



เอกสารโครงงานฉบับสมบูรณ์  
ระบบสแกนคิวอาร์โค้ดสั่งอาหารผ่านเว็บไซต์

จัดทำโดย

นายพิรัชย์	ชัยรัตน์	663380223-3
นายยศภัทร	สองเมือง	663380230-6
นางสาวธัญชนก	กุ่มกะ	663380506-1
นางสาวปิยะธิดา	วิจิตรจันทร์	663380508-7
นายเอกมงคล	พลเสนา	663380515-0

อาจารย์ที่ปรึกษา : รศ.ดร.ปัญญาพล หอระตะ

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา CP353002 หลักการออกแบบพัฒนาซอฟต์แวร์  
ภาคเรียนที่ 1 ปีการศึกษา 2568  
สาขาวิชาวิทยาการคอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น

## คำนำ

โครงการนี้เป็นส่วนหนึ่งของรายวิชา CP353002 หลักการออกแบบพัฒนาซอฟต์แวร์ เพื่อศึกษาการพัฒนาเว็บแอปพลิเคชันด้วย Rest API ใน Spring Boot ซึ่งเป็นเทคโนโลยีที่ได้รับความนิยมอย่างแพร่หลายในปัจจุบัน เนื่องจากมีความสามารถในการพัฒนาเว็บที่รวดเร็ว มีความยืดหยุ่น และรองรับการทำงานร่วมกับฐานข้อมูลได้อย่างมีประสิทธิภาพ

โดยโครงการนี้มุ่งเน้นการสร้างระบบสั่งอาหารผ่านคิวอาร์โค้ดสั่งอาหาร ซึ่งจะช่วยอำนวยความสะดวกให้กับผู้ใช้ โดยผู้ใช้สามารถเลือกเมนู และหลังจากเลือกเมนูแล้วยังสามารถสร้างใบเสร็จสำหรับการชำระเงินได้อีกด้วย คณะผู้จัดทำได้ออกแบบระบบให้มีความใช้งานง่ายทั้งฝั่งผู้ใช้และฝั่งแอดมิน และมีการจัดการข้อมูลได้อย่างเป็นระบบผ่านการใช้ Spring Boot ร่วมกับเทคโนโลยีอื่นๆ เช่น HTML, CSS, JavaScript และมีการใช้ฐานข้อมูลเป็น MySQL

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่าโครงการนี้จะเป็นประโยชน์ต่อผู้ที่สนใจในการพัฒนาเว็บแอปพลิเคชันด้วย Spring Boot และสามารถนำความรู้ที่ได้ไปประยุกต์ใช้ในการสร้างสรรค์ระบบอื่นๆได้อย่างมีประสิทธิภาพ

คณะผู้จัดทำ

## สารบัญ

เรื่อง	หน้า
คำนำ	ก
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 แนวคิด และทฤษฎีที่เกี่ยวข้อง	3
2.1 ทฤษฎีที่เกี่ยวข้อง	3
บทที่ 3 วิธีการดำเนินงาน	4
3.1 เครื่องมือและเทคโนโลยีที่ใช้ในการพัฒนา	4
3.2 ขั้นตอนการดำเนินงาน	4
3.3 แผนผังการทำงานของระบบ	5
บทที่ 4 ผลการดำเนินงาน	9
4.1 หน้าเว็บไซต์ในส่วนของลูกค้า	9
4.2 หน้าเว็บไซต์ในส่วนของพนักงาน	11
4.3 วิธี Deploy Spring Boot และ MySQL บน Railway	15
บทที่ 5 สรุปผล และข้อเสนอแนะ	18
5.1 สรุปผล	18
5.2 ข้อเสนอแนะ	18
เอกสารอ้างอิง	19

## สารบัญภาพ

ภาพที่	หน้า
ภาพที่ 1 Use Case Diagram	5
ภาพที่ 2 System Sequence Diagram	7
ภาพที่ 3 Domain Model	7
ภาพที่ 4 ER Diagram	8
ภาพที่ 5 หน้าสั่งอาหารสำหรับลูกค้า	9
ภาพที่ 6 หน้าสั่งอาหารสำหรับลูกค้า (แสดงรายการที่สั่งแล้ว)	10
ภาพที่ 7 หน้าเข้าสู่ระบบของพนักงาน	11
ภาพที่ 8 หน้าแสดงผลห้องครัว	11
ภาพที่ 9 หน้าคิดเงินและ Dashboard ยอดขาย	12
ภาพที่ 10 หน้าคิดเงินและ Dashboard ยอดขาย (ยืนยันการรับชำระเงิน)	13
ภาพที่ 11 หน้าคิดเงิน (บันทึกการรับชำระเงินเรียบร้อยแล้ว)	14
ภาพที่ 12 การปรับค่า Datasource	15
ภาพที่ 13 การ Deploy โดยเลือกผ่าน Repository บน Github	15
ภาพที่ 14 Environment Variables ที่ถูกสร้างขึ้น	16
ภาพที่ 15 การเพิ่มตัวแปรใหม่ในโปรเจกต์ เพื่อทำการ Map	16
ภาพที่ 16 ตัวอย่าง URL ที่ได้จาก Railway	17

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของโครงการ

ปัจจุบันเทคโนโลยีได้เข้ามามีบทบาทในชีวิตประจำวันเป็นอย่างมาก โดยเฉพาะบริการที่เน้นความสะดวก รวดเร็ว และลดการพึ่งพาแรงงานมนุษย์ ตัวอย่างที่เห็นได้ชัดคือ การสั่งอาหารผ่านระบบออนไลน์ ที่ช่วยให้ลูกค้าสามารถเลือกเมนูและทำการสั่งซื้อได้ด้วยตนเองผ่านสมาร์ทโฟน โดยไม่จำเป็นต้องรอพนักงานมารับออเดอร์แบบเดิม

จากแนวโน้มดังกล่าว คณะผู้จัดทำจึงเห็นความสำคัญของการนำเทคโนโลยีมาประยุกต์ใช้เพื่อยกระดับประสบการณ์การรับประทานอาหารในร้าน จึงได้พัฒนา ระบบสแกนคิวอาร์โค้ดสั่งอาหารผ่านเว็บไซต์ ซึ่งแต่ละโต๊ะจะได้รับคิวอาร์โค้ดเฉพาะของตนเอง เมื่อลูกค้าทำการสั่งอาหาร รายการจะถูกส่งไปยังหน้าจอของครัวโดยอัตโนมัติ เพื่อให้พนักงานจัดเตรียมอาหารได้อย่างรวดเร็ว พร้อมทั้งมีระบบสำหรับพนักงานหน้าร้านในการตรวจสอบสถานะออเดอร์และดำเนินการคิดเงินได้อย่างสะดวก

โครงการนี้พัฒนาด้วย Spring Boot Framework ซึ่งทำหน้าที่ทั้งฝั่ง Backend และ Frontend เชื่อมต่อกับฐานข้อมูลผ่าน RESTful API โดยออกแบบโครงสร้างข้อมูลให้มีความสัมพันธ์ในรูปแบบ One-to-One และ One-to-Many อย่างชัดเจน เพื่อให้ระบบมีความเสถียร ปลอดภัย และสามารถแลกเปลี่ยนข้อมูลระหว่างผู้ใช้งานได้อย่างมีประสิทธิภาพ อีกทั้งยังช่วยลดขั้นตอนการทำงานของพนักงานภายในร้าน เพิ่มความรวดเร็วในการให้บริการ และยกระดับประสบการณ์ของลูกค้า

