



## Software Requirements Specification (Quickbites Ordering system)

### จัดทำโดย

6604062630111	นายจุมพลภัทร์	สาเกตุล
6604062630251	นายธีรชฌานนท์	ศรีธรรมยศ
6604062630358	นายพงษ์พัฒน์	บางข่า
6604062630561	นายอชิตพล	แทนโป
6604062630579	นายอริป	ยวงแก้ว

### เสนอ

ผู้ช่วยศาสตราจารย์ สถิตย์ ประสมพันธ์

รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชา 040613306 วิศวกรรมซอฟต์แวร์  
ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ปีการศึกษา 2/2567

## สารบัญ

	หน้า
1. Introduction	2
1.1 ที่มาและความสำคัญของระบบ	2
1.2 Stakeholder	2
1.3 Software Purpose	3
1.4 Software Scope	3
2. System Architecture	4
3. User Interface Prototype	5-7
4. Specification Requirements	8
4.1 User Requirement	8
4.2 System Requirement	8
4.3 Functional Requirement	9
4.4 Non-Functional Requirement	9
5. แผนภาพยูสเคส (Use - Case Diagram)	10
6. แผนภาพคลาส(Class Diagram)	11
7. แผนภาพกิจกรรม (Activity Diagram)	12-23
8. แผนภาพการปรับใช้ (Deployment Diagram)	24
9. แผนภาพความสัมพันธ์ของเอนทิตี (ER Diagram)	25
10. User Acceptance Test	26-38
11. Process Model	39
12. Risk Management	40
13. Risk Planning	41
14. Version Control	42-43
15. QA (Quality Assurance)	44-45

## Introduction

### 1.1 ที่มาและความสำคัญของระบบ

ในปัจจุบัน ธุรกิจร้านอาหารและภัตตาคารต้องการระบบที่ช่วยเพิ่มความสะดวกให้แก่ลูกค้าและลดภาระงานของพนักงาน ระบบการสั่งอาหารแบบเดิมที่ใช้พนักงานรับออเดอร์อาจทำให้เกิดข้อผิดพลาด เช่น รับออเดอร์ผิด คิวการสั่งอาหารล่าช้า หรือเกิดความสับสนในการให้บริการ ด้วยเหตุนี้ Quickbites จึงถูกพัฒนาขึ้นเพื่อตอบสนองต่อการให้บริการที่รวดเร็วและแม่นยำยิ่งขึ้น Quickbites เป็นระบบที่ช่วยให้ลูกค้าสามารถใช้สมาร์ทโฟนสแกน QR Code เพื่อเข้าถึงเมนูดิจิทัลและสั่งอาหารได้โดยตรง ข้อมูลคำสั่งซื้อจะถูกส่งไปยังครัวและบันทึกลงฐานข้อมูลโดยอัตโนมัติ นอกจากนี้ ระบบยังสามารถบันทึกข้อมูลยอดขาย รายรับ-รายจ่าย และสถิติทางการเงินของร้านได้อย่างเป็นระบบ ทำให้เจ้าของร้านสามารถวิเคราะห์ยอดขายและวางแผนธุรกิจได้อย่างมีประสิทธิภาพ

### 1.2 Stakeholder

#### 1. ลูกค้า :

- เป็น Stakeholder ที่สำคัญ เพราะระบบจะมุ่งเน้นการให้บริการที่สะดวกและประทับใจ โดยคาดหวังการใช้งานที่ง่าย รวดเร็ว และรองรับการชำระเงินทั้งแบบออนไลน์และเงินสด

#### 2. เจ้าของร้าน/ผู้จัดการ :

- มีบทบาทในการตัดสินใจ เกี่ยวกับการใช้งานระบบ การพัฒนา รวมถึงการปรับปรุงฟีเจอร์ใหม่ๆ และวางแผนในการทำงานของร้าน

#### 3. พนักงานในร้านอาหาร :

- มีบทบาทหน้าที่รับออเดอร์, เสิร์ฟอาหาร, ตรวจสอบคำสั่งซื้อรายการอาหาร และให้บริการลูกค้า โดยจะมี Quickbites มาช่วยในส่วนของการบริการให้สะดวกและง่ายมากยิ่งขึ้น

#### 4. ทีมพัฒนาระบบ :

- ทีมพัฒนาระบบต้องรับฟัง Feedback และปรับปรุงฟังก์ชันให้ตรงกับความต้องการของลูกค้าและร้านพัฒนาและดูแลระบบให้ทำงานได้อย่างราบรื่น โดยต้องมั่นใจว่าระบบปลอดภัย เสถียร และใช้งานง่าย

### 1.3 Software Purpose

วัตถุประสงค์ของเอกสารฉบับนี้คือการกำหนด **ข้อกำหนดซอฟต์แวร์ (Software Requirements Specification - SRS)** สำหรับ **QuickBites** ซึ่งเป็นระบบสั่งอาหารออนไลน์ผ่านเว็บไซต์ด้วยการสแกนQR Code **QuickBites** มีเป้าหมายเพื่อให้กระบวนการสั่งอาหารเป็นเรื่องง่ายและมีประสิทธิภาพมากขึ้น โดยเชื่อมต่อลูกค้ากับร้านอาหารและช่วยให้การจัดการออเดอร์เป็นไปอย่างราบรื่น

### 1.4 Software Scope

QuickBites เป็นแพลตฟอร์มออนไลน์ที่ช่วยให้ลูกค้าสามารถเรียกดูเมนู สั่งอาหาร และชำระเงินได้ โดยระบบประกอบด้วย 2 ระบบหลัก :

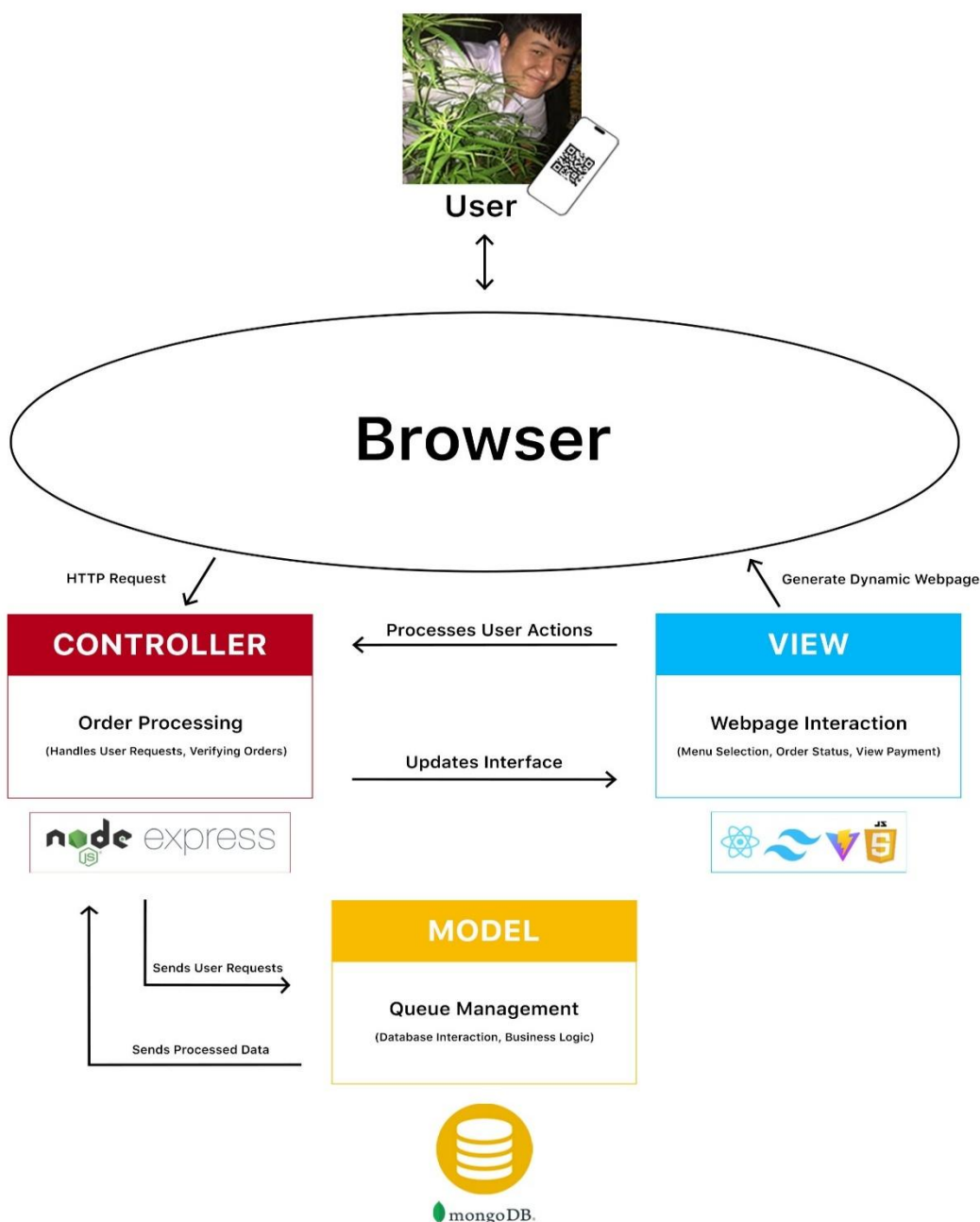
1. **ระบบสำหรับลูกค้า :** สำหรับให้ลูกค้าเรียกดูร้านอาหาร สั่งอาหาร และทำการชำระเงิน :
2. **ระบบสำหรับร้านอาหาร/ผู้ดูแลระบบ :** สำหรับร้านอาหารใช้จัดการออเดอร์ อัปเดตเมนู และให้ผู้ดูแลระบบใช้ตรวจสอบและดูแลการดำเนินงานของแพลตฟอร์ม

โดยมีสิ่งที่มีอยู่ในระบบ :

- อินเทอร์เฟซลูกค้า สำหรับเรียกดูเมนู สั่งอาหาร และตรวจสอบสถานะคำสั่งซื้อ
- ระบบจัดการออเดอร์ ที่ส่งคำสั่งซื้อไปยังครัวโดยอัตโนมัติ
- ระบบเรียกเก็บเงิน เพื่อให้ลูกค้าตรวจสอบและดำเนินการชำระเงิน
- แผงควบคุมสำหรับผู้ดูแลระบบ ให้พนักงานร้านอาหารสามารถจัดการเมนู ติดตามคำสั่งซื้อ และอัปเดตสถานะออเดอร์

