าเมนูอาหาร ไรเดอร์ และราคาอาหาร โดยกระบวนการนี้ จะทำเพียงครั้งเดียวเท่านั้น เว้นเสียแต่ว่า จะมีการปรับราคาหรือสูตรอาหาร

2.1 คลิ๊กไปที่เมนู Material ทางฝั่งซ้ายของโปรแกรม

Graphical user interface, application

Description automatically generated

รูปประกอบ 4.16 ตำแหน่งเมนู Material

2.2 จากนั้น ทำการเพิ่มวัตถุดิบภายในร้านทีละอย่าง สามารถทำได้โดยการกรอกลงไปภายในช่องข้อมูลด้านบน จากซ้ายไปขวา ช่องที่ต้องกรอกจะเป็น ชื่อ ประเภท และหน่วย แล้วกดปุ่ม ADD

ชื่อ คือชื่อของวัตถุดิบที่จะเพิ่ม

ประเภท คือ ประเภทว่าเป็น Ingredient (วัตถุดิบ) หรือ Packaging (บรรจุภัณฑ์)

หน่วย คือ หน่วยเรียกของวัตถุดิบนั้นๆ เช่น ชิ้น ตัว กรัม ขวด มิลลิกรัม A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 4.17 การสร้างวัตถุดิบ

2.2.1 เมื่อเพิ่มสำเร็จข้อมูลจะถูกเพิ่มไปยังตารางด้านล่าง

2.2.2 ผู้ใช้สามารถกดปุ่มดินสอสีเขียวทางด้านขวาเพื่อแก้ไขข้อมูลได้

2.2.3 ผู้ใช้สามารถกดปุ่มถังขยะสีแดงทางด้านขวาเพื่อลบข้อมูลได้

2.3 ทำซ้ำไปเรื่อยๆจนกว่าจะลงทะเบียนวัตถุดิบและบรรจุภัณฑ์ที่ใช้ภายในร้านจนหมด

2.4 ไปที่เมนู Item ทางด้านซ้ายของโปรแกรม

Graphical user interface, application

Description automatically generated

รูปประกอบ 4.18 ตำแหน่งเมนู Item

2.5 กดปุ่ม Create ทางด้านขวาเพื่อเริ่มสร้างอาหาร Graphical user interface, application

