**บทที่ 1**

**บทนำ**

1. **ความเป็นมาและความสำคัญของโครงงาน**

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

ในปัจจุบัน platform การส่งอาหารออนไลน์ในประเทศไทยมีหลากหลายเป็นอย่างมาก เช่น Grab Food, Lineman Wongnai, ShopeeFood, Robinhood โดยแต่ละบริษัทก็มีระบบการให้บริการที่แตกต่างกันไป ส่งผลให้การทำบัญชี การคิดรายรับ รายจ่าย กำไร ต้นทุน หรือบัญชีการเงินต่างๆในร้านค้าเกิดความซับซ้อนมากกว่าการขายหน้าร้านปกติทั่วไป

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

1. **วัตถุประสงค์ของโครงงาน**

ระบบจัดการร้านอาหารเดลิเวอรี่สามารถ

1. จัดเก็บสูตรอาหารและคำนวณต้นทุนแต่ละเมนูได้โดยอ้างอิงจากราคาวัสดุล่าสุด
2. บันทึกรายจ่ายของร้านได้โดยมีแบ่งแยกประเภทชัดเจน
3. บันทึกคำสั่งซื้อได้โดยที่ง่าย และคำนวณค่าธรรมเนียมต่างๆอัตโนมัติ
4. สรุปค่าแรง กำไร และรายได้สุทธิ ของแต่ละวันได้
5. แสดงข้อมูลสถิติการสั่งอาหารจากคำสั่งซื้อต่างๆ
6. แสดงข้อมูลรายจ่ายของร้านตามช่วงเวลาต่างๆ
7. จัดการสต๊อคสินค้าได้ และมีการเตือนเมื่อสต๊อคใดต่ำกว่ากำหนด
8. **ขอบเขตของโครงงาน**
   1. **ด้านประชากร**
9. เจ้าของร้าน
10. พนักงานร้าน ‘สมายดอง’
    1. **ด้านเทคโนโลยีที่ใช้ในการพัฒนาโครงงาน**
       1. **Hardware**

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

- Smart Phone (Xiaomi 10T pro)

* + 1. **Software**

- Microsoft Windows 10 Pro build 19043

- MongoDB 5.0

- Express.js 4.0

- React 18.0.0

- NodeJS 16.14.2 LTS

- Visual Studio Code 1.66.2

- Notepad++ 8.3.3

* + 1. **Technology**

- Model View Controller

- REST API

- Cloud Database (MongoDB)

- Node.JS

- Framework (Express, React)