## System Architecture

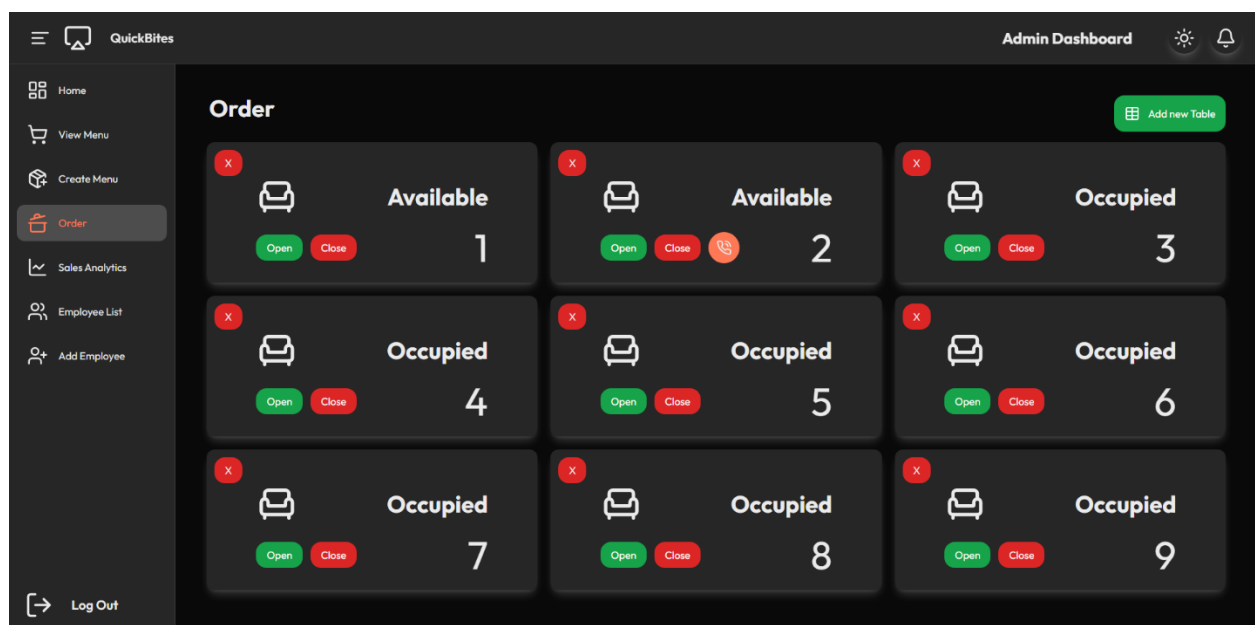
### QUICKBITE SYSTEM ARCHITECTURE



รายละเอียด System Architecture ของ QuickBites ระบบ QuickBite ประกอบด้วยส่วนประกอบหลัก 3 ส่วนตามรูปแบบ MVC (Model-View-Controller) :

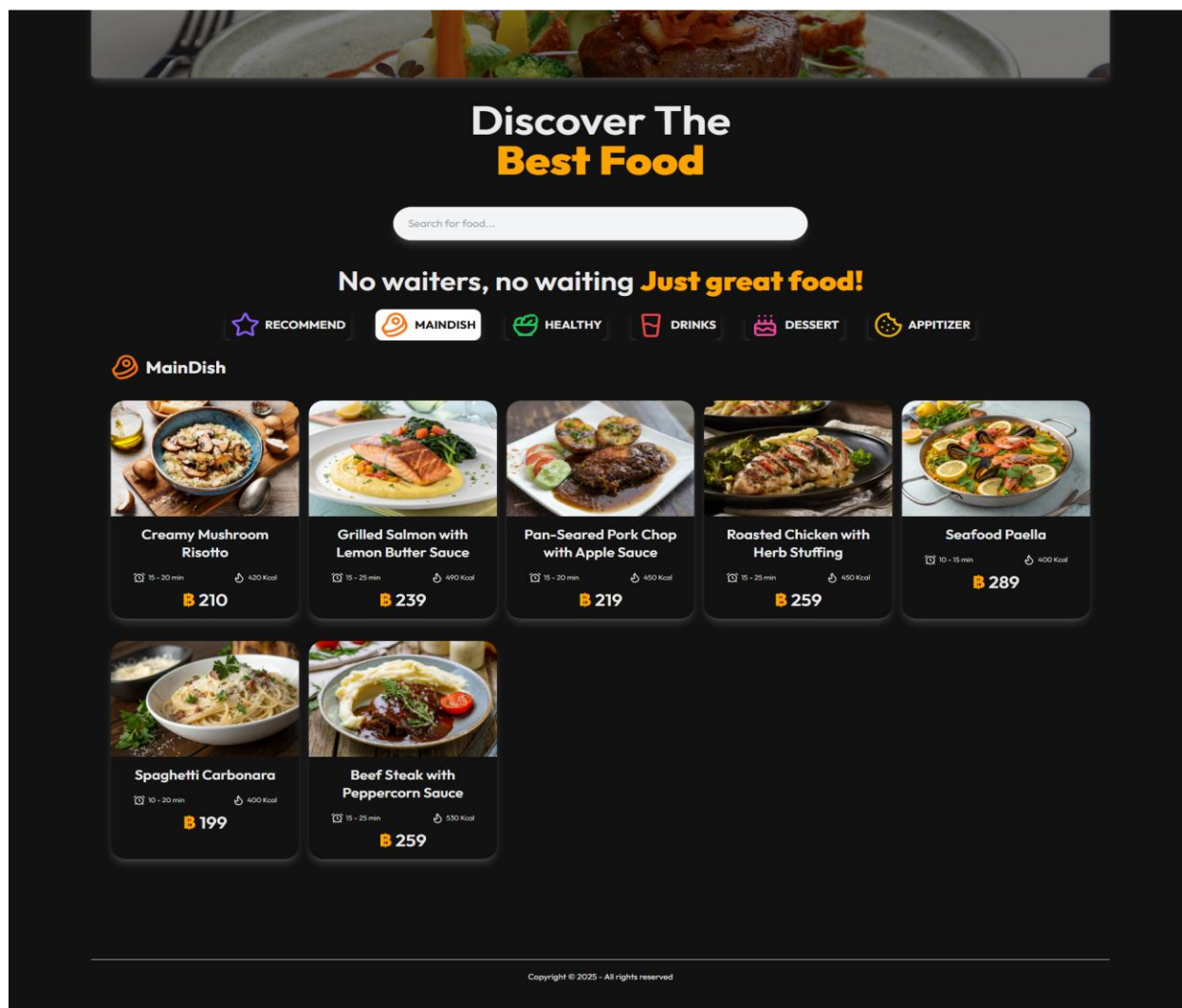
- 1. User (ผู้ใช้)** ผู้ใช้เข้าระบบผ่านเว็บไซต์ทำการส่งคำขอ HTTP ไปยังระบบ
- 2. Browser (เว็บเบราว์เซอร์)** ทำหน้าที่แสดงอินเทอร์เฟซผู้ใช้ รับการโต้ตอบจากผู้ใช้และส่งคำขอไปยัง Controller
- 3. Controller (หน่วยประมวลผลหลักของระบบ)** ทำหน้าที่จัดการคำขอจากผู้ใช้
- 4. View** ส่วนที่จัดการการแสดงผลและการโต้ตอบกับผู้ใช้
- 5. Model** ทำหน้าจําลองการติดต่อกับระบบข้อมูล โดยใช้ Express.js เป็นเฟรมเวิร์กหลัก
- 6. ฐานข้อมูล (mongoDB)** ใช้ MongoDB

## User Interface Prototype



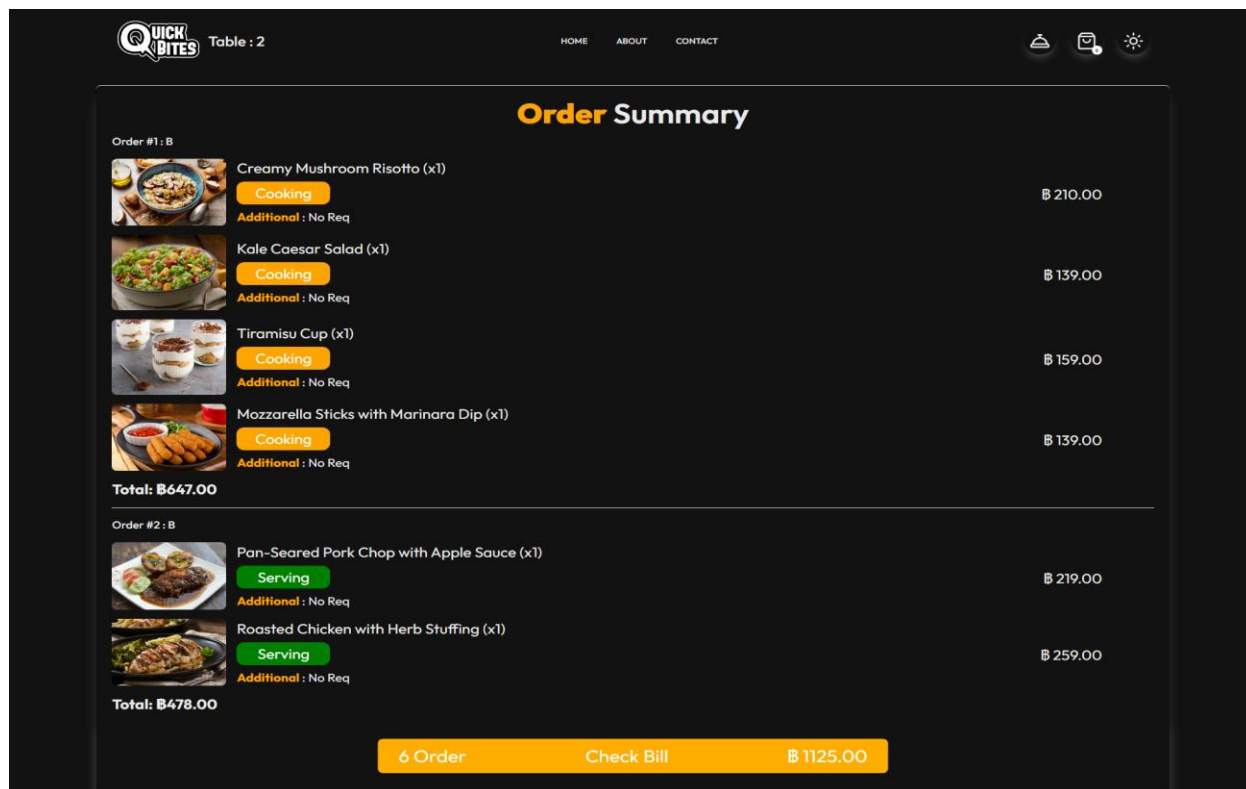
Admin Dashboard UI ของระบบQuickbites ในหน้าHome pages จะมีการแสดงผลதியอดขายที่ผ่านๆมาของร้านซึ่งสามารถใช้ประโยชน์ได้หลายทางเช่น คำนวณการขายรายการอาหาร ด้านธุรกิจต่างๆ เป็นต้น มีแสดงคำสั่งซื้อOrderล่าสุดจากลูกค้า และรายการอาหารที่เป็นที่นิยม

ในหน้าต่าง Order จะสามารถเปลี่ยนสถานะโต๊ะได้ โดยจะสามารถรับการแจ้งเตือนคำสั่งซื้อจากลูกค้าได้จากหน้านี้เช่นกัน



ส่วนของเว็บลูกค้า จะมีการแสดงผลเมนู ประเภท รูปภาพ ชื่อ ราคา เวลาที่ต้องใช้ในการทำ แคลลอรีของอาหารต่างๆภายในร้าน โดยสามารถเพิ่ม/ลบ หรือ ใส่รายละเอียดต่างๆเข้าไปในรายการอาหารที่จะสั่งได้ และมีช่องค้นหาเมนูอาหารเพื่อพิมพ์ค้นหาอาหารที่ต้องการอย่างสะดวกและง่าย รวมถึงหน้านี้จะมีช่องทางติดต่อของ

ทางร้าน ไอคอนตะกร้าเพื่อที่จะเข้าไปดูรายการอาหารที่ใส่ตะกร้าเอาไว้แล้ว ไอคอนปรับเปลี่ยนธีมDark/Lightของเว็บ สามารถกดสั่งอาหารหลายๆครั้งได้โดยไม่จำเป็นต้องกดสั่งทีเดียวทั้งหมด เพื่อความยืดหยุ่นในการสั่งอาหาร



หน้าต่างOrder Summaryของลูกค้า จะมีการแสดงผลบอกจำนวนรายการอาหารที่สั่ง สถานะของอาหารที่สั่งไป โดยจะมีการอัปเดตสถานะของอาหารผ่านทางEmployeeและAdminในAdmin Dashboard แสดงผลราคารวมทั้งหมดของอาหารที่สั่งและปุ่มกดเพื่อเข้าสู่กระบวนการชำระเงิน โดยส่วนของแถบไอคอนด้านบนของหน้าOrder Summaryจะทำหน้าที่เหมือนกับหน้าHomeทุกประการ