Description automatically generated

รูปประกอบ 4.19 ตำแหน่งปุ่มสร้างอาหาร

2.6 สร้างอาหารโดยกรอกข้อมูลดังนี้

2.6.1 ช่องว่างช่องแรก กรอกชื่ออาหารที่ต้องการ

2.6.2 กรอกวัตถุดิบที่ใช้

2.6.2.1 กดปุ่ม  เพื่อเพิ่มวัตถุดิบ

2.6.2.2 ในช่อง Material สามารถกดปุ่มสามเหลี่ยมเล็กๆหรือพิมพ์หาของที่ใช้

Graphical user interface

Description automatically generated

รูปประกอบ 4.20 Searchable dropdown

2.6.2.2.1 หากไม่มีวัตถุดิบที่ต้องการ กลับไปทำข้อ 2.2

2.6.2.3 ในช่องจำนวน ให้กรอกจำนวนของที่ใช้ต่ออาหาร 1 ชุด

2.6.2.4 สามารถกดปุ่ม  เพื่อเปิดเครื่องคิดเลขได้

2.6.3 กดปุ่ม  ทางด้านขวา เพื่อบันทึกอาหาร

2.6.4 เมื่อเพิ่มสำเร็จข้อมูลจะถูกเพิ่มไปยังตาราง

2.6.5 ผู้ใช้สามารถกดปุ่มดินสอสีเขียวทางด้านขวาเพื่อแก้ไขข้อมูลได้

2.6.6 ผู้ใช้สามารถกดปุ่มถังขยะสีแดงทางด้านขวาเพื่อลบข้อมูลได้

2.6.7 ทำซ้ำจนมีอาหารเท่าที่ต้องการ

2.7 สร้างเมนูอาหารโดยกรอกข้อมูลดังนี้

2.7.1 ช่องว่างช่องแรก กรอกชื่อเมนูอาหารที่ต้องการ

2.7.2 ช่อง Work Cost กรอกต้นทุนค่าแรงที่ใช้ในการทำเมนูนี้

2.7.3 ช่อง Gas Cost กรอกต้นทุนค่าน้ำมันรถที่ใช้ในการทำเมนูนี้

2.7.4 กรอกอาหารที่ใช้ประกอบเมนูนี้

2.7.4.1 กดปุ่ม  เพื่อเพิ่มวัตถุดิบ

2.7.4.2 ในช่อง Item สามารถกดปุ่มสามเหลี่ยมเล็กๆหรือพิมพ์หาของที่ใช้

A screenshot of a computer

Description automatically generated with medium confidence

2.7.4.2.1 หาไม่มีอาหารที่ต้องการ กลับไปทำข้อ 2.6

2.7.4.3 ในช่องจำนวน ให้กรอกจำนวนของที่ใช้ต่อเมนู 1 เมนู

2.7.4.4 สามารถกดปุ่ม  เพื่อเปิดเครื่องคิดเลขได้

2.7.5 กดปุ่ม  ทางด้านขวา เพื่อบันทึกอาหาร

2.7.6 เมื่อเพิ่มสำเร็จข้อมูลจะถูกเพิ่มไปยังตาราง

2.7.7 ผู้ใช้สามารถกดปุ่มดินสอสีเขียวทางด้านขวาเพื่อแก้ไขข้อมูลได้

2.7.8 ผู้ใช้สามารถกดปุ่มถังขยะสีแดงทางด้านขวาเพื่อลบข้อมูลได้

2.8 ไปที่เมนู Rider&Pricing เพื่อตั้งค่าไรเดอร์

Graphical user interface, application

Description automatically generated

รูปประกอบ 4.21 ตำแหน่งเมนู Rider&Pricing

2.8.1 กดปุ่ม  ทางซ้ายเพื่อเปิดเมนูสร้างไรเดอร์

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 4.22 ตำแหน่งปุ่มสร้างแพลตฟอร์ม

2.8.2 กรอกข้อมูลดังนี้ เพื่อสร้างไรเดอร์

Name: ชื่อไรเดอร์

Fees: ค่าธรรมเนียม (%)

Tax: ภาษีค่าธรรมเนียม (%)