1. **ขอบเขตของโครงงาน**
2. ระบบตรวจสอบสิทธิในการเข้าใช้งานซอฟต์แวร์ (Authentication or Login)
3. การจัดการวัตถุดิบ
   1. เพิ่มรายการวัตถุดิบ (ชื่อ, ราคา, จำนวน, หน่วย)
   2. แก้ไข/ปรับปรุงรายการวัตถุดิบ
   3. ลบรายการวัตถุดิบ
   4. ค้นหาวัตถุดิบ
4. การจัดการบรรจุภัณฑ์
   1. เพิ่มรายการบรรจุภัณฑ์ (ชื่อ, ราคา, จำนวน, หน่วย)
   2. แก้ไข/ปรับปรุงรายการบรรจุภัณฑ์
   3. ลบรายการบรรจุภัณฑ์
   4. ค้นหาบรรจุภัณฑ์
5. การจัดการสูตรอาหาร
   1. เพิ่มสูตรอาหาร (ชื่ออาหาร, วัตถุดิบ/บรรจุภัณฑ์+จำนวนที่ใช้, ค่าแรง)
   2. แก้ไข/ปรับปรุงสูตรอาหาร
   3. ลบสูตรอาหาร
   4. ค้นหาสูตรอาหาร
6. การจัดการเมนูอาหาร
   1. เพิ่มเมนูอาหาร (ชื่อเมนู/ชุดเซ็ต, อาหารในเมนู)
   2. แก้ไขเมนูอาหาร
   3. ลบเมนูอาหาร
   4. ค้นหาเมนูอาหาร
7. การจัดการระบบส่งอาหาร
   1. เพิ่มระบบส่งอาหาร (ชื่อ, วิธีการคิดบัญชี)
      1. วิธีการคิดบัญชีจะประกอบด้วย ค่าธรรมเนียมตามออเดอร์ ค่าสมาชิกรายเดือน ฯลฯ
   2. แก้ไขระบบส่งอาหาร (ชื่อ, วิธีการคิดบัญชี, ยอดเงินคงเหลือ)
   3. ลบระบบส่งอาหาร
   4. ยกเลิกระบบส่งอาหารชั่วคราว
8. ระบบจัดการสต๊อคอาหารพร้อมส่ง
   1. กดเลือกเมนูที่ต้องการเพื่อเพิ่มสต๊อคอาหาร
      1. เมื่อกดเพิ่มสต๊อคเมนูใดๆ ระบบจะหักสต๊อควัตถุดิบ/บรรจุภัณฑ์ที่เมนูนั้นๆใช้
   2. สต๊อคสินค้าจะลดลงอัตโนมัติเมื่อมีการกดเพิ่มออเดอร์สินค้า
9. ระบบจัดการรายจ่าย
   1. ส่วนจัดการหมวดหมู่รายจ่าย
      1. เพิ่มหมวดหมู่ (ชื่อหมวด, เลือกหมวด Parent (ถ้ามี) )
      2. แก้ไขหมวดหมู่ (ชื่อหมวด, หมวด Parent)
      3. ลบหมวดหมู่
   2. การจัดการรายจ่าย
      1. เมื่อเลือกหมวดวัตถุดิบจะต้องเลือกวัตถุดิบจากรายชื่อที่เพิ่มไว้แล้ว จากนั้นใส่ราคาที่ซื้อ จำนวนที่ได้ และที่มา
      2. เมื่อเลือกหมวดบรรจุภัณฑ์จะต้องเลือกบรรจุภัณฑ์จากรายชื่อที่เพิ่มไว้แล้วจากนั้นใส่ราคาที่ซื้อและจำนวนที่ได้ และที่มา
      3. เมื่อเลือกหมวดอื่นๆจะต้องใส่ ชื่อ ราคา และที่มา(มีหรือไม่มีก็ได้)
      4. ด้านล่างของส่วน input จะเป็นรายการรายจ่ายล่าสุด n รายการ (ตามแต่ผู้ใช้เลือกจาก dropdown) เมื่อคลิกที่รายการล่าสุดเหล่านั้น ระบบจะคัดลอกข้อมูลทั้งหมดขึ้นไปใน input ด้านบน เพื่อเพิ่มรายการอย่างรวดเร็ว
10. ระบบบันทึกออเดอร์
    1. เมื่อมีออเดอร์เกิดขึ้น ผู้ใช้จะต้องเพิ่มรายการโดย
       1. เลือกช่องทางระบบส่งอาหาร
       2. เลือกเมนูในออเดอร์ที่ได้รับ
       3. ใส่จำนวนของแต่ละเมนู
    2. เมนูในคำสั่งซื้อที่ได้รับจะถูกเก็บเป็นสถิติการขายทันที
    3. ยอดรวมของแต่ละออเดอร์จะถูกคำนวณโดยอิงจากกฏการคำนวณของแต่ละระบบส่งอาหาร
    4. ยอดรวมของแต่ละออเดอร์จะถูกเพิ่มเข้าไปในกระเป๋าเงินของระบบส่งอาหารนั้นๆ
    5. แต่ละออเดอร์ที่ถูกเพิ่มจะมีปุ่มที่ท้ายรายการเพื่อยืนยันว่าออเดอร์นั้นๆได้รับเงินจากระบบส่งอาหารแล้ว
    6. ออเดอร์ต้องได้รับการยืนยันรับเงินก่อนถึงจะเพิ่มเป็นรายรับของวันที่ยืนยัน
    7. เมื่อออเดอร์ได้รับการยืนยันรับเงิน ระบบจะหักยอดออกจากกระเป๋าเงินระบบส่งอาหารนั้นๆ
11. การคำนวณต้นทุนและกำหนดราคาสินค้า
    1. แสดงผลของแต่ละเมนู ประกอบด้วย
       1. ระบบส่งอาหาร
       2. ชื่อเมนู
       3. ต้นทุนวัตถุดิบ
       4. ต้นทุนบรรจุภัณฑ์
       5. ต้นทุนแรงงาน
       6. ต้นทุนน้ำมัน
       7. ต้นทุนทั้งหมด (iii + iv + v +vi)
       8. ราคาขาย (สามารถแก้ไขได้ทันทีตรงนี้)
       9. ยอดเงินที่จะได้รับ (ราคาขาย - ค่าธรรมเนียม)
       10. กำไรสุทธิ (ยอดเงินที่จะได้รับ - ต้นทุนทั้งหมด)
    2. ผู้ใช้มีสิทธิ์เลือก
       1. ระบบส่งอาหาร หากไม่เลือกจะแสดงทั้งหมด
       2. เป้าหมายกำไรที่ต้องการ
       3. วิธีการคำนวณ เมื่อเลือกแล้วจะถูกบันทึกไว้เพื่อใช้แสดงผลครั้งต่อไปและการแสดงผลบนหน้า Dashboard (ค่าเริ่มต้นคือ ใช้ราคาล่าสุด)
          1. ใช้ราคาวัตถุดิบ/บรรจุภัณฑ์ล่าสุด
          2. ใช้ราคาวัตถุดิบ/บรรจุภัณฑ์เฉลี่ยจาก 1 อาทิตย์ที่ผ่านมา
          3. ใช้ราคาวัตถุดิบ/บรรจุภัณฑ์เฉลี่ยจาก 1 เดือนที่ผ่านมา
       4. เมนูที่มีกำไรสุทธิไม่ถึงตามเป้าหมายที่ตั้งไว้จะแสดงผลด้วยพื้นหลังที่แตกต่างจากรายการอื่น
12. หน้า Dashboard
    1. แสดงผลเงินคงเหลือในกระเป๋าของแต่ละระบบส่งอาหาร
    2. แสดงผลบัญชีรายวัน
       1. ค่าแรง
       2. กำไร
       3. ต้นทุน
       4. ยอดที่ควรได้รับ
    3. แสดงผลค่าน้ำมันคงเหลือ และปุ่มกด เมื่อกดที่ปุ่มจะเป็นการบอกระบบว่าจะใช้เงินตรงส่วนนี้ ระบบจะถามจำนวน (Default คือค่าน้ำมันคงเหลือทั้งหมด) เมื่อกดยืนยัน ระบบจะหักค่าน้ำมันคงเหลือออกและเพิ่มเป็นรายจ่ายค่าน้ำมันโดยอัตโนมัติ
13. หน้าแสดงสถิติการขายแบบ แสดงผลเป็น
    1. กราฟและตาราง
    2. ตามรายการอาหารหรือตามรายการเมนู
    3. รายวันหรือรายอาทิตย์หรือรายเดือนหรือรายปี
14. **กรอบแนวคิดการพัฒนาโครงงาน**
    1. **แบบจำลองกระบวนการ (Software Process Model)** เลือกใช้ Waterfall Model ซึ่งประกอบไปด้วย การวิเคราะห์ความต้องการ การออกแบบระบบ การพัฒนาระบบ การทดสอบระบบ การนำระบบไปใช้ และการปรับปรุงแก้ไข เนื่องจากต้องจัดทำเอกสารโครงการ ซึ่งจำเป็นต้องมีการวางแผนการพัฒนาล่วงหน้าอยู่แล้ว โมเดล Waterfall ที่ต้องการการวางแผนล่วงหน้าจึงจัดได้ว่าเหมาะสมกับรูปแบบการทำงานมาก

Diagram

Description automatically generated

* 1. **แผนภูมิแกนต์ (Gantt Chart)**

A picture containing chart

Description automatically generated

1. **ประโยชน์ที่ได้รับ**
   1. ร้านค้ามีระบบคำนวณต้นทุนที่เปลี่ยนแปลงตามราคาวัตถุดิบและวัสดุต่างๆ
   2. ร้านค้าสามารถตรวจสอบรายรับรายจ่ายได้อย่างชัดเจน
   3. ร้านค้ามีรายงานสรุปยอดค่าแรง กำไร และรายได้สุทธิรายวันเพื่อจัดการบัญชีรายวัน
   4. ร้านค้าสามารถตรวจสอบสถิติการขายต่างๆ เพื่อนำข้อมูลไปใช้ต่อไปได้
   5. ร้านค้าได้รับแจ้งเตือนเมื่อสต๊อคสินค้าต่ำกว่าที่กำหนด
2. **นิยามศัพท์เฉพาะ**
   1. ผู้ใช้ คือ ผู้จัดการหรือเจ้าของร้านสมายดอง
   2. สูตรอาหาร คือการเก็บข้อมูลอาหารในร้านว่าประกอบด้วยวัตถุดิบหรือบรรจุภัณฑ์ใดบ้าง เช่น สูตรอาหาร ‘ข้าวญี่ปุ่น’ ต้องมีส่วนประกอบคือ ข้าวญี่ปุ่น, น้ำเปล่า, งาดำ, ไฟฟ้า, ถ้วยข้าว, ฝาถ้วย
   3. เมนูอาหาร คือการเก็บข้อมูลเมนูต่างๆในร้าน ว่าเมนูนั้นๆประกอบด้วยอาหารใดบ้าง เช่น เมนู “กุ้งดองซีอิ๊ว” ประกอบด้วย กุ้งดอง, น้ำจิ้มซีฟู๊ด, พริก, กระเทียม
   4. ระบบส่งอาหาร คือ platform สำหรับการสั่งอาหารต่างๆเช่น Grab, Lineman, Foodpanda, ShopeeFood, Robinhood