#### 1.2 วัตถุประสงค์ของโครงการ

1. เพื่อพัฒนาและออกแบบเว็บไซต์สำหรับการสั่งอาหารออนไลน์ภายในร้านอาหาร โดยลูกค้าสามารถสแกนคิวอาร์โค้ดที่โต๊ะเพื่อเข้าหน้าเมนูและทำการสั่งอาหารได้ด้วยตนเอง
2. เพื่อศึกษาการพัฒนาเว็บแอปพลิเคชันโดยใช้ Spring Boot Framework ร่วมกับ RESTful API สำหรับเชื่อมต่อและแลกเปลี่ยนข้อมูลระหว่างฝั่งผู้ใช้งานกับฐานข้อมูล
3. เพื่อออกแบบโครงสร้างฐานข้อมูลที่มีความสัมพันธ์แบบ One-to-One และ One-to-Many อย่างเหมาะสม เพื่อให้ระบบสามารถจัดการข้อมูลการสั่งอาหารได้อย่างถูกต้องและมีประสิทธิภาพ

### 1.3 ขอบเขตของโครงการ

1. ระบบจะพัฒนาในรูปแบบเว็บไซต์สั่งอาหารออนไลน์ที่สามารถเข้าถึงได้ผ่านอุปกรณ์สมาร์ตโฟนหรือคอมพิวเตอร์
2. ลูกค้าสามารถ สแกนคิวอาร์โค้ดที่โต๊ะอาหาร เพื่อเข้าสู่หน้าร้านและทำการสั่งเมนูได้ด้วยตนเอง
3. ระบบจะมีส่วนติดต่อผู้ใช้ (Frontend) สำหรับลูกค้าในการดูเมนูอาหาร เพิ่มรายการสั่งซื้อ และยืนยันการสั่งอาหาร
4. ฝ่ายร้านอาหารจะมีระบบจัดการ (Backend) ตรวจสอบรายการสั่งซื้อของลูกค้า กดยืนยันรายการสั่งซื้อที่เสร็จแล้ว ดูสรุปยอดบิลที่ลูกค้าต้องชำระ พิมพ์ใบเสร็จและดู Dashboard ยอดขาย
5. ขอบเขตของโครงการจะครอบคลุมถึงการพัฒนาระบบหลักของการสั่งอาหาร การดูรายการสั่งซื้อที่เข้ามา การสรุปยอดบิลและ Dashboard สรุปยอดขาย
6. โครงการนี้เป็นการพัฒนาเพื่อการศึกษาและทดลองใช้งานในสภาพแวดล้อมจำลองไม่ได้เชื่อมต่อกับระบบร้านอาหารจริง

### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. พัฒนาเว็บไซต์สั่งอาหารออนไลน์โดยลูกค้าสามารถสแกนคิวอาร์โค้ดเพื่อสั่งอาหารบนเว็บไซต์ได้
2. ลดความสับสนและประหยัดเวลาในการรับออเดอร์อาหารของลูกค้า
3. อำนวยความสะดวกให้แก่พนักงานและลูกค้าในร้านอาหาร

## บทที่ 2

### แนวคิด และทฤษฎีที่เกี่ยวข้อง

#### 2.1 ทฤษฎีที่เกี่ยวข้อง

##### 2.1.1 Rest API

REST API (Representational State Transfer Application Programming Interface) คือ รูปแบบของ API ที่ออกแบบตามหลักการของสถาปัตยกรรม REST เพื่อให้ระบบต่าง ๆ สามารถสื่อสาร และแลกเปลี่ยนข้อมูลกันผ่านเครือข่าย เช่น อินเทอร์เน็ต ได้อย่างมีประสิทธิภาพ โดย

REST (Representational State Transfer) เป็นชุดข้อมูลของข้อจำกัดทางสถาปัตยกรรม ไม่ใช่ โปรโตคอลหรือมาตรฐาน ใช้กำหนดวิธีการออกแบบ API ให้มีความเรียบง่ายและมีโครงสร้างชัดเจน โดยทั่วไป REST ใช้ HTTP เป็นช่องทางหลักในการสื่อสาร และมักส่งข้อมูลในรูปแบบ JSON

API (Application Programming Interface) คือชุดของกฎและวิธีการที่กำหนดขึ้น เพื่อให้โปรแกรมหนึ่งสามารถสื่อสารหรือเรียกใช้งานฟังก์ชันของอีกโปรแกรมหนึ่งได้ โดยไม่จำเป็นต้องรู้ รายละเอียดการทำงานภายใน

##### 2.1.2 CRUD

CRUD เป็นคำย่อที่มาจาก 4 คำ คือ CREATE, READ, UPDATE และ DELETE ซึ่งคำเหล่านี้ อธิบายถึงการดำเนินการที่จำเป็น 4 ประการในกระบวนการพื้นฐานที่ใช้งานกับฐานข้อมูล โดยแต่ละคำมีความหมาย ดังนี้

1. CREATE การสร้าง เป็นการเพิ่มข้อมูลใหม่เข้าสู่ฐานข้อมูล
2. READ การอ่าน การดึงข้อมูลจากฐานข้อมูลมาแสดงผล
3. UPDATE การแก้ไขข้อมูลที่มีอยู่ในฐานข้อมูล
4. DELETE การลบข้อมูลออกจากฐานข้อมูล

## บทที่ 3

### วิธีการดำเนินงาน

การดำเนินการวิจัยในโครงการ “การพัฒนาเว็บไซต์สั่งอาหารด้วย Spring Boot” มีขั้นตอนและกระบวนการที่ชัดเจน เพื่อให้ได้เว็บไซต์ที่มีประสิทธิภาพ ตอบสนองต่อความต้องการของผู้ใช้ และสามารถใช้งานได้จริง โดยมีรายละเอียดดังนี้

#### 3.1 เครื่องมือและเทคโนโลยีที่ใช้ในการพัฒนา

1. ภาษาโปรแกรม: Java , HTML , CSS , JavaScript
2. Framework: Spring Boot
3. เครื่องมือพัฒนา: VS Code
4. ฐานข้อมูล: MySQL