Gas Cost: ต้นทุนค่าน้ำมันสำหรับทุกเมนูในไรเดอร์นี้

2.8.3 กดปุ่ม  ทางขวาของหน้าจอเพื่อบันทึกไรเดอร์ใหม่

2.8.4 ทำซ้ำจนได้ไรเดอร์ครบที่ต้องการ

2.9 ตั้งค่าราคา

2.9.1 เลือกไรเดอร์ที่ต้องการแก้ไขราคา

A screenshot of a computer

Description automatically generated with medium confidence

รูปประกอบ 4.23 การเลือกไรเดอร์

2.9.2 ดับเบิ้ลคลิ๊กที่ราคาของเมนูที่ต้องการแก้ไข

Application

Description automatically generated with low confidence

รูปประกอบ 4.24 ตำแหน่งในการแก้ไขราคา

2.9.3 แก้ไขราคาเป็นราคาที่ต้องการแล้วกด Enter หรือคลิ๊กที่พื้นที่ว่างข้างนอก

2.9.4 ทำซ้ำจนแก้ไขราคาที่ต้องการจนครบ

2.10 ตั้งค่าหมวดหมู่รายจ่าย

2.10.1 ไปที่เมนู Expense

Graphical user interface, application

Description automatically generated

รูปประกอบ 4.25 ตำแหน่งเมนู Expense

2.10.2 กดที่ Category Edit ทางด้านขวาของหน้าจอ

A screenshot of a computer screen

Description automatically generated with medium confidence

รูปประกอบ 4.26 ตำแหน่งปุ่มแก้ไขหมวดหมู่รายจ่าย

2.10.3 หากต้องการเพิ่มหมวดหมู่หลัก ให้ใส่ชื่อหมวดหมู่ที่ช่อง Name แล้วกด  บันทึก

2.10.4 หากต้องการเพิ่มหมวดหมู่รอง

2.10.4.1 กรอกชื่อที่ต้องการ

2.10.4.2 เลือกหมวกหมู่หลักจากช่อง Parent

2.10.4.3 กดบันทึก 

2.10.5 ผู้ใช้สามารถแก้ไขหมวดหมู่ได้โดยการกดที่ชื่อหมวดหมู่ด้านล่าง

2.10.6 ผู้ใช้สามารถลบหมวดหมู่ได้โดยการกด  ทางด้านขวา

2.11 เสร็จสิ้นขั้นตอนการตั้งค่าร้านค้าครั้งแรก

3. การจัดการรายจ่าย

3.1 ไปที่เมนู Expense

3.2 กดปุ่ม Create ทางด้านขวาของหน้าจอเพื่อเพิ่มรายจ่าย

Graphical user interface, application

Description automatically generated

รูปประกอบ 4.27 ตำแหน่งปุ่มสร้างคำสั่งซื้อ

3.2.1 เลือกหมวดหมู่ที่ช่อง Category

3.2.1.1 กรณีที่หมวดหมู่ที่เลือกมีหมวดหมู่รอง สามารถเลือกหมวดหมู่รองได้

3.2.2 เลือก material ในกรณีที่เลือกหมวดหมู่เป็น Ingredient หรือ Packaging

3.2.3 กรอกชื่อหรือคำอธิบายเมื่อเลือกหมวดหมู่อื่น

3.2.4 กรอกจำนวนที่ซื้อ

3.2.5 กรอกยอดรวมทั้งหมด

3.2.6 สามารถเลือกวันเวลาทึ่ค่าใช้จ่ายเกิดขึ้นได้ หากปล่อยว่างจะใช้วันเวลาปัจจุบัน

3.2.7.1 กด Save & Add new expense เพื่อบันทึกและเตรียมสร้างรายการถัดไป

3.2.7.2 กด Save & Go back เพื่อบันทึกและกลับไปยังหน้าตารางรายการ

3.3 ผู้ใช้สามารถกดปุ่ม  เพื่อแก้ไขรายการที่เลือกได้

3.4 ผู้ใช้สามารถกด  เพื่อสร้างรายการใหม่จากข้อมูลของรายการที่เลือก

3.5 ผู้ใช้สามารถกด  เพื่อลบรายการที่เลือกได้

4. การจัดการคำสั่งซื้อ

4.1 กดที่ Order เมนูทางด้านซ้ายของโปรแกรม

Graphical user interface, application

Description automatically generated

รูปประกอบ 4.28 ตำแหน่งเมนู Order

4.2 สร้างคำสั่งซื้อใหม่

4.2.1 เลือกไรเดอร์ที่ต้องการ

Graphical user interface

Description automatically generated with medium confidence

รูปประกอบ 4.29 ส่วนสำหรับเลือกไรเดอร์

4.2.2 คลิ๊กซ้ายที่เมนูที่ต้องการเพื่อเพิ่มเมนูเข้าสู่คำสั่งซื้อ

Graphical user interface

Description automatically generated

รูปประกอบ 4.30 ส่วนสำหรับเลือกเมนูอาหาร

4.2.3 ผู้ใช้สามารถกดที่เมนูซ้ำได้เพื่อเพิ่มจำนวนเมนูเดิมลงในคำสั่งซื้อ

4.2.4 ผู้ใช้สามารถใช้ปุ่ม  เพื่อเพิ่มหรือลดจำนวนเมนูอาหารได้

4.2.5 ผู้ใช้สามารถเลือกวันเวลาที่คำสั่งซื้อเกิดหรือปล่อยว่างเพื่อใช้เวลาปัจจุบันได้