**บทที่ 2**

**แนวคิดทฤษฎีและงานวิจัยที่เกี่ยวข้อง**

**(Literature Review หรือ ทบทวนวรรณกรรม)**

**2.1 แนวคิดทฤษฎี**

**2.1.1 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

**2.1.2 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 สามารถรองรับการใช้งานในอนาคตได้ดี

**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

**2.1.3 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

**2.1.4 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

**2.1.5 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 คือการกำหนดรูปแบบการจัดเก็บข้อมูล

2. View คือส่วนของการแสดงผล

3. Controller คือส่วนของการประมวลผล

**2.1.6 ฐานข้อมูลแบบ NoSQL**

**เนื้อหา :**

ฐานข้อมูล NoSQL ถูกสร้างขึ้นเป็นพิเศษสำหรับเฉพาะ แบบจำลองข้อมูล และมีตารางเวลาที่ยืดหยุ่นสำหรับการสร้างแอปพลิเคชันที่ทันสมัย ฐานข้อมูล NoSQL ได้รับการยอมรับอย่างกว้างขวางถึงความสะดวกในการพัฒนาฟังก์ชันและประสิทธิภาพสำหรับ ข้อมูลขนาดใหญ่. ใช้โมเดลข้อมูลที่หลากหลายรวมถึงเอกสารแผนภูมิค่าคีย์ในหน่วยความจำและการค้นหา

ฐานข้อมูล NoSQL ใช้โมเดลข้อมูลที่หลากหลายในการเข้าถึงและจัดการข้อมูล เช่น เอกสาร แผนภูมิ ค่าคีย์ ในหน่วยความจำ และการค้นหา นอกจากนี้ ฐานข้อมูลประเภทนี้ยังได้รับการปรับให้เหมาะสมเป็นพิเศษสำหรับแอปพลิเคชันที่ต้องการข้อมูลจำนวนมาก เวลาแฝงต่ำ และแบบจำลองข้อมูลที่ยืดหยุ่น เราบรรลุเป้าหมายนี้โดยเฉพาะอย่างยิ่งโดยการลดข้อจำกัดความสอดคล้องของข้อมูลบางอย่างที่ทราบในฐานข้อมูลอื่น

**แหล่งอ้างอิง :**

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

**สรุปความรู้ :**

ฐานข้อมูลแบบ NoSQL คือฐานข้อมูลที่ไม่มีการกำหนดความสัมพันธ์ระหว่างข้อมูล ทำให้มีความยืดหยุ่นสูง และสามารถขยายขนาดได้ง่ายกว่าฐานข้อมูลแบบ SQL

**2.1.7 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

**2.1.8 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

**2.1.9 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

**2.1.10 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

**2.2 งานวิจัยที่เกี่ยวข้อง**

**2.2.1 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)

**2.2.2 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)

**2.2.3 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)

**2.2.4 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 2.2.1 – 2.2.4 :**

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

**2.2.5 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

**บทที่ 3**

**วิธีดำเนินโครงงาน (Research Methodology)**

**3.1 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

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

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

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

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

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

**3.2 แผนภาพกระแสข้อมูล (Dataflow Diagram)**

**Diagram

Description automatically generated**

3.2.1 การยืนยันตัวตน : ผู้ใช้ส่งข้อมูลการเข้าสู่ระบบ -> ระบบดึงข้อมูลผู้ใช้จากฐานข้อมูล -> ระบบประมวลผล -> ระบบส่งข้อมูลการล็อกอินกลับไปที่ผู้ใช้

3.2.2 การจัดการข้อมูลสินค้า : ผู้ใช้ส่งข้อมูลสินค้าเข้าสู่ระบบ -> ระบบดึงข้อมูลผู้ใช้จากฐานข้อมูล -> ระบบบันทึกข้อมูลสินค้าและข้อมูลผู้ใช้ลงในฐานข้อมูลสินค้า

3.2.3 การจัดการสต๊อคสินค้า : ระบบดึงข้อมูลสต๊อคสินค้าให้ผู้ใช้ -> ผู้ใช้ส่งข้อมูลสต๊อคสินค้าเข้าสู่ระบบ -> ระบบดึงข้อมูลผู้ใช้จากฐานข้อมูล -> ระบบบันทึกข้อมูลสต๊อคและข้อมูลผู้ใช้ลงในฐานข้อมูลสต๊อคสินค้า

3.2.4 การจัดการหมวดหมู่รายจ่าย : ระบบดึงข้อมูลหมวดหมู่รายจ่ายจากฐานข้อมูลหมวดหมู่รายจ่ายให้ผู้ใช้ -> ผู้ใช้ส่งข้อมูลหมวดหมู่รายจ่ายเข้าสู่ระบบ -> ระบบดึงข้อมูลผู้ใช้จากฐานข้อมูล -> ระบบบันทึกข้อมูลหมวดหมู่รายจ่ายและข้อมูลผู้ใช้ลงในฐานข้อมูลหมวดหมู่รายจ่าย

3.2.5 การบันทึกรายจ่าย : ระบบดึงข้อมูล รายชื่อหมวดหมู่รายจ่ายจากฐานข้อมูลหมวดหมู่รายจ่าย, รายชื่อวัตถุดิบจากฐานข้อมูลวัตถุดิบ และรายชื่อบรรจุภัณฑ์จากฐานข้อมูลบรรจุภัณฑ์ให้ผู้ใช้ -> ผู้ใช้ส่งข้อมูลรายจ่ายให้ระบบ -> ระบบบันทึกข้อมูลรายจ่ายและข้อมูลผู้ใช้ลงในฐานข้อมูลรายจ่าย

3.2.6 การบันทึกคำสั่งซื้อ : ระบบดึงข้อมูล รายชื่อระบบส่งอาหารจากฐานข้อมูลระบบส่งอาหาร และรายชื่อเมนูจากฐานข้อมูลเมนูอาหารให้ผู้ใช้ -> ผู้ใช้ส่งข้อมูลคำสั่งซื้อให้ระบบ -> ระบบดึงข้อมูลราคาสินค้าจากฐานข้อมูลราคาสินค้า -> ระบบบันทึกข้อมูลคำสั่งซื้อและข้อมูลผู้ใช้ลงในฐานข้อมูลคำสั่งซื้อ -> ระบบปรับปรุงข้อมูลในฐานข้อมูลสต๊อคสินค้า -> ระบบปรับปรุงข้อมูลในฐานข้อมูลบัญชีเงิน