#### 3.2 ขั้นตอนการดำเนินงาน

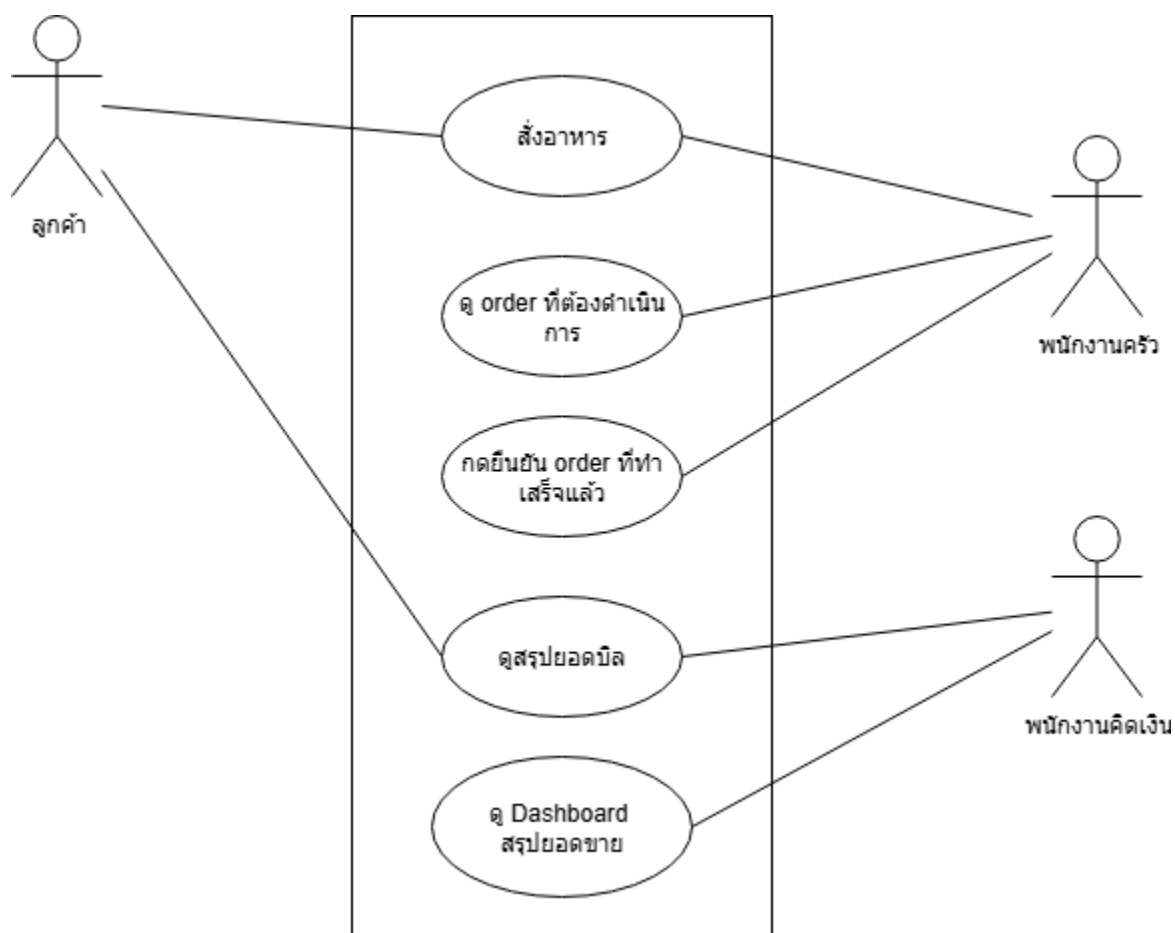
1. ศึกษาความต้องการของระบบ (Requirement Analysis)
  - วิเคราะห์ฟังก์ชันหลักที่จำเป็น เช่น การสั่งอาหารในฝั่งผู้ใช้ การรับออเดอร์ในฝั่งครัว
2. ออกแบบระบบ (System Design)
  - ออกแบบฐานข้อมูล (Database Schema)
  - ออกแบบโครงสร้างเว็บแอปพลิเคชัน (Architecture)
  - ออกแบบหน้าตาเว็บไซต์ (UI/UX Design)
3. พัฒนาระบบ (System Development)
  - ใช้ Spring Boot ในการพัฒนาในฝั่งเซิร์ฟเวอร์
  - ใช้ HTML , CSS , JavaScript ในการพัฒนาหน้าเพจสำหรับฝั่งใช้งาน
  - เชื่อมต่อกับฐานข้อมูล MySQL
4. ทดสอบระบบ (System Testing)
  - ทดสอบการทำงานของแต่ละฟังก์ชัน
  - ทดสอบการทำงานแบบรวมทั้งระบบ
  - แก้ไขข้อผิดพลาดที่พบ
5. ประเมินผลและปรับปรุง (Evaluation and Improvement)



- ประเมินการใช้งานจากผู้ทดสอบ
- ปรับปรุงส่วนที่ยังไม่สมบูรณ์
- สรุปผลการดำเนินงาน

### 3.3 แผนผังการทำงานของระบบ

#### 1. Use Case Diagram



ภาพที่ 1 Use Case Diagram

## รายละเอียด Use case

Use case UC01: Normal Order Process

Primary Actor: ลูกค้า

Stakeholder and Interest:

- ลูกค้า: ต้องการระบบสั่งอาหารที่สะดวก รวดเร็ว
- พนักงานครัว: ต้องการระบบที่สามารถดูคำสั่งซื้อที่เข้ามาได้รวดเร็ว และใช้งานง่าย
- พนักงาน: ต้องการระบบที่สามารถสรุปยอดบิล และมีหน้า Dashboard ที่สรุปข้อมูลให้เข้าใจง่าย

Preconditions:

- ลูกค้าต้องมีโต๊ะนั่งก่อนจึงจะทำการสั่งอาหารได้
- พนักงานกับพนักงานครัวต้องล็อกอินเข้าระบบไว้แล้ว

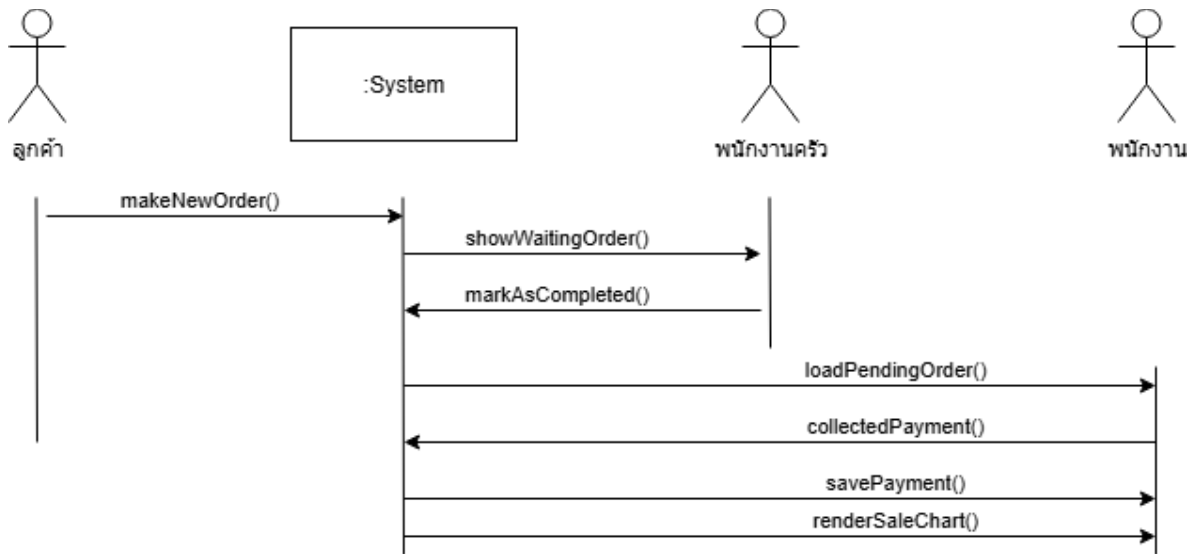
Postconditions:

- ระบบมีการบันทึกว่าลูกค้าสั่งอะไรไปบ้าง
- ระบบมีการบันทึกว่าแต่ละคำสั่งซื้อที่เข้าไปสู่ห้องครัวมีสถานะเป็นอย่างไร
- ระบบมีการบันทึกว่าคำสั่งซื้อนั้นชำระเงินเรียบร้อยแล้วหลังจากลูกค้าชำระเงินที่เคาน์เตอร์

Main Success Scenario:

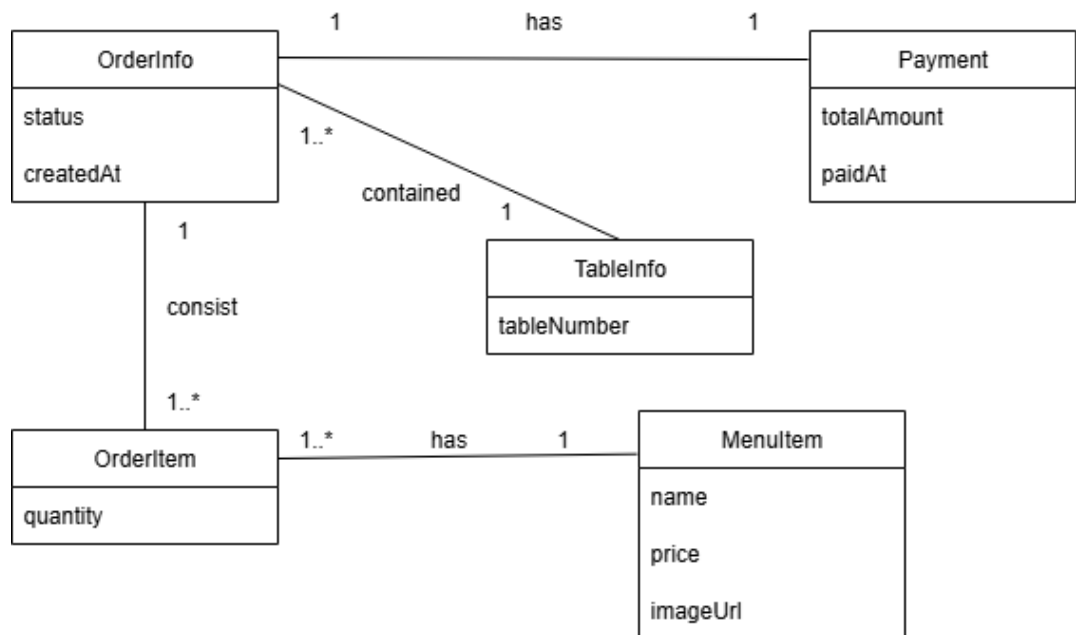
1. ลูกค้าสั่งอาหารแล้วกดส่งคำสั่งซื้อ
2. ระบบส่งคำสั่งซื้อไปที่พนักงานครัว
3. พนักงานครัวดำเนินการตามคำสั่งซื้อที่เข้ามาและกดยืนยันคำสั่งซื้อที่ทำเสร็จแล้ว
4. ลูกค้าได้รับอาหารตามที่สั่ง
5. ลูกค้าไปชำระเงินที่เคาน์เตอร์
6. พนักงานดูคำสั่งซื้อที่รอรับชำระ และรับชำระเงินจากลูกค้า
7. พนักงานกดยืนยันรับชำระเงินที่ระบบ
8. ระบบบันทึกการชำระเงินพร้อมทั้งแสดง Dashboard