4.2.6 กดปุ่ม  เพื่อบันทึกคำสั่งซื้อลงในระบบ

4.3 ผู้ใช้สามารถกดปุ่ม  เพื่อลบคำสั่งซื้อที่ไม่ต้องการได้

5. สถิติการขาย

5.1 ไปที่เมนู Dashboard

Graphical user interface, application

Description automatically generated

รูปประกอบ 4.31 ตำแหน่งเมนู Dashboard

5.2 ผู้ใช้สามารถเลือกไรเดอร์ที่ต้องการได้

A picture containing graphical user interface

Description automatically generated

รูปประกอบ 4.32 การเลือกไรเดอร์ใน Dashboard

5.3 ผู้ใช้สามารถเลือกขอบเขตระยะเวลาได้

Graphical user interface, application

Description automatically generated

รูปประกอบ 4.33 การเลือกขอบเขตเวลา

D = คำสั่งซื้อที่เกิดขึ้นใน 24 ชั่วโมงที่ผ่านมา

W = คำสั่งซื้อที่เกิดขึ้นใน 7 วันที่ผ่านมา

M = คำสั่งซื้อที่เกิดขึ้นใน 1 เดือนที่ผ่านมา

Y = คำสั่งซื้อที่เกิดขึ้นใน 1 ปีที่ผ่านมา

5.4 ผู้ใช้สามารถกดปุ่ม เพื่อเปิดหรือปิดโหมดแสดงสถิติผ่านกราฟแท่งได้

Chart

Description automatically generated

รูปประกอบ 4.34 กราฟสถิติการขาย

# บทสรุปและข้อเสนอแนะ (Conclusions and Recommendations)

## วัตถุประสงค์หรือประเด็นสำคัญในการพัฒนา

โปรเจคระบบจัดการร้านอาหารเดลิเวอรี่ ถูกพัฒนาขึ้นเพื่อช่วยเหลือร้านอาหารรูปแบบใหม่ ที่ได้รับความนิยมมากขึ้นในช่วงปี 2021 ที่ผ่านมา ร้านอาหารประเภทใหม่ที่กล่าวถึงนี้ คือร้านอาหารที่ไม่มีหน้าร้าน ไม่มีโต๊ะรองรับลูกค้า เพียงเปิดเมนูตามแพลตฟอร์มส่งอาหารออนไลน์ต่างๆ ร้านอาหารประเภทนี้จะพบเจอปัญหาที่ว่า แพลตฟอร์มมีมากมายหลายบริษัท แต่ละบริษัทก็ต่างคิดค่าบริการ ภาษี ค่าสมาชิก แตกต่างกันไป ทำให้ต้นทุนอาหารที่ผลิต ราคาอาหารที่ขาย และค่าใช้จ่ายอื่นๆ มีความซับซ้อนมากกว่าร้านอาหารทั่วไปที่ไม่มีบริการส่งอาหารออนไลน์ หรือใช้แพลตฟอร์มส่งอาหารแค่แพลตฟอร์มเดียว

ในตัวอย่างร้านอาหาร ‘สมายดอง’ ที่ผู้จัดทำได้เลือกใช้ในการวิจัยข้อมูล ทางร้านไม่มี่ระบบ POS เนื่องจากระบบ POS ทั่วไปไม่รองรับการที่เมนูหนึ่งเมนูมีราคา 4 – 5 ราคา ตามแต่แพลตฟอร์มส่งอาหารที่เลือกใช้ ทำให้ทางร้านต้องใช้ระบบการบันทึกคำสั่งซื้อที่ยุ่งยาก วุ่นวาย ไม่สะดวกในการใช้งาน

นอกจากนี้ ทางร้านเองยังประสบปัญหาในการบันทึกข้อมูลคำสั่งซื้อในระยะยาว และจัดทำเป็นสถิติการขาย เพื่อปรับปรุงร้านและจัดทำโปรโมชั่นต่างๆ โดยทั่วไปแล้ว แต่ละแพลตฟอร์มก็มีการเก็บสถิติของตัวเอง แต่วิธีการเก็บเหล่านี้ก็ไม่เป็นมาตรฐานเดียวกัน การค้นหาข้อมูลก็ยากลำบาก ทำให้การรวบรวมข้อมูลการขายมาไว้ ณ ที่เดียวกันไม่ใช่เรื่องง่าย

อีกปัญหาหนึ่งที่ทางร้านพบเจอคือ ระบบที่ใช้บันทึกข้อมูลการขายอยู่ส่วนหนึ่ง ระบบที่ใช้บันทึกรายจ่ายภายในร้านก็อยู่อีกส่วนหนึ่ง เมื่อทางร้านต้องการจะวิเคราะห์ข้อมูลเพื่อนำไปใช้ในการจัดการบัญชีหรืออื่นๆ ก็ต้องค่อยๆเก็บข้อมูลทีละระบบ ก่อนที่จะนำมารวมกันแล้ววิเคราะห์ออกมาได้ และระบบเหล่านี้ก็ยังมีค่าใช้จ่ายหรือค่าสมาชิกรายเดือนแยกกันไปต่างหาก

ส่วนของระบบบันทึกรายจ่ายแต่เดิมของร้านนั้น ทางร้านเองค่อนข้างพอใจกับระบบเดิม แต่หลังจากการวิเคราะห์การทำงานแล้วนั้น ผู้จัดทำเล็งเห็นว่า ระบบเดิมยังไม่มีการจัดเก็บข้อมูลที่เป็นมาตรฐานมากพอ ทำให้การค้นหาข้อมูลรายจ่ายมีความยากลำบากอยู่บ้าง และบางรายจ่ายที่เกิดขึ้นซ้ำๆ ก็ยังคงต้องกรอกข้อมูลใหม่ตั้งแต่ต้นทุกวันเพื่อบันทึกรายจ่ายนั้น

## ผลการพัฒนาสามารถแก้ปัญหาได้มากน้อยเพียงใด

โปรแกรมจัดการร้านอาหารเดลิเวอรี่ที่พัฒนาขึ้นมานั้น ได้รวบรวมระบบทุกอย่างที่ทางร้านต้องการไว้ ทำให้สามารถทำงานได้อย่างครบวงจรภายในที่เดียว ไม่ว่าจะเป็นการจัดเก็บข้อมูลสูตรอาหาร การบันทึกรายจ่าย การบันทึกคำสั่งซื้อ และการทำสถิติการขาย