## Specification Requirements

### 4.1 User Requirement

#### 4.1.1 ระบบสำหรับลูกค้า:

- เรียกดูเมนูและดูรายละเอียดคำอธิบายของรายการอาหาร
- สั่งอาหารและติดตามสถานะออเดอร์แบบเรียลไทม์
- เรียกพนักงานเสิร์ฟ
- ชำระเงินผ่านช่องทาง สแกนQR Code และ เงินสด

#### 4.1.2 ระบบสำหรับร้านอาหาร/ผู้ดูแลระบบ:

- สำหรับร้านอาหาร
  - รับและดำเนินการคำสั่งซื้อจากลูกค้าผ่านระบบ
- สำหรับผู้ดูแลระบบ
  - เพิ่มและลบเมนูในระบบ
  - อัปเดตและอัปโหลดรายการเมนูพร้อมราคาและคำอธิบาย
  - ติดตามสถิติยอดขายของอาหาร
  - เพิ่มหมายเลขโต๊ะของร้านอาหารลงในระบบ
  - เรียกดูข้อมูลของอาหารภายในระบบ

### 4.2 System Requirement

- การยืนยันตัวตนและการอนุญาต: ระบบล็อกอินที่ปลอดภัยสำหรับผู้ดูแลระบบ
- การจัดการฐานข้อมูล: การจัดเก็บข้อมูลผู้ใช้ รายละเอียดเมนู ประวัติออเดอร์ ความคิดเห็นในที่เดียว
- ระบบการแจ้งเตือน: การแจ้งเตือนผ่านอีเมลสำหรับการอัปเดตสถานะออเดอร์และความคิดเห็น
- การอัปเดตแบบเรียลไทม์ : ให้แน่ใจว่าคำสั่งซื้อต่าง ๆ ได้รับการอัปเดตในเวลาจริง

### 4.3 Functional Requirement

1. การสแกน QR Code : ลูกค้าควรสามารถสแกน QR Code ที่โต๊ะของตนเพื่อเข้าถึงเว็บไซต์ (เช่น การเรียกดูเมนู, การสั่งอาหาร, การดูสถานะออเดอร์แบบเรียลไทม์ ฯลฯ)

2. การจัดการคำสั่งซื้อ: ควรสามารถจัดการคำสั่งซื้อที่ลูกค้าสั่ง โดยการอัปเดตสถานะของคำสั่ง (สั่ง -> กำลังทำอาหาร -> พร้อมเสิร์ฟ -> เสิร์ฟแล้ว)

3. การจัดการเมนูของร้าน: ผู้จัดการระบบร้านอาหารควรสามารถจัดการเมนูของร้านได้ โดยการเพิ่ม, อัปเดต และลบรายการเมนู รวมถึงการเข้าถึงประวัติคำสั่งซื้อเพื่อทำการวิเคราะห์ยอดขาย

4. การชำระเงิน: ระบบควรสามารถทำการสรุปการชำระเงินอาหาร โดยการสร้างใบแจ้งค่าชำระสำหรับแต่ละโต๊ะ

### 4.4 Non-Functional Requirement

1. ประสิทธิภาพ(Performance): ระบบควรประมวลผลคำสั่งซื้อและโหลดอินเทอร์เฟซผู้ใช้ภายใน 5 วินาที

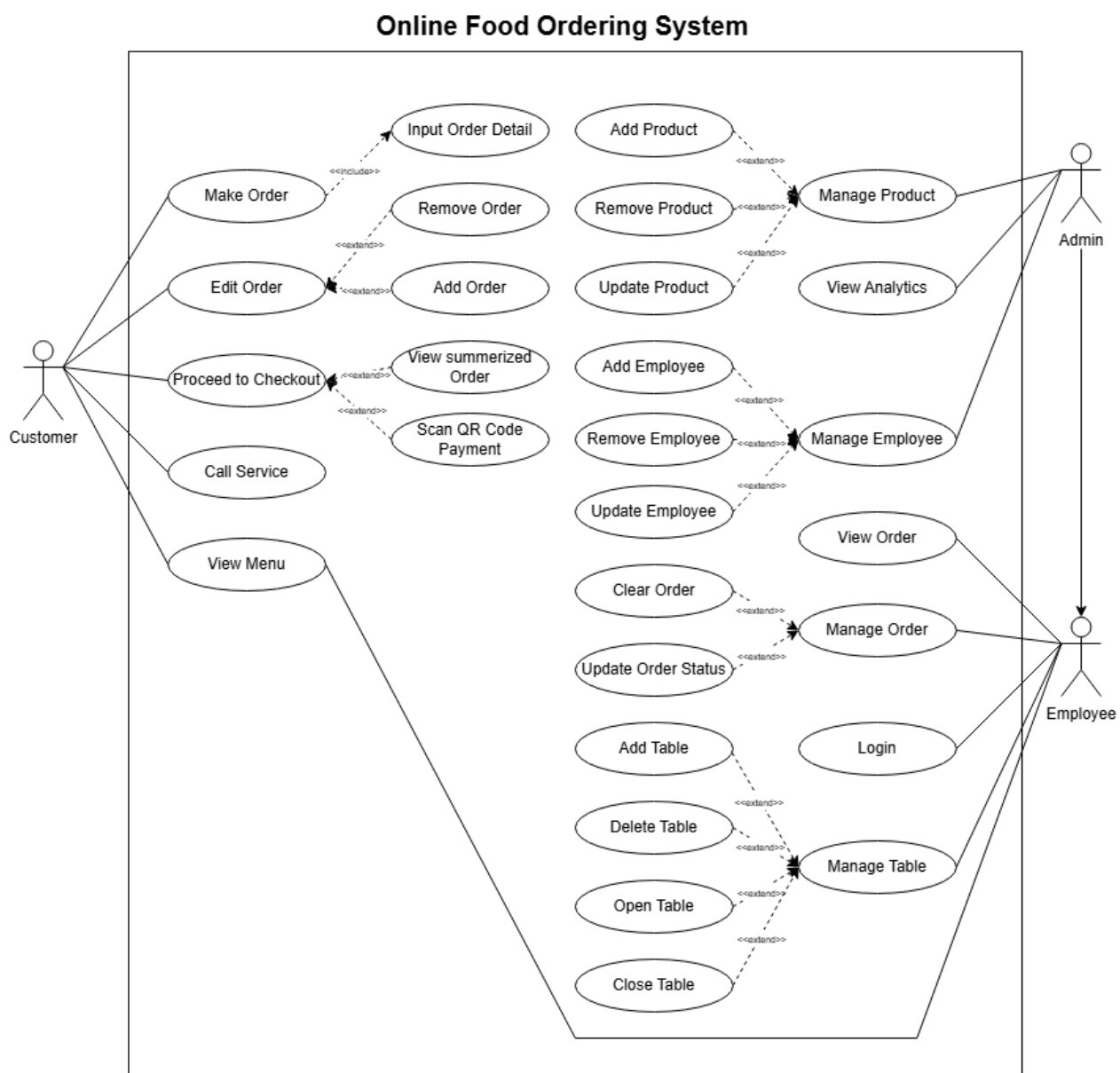
2. ความเสถียรภาพ(Reliability): ระบบควรมีอัตราการทำงานที่ 99.9% เพื่อให้แน่ใจว่าระบบพร้อมใช้งานตลอดเวลาในการดำเนินงานของร้านอาหาร หากการเชื่อมต่ออินเทอร์เน็ตขาดหายไป คำสั่งซื้อล่าสุดยังคงถูกคิวไว้และระบบสามารถรองรับผู้ใช้งานพร้อมกันอย่างน้อย 100 คนในร้านอาหาร

3. การใช้งาน(Usability): อินเทอร์เฟซผู้ใช้ควรเป็นแบบ **mobile-first**, รองรับการใช้งานได้ดีบนหน้าจอขนาดต่าง ๆ รวมถึงสมาร์ทโฟนและแท็บเล็ตและกระบวนการสั่งอาหารควรมีการออกแบบที่ใช้ทำงานง่าย ทำให้ลูกค้าสามารถสั่งอาหารได้ใน **3 ขั้นตอนหรือน้อยกว่า** ตั้งแต่การเลือกเมนูจนถึงการยืนยันคำสั่งซื้อ

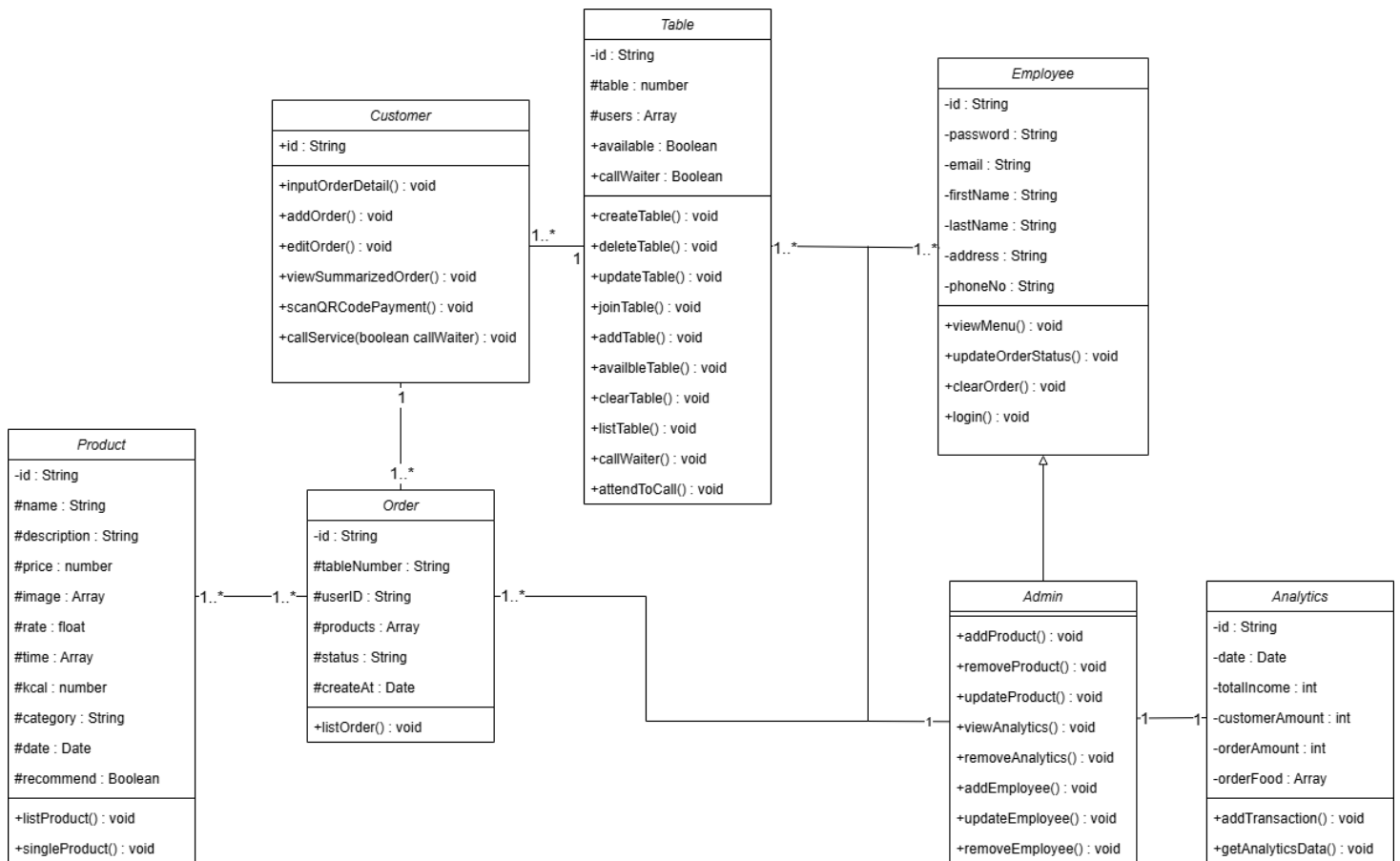
## แผนภาพยูสเคส (Use – Case Diagram)