## 2. System Sequence Diagram



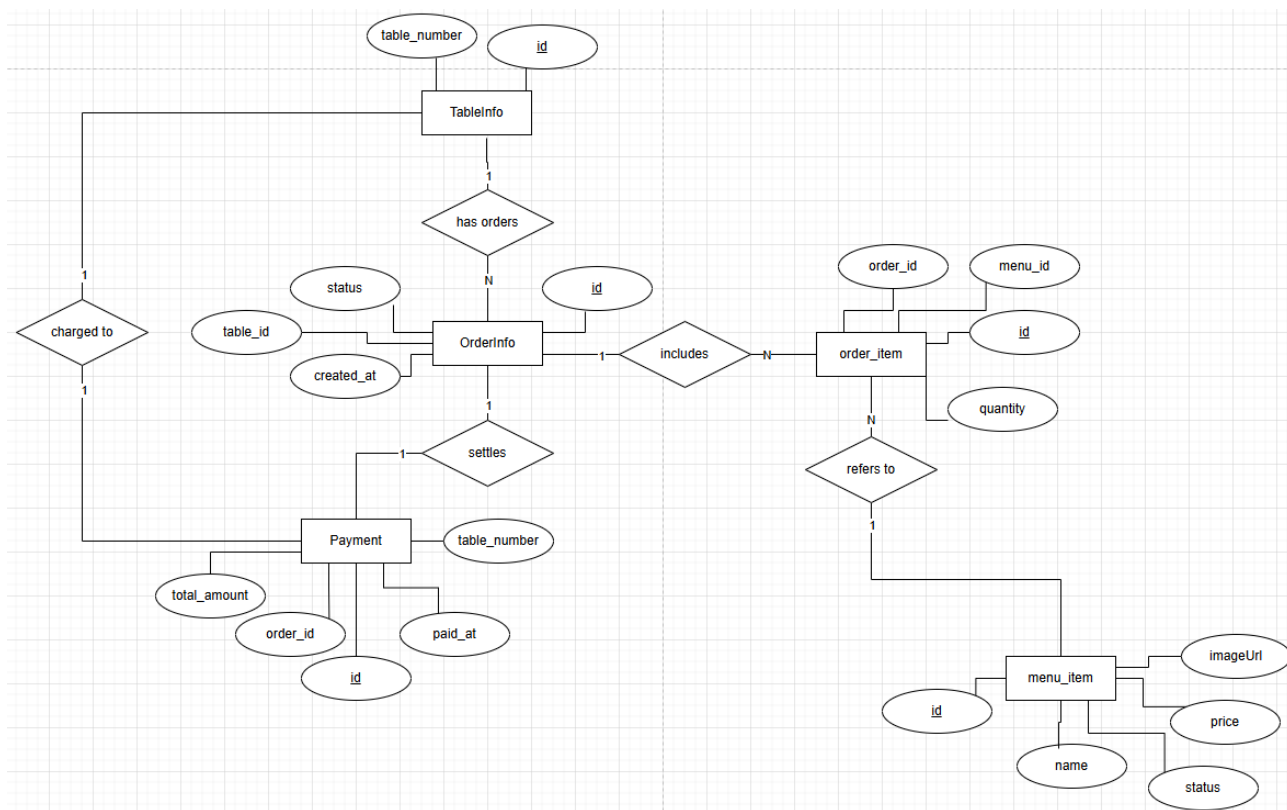
ภาพที่ 2 System Sequence Diagram

## 3. Domain Model



ภาพที่ 3 Domain Model

## 4. ER Diagram



ภาพที่ 4 ER Diagram

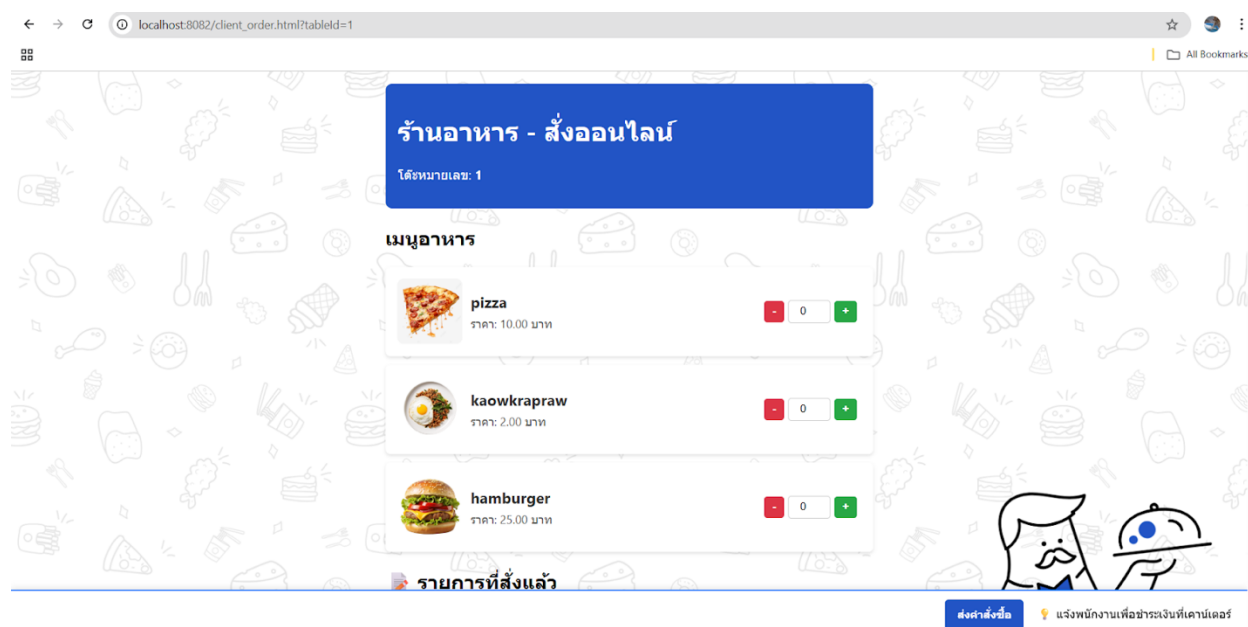
## บทที่ 4

### ผลการดำเนินงาน

โครงการ “ระบบสแกนคิวอาร์โค้ดสั่งอาหารผ่านเว็บไซต์” ซึ่งทำการพัฒนาโดยใช้ Spring Boot Framework ในการทำหน้าที่ทั้งฝั่ง Backend และ Frontend เชื่อมต่อกับฐานข้อมูลผ่าน RESTful API ได้มีผลการวิจัยในด้านการออกแบบและพัฒนาเว็บไซต์ ดังนี้