ระบบบันทึกรายจ่าย ถูกสร้างขึ้นเพื่อคงไว้ซึ่งทุกอย่างที่ผู้ใช้ชื่นชอบในระบบเก่า และเพิ่มเติมคำสั่งใหม่ๆ ที่จะช่วยให้การทำงานง่ายขึ้น เช่น ปุ่ม Repeat ที่ปรากฏที่รายจ่ายเก่าๆ ที่เมื่อกดไปแล้ว ระบบจะเพิ่มรายการใหม่ในเวลาปัจจุบัน ด้วยข้อมูลของรายจ่ายที่เลือก หรือ การเลือกวัตถุดิบจากฐานข้อมูลโดยตรงแทนการพิมพ์ชื่อสินค้าที่ซื้อ ทำให้การเก็บข้อมูลมีมาตรฐาน ค้นหาได้ง่าย แต่ยังคงระบบพิมพ์ชื่อรายจ่ายเอาไว้ เมื่อเลือกหมวดหมู่อื่นๆที่ไม่ใช่วัตถุดิบ ทำให้ระบบยังมีความยืดหยุ่นอยู่เช่นเดิม ทำให้ปัญหาในการค้นหารายจ่าย และความยุ่งยากในการทำซ้ำรายจ่ายหมดไป

ระบบบันทึกคำสั่งซื้อ ถูกออกแบบสำหรับการใช้งานที่สะดวก รวดเร็ว และเข้าถึงได้ง่าย ในส่วนระบบบันทึกคำสั่งซื้อนั้น ได้ออกแบบมาให้ responsive เป็นพิเศษ ทำให้สามารถใช้งานได้ง่ายทั้งใน Desktop PC หรือ Mobile Device นอกจากนี้ ในส่วนของการบันทึกข้อมูลนั้น ผู้ใช้สามารถเพิ่มข้อมูลได้เพียงการกดเลือกเท่านั้น ไม่จำเป็นต้องพิมพ์ค่าข้อมูลใดๆ ส่วนราคา ยอดรวม และรายละเอียดอื่นๆนั้น ระบบจะคำนวณให้โดยอัตโนมัติ จากการที่ผู้ใช้ได้กำหนดราคาต่างๆไว้ล่วงหน้า ตามแพลตฟอร์มส่งอาหารที่เลือกใช้ ทำให้ปัญหาในการบันทึกคำสั่งซื้อ และความยุ่งยากในการเข้าถึงระบบบันทึกคำสั่งซื้อหมดไป

ระบบยังสามารถแสดงสถิติการขายได้ในทุกที่ทุกเวลา ทั้งในรูปแบบตาราง และรูปแบบกราฟ โดยผู้ใช้สามารถเลือกได้ว่าต้องการจะดูสถิติการขายจากแพลตฟอร์มใดบ้าง และยังเลือกช่วงเวลาที่ต้องการได้อีกด้วย ทำให้จากเดิมที่การรวบรวมข้อมูลการขายต่างๆที่เป็นไปไม่ได้ หรือเป็นไปได้ยาก สามารถทำได้ง่ายๆ เพียงกดเลือกสิ่งที่ต้องการ

## สิ่งที่คาดหวังที่ระบบจะสามารถช่วยในการทำงานได้มากน้อยเพียงใด

ผู้จัดทำคาดหวังว่าระบบจัดการร้านอาหารเดลิเวอรี่นี้ จะช่วยให้ทางร้านสามารถจัดการร้านได้อย่างสะดวก รวดเร็ว แล้วถูกต้องมากยิ่งขึ้น เนื่องด้วยเหตุผลดังนี้

1. ระบบบันทึกสูตรอาหารที่ละเอียดแม่นยำ จะช่วยให้ทางร้านสามารถทำอาหารในปริมาณที่ต้องการได้ง่ายมากขึ้นกว่าระบบถ้วยตวงมาก
2. ระบบบันทึกรายจ่ายที่มีมาตรฐานและสามารถทำซ้ำรายจ่ายได้อย่างง่ายดาย จะช่วยให้เวลาที่ใช้ในการลงบัญชีรายจ่ายน้อยลง และการค้นหาข้อมูลสินค้าย้อนหลังทำได้ง่ายและแม่นยำขึ้น
3. ระบบบันทึกคำสั่งซื้อที่ใช้ได้ง่าย และเข้าถีงได้จากโทรศัพท์มือถือ จะช่วยให้เวลาที่ใช้ในการบันทึกคำสั่งซื้อน้อยลงมากๆ และถูกต้องมากขึ้น เมื่อเทียบกับระบบเก่าที่ต้องกวาดหาข้อมูลจากในตารางและบันทึกเองด้วยมือ นอกจากนี้ทางร้านจะมีประวัติการขายให้ดูย้อนหลังได้อีกด้วย
4. ระบบแสดงสถิติการขาย จะช่วยให้ทางร้านสามารถตรวจสอบยอดขายได้ตามต้องการ ทั้งการแบ่งสถิติตามแพลตฟอร์ม หรือกระทั่งการเลือกช่วงระยะเวลาที่ต้องการ จากเดิมที่การตรวจสอบเพื่อทำสถิติในระยะหนึ่งเดือนทำได้ยาก และในระยะหกเดือนขึ้นไปไม่สามารถตรวจสอบได้เลย