**3.3 โครงสร้างซอฟต์แวร์ (Software Structure)**

**Diagram

Description automatically generated**

1. เมื่อเข้ามาในระบบ ผู้ใช้จะต้องผ่านการตรวจสอบตัวตน
2. เมื่อผ่านการตรวจสอบตัวตนแล้วจะสามารถเข้าถึงส่วนต่างๆดังนี้
   1. หน้า Dashboard ที่รวบรวมข้อมูลสถิติต่างๆไว้
   2. หน้าจัดการวัตถุดิบ ภายในประกอบด้วย
      1. หน้ารายชื่อที่สามารถค้นหาและลบได้
      2. ส่วนสำหรับเพิ่มรายการ
      3. ส่วนสำหรับแก้ไขรายการ
   3. หน้าจัดการบรรจุภัณฑ์ ภายในประกอบด้วย
      1. หน้ารายชื่อที่สามารถค้นหาและลบได้
      2. ส่วนสำหรับเพิ่มรายการ
      3. ส่วนสำหรับแก้ไขรายการ
   4. หน้าจัดการสูตรอาหาร ภายในประกอบด้วย
      1. หน้ารายชื่อที่สามารถค้นหาและลบได้
      2. ส่วนสำหรับเพิ่มรายการ
      3. ส่วนสำหรับแก้ไขรายการ
   5. หน้าจัดการเมนูอาหาร ภายในประกอบด้วย
      1. หน้ารายชื่อที่สามารถค้นหาและลบได้
      2. ส่วนสำหรับเพิ่มรายการ
      3. ส่วนสำหรับแก้ไขรายการ
   6. หน้าจัดการระบบส่งอาหาร ภายในประกอบด้วย
      1. หน้ารายชื่อที่สามารถค้นหาและปิดการใช้งานระบบได้
      2. ส่วนสำหรับเพิ่มระบบ
      3. ส่วนสำหรับแก้ไขระบบ
   7. หน้าจัดการสต๊อคสินค้า ภายในประกอบด้วย
      1. หน้ารายชื่อที่สามารถลดสต๊อคสินค้าได้
      2. ส่วนสำหรับเพิ่มสต๊อคสินค้า
   8. หน้าคำนวณต้นทุนและกำหนดราคา ภายในประกอบไปด้วย
      1. ตารางแสดงต้นทุน
      2. ส่วนสำหรับกำหนดราคาสินค้า
   9. หน้าเพิ่มคำสั่งซื้อ

**3.4. ข้อกำหนดซอฟต์แวร์ (Input/Output Specifications)**

3.4.1 โมดูลการเข้าสู่ระบบ (Login / Sign in)

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

3.4.2 โมดูลการเพิ่มวัตถุดิบ

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. ชื่อวัตถุดิบ  2. หน่วย (กรัม/ชิ้น/ถุง ฯลฯ) | 1. บันทึกข้อมูลลงฐานข้อมูล  2. บันทึกผู้เพิ่มข้อมูล | 1. ข้อมูลเข้าสู่ระบบ แสดงข้อความ “เพิ่มสำเร็จ” |

3.4.3 โมดูลการแก้ไขวัตถุดิบ

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกวัตถุดิบ  2. ชื่อวัตถุดิบ  3. หน่วย (กรัม/ชิ้น/ถุง ฯลฯ) | 1. แก้ไขข้อมูลในฐานข้อมูล  2. เพิ่มข้อมูลผู้แก้ไขล่าสุด | 1. ข้อมูลได้รับการแก้ไข แสดงข้อความ “แก้ไขสำเร็จ” |

3.4.4 โมดูลการลบวัตถุดิบ

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกวัตถุดิบ | 1. แก้ไขสถานะข้อมูลเป็น ยกเลิก  2. เพิ่มข้อมูลผู้ยกเลิก | 1. ข้อมูลถูกลบ แสดงข้อความ “ลบสำเร็จ” |

3.4.5 โมดูลการเพิ่มบรรจุภัณฑ์

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. ชื่อบรรจุภัณฑ์  2. หน่วย (กรัม/ชิ้น/ถุง ฯลฯ) | 1. บันทึกข้อมูลลงฐานข้อมูล  2. บันทึกผู้เพิ่มข้อมูล | 1. ข้อมูลเข้าสู่ระบบ แสดงข้อความ “เพิ่มสำเร็จ” |

3.4.6 โมดูลการแก้ไขบรรจุภัณฑ์

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกบรรจุภัณฑ์  2. ชื่อบรรจุภัณฑ์  3. หน่วย (กรัม/ชิ้น/ถุง ฯลฯ) | 1. แก้ไขข้อมูลในฐานข้อมูล  2. เพิ่มข้อมูลผู้แก้ไขล่าสุด | 1. ข้อมูลได้รับการแก้ไข แสดงข้อความ “แก้ไขสำเร็จ” |

3.4.7 โมดูลการลบบรรจุภัณฑ์

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกบรรจุภัณฑ์ | 1. แก้ไขสถานะข้อมูลเป็น ยกเลิก  2. เพิ่มข้อมูลผู้ยกเลิก | 1. ข้อมูลถูกลบ แสดงข้อความ “ลบสำเร็จ” |

3.4.8 โมดูลการเพิ่มสูตรอาหาร

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. ชื่อสูตรอาหาร  2. ส่วนประกอบ  2.1 เลือกส่วนประกอบที่ใช้  2.1.1 เลือกจากรายชื่อส่วนประกอบหรือบรรจุภัณฑ์  2.2 ปริมาณที่ใช้  3. ค่าแรงงาน  4. ค่าน้ำมัน  5. เลือกว่าเป็นสูตรที่ต้องนับสต๊อคหรือไม่ | 1. บันทึกข้อมูลลงฐานข้อมูล  2. บันทึกผู้เพิ่มข้อมูล | 1. ข้อมูลเข้าสู่ระบบ แสดงข้อความ “เพิ่มสำเร็จ” |

3.4.9 โมดูลการแก้ไขสูตรอาหาร

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกสูตรอาหาร  2. ชื่อสูตรอาหาร  3. เพิ่ม/ลบส่วนประกอบ  3.1 ส่วนประกอบที่เลือก  3.1.1 เลือกจากรายชื่อส่วนประกอบหรือบรรจุภัณฑ์  3.2 ปริมาณที่ใช้  4. ค่าแรงงาน  5. ค่าน้ำมัน  6. เลือกว่าเป็นสูตรที่ต้องนับสต๊อคหรือไม่ | 1. แก้ไขข้อมูลในฐานข้อมูล  2. เพิ่มข้อมูลผู้แก้ไขล่าสุด | 1. ข้อมูลได้รับการแก้ไข แสดงข้อความ “แก้ไขสำเร็จ” |

3.4.10 โมดูลการลบสูตรอาหาร

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกสูตรอาหาร | 1. แก้ไขสถานะข้อมูลเป็น ยกเลิก  2. เพิ่มข้อมูลผู้ยกเลิก | 1. ข้อมูลถูกลบ แสดงข้อความ “ลบสำเร็จ” |

3.4.11 โมดูลการเพิ่มเมนูอาหาร

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. ชื่อเมนูอาหาร  2. เลือกส่วนประกอบจากรายชื่อสูตรอาหาร | 1. บันทึกข้อมูลลงฐานข้อมูล  2. บันทึกผู้เพิ่มข้อมูล | 1. ข้อมูลเข้าสู่ระบบ แสดงข้อความ “เพิ่มสำเร็จ” |