#### 4.1 หน้าเว็บไซต์ในส่วนของลูกค้า

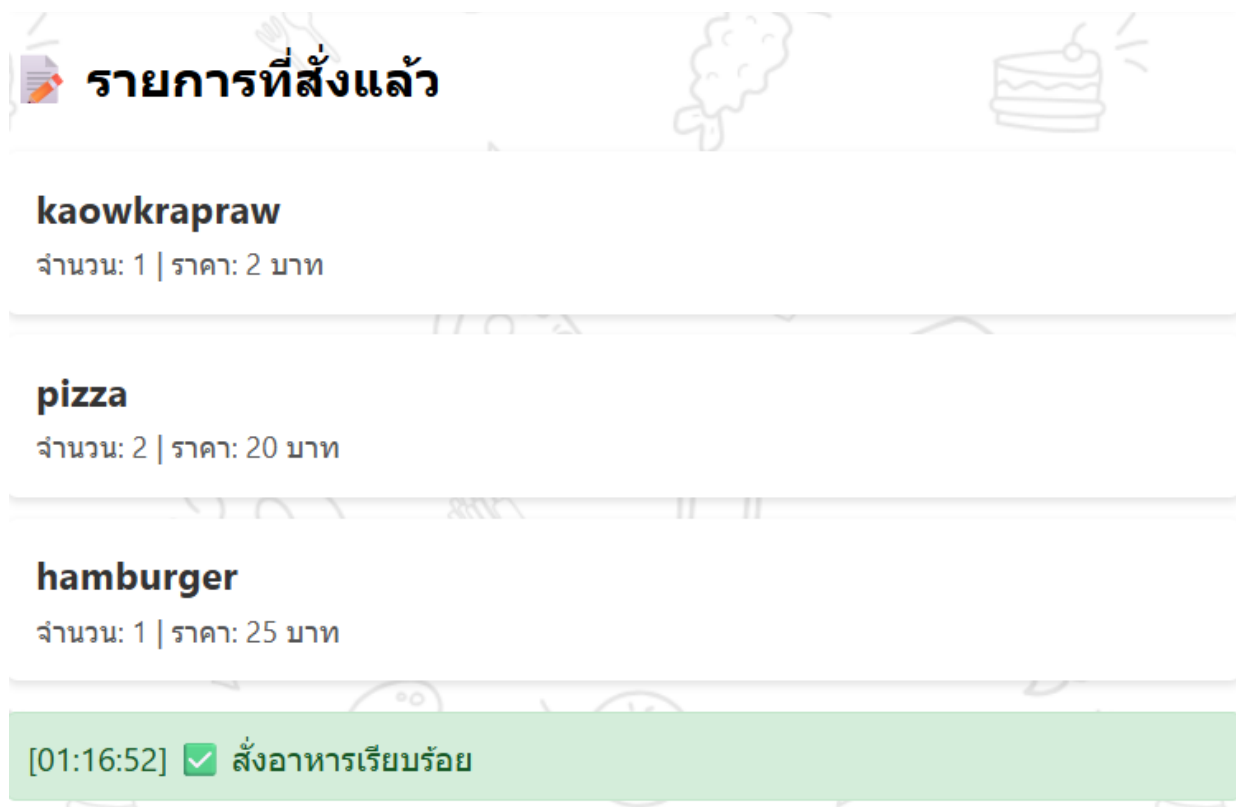
##### 4.1.1 หน้าสั่งอาหารสำหรับลูกค้า



ภาพที่ 5 หน้าสั่งอาหารสำหรับลูกค้า

เมื่อลูกค้าทำการสแกนคิวอาร์โค้ดบนโต๊ะอาหาร จะพบกับหน้าสั่งอาหารสำหรับลูกค้า ซึ่งจะมีการแสดงเมนูอาหารต่างๆ ราคา และลูกค้าสามารถปรับเพิ่มหรือลดจำนวนเมนูอาหารที่ต้องการสั่งได้ จากนั้นลูกค้าสามารถกดปุ่มส่งคำสั่งซื้อ เพื่อส่งรายการอาหารที่สั่งไปยังครัวต่อไป

#### 4.1.2 หน้าสั่งอาหารสำหรับลูกค้าหลังสั่งอาหารเสร็จแล้ว

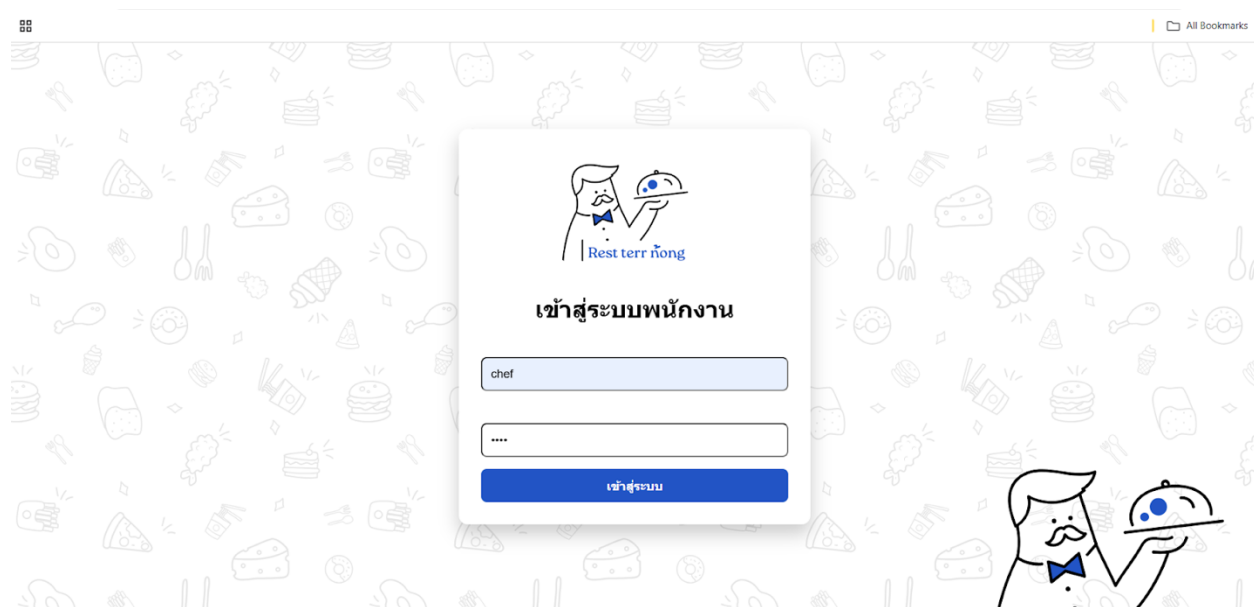


ภาพที่ 6 หน้าสั่งอาหารสำหรับลูกค้า (แสดงรายการที่สั่งแล้ว)

หลังลูกค้ากดปุ่มส่งคำสั่งซื้อจะมีการแจ้งเตือนว่าสั่งอาหารเรียบร้อยแล้วพร้อมทั้งแสดงรายการที่สั่งแล้ว โดยมีการแสดงชื่ออาหาร จำนวนและราคา