การทำงานของระบบสั่งอาหารออนไลน์ผ่านเว็บไซต์ด้วยการสแกนQR CodeของQuickbites

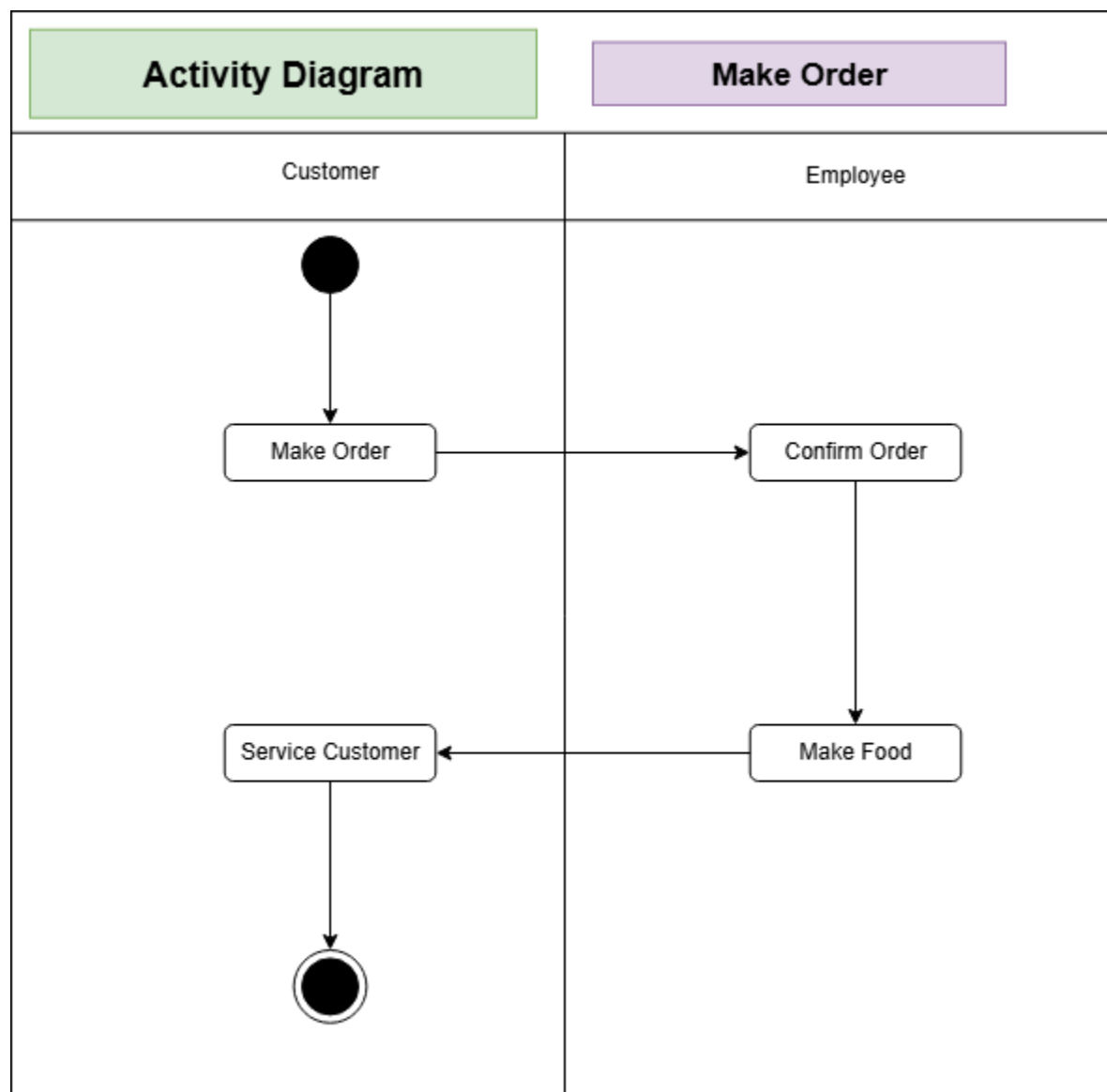
สามารถแสดงได้ด้วยแผนภาพยูสเคสดังนี้



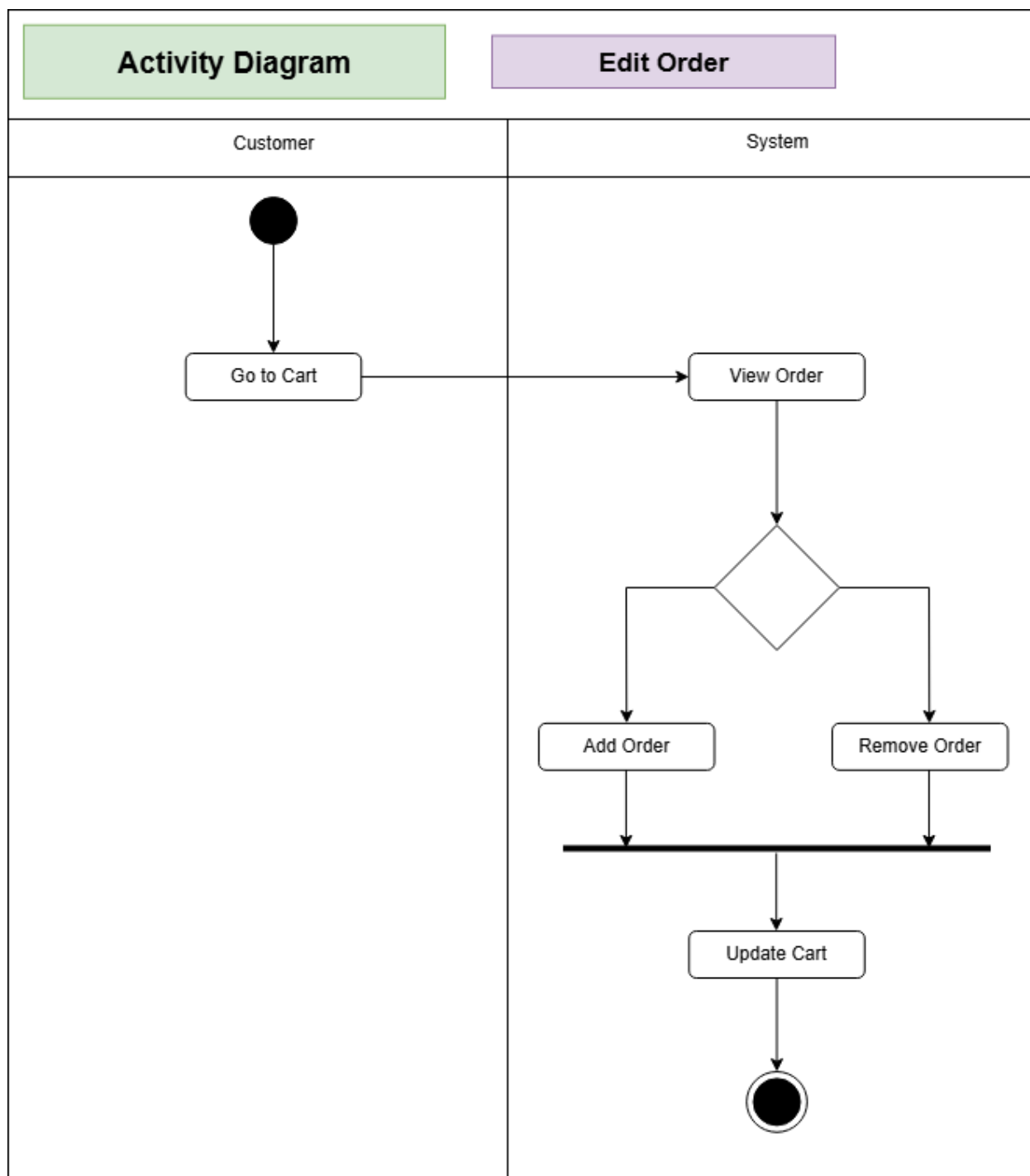
## แผนภาพคลาส (Class Diagram)



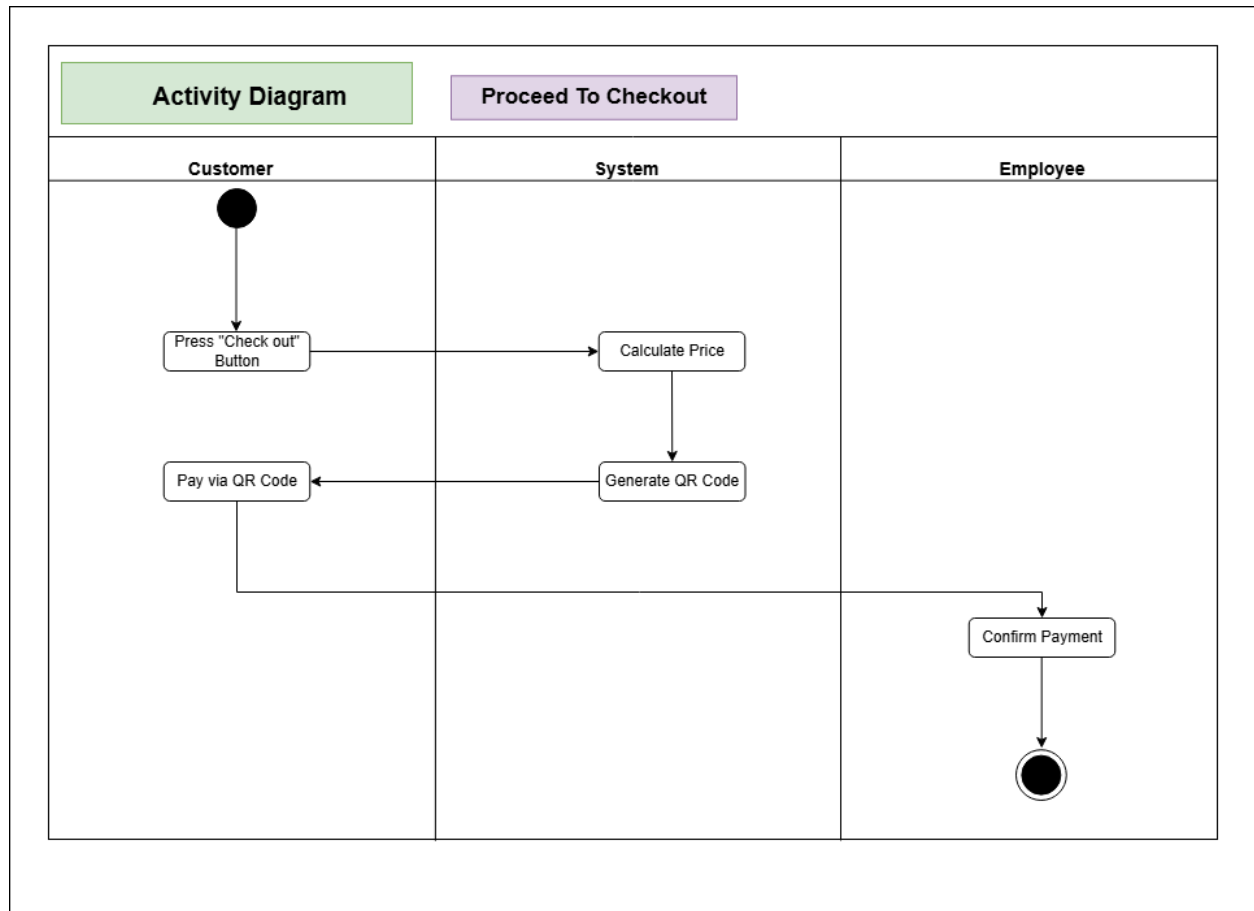
## แผนภาพกิจกรรม (Activity Diagram)