3.4.12 โมดูลการแก้ไขเมนูอาหาร

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกเมนูอาหาร  2. ชื่อเมนูอาหาร  3. เพิ่ม/ลบส่วนประกอบจากรายชื่อสูตรอาหาร | 1. แก้ไขข้อมูลในฐานข้อมูล  2. เพิ่มข้อมูลผู้แก้ไขล่าสุด | 1. ข้อมูลได้รับการแก้ไข แสดงข้อความ “แก้ไขสำเร็จ” |

3.4.13 โมดูลการลบเมนูอาหาร

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกเมนูอาหาร | 1. แก้ไขสถานะข้อมูลเป็น ยกเลิก  2. เพิ่มข้อมูลผู้ยกเลิก | 1. ข้อมูลถูกลบ แสดงข้อความ “ลบสำเร็จ” |

3.4.14 โมดูลการเพิ่มระบบส่งอาหาร

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. ชื่อระบบส่งอาหาร  2. ค่าธรรมเนียม  3. ภาษี  4. ค่าสมาชิกรายเดือน  5. ยอดเงินคงเหลือ | 1. บันทึกข้อมูลลงฐานข้อมูลระบบส่งอาหาร  2. สร้างข้อมูลใหม่ในฐานข้อมูลบัญชีเงิน  3. บันทึกผู้เพิ่มข้อมูล | 1. ข้อมูลเข้าสู่ระบบ แสดงข้อความ “เพิ่มสำเร็จ” |

3.4.15 โมดูลการแก้ไขระบบส่งอาหาร

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกระบบส่งอาหาร  2. ชื่อระบบส่งอาหาร  3. ค่าธรรมเนียม  4. ภาษี  5. ค่าสมาชิกรายเดือน  6. ยอดเงินคงเหลือ | 1. แก้ไขข้อมูลในฐานข้อมูลระบบส่งอาหาร  2. แก้ไขข้อมูลในฐานข้อมูลบัญชีเงิน  2. เพิ่มข้อมูลผู้แก้ไขล่าสุด | 1. ข้อมูลได้รับการแก้ไข แสดงข้อความ “แก้ไขสำเร็จ” |

3.4.16 โมดูลการลบระบบส่งอาหาร

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกระบบส่งอาหาร | 1. แก้ไขสถานะข้อมูลเป็น ยกเลิก  2. เพิ่มข้อมูลผู้ยกเลิก | 1. ข้อมูลถูกลบ แสดงข้อความ “ลบสำเร็จ” |

3.4.17 โมดูลการหยุดใช้งานระบบส่งอาหารชั่วคราว

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกระบบส่งอาหาร | 1. แก้ไขข้อมูลสถานะในฐานข้อมูลเป็น หยุดชั่วคราว | 1. ข้อมูลได้รับการแก้ไข แสดงข้อความ “{ชื่อระบบ} หยุดใช้งานชั่วคราว” |

3.4.18 โมดูลการเพิ่มสต๊อคอาหารพร้อมส่ง

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. กดปุ่ม + หน้าจำนวนสต๊อคคงเหลือ | 1. เพิ่มจำนวนสต๊อค 1 ในฐานข้อมูล | 1. ข้อมูลได้รับการแก้ไข |

3.4.19 โมดูลการลดสต๊อคอาหารพร้อมส่ง

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. กดปุ่ม - หน้าจำนวนสต๊อคคงเหลือ | 1. ลดจำนวนสต๊อคที่เก่าที่สุดลง 1 ในฐานข้อมูล | 1. ข้อมูลได้รับการแก้ไข |

3.4.20 โมดูลการเพิ่มหมวดหมู่รายจ่าย

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. ชื่อหมวดหมู่  2. เลือกว่าเป็นหมวดหมู่ย่อยของหมวดใด หรือไม่เลือก | 1. บันทึกข้อมูลลงฐานข้อมูล  2. บันทึกผู้เพิ่มข้อมูล | 1. ข้อมูลเข้าสู่ระบบ แสดงข้อความ “เพิ่มสำเร็จ” |

3.4.21 โมดูลการแก้ไขหมวดหมู่รายจ่าย

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกหมวดหมู่  2. ชื่อหมวดหมู่  3. เลือกว่าเป็นหมวดหมู่ย่อยของหมวดใด หรือไม่เลือก | 1. แก้ไขข้อมูลในฐานข้อมูล  2. เพิ่มข้อมูลผู้แก้ไขล่าสุด | 1. ข้อมูลได้รับการแก้ไข แสดงข้อความ “แก้ไขสำเร็จ” |

3.4.22 โมดูลการลบหมวดหมู่รายจ่าย

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกหมวดหมู่ | 1. แก้ไขสถานะข้อมูลเป็น ยกเลิก  2. เพิ่มข้อมูลผู้ยกเลิก | 1. ข้อมูลถูกลบ แสดงข้อความ “ลบสำเร็จ” |

3.4.23 โมดูลการเพิ่มรายจ่าย

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1.a เลือกวัตถุดิบที่เป็นรายจ่าย  1.b เลือกบรรจุภัณฑ์ที่เป็นรายจ่าย  1.c เลือกหมวดหมู่รายจ่าย  1.c.1 ชื่อรายจ่าย  2. จำนวนที่ได้  3. ราคาที่ซื้อ  4. สถานที่ซื้อหรือปล่อยว่าง | 1. บันทึกข้อมูลลงฐานข้อมูล  2. บันทึกผู้เพิ่มข้อมูล | 1. ข้อมูลเข้าสู่ระบบ แสดงข้อความ “เพิ่มสำเร็จ” |

3.4.24 โมดูลการแก้ไขรายจ่าย

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกรายจ่าย  2.1 เลือกวัตถุดิบที่เป็นรายจ่าย  หรือ เลือกบรรจุภัณฑ์ที่เป็นรายจ่าย หรือ เลือกหมวดหมู่รายจ่ายและชื่อรายจ่าย  3. จำนวนที่ได้  4. ราคาที่ซื้อ  5. สถานที่ซื้อหรือปล่อยว่าง | 1. แก้ไขข้อมูลในฐานข้อมูล  2. เพิ่มข้อมูลผู้แก้ไขล่าสุด | 1. ข้อมูลได้รับการแก้ไข แสดงข้อความ “แก้ไขสำเร็จ” |

3.4.25 โมดูลการลบรายจ่าย

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกรายจ่าย | 1. แก้ไขสถานะข้อมูลเป็น ยกเลิก  2. เพิ่มข้อมูลผู้ยกเลิก | 1. ข้อมูลถูกลบ แสดงข้อความ “ลบสำเร็จ” |

3.4.26 โมดูลการเพิ่มออเดอร์

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกระบบส่งอาหาร  2. เลือกเมนู  3. จำนวน | 1. บันทึกข้อมูลลงฐานข้อมูล  2. ลดสต๊อคตามจำนวน หากเมนูที่เลือกมีการนับสต๊อค  3. เพิ่มจำนวนเงินลงบัญชีระบบส่งอาหาร  4. บันทึกข้อมูลผู้เพิ่มออเดอร์ | 1. ข้อมูลบันทึกสำเร็จ แสดงข้อความ “เพิ่มออเดอร์สำเร็จ” |