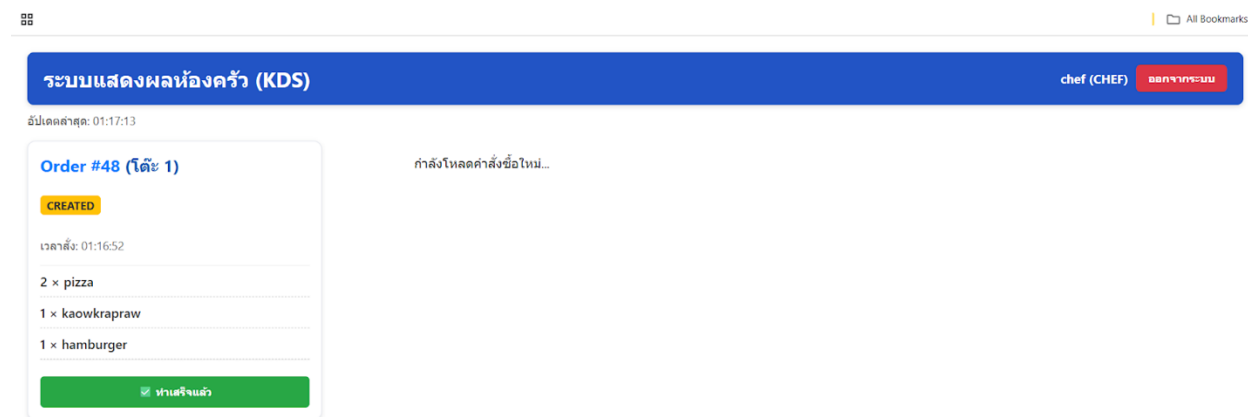
## 4.2 หน้าเว็บไซต์ในส่วนของพนักงาน

### 4.2.1 หน้าเข้าสู่ระบบของพนักงาน



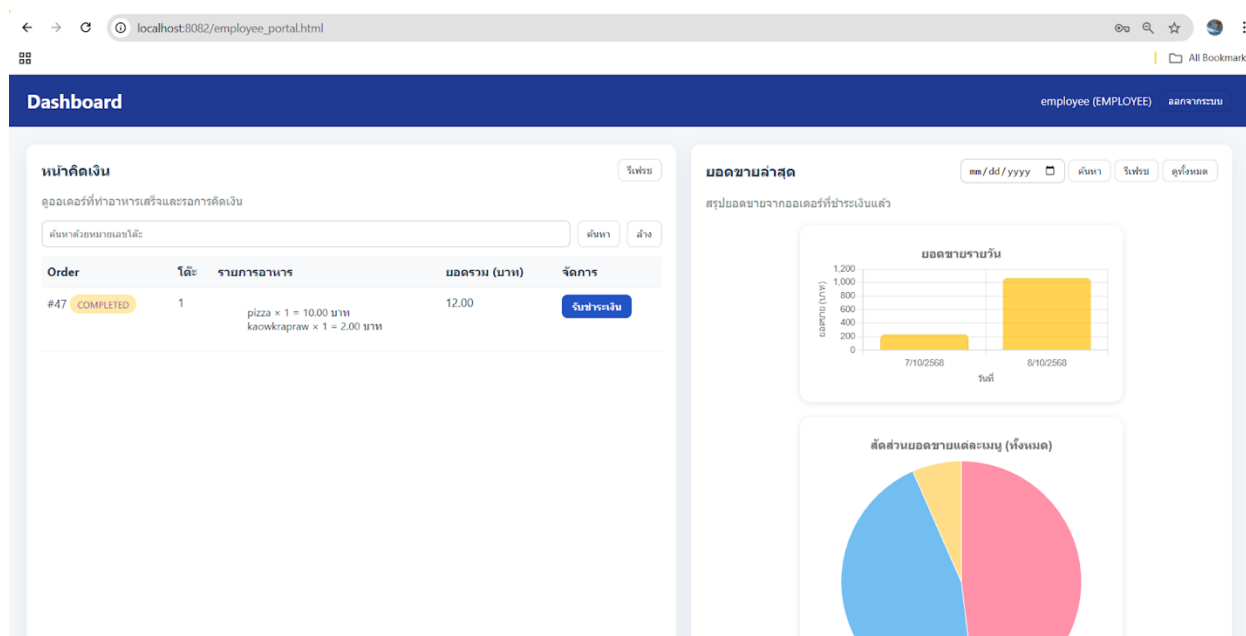
ภาพที่ 7 หน้าเข้าสู่ระบบของพนักงาน

ก่อนพนักงานเข้าใช้งานต้องมีการเข้าสู่ระบบก่อน โดยทำการกรอกข้อมูลผู้ใช้และรหัสผ่านให้ถูกต้อง จากนั้นกดเข้าสู่ระบบ



ภาพที่ 8 หน้าแสดงผลห้องครัว

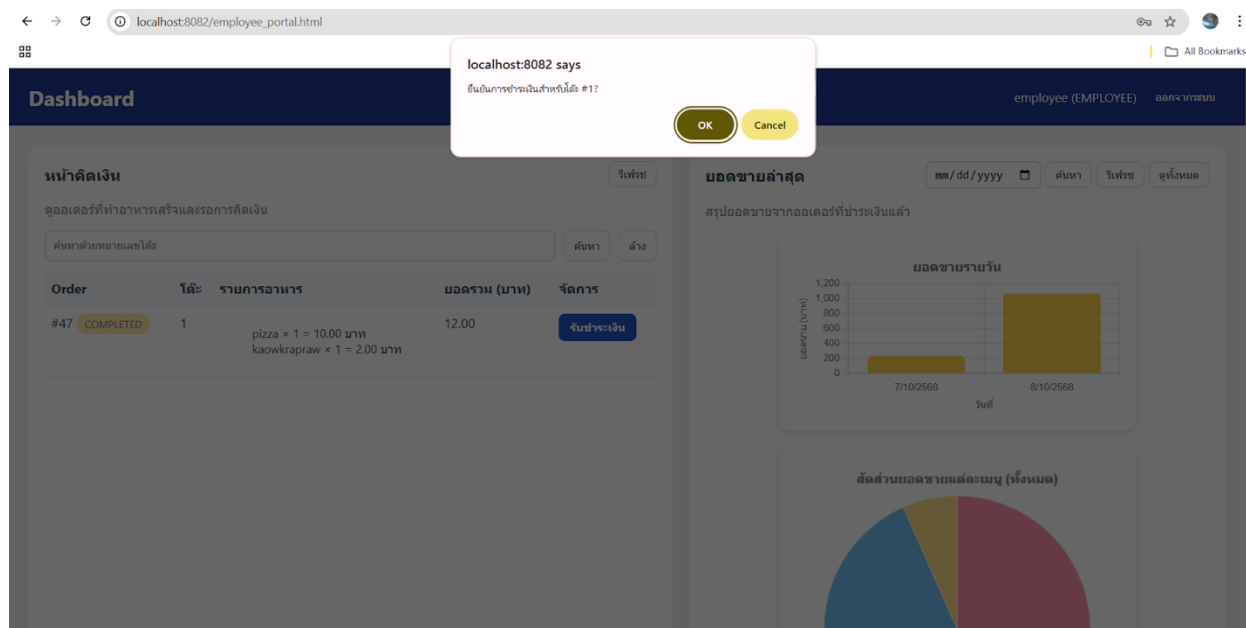
เมื่อพนักงานครัวเข้าสู่ระบบจะพบกับหน้าแสดงผลห้องครัว ซึ่งจะทำการแสดงคำสั่งซื้อที่เข้ามา โดยจะมีรายละเอียดคำสั่งซื้อประกอบไปด้วย หมายเลขคำสั่งซื้อ หมายเลขโต๊ะที่ทำการสั่ง สถานะของคำสั่งซื้อ เวลาที่สั่ง รายการอาหาร โดยพนักงานครัวสามารถกดปุ่มทำเสร็จแล้ว เพื่อยืนยันคำสั่งซื้อที่ทำเสร็จแล้วได้