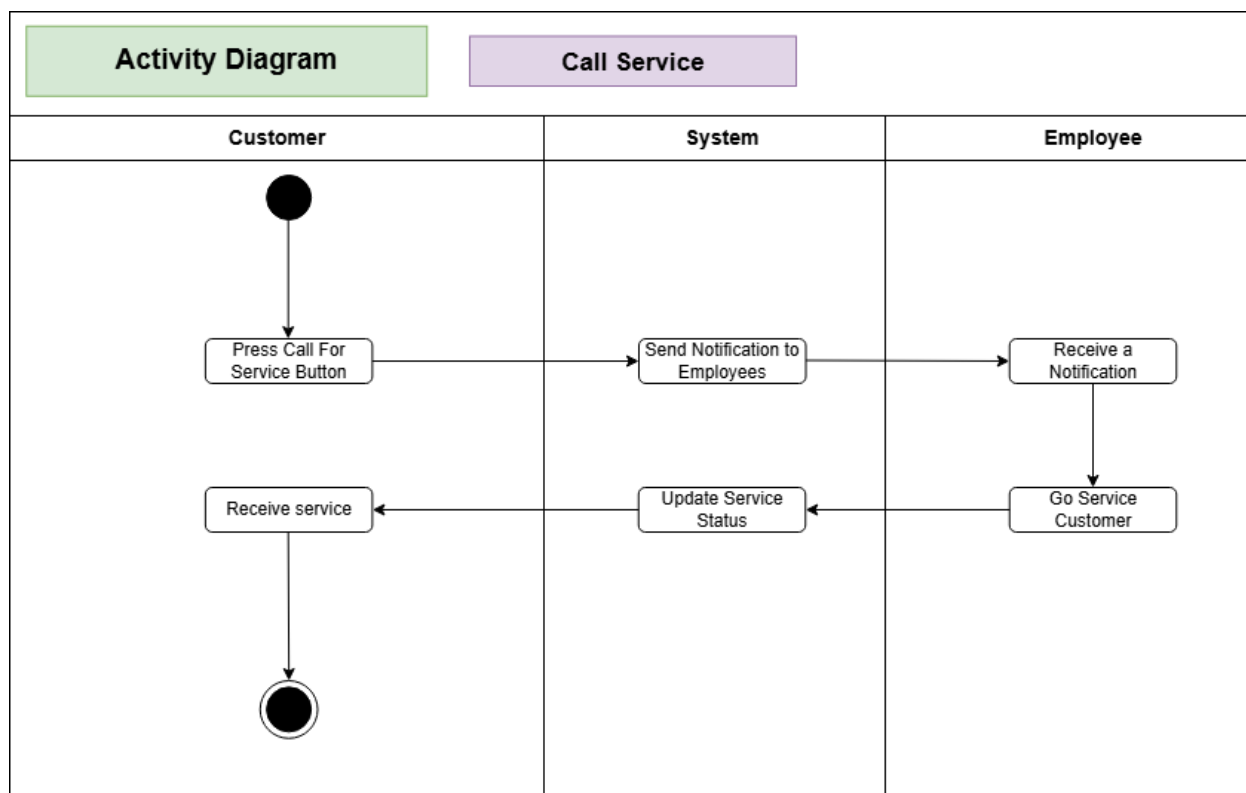
## แผนภาพกิจกรรม (Activity Diagram)