## ปัญหาและข้อเสนอแนะ

ปัญหาที่พบเจอสามารถแบ่งออกได้เป็น 2 ส่วนคือ ปัญหาทางกระบวนการทำงานของทางร้าน และปัญหาทางเทคนิคในการพัฒนาโปรแกรม

**ปัญหาทางกระบวนการทำงาน**

1. Flow การทำงานของระบบเดิมมีความซับซ้อนและไม่ตรงกับหลักในการพัฒนาโปรแกรมโดยทั่วไป และทางร้านเองก็ยืนยันว่าไม่ต้องการจะเป็นระบบการทำงานตรงนี้ เนื่องจากความเคยชินและความเชี่ยวชาญของทั้งพนักงานและเจ้าของร้านเอง ปัญหานี้ได้รับการแก้ไขโดย การออกแบบโครงสร้างโปรแกรมอย่างละเอียด เพื่อให้ตรงกับความต้องการของผู้ใช้ และความเข้าใจของผู้จัดทำ
2. ทางร้านไม่มีการตัดต้นทุนค่าแรงเป็นรายเดือน แต่ใช้วิธีกำหนดให้แต่ละเมนูอาหาร มีต้นทุนค่าแรงที่แตกต่างกันออกไป ปัญหานี้ได้รับการแก้ไขโดย การสร้าง field ใหม่ในฐานข้อมูลเมนูอาหาร เพื่อเก็บต้นทุนค่าแรงในแต่ละเมนู
3. ทางร้านไม่มีการตัดต้นทุนค่าน้ำมันตามการใช้งานจริง แต่ใช้วิธีกำหนดให้แต่ละเมนูอาหาร มีต้นทุนค่าน้ำมันที่แตกต่างกันออกไป ปัญหานี้ได้รับการแก้ไขโดย การสร้าง field ใหม่ในฐานข้อมูลเมนูอาหาร เพื่อเก็บต้นทุนค่าน้ำมันในแต่ละเมนู
4. ทางร้านต้องการให้บางแพลตฟอร์ม ไม่ใช้ต้นทุนค่าน้ำมันจากเมนูอาหาร แต่ทุกเมนูจะใช้ต้นทุนค่าน้ำมันเท่ากัน เพียงแต่ แต่ละแพลตฟอร์มก็จะมีต้นทุนค่าน้ำมันตรงนี้ที่แตกต่างกันออกไป ปัญหานี้ได้รับการแก้ไขโดย การที่กำหนด field ใหม่ในฐานข้อมูลแพลตฟอร์มส่งอาหาร แล้วให้ระบบคำนวณต้นทุนจากตรงนี้ก่อน หากแพลตฟอร์มไม่มีค่านี้ จึงจะใช้ค่าต้นทุนน้ำมันจากเมนูอาหารนั้นๆ