ภาพที่ 9 หน้าคิดเงินและ Dashboard ยอดขาย

เมื่อพนักงานเข้าสู่ระบบจะพบกับหน้าคิดเงินและ Dashboard ยอดขาย ซึ่งจะทำการแบ่งเป็น 2 ส่วนได้แก่ ส่วนของหน้าคิดเงินและส่วนของหน้า Dashboard ยอดขาย โดยหน้าคิดเงินจะทำการแสดงคำสั่งซื้อที่รอการรับชำระเงิน จะมีการแสดงรายละเอียดหมายเลขคำสั่งซื้อ สถานะคำสั่งซื้อ โต๊ะที่ทำการสั่ง รายการอาหารที่สั่งซื้อ ยอดรวมค่าอาหาร พนักงานสามารถกดปุ่มรับชำระเงินเพื่อรับชำระเงินค่าอาหารของลูกค้าได้ และในส่วน of หน้า Dashboard ยอดขายจะทำการแสดงข้อมูลสรุปยอดขายในรูปแบบของกราฟแท่งและสัดส่วนยอดขายแต่ละเมนูในรูปแบบของกราฟวงกลม





ภาพที่ 10 หน้าคิดเงินและ Dashboard ยอดขาย (ยืนยันการรับชำระเงิน)

เมื่อพนักงานกดปุ่มรับชำระเงินจะมีการหน้าต่างแจ้งเตือนอีกครั้งว่ายืนยันการรับชำระเงินสำหรับคำสั่งซื้อนั้นหรือไม่

## หน้าคิดเงิน

รีเฟรช

ดูออเดอร์ที่ทำอาหารเสร็จและรอการคิดเงิน

ค้นหาด้วยหมายเลขโต๊ะ

ค้นหา

ล้าง

Order	โต๊ะ	รายการอาหาร	ยอดรวม (บาท)	จัดการ
-------	------	-------------	--------------	--------

ไม่มีออเดอร์ที่รอคิดเงิน

✕ ปิดใบเสร็จ

## ใบเสร็จรวมโต๊ะ #1

ออเดอร์ที่รวม: 47

- kaowkrapraw × 1 = 2.00 บาท
- pizza × 1 = 10.00 บาท

ยอดรวม: 12.00 บาท

เวลาชำระเงิน: 20/10/2568 01:15:58

บันทึกการชำระเงินเรียบร้อยแล้ว

พิมพ์ใบเสร็จลูกค้า

ภาพที่ 11 หน้าคิดเงิน (บันทึกการรับชำระเงินเรียบร้อยแล้ว)

เมื่อพนักงานกดปุ่ม OK ที่หน้าต่างแจ้งเตือนยืนยันการรับชำระเงิน หน้าคิดเงินจะทำการแสดงใบเสร็จรวม ซึ่งจะแสดงรายการอาหารที่สั่ง จำนวนของแต่ละรายการอาหาร ราคาแต่ละเมนู ยอดรวมค่าอาหาร และเวลาชำระเงิน พนักงานสามารถกดปุ่มพิมพ์ใบเสร็จลูกค้าเพื่อพิมพ์ใบเสร็จต่อไปได้

### 4.3 วิธี Deploy Spring Boot และ MySQL บน Railway

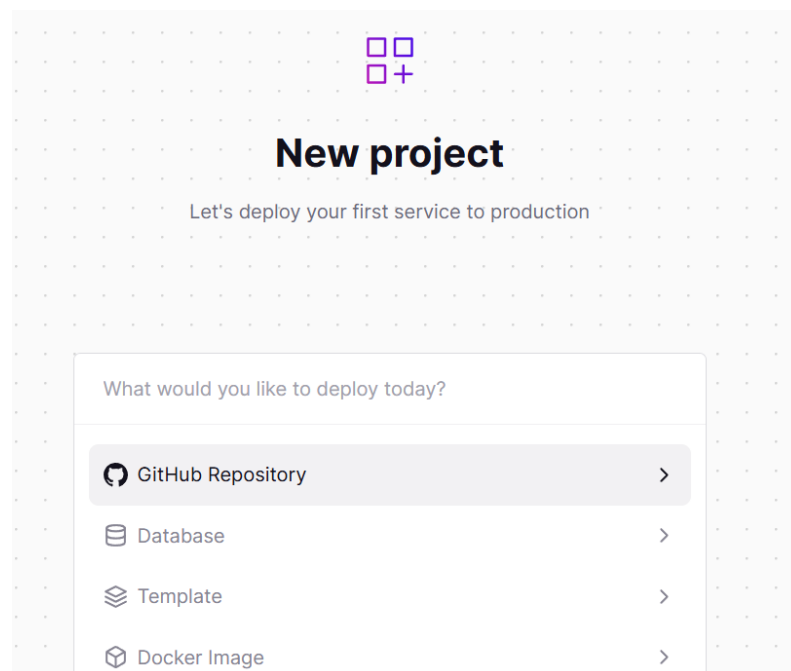
4.3.1 เตรียมโปรเจกต์โดยการตรวจสอบโครงสร้างว่า pom.xml ใช้ Spring Boot plugin และสามารถสร้าง executable JAR ได้ ซึ่งใช้คำสั่ง mvn clean package -DskipTests บน Terminal ผลลัพธ์จะอยู่ที่โฟลเดอร์ target