## แผนภาพกิจกรรม (Activity Diagram)

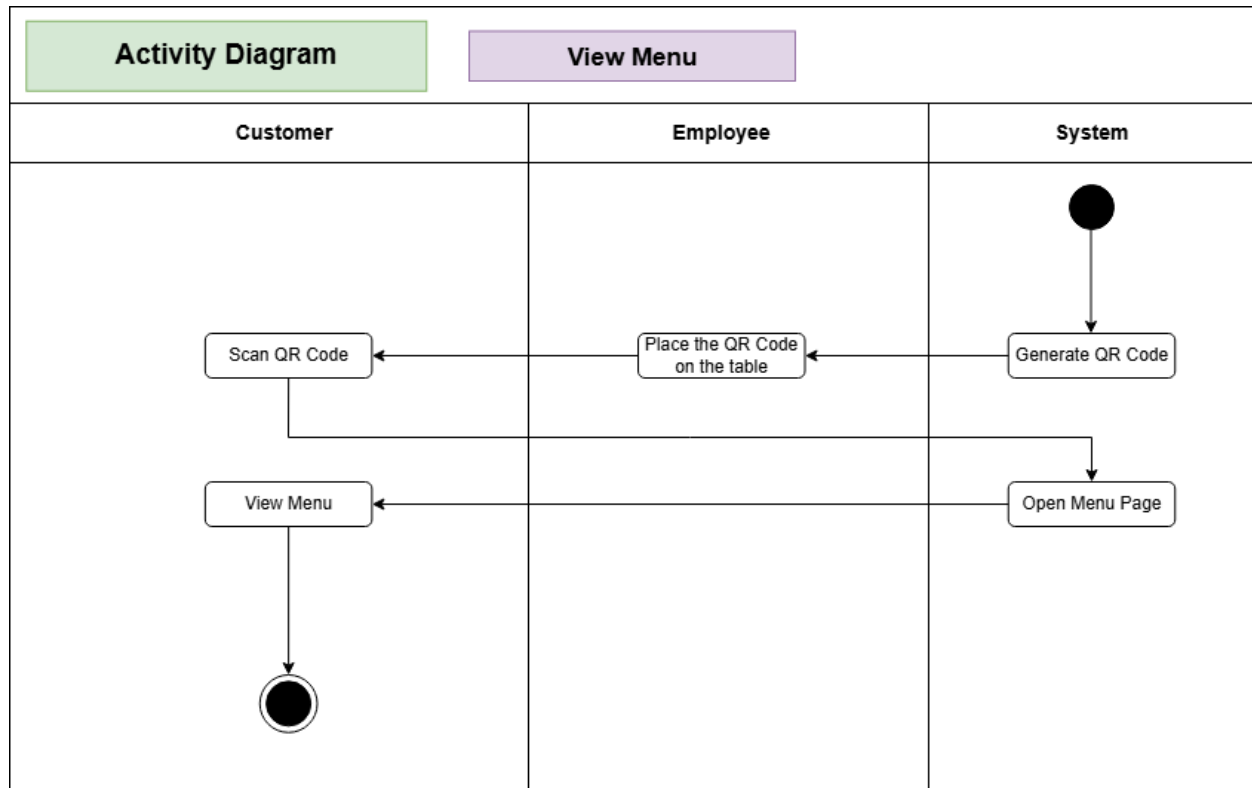


## แผนภาพกิจกรรม (Activity Diagram)

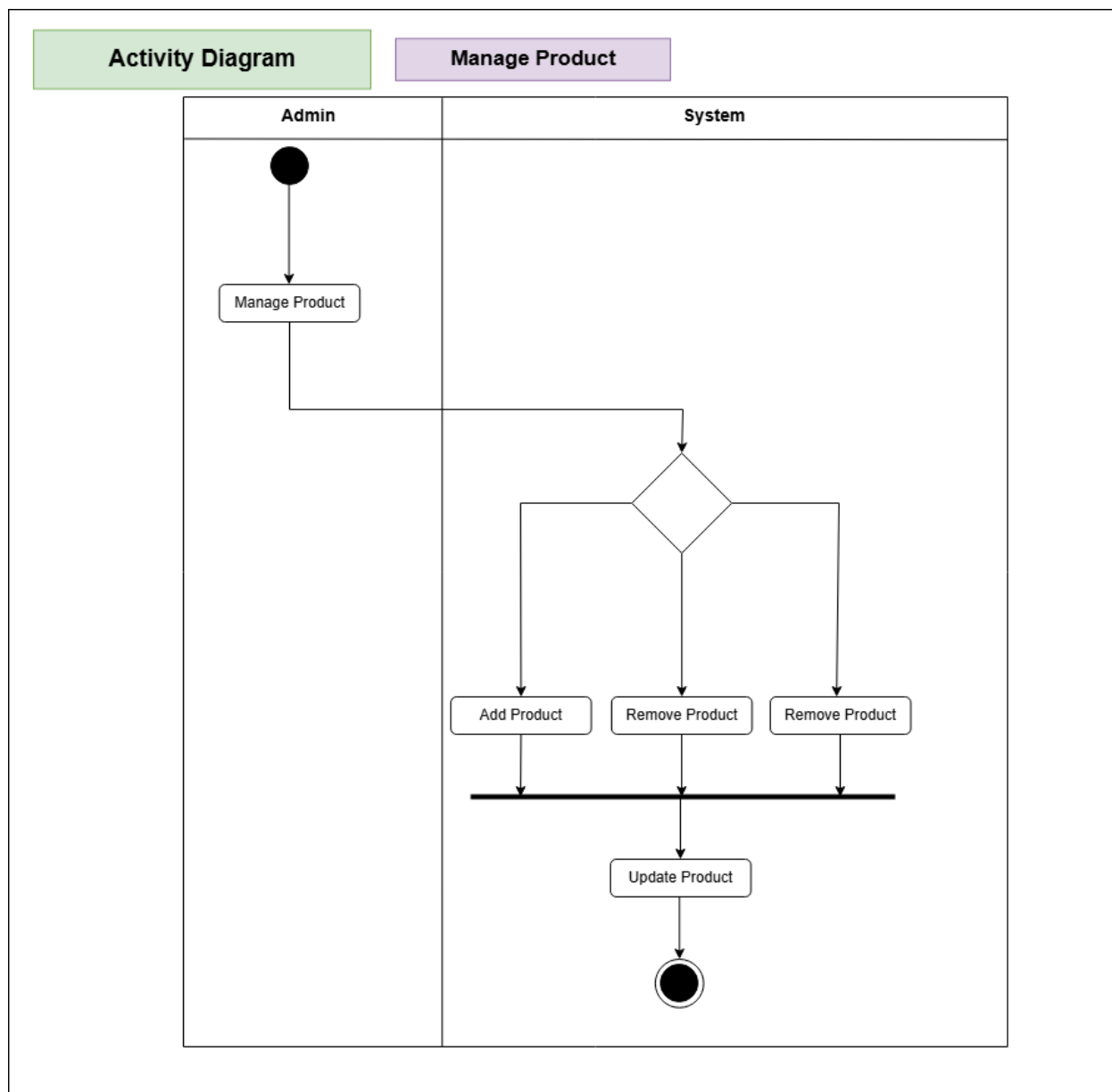




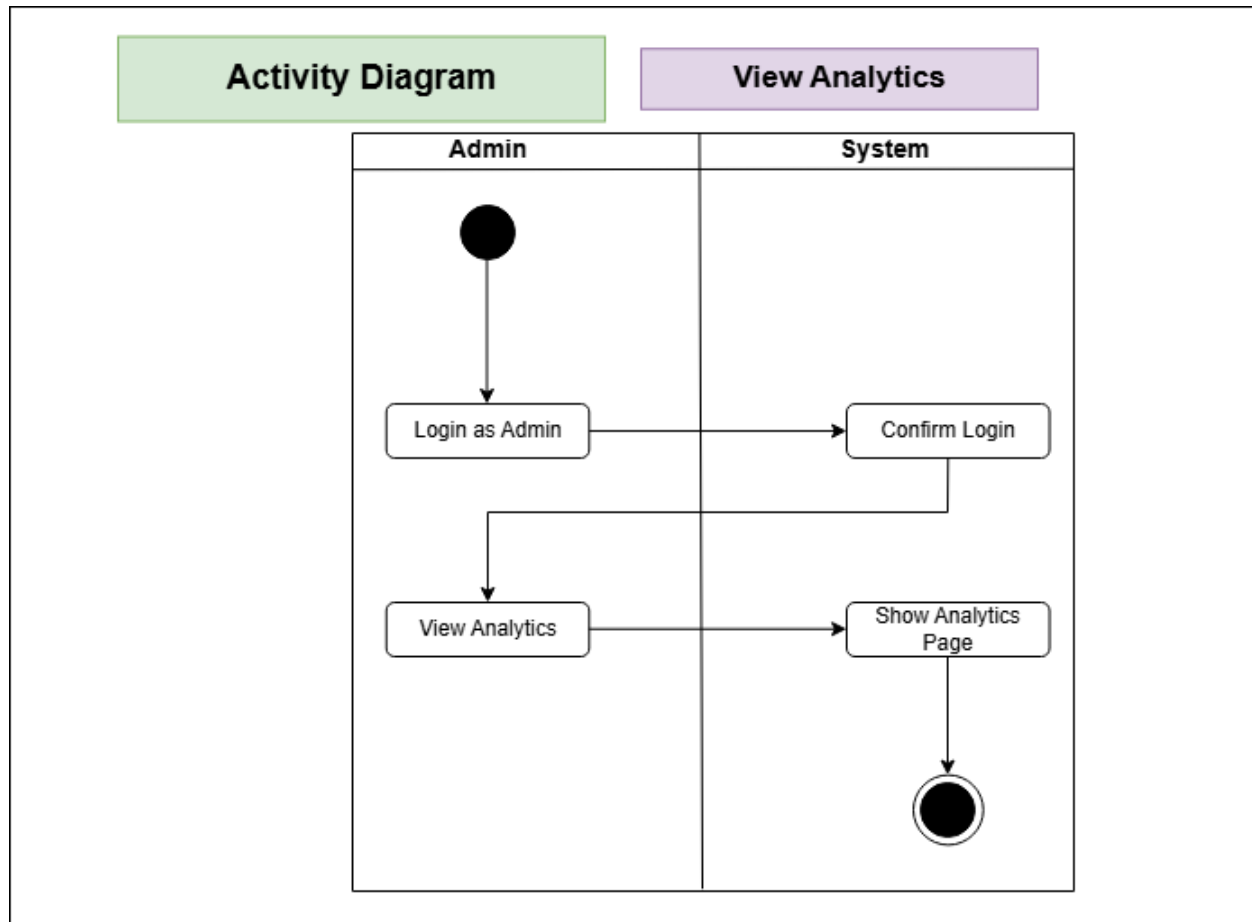
## แผนภาพกิจกรรม (Activity Diagram)



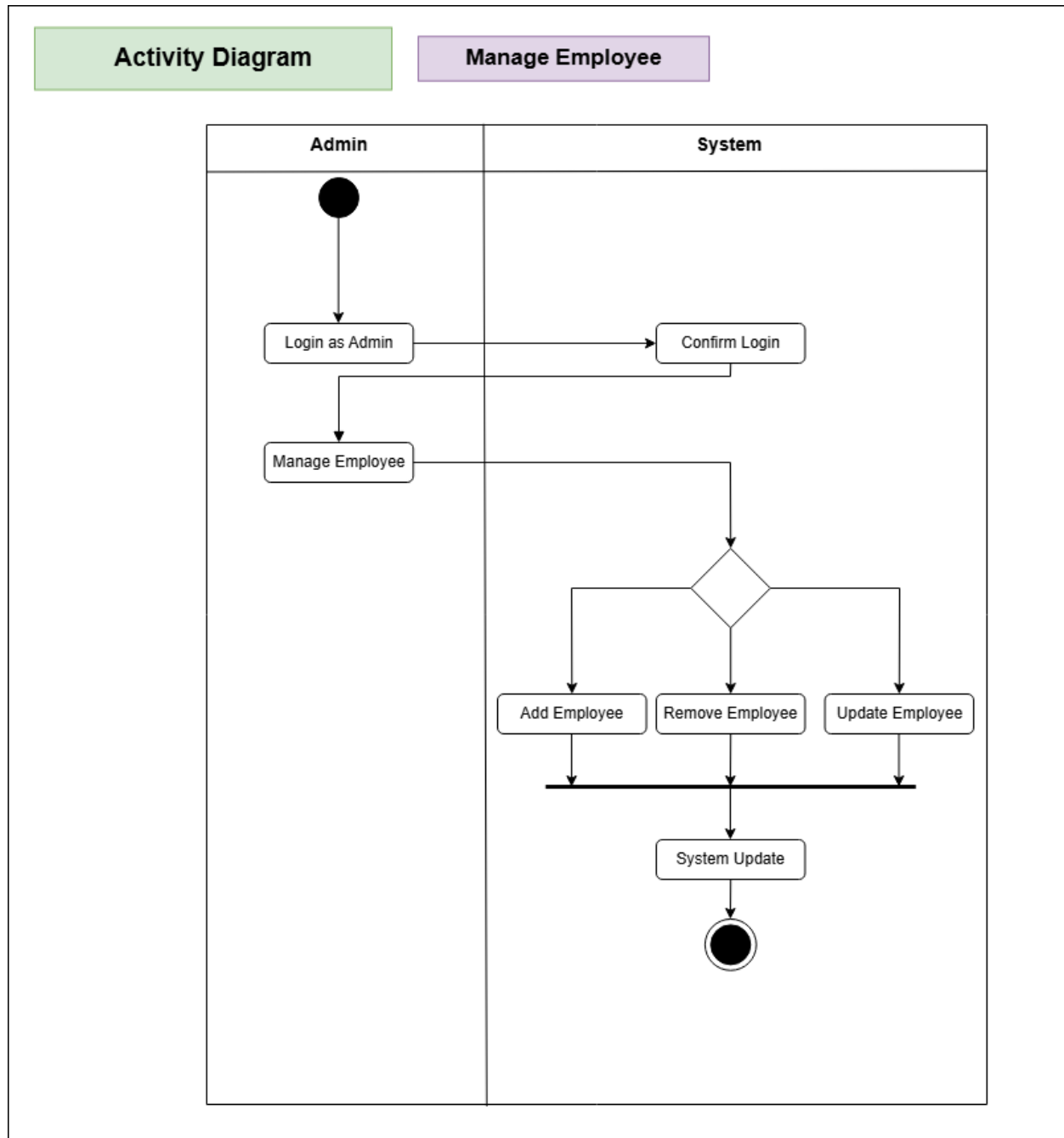
## แผนภาพกิจกรรม (Activity Diagram)



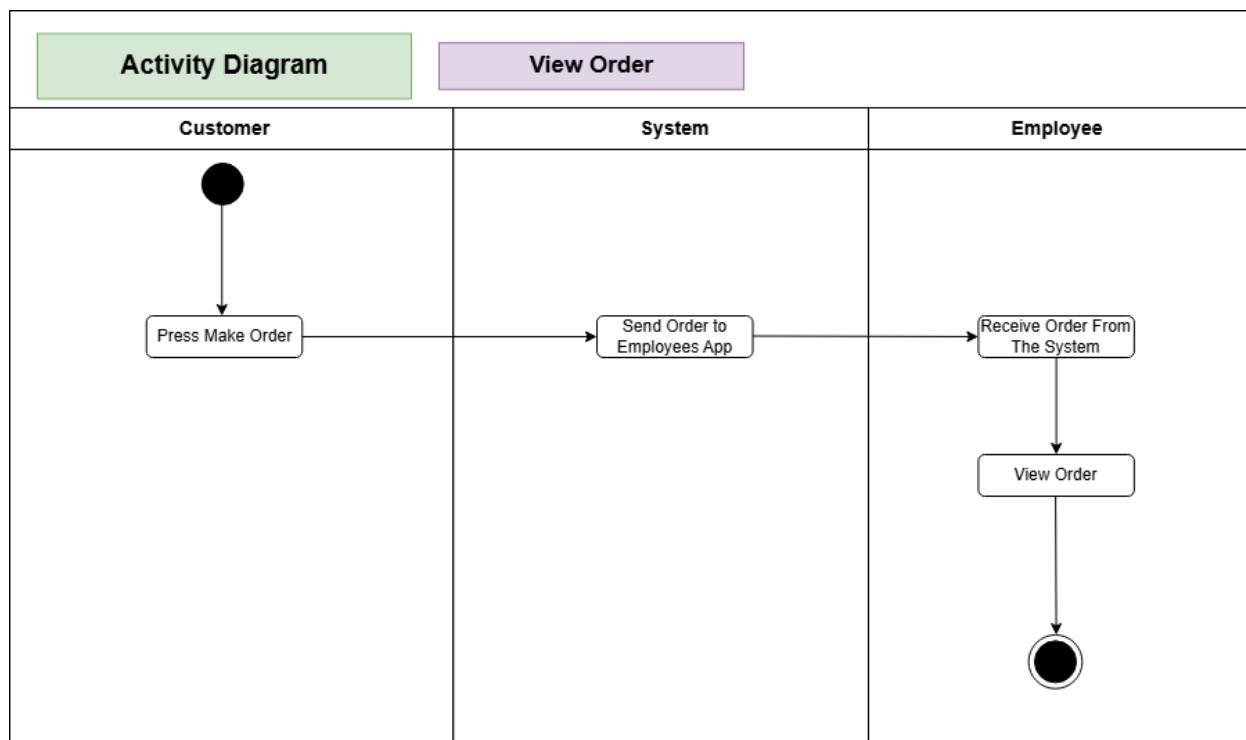
## แผนภาพกิจกรรม (Activity Diagram)



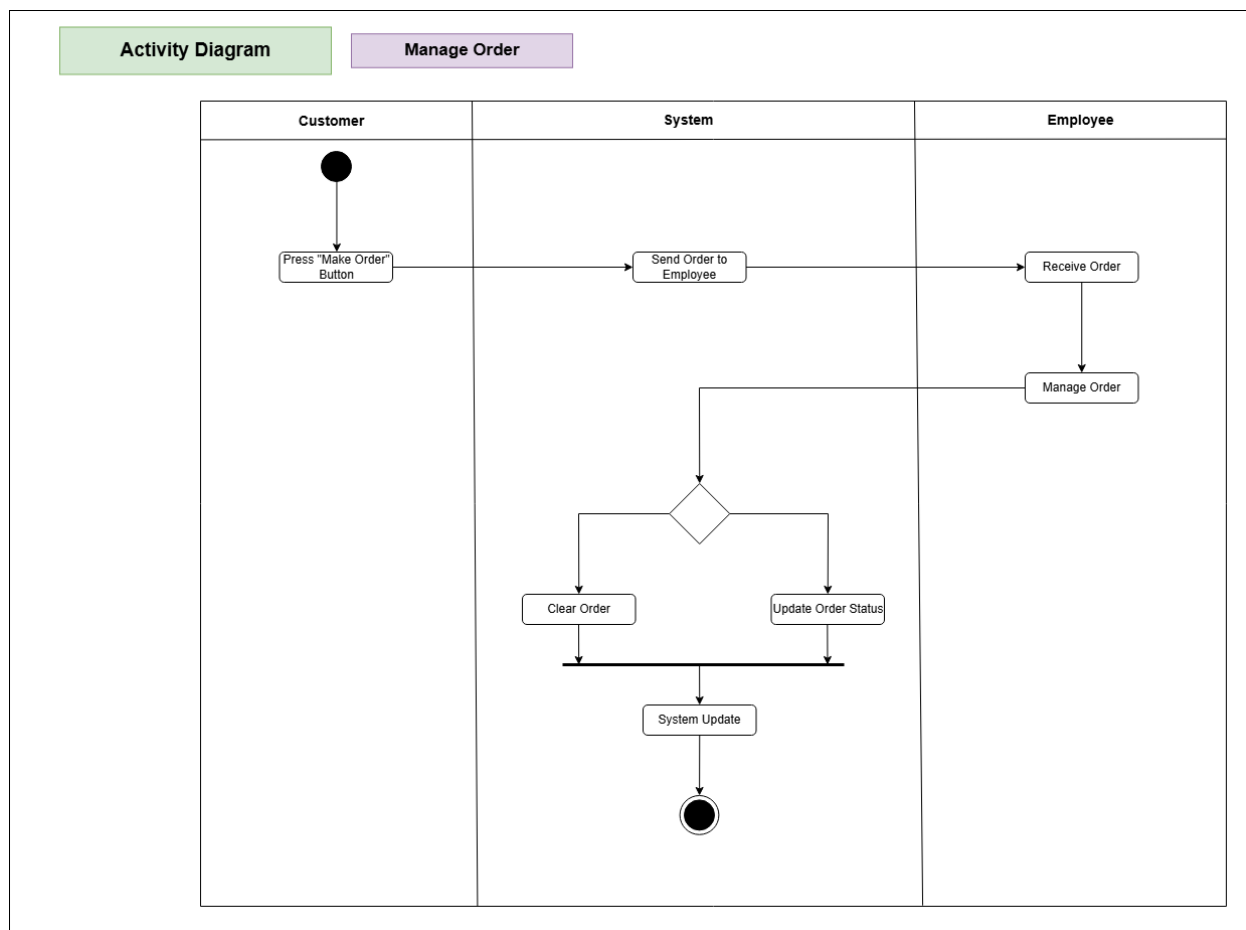
## แผนภาพกิจกรรม (Activity Diagram)