3.4.27 โมดูลการแก้ไขออเดอร์

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกออเดอร์  2. เลือกระบบส่งอาหาร  3. เลือกเมนู  4. จำนวน | 1. แก้ไขข้อมูลลงฐานข้อมูล  2. แก้ไขสต๊อคตามจำนวน หากเมนูที่เลือกมีการนับสต๊อค  3. แก้ไขจำนวนเงินลงบัญชีระบบส่งอาหาร  4. บันทึกข้อมูลผู้เพิ่มออเดอร์ | 1. แก้ไขข้อมูลสำเร็จ แสดงข้อความ “แก้ไขออเดอร์สำเร็จ” |

3.4.28 โมดูลการลบออเดอร์

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกออเดอร์ | 1. เพิ่มจำนวนเงินลงบัญชีระบบส่งอาหาร  2. เพิ่มสต๊อคตามจำนวนในออเดอร์ หากเมนูในออเดอร์มีการนับสต๊อค  3. ลบข้อมูลออเดอร์ออกจากฐานข้อมูล | 1. ข้อมูลถูกลบ แสดงข้อความ “ลบสำเร็จ” |

3.4.29 โมดูลการแก้ไขราคา

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกระบบส่งอาหาร  2. เลือกเมนูอาหาร  3. ราคาอาหาร | 1. บันทึกข้อมูลลงฐานข้อมูล  2. บันทึกผู้แก้ไขข้อมูล | 1. ข้อมูลบันทึกสำเร็จ แสดงข้อความ “กำหนดราคาสำเร็จ” |

3.4.30 โมดูลการคำนวณราคา

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกระบบส่งอาหาร  2. กำไรที่ต้องการ  3. เลือกเปอร์เซ็นต์หรือบาท | 1. ดึงข้อมูลจากฐานข้อมูล เมนูอาหาร, สูตรอาหาร, บรรจุภัณฑ์ และ วัตถุดิบ  2. คำนวณต้นทุนของแต่ละเมนูโดยการนำวัตถุดิบและบรรจุภัณฑ์ที่ใช้คูณด้วยจำนวนและคูณด้วยราคา  3. คำนวณยอดเงินที่จะได้รับ โดยนำราคาขายลบด้วยค่าธรรมเนียมและภาษีค่าธรรมเนียม  3. คำนวณกำไรโดยนำยอดเงินที่จะได้รับลบด้วยต้นทุน  4. คำนวณกำไรเป้าหมายโดยนำต้นทุนบวกด้วยกำไรที่ต้องการ | 1. แสดงตารางข้อมูลของแต่ละเมนู โดยแสดงผล ต้นทุนวันถุดิบ ต้นทุนบรรจุภัณฑ์ ต้นทุนแรงงาน ต้นทุนน้ำมัน ต้นทุนทั้งหมด ราคาขาย ยอดเงินที่จะได้รับ กำไร  2. แสดงผลกำไรเป็นสีแดงหากกำไรต่ำกว่ากำไรเป้าหมาย |

3.4.31 โมดูลการกำหนดราคาขายผ่านหน้าคำนวณราคา

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. ราคา | 1. บันทึกข้อมูลลงฐานข้อมูล  2. บันทึกผู้แก้ไขข้อมูล | 1. ข้อมูลบันทึกสำเร็จ แสดงข้อความ “กำหนดราคาสำเร็จ” |

3.4.32 โมดูล Dashboard - ยอดเงินคงเหลือ

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
|  | 1. ดึงข้อมูลยอดเงินคงเหลือจากฐานข้อมูลบัญชีเงิน | 1. แสดงยอดเงินคงเหลือของแต่ละระบบส่งอาหารและบัญชีค่าน้ำมัน |

3.4.33 โมดูล Dashboard - คำนวณบัญชีรายวัน

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. วัน เดือน ปี | 1. ดึงข้อมูลออเดอร์วันที่เลือกจากฐานข้อมูลออเดอร์  2. คำนวณ ค่าแรง ต้นทุน กำไร ยอดเงินที่จะได้รับ จากข้อมูลที่ดึงมา | 1. แสดงตารางของแต่ละระบบส่งอาหาร แต่ละตารางจะแสดง  1.1 ห้วตารางเป็น ชื่อระบบส่งอาหาร – เปอร์เซนต์ค่าธรรมเนียม+ภาษีค่าธรรมเนียม  1.2 เมนูอาหาร ประกอบด้วย  1.2.1 ชื่อเมนูอาหาร  1.2.2 ราคาขายต่อชิ้น  1.2.3 ต้นทุนทั้งหมด  1.2.4 ค่าแรงทั้งหมด  1.2.5 กำไรทั้งหมด  1.2.6 ยอดเงินที่จะได้รับทั้งหมด |

3.4.34 โมดูล Dashboard - สถิติการขาย

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| 1. เลือกให้แสดงตามสูตรอาหารหรือเมนูอาหาร  2. เลือกการแสดงผลเป็น รายวัน, รายอาทิตย์, รายเดือน, หรือรายปี  3. เลือกช่วงเวลา | 1. ดึงข้อมูลออเดอร์ตามการแสดงผลและช่วงเวลาที่เลือก  2. นับจำนวนการขายตามสูตรอาหารหรือเมนูอาหาร | 1. แสดงผลตารางข้อมูลการขายตามที่เลือก  2. แสดงกราฟข้อมูลการขายตามที่เลือก |

**3.5 E-R Diagram**

*Diagram

Description automatically generated*

**3.6 ตารางฐานข้อมูล (Tables)**

3.6.1 ตารางผู้ใช้ (Users)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **ชื่อฟิลด์** | **ชนิดข้อมูล** | **คำอธิบาย** | **ข้อมูลตัวอย่าง** |
| 1 | \_id (PK) | ObjectId || Varchar | PK สร้างอัตโนมัติโดย MongoDB หรือกำหนดเอง | ObjectId(“62861e401904f7babfb13d”) |
| 2 | Username | Varchar | ชื่อผู้ใช้ | iamowner |
| 3 | Password | Varchar | รหัสผ่านที่เข้ารหัสแล้ว | $2y$10$/uO.IASKMRL.RQSOrgWZ5.Lx  u6elu1WET0.w9zWznlLwPfQgfwUEW |
| 4 | isActive | Boolean | สถานะข้อมูล  True : ยังใช้งานอยู่  False : ยกเลิกการใช้งานแล้ว | True |
| 5 | modifiedBy | Varchar | เก็บชื่อผู้ใช้ของผู้ที่แก้ไขข้อมูลล่าสุด | iamowner |