**ปัญหาทางเทคนิค**

1. ผู้จัดทำต้องการจะใช้ MERN Stack ในการพัฒนาโปรแกรม เพื่อเรียนรู้ JavaScript ในเชิงลึก และเรียนรู้เทคโนโลยีใหม่ๆที่ไม่เคยได้ลองใช้ ทั้ง ฐานข้อมูลแบบ NoSQL, การทำงานกับ JavaScript Library, การใช้ Framework ในการเขียน REST API และอื่นๆอีกมากมาย ทำให้คุณภาพของโค้ดภายในโปรแกรมนั้นมีความไม่เท่ากัน โดยที่ผลงานหรือ components ในช่วงแรกจะมีคุณภาพต่ำ และผลงานต่อๆไปก็มีคุณภาพสูงขึ้นเรื่อยๆ ตามการเรียนรู้ของผู้จัดทำ ปัญหานี้ได้รับการแก้ไขโดย การกลับมารีแฟคเตอร์ (refactor) หรือกระทั่งรีเวิร์ค (rework) โค้ดเก่าๆใหม่ทั้งหมด ทำให้คุณภาพโค้ดมีความใกล้เคียงกัน
2. หมวดหมู่ค่าใช้จ่ายมีการเก็บข้อมูลแบบ Self one to many กล่าวคือ ในข้อมูลหนึ่งอาจอ้างถึงข้อมูลอื่นๆใน collection เดียวกัน เป็นปัญหาในส่วนของการทำ Sub-category เมื่อผู้ใช้ต้องการลบหมวดหมู่ที่เป็น Sub-category จะมีความยุ่งยากมากกว่าปกติ หรือเมื่อผู้ใช้ต้องการเปลี่ยน Parent ของ Sub-category การโยกย้าย Embedded documents นั้นไม่ใช่เรื่องง่าย ปัญหานี้ได้รับการแก้ไขโดย การปรับเปลี่ยนแนวคิดและ Algorithm ใหม่ทั้งหมด แทนที่จะใช้ Embedded documents เพื่อเก็บข้อมูล Sub-cateogry ก็ให้ทุก category อยู่ใน level ที่เท่ากันทั้งหมด และเฉพาะ category ที่ไม่มี parent เท่านั้น ระบบจึงจะจัดว่าเป็น Main category
3. Form ในการเก็บส่วนประกอบของอาหารหรือเมนูอาหารนั้นมีความ dynamic ที่สูงมาก ผู้ใช้สามารถเพิ่มและลดจำนวน field ได้ตามชอบ ทำให้การทำ form component ทำได้ยาก และการส่งต่อข้อมูลก็มีความซับซ้อน เนื่องจากที่ผ่านมา แต่ละ input ใน form จะใช้ state แยกกัน ปัญหานี้ได้รับการแก้ไขโดย การกำหนด state ให้เป็น Array และส่งต่อข้อมูลด้วยค่า index ของ input ที่ loop ขึ้นมา
4. การกำหนดราคาของแต่ละเมนูมีความคล่องตัวต่ำ ทำให้ใช้เวลามากในการกำหนดราคาของทุกเมนูในทุกๆแพลตฟอร์ม ปัญหานี้ได้รับการแก้ไขโดย การเปลี่ยนจากการกรอกข้อมูลในรูปแบบเดิมมาเป็นการใช้ DataGrid ของ MaterialUI และใช้ feature ที่ชื่อว่า cell editing
5. ในการที่ React จะทำการ re-render component แต่ละครั้งนั้น React จะเปรียบเทียบ state แต่ละ state ว่ามีการเปลี่ยนอะไรหรือไม่ หากไม่เปลี่ยนก็จะไม่มีการ re-render เกิดขึ้น ซึ่งใน JavaScript การเปรียบเทียบค่าระหว่าง Object กับ Object หรือ Array กับ Array จะเปรียบเทียบกันด้วยค่า reference กล่าวคือ จะไม่มีการเปรียบเทียบค่าภายในของ Object เรียกว่า Shallow compare ปัญหาที่เกิดขึ้นคือ เมื่อมีการเปลี่ยนแปลงข้อมูลภายใน state แบบ Object หรือ Array ไม่ว่าค่าภายในจะแตกต่างจากเดิมเท่าไร React ก็จะไม่ทำการ re-render components ปัญหานี้ได้รับการแก้ไขโดย ทุกๆการแก้ไข state ที่เป็น Object หรือ Array จะต้องเป็นการสร้างใหม่ทุกครั้งโดยการใช้ spread operator ช่วย แทนที่จะกำหนดค่าเข้าไปในตัวแปรโดยตรง
6. หลังจากผู้จัดทำได้เรียนรู้และทำความเข้าใจ React มากขึ้น ผู้จัดทำจึงได้ข้อสรุปว่า การพัฒนาโปรแกรมในลักษณะนี้ ไม่มีความจำเป็นต้องต้องใช้ Redux state management ปัญหานี้ไม่ได้รับการแก้ไข เนื่องจากหากต้องการนำ Redux ออก จะต้องพัฒนาโปรแกรมใหม่ทั้งหมดตั้งแต่ต้น และทางผู้จัดทำก็ไม่เห็นถึงผลกระทบในภาพรวม จึงไม่มีความจำเป็นต้องเริ่มพัฒนาใหม่ทั้งหมดเพื่อแก้ปัญหานี้ แต่หากต้องการจะแก้ไข สิ่งที่ต้องทำคือ เอาระบบ Redux ออกทั้งหมด แล้วทดแทนด้วยการให้ component แม่ fetch ข้อมูลจาก API แล้วใช้ Context hook provide ข้อมูลลงไปที่ component ลูก

# บรรณานุกรม

Jirayut Intachai. (2564). **HTML คืออะไร? ทำไมคนเขียนเว็บไซต์ต้องใช้งาน.** จาก Terrestrial: https://goterrestrial.com/2021/05/19/what-is-html/