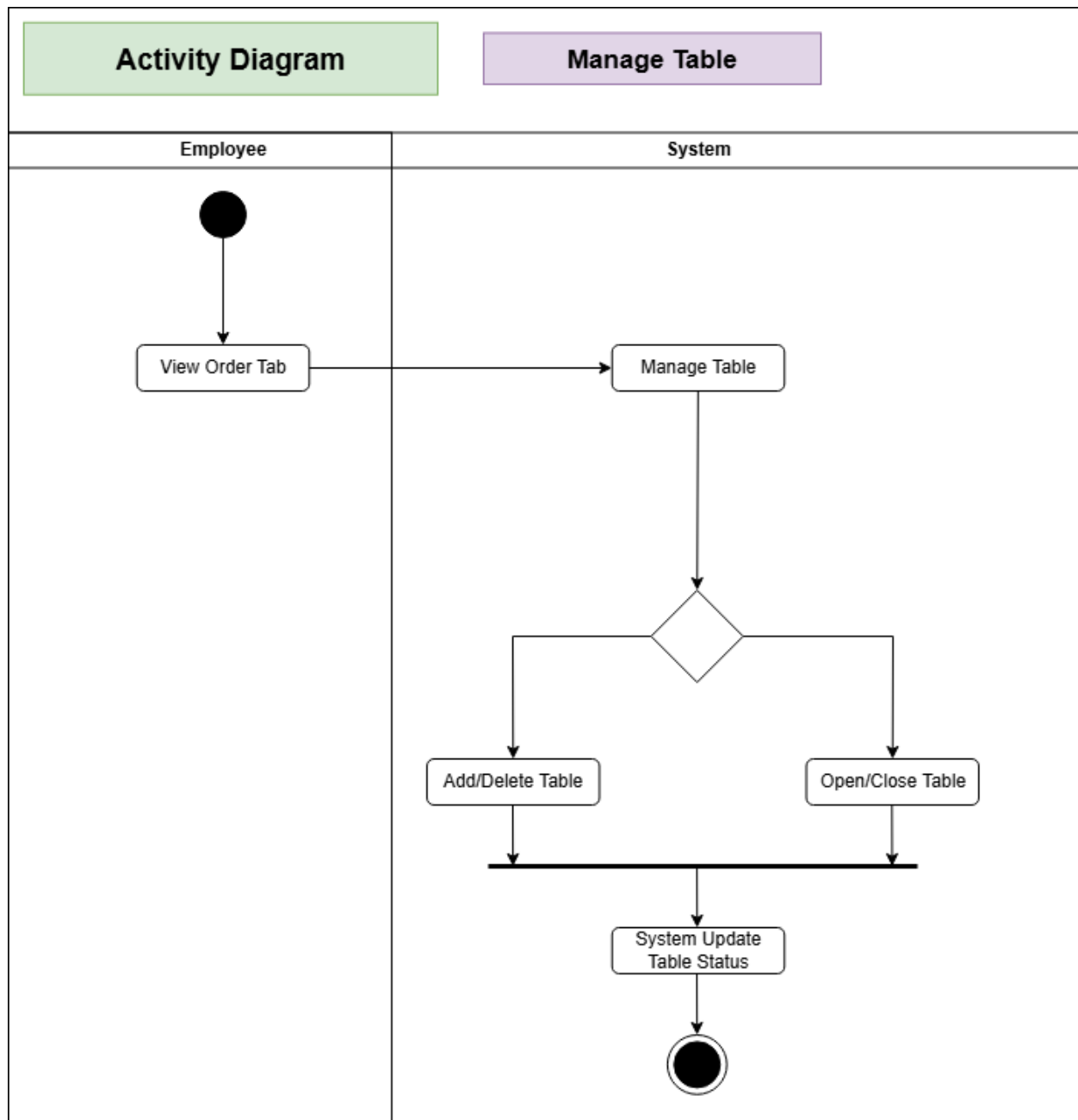
## แผนภาพกิจกรรม (Activity Diagram)



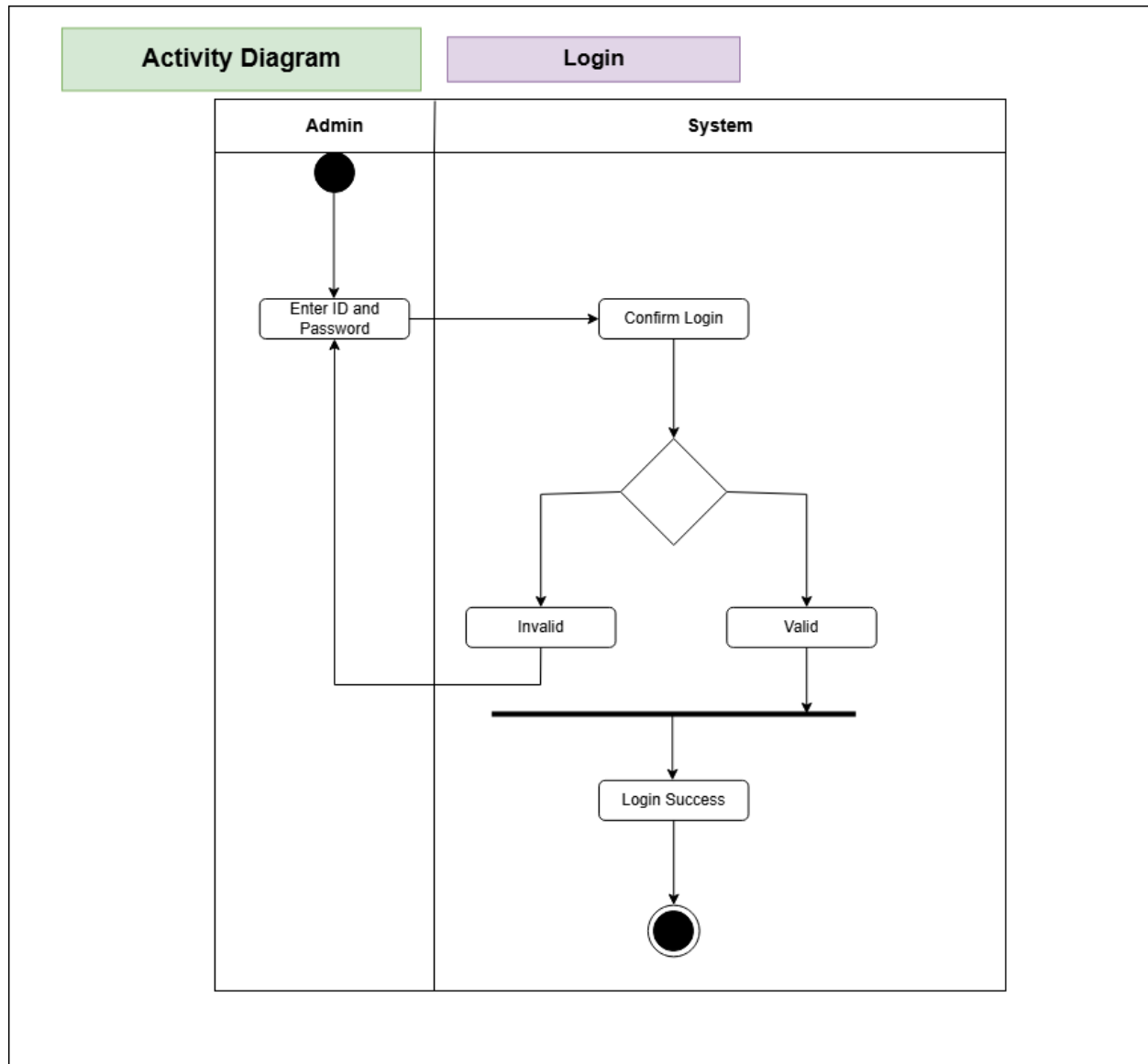
## แผนภาพกิจกรรม (Activity Diagram)