3.6.1 ตารางวัตถุดิบ (Ingredients)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **ชื่อฟิลด์** | **ชนิดข้อมูล** | **คำอธิบาย** | **ข้อมูลตัวอย่าง** |
| 1 | \_id (PK) | ObjectId || Varchar | PK สร้างอัตโนมัติโดย MongoDB หรือกำหนดเอง | ObjectId(“62861e401904f7babfb13d”) |
| 2 | Name | Varchar | ชื่อวัตถุดิบ | ซีอิ๊วขาว |
| 3 | Unit | Varchar | หน่วย เช่น กรัม ชิ้น | มิลลิลิตร |
| 4 | isActive | Boolean | สถานะข้อมูล  True : ยังใช้งานอยู่  False : ยกเลิกการใช้งานแล้ว | True |
| 5 | modifiedBy | Varchar | เก็บชื่อผู้ใช้ของผู้ที่แก้ไขข้อมูลล่าสุด | iamowner |

3.6.2 ตารางบรรจุภัณฑ์ (Packaging)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **ชื่อฟิลด์** | **ชนิดข้อมูล** | **คำอธิบาย** | **ข้อมูลตัวอย่าง** |
| 1 | \_id (PK) | ObjectId || Varchar | PK สร้างอัตโนมัติโดย MongoDB หรือกำหนดเอง | ObjectId(“62861e401904f7babfb13d”) |
| 2 | Name | Varchar | ชื่อบรรจุภัณฑ์ | ถุงซีลขนาด 8x6 |
| 3 | Unit | Varchar | หน่วย เช่น กรัม ชิ้น | ถุง |
| 4 | isActive | Boolean | สถานะข้อมูล  True : ยังใช้งานอยู่  False : ยกเลิกการใช้งานแล้ว | True |
| 5 | modifiedBy | Varchar | เก็บชื่อผู้ใช้ของผู้ที่แก้ไขข้อมูลล่าสุด | iamowner |

3.6.3 ตารางสูตรอาหาร (Recipes)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **ชื่อฟิลด์** | **ชนิดข้อมูล** | **คำอธิบาย** | **ข้อมูลตัวอย่าง** |
| 1 | \_id (PK) | ObjectId || Varchar | PK สร้างอัตโนมัติโดย MongoDB หรือกำหนดเอง | ObjectId(“62861e401904f7babfb13d”) |
| 2 | Name | Varchar | ชื่อสูตรอาหาร | น้ำจิ้มซีฟู๊ด |
| 3 | Components | Array of objects | ส่วนประกอบในสูตรอาหาร | [  {Type: ‘Ing’, ID: ‘1005’, Amount: 5},  {Type: ‘Pkg’, ID: ‘1017’, Amount: 1}, ] |
| 4 | isCount | Boolean | ต้องการให้นับสต๊อคอาหารนี้หรือไม่  True: นับ  False: ไม่นับ | True |
| 5 | isActive | Boolean | สถานะข้อมูล  True : ยังใช้งานอยู่  False : ยกเลิกการใช้งานแล้ว | True |
| 6 | modifiedBy | Varchar | เก็บชื่อผู้ใช้ของผู้ที่แก้ไขข้อมูลล่าสุด | iamowner |

3.6.3 ตารางเมนูอาหาร (Menus)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **ชื่อฟิลด์** | **ชนิดข้อมูล** | **คำอธิบาย** | **ข้อมูลตัวอย่าง** |
| 1 | \_id (PK) | ObjectId || Varchar | PK สร้างอัตโนมัติโดย MongoDB หรือกำหนดเอง | ObjectId(“62861e401904f7babfb13d”) |
| 2 | Name | Varchar | ชื่อเมนู | ชุดกุ้งดองสาหร่ายซูชิ |
| 3 | Components | Array of objects | ส่วนประกอบในสูตรอาหาร ประกอบด้วย ไอดีสูตรอาหาร และจำนวนที่ใช้ | [  { RecipeID: ‘1001’, Amount: 1}, { RecipeID: ‘1002’, Amount: 1}, { RecipeID: ‘1003’, Amount: 2}, ] |
| 4 | isActive | Boolean | สถานะข้อมูล  True : ยังใช้งานอยู่  False : ยกเลิกการใช้งานแล้ว | True |
| 5 | modifiedBy | Varchar | เก็บชื่อผู้ใช้ของผู้ที่แก้ไขข้อมูลล่าสุด | iamowner |

3.6.4 ตารางระบบส่งอาหาร (Deliveries)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **ชื่อฟิลด์** | **ชนิดข้อมูล** | **คำอธิบาย** | **ข้อมูลตัวอย่าง** |
| 1 | \_id (PK) | ObjectId || Varchar | PK สร้างอัตโนมัติโดย MongoDB หรือกำหนดเอง | ObjectId(“62861e401904f7babfb13d”) |
| 2 | Name | Varchar | ชื่อระบบ | Shopee Food |
| 3 | Fees | Double | ค่าธรรมเนียมเป็นเปอร์เซ็นต์ | 32.1 |
| 4 | FeesTax | Double | ภาษีค่าธรรมเนียมเป็นเปอร์เซ็นต์ | 7 |
| 5 | isActive | Boolean | สถานะข้อมูล  True : ยังใช้งานอยู่  False : ยกเลิกการใช้งานแล้ว | True |
| 6 | modifiedBy | Varchar | เก็บชื่อผู้ใช้ของผู้ที่แก้ไขข้อมูลล่าสุด | iamowner |

3.6.5 ตารางราคาอาหาร (Prices)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **ชื่อฟิลด์** | **ชนิดข้อมูล** | **คำอธิบาย** | **ข้อมูลตัวอย่าง** |
| 1 | \_id (PK) | ObjectId || Varchar | PK สร้างอัตโนมัติโดย MongoDB หรือกำหนดเอง | ObjectId(“62861e401904f7babfb13d”) |
| 2 | MenuID | Varchar | ไอดีของเมนูอาหาร | 1017 |
| 3 | DeliveryID | Varchar | ไอดีของระบบส่งอาหาร | 1076 |
| 4 | Amount | Double | ราคาอาหาร (บาท) | 189 |
| 5 | isActive | Boolean | สถานะข้อมูล  True : ยังใช้งานอยู่  False : ยกเลิกการใช้งานแล้ว | True |
| 6 | modifiedBy | Varchar | เก็บชื่อผู้ใช้ของผู้ที่แก้ไขข้อมูลล่าสุด | iamowner |

3.6.6 ตารางบัญชีเงิน (Balance)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **ชื่อฟิลด์** | **ชนิดข้อมูล** | **คำอธิบาย** | **ข้อมูลตัวอย่าง** |
| 1 | \_id (PK) | ObjectId || Varchar | PK สร้างอัตโนมัติโดย MongoDB หรือกำหนดเอง | ObjectId(“62861e401904f7babfb13d”) |
| 2 | Name | Varchar | ชื่อบัญชี | น้ำมัน |
| 3 | Amount | Varchar | จำนวนเงินคงเหลือ | 69 |
| 4 | isActive | Boolean | สถานะข้อมูล  True : ยังใช้งานอยู่  False : ยกเลิกการใช้งานแล้ว | True |
| 5 | modifiedBy | Varchar | เก็บชื่อผู้ใช้ของผู้ที่แก้ไขข้อมูลล่าสุด | iamowner |