4.3.2 ปรับค่า Datasource ให้รองรับ Environment Variables โดยทำการแก้ไขใน application.properties

```
spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:${MYSQL_PORT:3306}/${MYSQL_DATABASE:mydb_demo}?use
spring.datasource.username=${MYSQL_USER:user01}
spring.datasource.password=${MYSQL_PASSWORD:s$crete01}
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

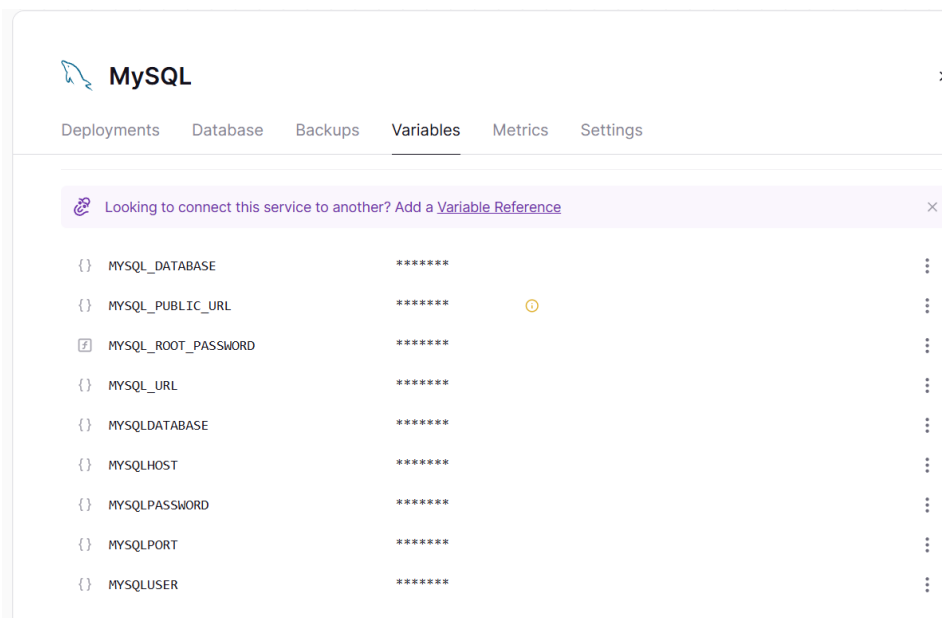
#### ภาพที่ 12 การปรับค่า Datasource

- 4.3.3 เตรียมไฟล์ Start Script
- 4.3.4 push โค้ดขึ้นบน Github โดยต้องทำการตรวจสอบว่า target ที่มีไฟล์ JAR ถูก commit
- 4.3.5 สร้าง Project และเชื่อม Github ใน Railway โดยคลิก New Project และกด Github Repository ทำการเลือก Repository ที่เราทำการ push ขึ้นบน Github



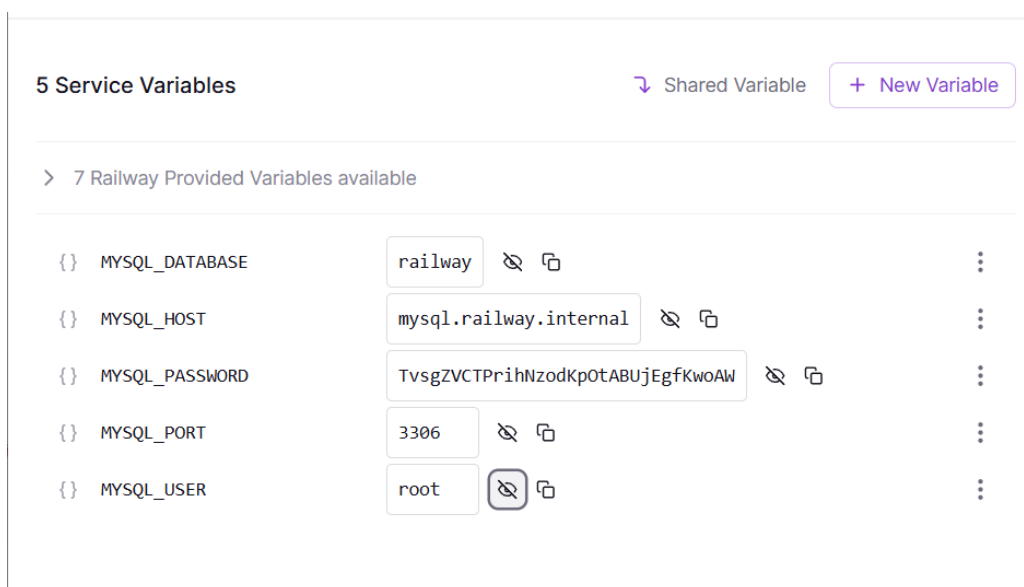
ภาพที่ 13 การ Deploy โดยเลือกผ่าน Repository บน Github

4.3.6 เพิ่ม MySQL Service บน Railway โดยทำการกด Database จากนั้นกดเลือก MySQL ซึ่งระบบจะสร้าง Environment Variables ให้



ภาพที่ 14 Environment Variables ที่ถูกสร้างขึ้น

4.3.7 การ Map Environment Variables โดยไปที่โปรเจกต์ที่ได้ทำการ Deploy กดตรง Variables และทำการเพิ่มตัวแปรใหม่ ดังภาพที่ 15



ภาพที่ 15 การเพิ่มตัวแปรใหม่ในโปรเจกต์ เพื่อทำการ Map

4.3.8 ตั้งค่า Start Command โดยไปที่ Project เลือก Setting และ Deployment เพื่อ Start Command จากนั้นทำการกรอกคำสั่งเพื่อรัน JAR

4.3.9 การ Deploy และการตรวจสอบการทำงานของระบบ โดยทำการเข้าสู่โปรเจกต์บน Railway จากนั้นทำการกดปุ่ม Deploy เพื่อสั่งการให้เริ่มกระบวนการ Deployment ทันที

4.3.10 การทดสอบการทำงานของระบบผ่าน Endpoint ภายนอก เมื่อระบบเริ่มต้นทำงานได้สมบูรณ์ Railway จะจัดสรร URL สำหรับการเข้าถึงแอปพลิเคชัน โดยอยู่ในรูปแบบดังตัวอย่างต่อไปนี้

A screenshot of a code editor showing a URL template: `https://<ชื่อโปรเจกต์>-production.up.railway.app`. The text is in a monospace font, and the placeholder text is in a lighter color.

ภาพที่ 16 ตัวอย่าง URL ที่ได้จาก Railway

ผู้พัฒนาสามารถใช้ URL ดังกล่าวเพื่อทดสอบการทำงานของระบบผ่านเบราว์เซอร์หรือเครื่องมือทดสอบ API เช่น Postman หากระบบทำงานถูกต้อง จะได้รับการตอบกลับ (Response) ที่แสดงถึงสถานะของระบบหรือข้อมูลที่ควรได้จาก API

## บทที่ 5

### สรุปผล และข้อเสนอแนะ

#### 5.1 สรุปผล

จากการปฏิบัติโครงงานครั้งนี้มีวัตถุประสงค์เพื่อพัฒนาและออกแบบเว็บไซต์สำหรับการสั่งอาหารออนไลน์ภายในร้านอาหาร โดยศึกษาการพัฒนาเว็บแอปพลิเคชันด้วย Spring Boot Framework ร่วมกับ RESTful API และออกแบบโครงสร้างฐานข้อมูลที่มีความสัมพันธ์แบบ One-to-One และ One-to-Many

จากผลการดำเนินงานพบว่าเว็บไซต์สำหรับการสั่งอาหารออนไลน์ภายในร้านอาหารสามารถอำนวยความสะดวกให้แก่ผู้ใช้งานได้ในระดับหนึ่ง โดยในส่วนของ ฟังก์ชันฝั่งพนักงาน สามารถดูบิลของแต่ละโต๊ะเพื่อสรุปยอดชำระเงินให้ลูกค้าได้ นอกจากนี้ยังสามารถดู แดชบอร์ดยอดขาย (Dashboard) และตรวจสอบได้ว่าเมนูใดเป็นเมนูขายดีในแต่ละวันสำหรับ ฟังก์ชันฝั่งพนักงานครัว สามารถดูรายการอาหารที่ลูกค้าสั่งเข้ามา และกดยืนยันเมื่อทำอาหารสำเร็จแล้ว ส่วน ฟังก์ชันฝั่งลูกค้า สามารถสแกนสั่งอาหารออนไลน์ผ่านเว็บไซต์ได้อย่างสะดวก

อย่างไรก็ตาม เว็บไซต์ยังคงมีบางฟังก์ชันที่สามารถพัฒนาและเพิ่มเติมได้ในอนาคต เพื่อให้ระบบมีความสมบูรณ์ยิ่งขึ้นและสามารถอำนวยความสะดวกแก่ผู้ใช้งานได้อย่างมีประสิทธิภาพสูงสุด

#### 5.2 ข้อเสนอแนะ

จากผลการปฏิบัติโครงงานครั้งนี้ สามารถเสนอแนะแนวทางการพัฒนาเว็บไซต์สำหรับการสั่งอาหารออนไลน์ภายในร้านอาหาร ให้มีสามารถตอบสนองความต้องการของผู้ใช้งาน ดังนี้

##### 5.2.1 การพัฒนาฟังก์ชันการใช้งาน

1. เพิ่มฟังก์ชันการจัดการเมนูโดยพนักงาน  
เพื่อให้พนักงานสามารถเพิ่ม ลบ หรือแก้ไขรายการอาหาร ราคา และรูปภาพเมนูได้โดยตรงผ่านหน้าเว็บไซต์ โดยไม่ต้องเข้าถึงฐานข้อมูลโดยตรง
2. เพิ่มฟังก์ชันที่ลูกค้าสามารถดูยอดรวมของรายการอาหารที่สั่งได้ด้วยตนเอง  
ก่อนที่พนักงานจะมาสรุปใบเสร็จ เพื่อให้ลูกค้าสามารถตรวจสอบยอดค่าใช้จ่ายเบื้องต้นได้อย่างสะดวกและโปร่งใส
3. พัฒนาระบบแจ้งเตือน  
เพื่อให้พนักงานและพนักงานครัวได้รับการแจ้งเตือนเมื่อมีออเดอร์ใหม่เข้ามา

## เอกสารอ้างอิง

What is a REST API?. (2020, May 8). Retrieved October 19, 2025, from

<https://www.redhat.com/en/topics/api/what-is-a-rest-api>

KongRuksiam Studio. (2024, Apr 1). รู้จักกับ API และ REST API (RESTful web service). Retrieved October 19, 2025, from

<https://kongruksiam.medium.com/rest-api-restful-web-services-10c98edf7b51>

Maria Shkutnik. (2024, November 15). What is CRUD? A Developer's Guide to the Basics of Web App Operations. Retrieved October 19, 2025, from <https://uibakery.io/blog/what-is-crud>