Kipakaporn. (2561**). CSS คืออะไร มีประโยชน์ อย่างไร.** จาก SoGoodWeb: https://blog.sogoodweb.com/Article/Detail/79237/CSS-คืออะไร-มีประโยชน์-อย่างไร

Worapong. (2563). **JavaScript คืออะไร ใช้งานอย่างไร.** จาก WebDoDee: https://www.webdodee.com/what-is-javascript/

**Framework คืออะไร โครงร่างหรือขอบเขตของระบบงาน.** (2561). จาก MindPHP: https://www.mindphp.com/developer/25-oop-php-framework/1957-framework-คืออะไร.html

**MVC คืออะไร ทำความเข้าใจรูปแบบการเขียนโปรแกรม. (2559).** จาก บริษัท โค๊ดบี จำกัด: h ttps://www.codebee.co.th/labs/mvc-คืออะไร-ทำความเข้าใจรู/

Wim Hoogenraad. (2562). **ฐานข้อมูล NoSQL คืออะไร.** จาก ITpedia: https://th.itpedia.nl/2019/06/03/wat-zijn-nosql-databases/

Chai Phonbopit. (2558). **MongoDB คืออะไร? + สอนวิธีใช้งานเบื้องต้น.** จาก Devahoy: https://www.devahoy.com/blog/2015/08/getting-started-with-mongodb

**Express.js เอ็กเพรส ดอทเจเอส คืออะไร Express.js คือ เป็น Web Application Framework ชื่อดังที่ได้รับความนิยมมาก สำหรับทำงานบน platform ของ Node.js.** จาก MindPHP: https://www.mindphp.com/คู่มือ/73-คืออะไร/3874-what-is-express-js.html

Mos Programmer. (2562). **มารู้จัก React กันเถอะ.** จาก Medium: https://mosprogramer.medium.com/มารู้จัก-react-กันเถอะ-eb04ab8b8ec1

**ทำความรู้จักกับ Node.js.** (2564). จาก MarcusCode: http://marcuscode.com/tutorials/nodejs/introducing-nodejs

ปริญญา อภัยภักดิ์ และณัฐพล เสาวพงษ์. **ระบบจัดการร้านอาหาร.** (ภาคนิพนธ์ปริญญาตรี คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยสยาม, 2562)

วิกรม ศรศรี และนำโชค ทองละมุล. **ระบบจัดการการสั่งอาหาร กรณีศึกษา ร้านอาหาร Zero.** (ปริญญาตรี คณะบริหารธุรกิจ มหาวิทยาลัยราชพฤกษ์, 2560)

ธีรพงศ์ ชูชื่น และจิรรัฐ ศรีโชค. **ระบบจัดการร้านอาหาร.** (ปริญญาตรี คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสำเด็จเจ้าพระยา, 2555)

ปรินทร อ่อนโก้ก, นิรัชชา บุญวิเทียน และ ชัญญานุช ศรีวรรณ์. **ระบบจัดการร้านกังฟูหม่าล่า เบียร์หิมะ.** (ปริญญาตรี คณะวิทยาการจัดการ มหาวิทยาลัยราชภัฏอุดรธานี, 2563)

วิรุฬห์ ศรีบริรักษ์. **เทคโนโลยีเภสัชสารสนเทศแสดงข้อมูลฉลากยาเอกสารกํากับยาแบบอัตโนมัติสําหรับบริหารจัดการคลังยาปฎิชีวนะเพื่อความปลอดภัย** (คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา, 2562)

# ประวัติย่อผู้พัฒนา

ชื่อ – สกุล : ธชิณ วิสุทธิมรรคกุล

รหัสนักศึกษา : 6101240160

หมายเลขโทรศัพท์ที่ติดต่อได้สะดวก : 083-622-2054

Email address : nickethachin@gmil.com

ส่วนโปรแกรมที่ได้พัฒนาในโครงงานนี้

* Database models
* Controllers
* REST API, Misc. API
* Backend routing
* Authentication & Middleware
* Dynamic navigating menu
* Register module
* Login module
* Dark mode theme
* Light & Dark mode switch
* Materials manager module
* Items manager module
* Itemsets manager module
* Expense category manager module
* Expenses manager module
* Orders manager module
* Dashboard module
* Frontend routing
* Redux state management

# ตัวอย่างเอกสาร

Graphical user interface, application, table, Excel

Description automatically generated

ตัวอย่างระบบบันทึกคำสั่งซื้อแบบเก่าของทางร้าน

Graphical user interface, application

Description automatically generated

ตัวอย่างระบบบันทึกรายจ่ายแบบเก่า