3.6.7 ตารางหมวดหมู่รายจ่าย (ExpenseCategory)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **ชื่อฟิลด์** | **ชนิดข้อมูล** | **คำอธิบาย** | **ข้อมูลตัวอย่าง** |
| 1 | \_id (PK) | ObjectId || Varchar | PK สร้างอัตโนมัติโดย MongoDB หรือกำหนดเอง | ObjectId(“62861e401904f7babfb13d”) |
| 2 | ParentId | Varchar | ไอดีหมวดหมู่แม่ | ObjectId(“62861e401904f7babfb13d”) |
| 3 | Name | Varchar | ชื่อหมวดหมู่ | อุปกรณ์ |
| 4 | isActive | Boolean | สถานะข้อมูล  True : ยังใช้งานอยู่  False : ยกเลิกการใช้งานแล้ว | True |
| 5 | modifiedBy | Varchar | เก็บชื่อผู้ใช้ของผู้ที่แก้ไขข้อมูลล่าสุด | iamowner |

3.6.8 ตารางรายจ่าย (ExpenseRecords)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **ชื่อฟิลด์** | **ชนิดข้อมูล** | **คำอธิบาย** | **ข้อมูลตัวอย่าง** |
| 1 | \_id (PK) | ObjectId || Varchar | PK สร้างอัตโนมัติโดย MongoDB หรือกำหนดเอง | ObjectId(“62861e401904f7babfb13d”) |
| 2 | CategoryId | Varchar | ไอดีหมวดหมู่รายจ่าย | ObjectId(“62861e401904f7babfb13d”) |
| 3 | Name | Varchar | ชื่อรายการจ่าย | เครื่องคั้นน้ำมะนาว |
| 4 | Amount | Double | จำนวนที่ซื้อ | 1 |
| 5 | Price | Double | ราคาสินค้า | 49 |
| 6 | Supplier | Varchar | สถานที่ซื้อ | Mr.DIY |
| 7 | isActive | Boolean | สถานะข้อมูล  True : ยังใช้งานอยู่  False : ยกเลิกการใช้งานแล้ว | True |
| 8 | modifiedBy | Varchar | เก็บชื่อผู้ใช้ของผู้ที่แก้ไขข้อมูลล่าสุด | iamowner |

3.6.9 ตารางคำสั่งซื้อ (OrderRecords)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **ชื่อฟิลด์** | **ชนิดข้อมูล** | **คำอธิบาย** | **ข้อมูลตัวอย่าง** |
| 1 | \_id (PK) | ObjectId || Varchar | PK สร้างอัตโนมัติโดย MongoDB หรือกำหนดเอง | ObjectId(“62861e401904f7babfb13d”) |
| 2 | Components | Array of objects | ส่วนประกอบในสูตรอาหาร โดยใน object จะประกอบด้วย ไอดีเมนูอาหาร และ จำนวนที่สั่ง | [  { MenuID: ‘420’, Amount: ‘1’ },  { MenuID: ‘6969’, Amount: ‘3’},  { MenuID: ‘8008’, Amount: ‘2’},  ] |
| 3 | isActive | Boolean | สถานะข้อมูล  True : ยังใช้งานอยู่  False : ยกเลิกการใช้งานแล้ว | True |
| 4 | modifiedBy | Varchar | เก็บชื่อผู้ใช้ของผู้ที่แก้ไขข้อมูลล่าสุด | iamowner |

3.6.10 ตารางสต๊อคอาหาร (Stock)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **ชื่อฟิลด์** | **ชนิดข้อมูล** | **คำอธิบาย** | **ข้อมูลตัวอย่าง** |
| 1 | \_id (PK) | ObjectId || Varchar | PK สร้างอัตโนมัติโดย MongoDB หรือกำหนดเอง | ObjectId(“62861e401904f7babfb13d”) |
| 2 | RecipeId | Varchar | ไอดีสูตรอาหาร | ObjectId(“62861e401904f7babfb13d”) |
| 3 | LotId | Integer | รหัสประจำล๊อต | 1001 |
| 4 | ManuDate | Date | วันที่ผลิต | 16/05/2022 |
| 5 | ExpDate | Date | วันหมดอายุ | 18/05/2022 |
| 6 | Amount | Integer | จำนวนคงเหลือ | 8 |
| 7 | isActive | Boolean | สถานะข้อมูล  True : ยังใช้งานอยู่  False : ยกเลิกการใช้งานแล้ว | True |
| 8 | modifiedBy | Varchar | เก็บชื่อผู้ใช้ของผู้ที่แก้ไขข้อมูลล่าสุด | iamowner |

**3.7 ส่วนต่อประสาน (User Interface / Screen Layout)**

3.7.1 เข้าสู่ระบบ

A screenshot of a computer

Description automatically generated with medium confidence

3.7.2 DashboardGraphical user interface, application

Description automatically generated

3.7.3 เพิ่ม/แก้ไข/ลบ วัตถุดิบ

Graphical user interface

Description automatically generated

3.7.4 เพิ่ม/แก้ไข/ลบ บรรจุภัณฑ์

Graphical user interface

Description automatically generated

3.7.5 เมนูสำหรับ เพิ่ม/แก้ไข/ลบ สูตรอาหารและเมนูอาหาร

Graphical user interface

Description automatically generated

3.7.6 เพิ่ม/แก้ไข สูตรอาหาร

Graphical user interface, website

Description automatically generated

3.7.7 เมนูสำหรับ เพิ่ม/แก้ไข/ลบ เมนูอาหาร

Graphical user interface

Description automatically generated with medium confidence

3.7.8 เพิ่ม/แก้ไข เมนูอาหาร

Graphical user interface

Description automatically generated

3.7.9 เพิ่ม/แก้ไข/ลบ ระบบส่งอาหาร

Graphical user interface

Description automatically generated

3.7.10 เพิ่ม/แก้ไข/ลบ หมวดหมู่รายจ่าย

Graphical user interface

Description automatically generated

3.7.11 เพิ่ม/แก้ไข/ลบ รายจ่าย

Graphical user interface, website

Description automatically generated

3.7.12 เมนูสำหรับ เพิ่ม/แก้ไข/ลบ คำสั่งซื้อ

Graphical user interface

Description automatically generated

3.7.13 เพิ่มคำสั่งซื้อ

A screenshot of a computer

Description automatically generated with medium confidence

3.7.14 แก้ไขคำสั่งซื้อ

A screenshot of a computer

Description automatically generated with medium confidence

3.7.15 เมนูจัดการสต๊อค

Graphical user interface

Description automatically generated

3.7.16 คำนวณและตั้งราคา

Graphical user interface

Description automatically generated with medium confidence