## แผนภาพกิจกรรม (Activity Diagram)

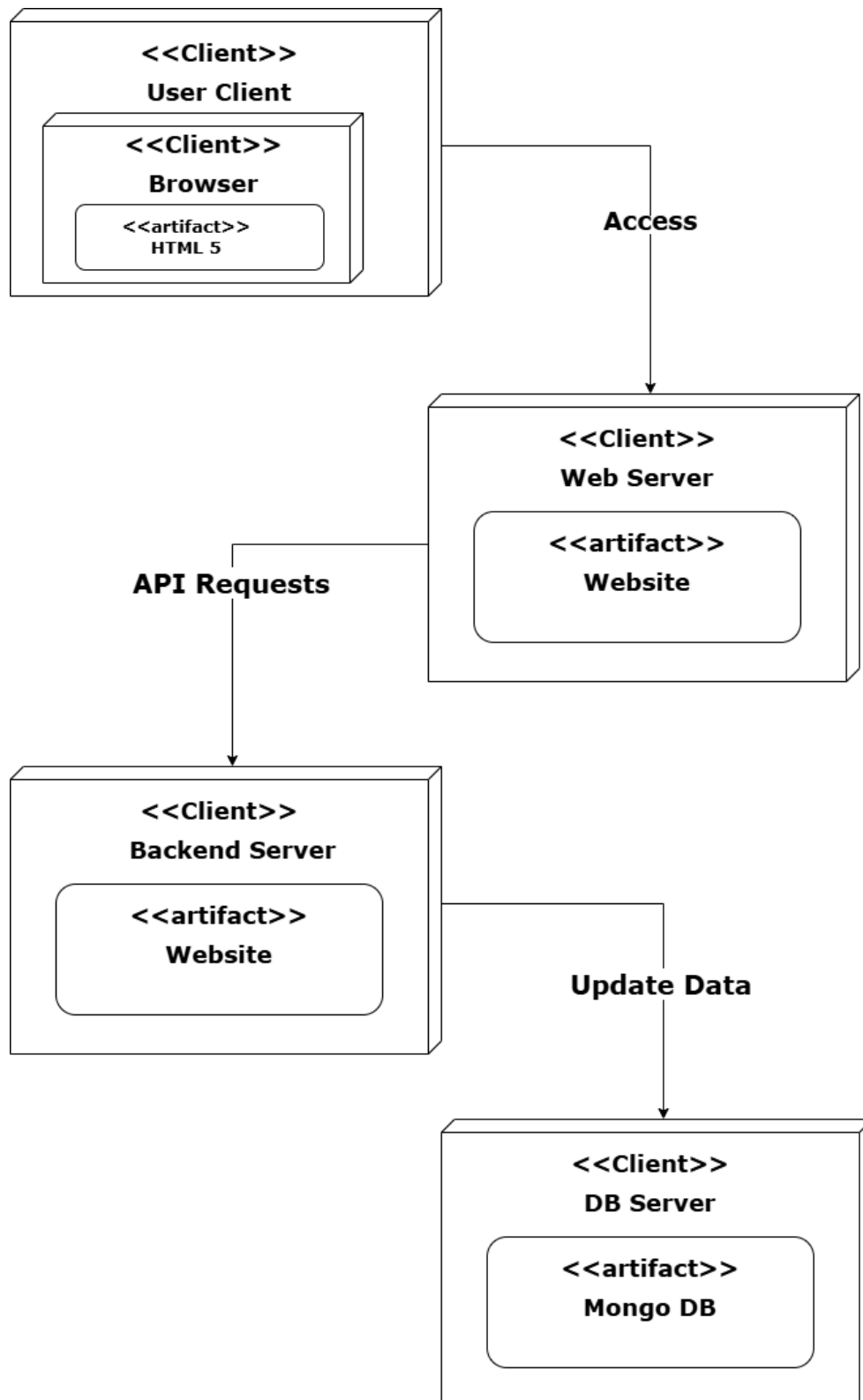


## แผนภาพกิจกรรม (Activity Diagram)

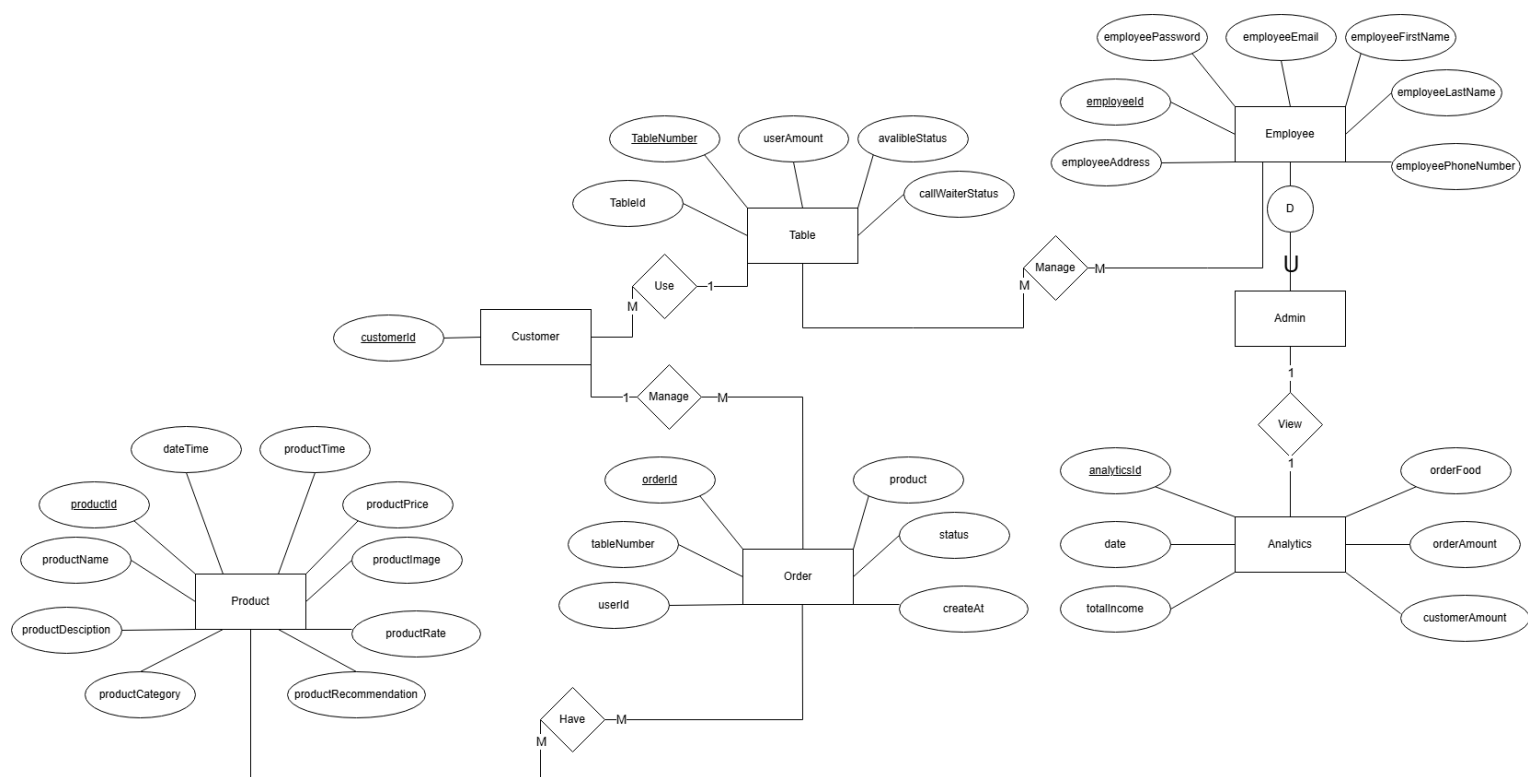




## แผนภาพการปรับใช้ (Deployment Diagram)



## แผนภาพความสัมพันธ์ของเอนทิตี (ER Diagram)



## User Acceptance Test

## Test Case 1: Admin Login

รหัสทดสอบ	ADMIN_LOG_01	วันที่ทดสอบ	28/3/2025
ระบบ/ส่วนงาน	การจัดการข้อมูลผู้ใช้งาน > ข้อมูลผู้ใช้งาน		
ชื่อการทำงาน	แอดมินล็อกอิน		
รหัสหน้าจอ-ชื่อหน้าจอ	ADM_LOG - ล็อกอินใช้งานในตำแหน่ง Admin		
คำอธิบาย	ทดสอบการเข้าสู่ระบบใน Account ของผู้ดูแลระบบ		
เงื่อนไขการทำงาน	อีเมลกับรหัส จะถูกกำหนดโดยผู้พัฒนาโปรเจ็คเท่านั้น		
ขั้นตอนการทำงาน	1. เข้าหน้าเว็บ QuickBites Dashboard 2. กรอก Email Address 3. กรอก Password 4. กดปุ่ม Login		
ผลที่คาดว่าจะได้รับ	เมื่อกรอกข้อมูลครบถ้วน และกดปุ่มล็อกอิน จะไปยังหน้า Home และขึ้นแจ้งว่า “Logged in as Admin”	ผลการทดสอบ:  ผ่าน	
การทดสอบการทำงานแบบไม่สมบูรณ์			
การกระทำ	ผลการทำงานที่คาดว่าจะได้รับ	ผลการทดสอบ	หมายเหตุ
1. กรอก Email Address โดยไม่มี “@”	จะแสดงตัวอักษรบนแถบขึ้นว่า “โปรดใส่ @ ในที่อยู่อีเมล”	ผ่าน	-
2. ไม่กรอกหรือกรอก Email Address ผิด	จะแสดงการแจ้งเตือนทางด้านบน ขาวว่า “Staff doesn't exist”	ผ่าน	-
3. กรอก Email Address ถูก แต่ใส่ Password ผิด	จะแสดงการแจ้งเตือนทางด้านบน ขาวว่า “Invalid admin credentials”	ผ่าน	-

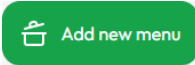


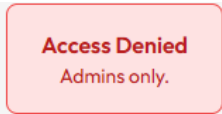
## Test Case 2: Employee Login

รหัสทดสอบ	EMP_LOG_01	วันที่ทดสอบ	28/3/2025
ระบบ/ส่วนงาน	การจัดการข้อมูลผู้ใช้งาน > ข้อมูลผู้ใช้งาน		
ชื่อการทำงาน	การล็อกอินของพนักงานร้าน		
รหัสหน้าจอ-ชื่อหน้าจอ	EMP_LOG - ล็อกอินใช้งานในตำแหน่ง Employee		
คำอธิบาย	ทดสอบการเข้าสู่ระบบใน Account ของพนักงานในร้าน		
เงื่อนไขการทำงาน	อีเมลกับรหัส จะถูกสร้างให้โดย Admin เท่านั้น		
ขั้นตอนการทำงาน	1. เข้าหน้าเว็บ QuickBites Dashboard 2. กรอก Email Address 3. กรอก Password 4. กดปุ่ม Login		
ผลที่คาดว่าจะได้รับ	เมื่อกรอกข้อมูลครบถ้วน และกดปุ่มล็อกอิน จะไปยังหน้า Home และขึ้นแจ้งว่า “Logged in as Employee”	ผลการทดสอบ: ผ่าน	
การทดสอบการทำงานแบบไม่สมบูรณ์			
การกระทำ	ผลการทำงานที่คาดว่าจะได้รับ	ผลการทดสอบ	หมายเหตุ
1. กรอก Email Address โดยไม่มี “@”	จะแสดงตัวอักษรบนแถบขึ้นว่า “โปรดใส่ @ ในที่อยู่อีเมล”	ผ่าน	-
2. ไม่กรอกหรือกรอก Email Address ผิด	จะแสดงการแจ้งเตือนทางด้านบน ขาวว่า “Staff doesn't exist”	ผ่าน	-
3. กรอก Email Address ถูก แต่ใส่ Password ผิด	จะแสดงการแจ้งเตือนทางด้านบน ขาวว่า “Invalid staff credentials”	ผ่าน	-

### Test Case 3: Show Menu (Staff)

รหัสทดสอบ	STAFF_MENU_01	วันที่ทดสอบ	28/3/2025
ระบบ/ส่วนงาน	การแสดงผลหน้าเว็บ > View menu		
ชื่อการทำงาน	การแสดงผลแถบเมนูของ QuickBites		
รหัสหน้าจอ-ชื่อหน้าจอ	1. MENU_SHOW1 – หน้าถึงหน้า View menu 2. MENU_SHOW2 – เลือกแสดงประเภทเมนู		
คำอธิบาย	ทดสอบการแสดงผลเมนูใน View Menu		
เงื่อนไขการทำงาน	ผู้ที่เข้าถึงหน้า View Menu จะต้องมียตำแหน่งเป็น Admin หรือ Employee เท่านั้น		
ขั้นตอนการทำงาน	1. เข้าสู่ระบบในตำแหน่ง Admin หรือ Employee 2. กดที่แถบ View Menu 3. กดที่แถบประเภท Menu		
ผลที่คาดว่าจะได้รับ	จะแสดงประเภทของเมนูอย่างถูกต้องและครบถ้วน	ผลการทดสอบ: ผ่าน	
การทดสอบการทำงานแบบไม่สมบูรณ์			
การกระทำ	ผลการทำงานที่คาดว่าจะได้รับ	ผลการทดสอบ	หมายเหตุ
1. กดซ้ำที่แถบประเภทเมนู	กลับมาแสดงเมนูทั้งหมด	ผ่าน	-

## Test Case 4: Menu Editing (Staff)

รหัสทดสอบ	STAFF_MENU_02	วันที่ทดสอบ	28/3/2025
ระบบ/ส่วนงาน	การแสดงผลหน้าเว็บ > Create menu		
ชื่อการทำงาน	การแก้ไขเมนูของ QuickBites		
รหัสหน้าจอ-ชื่อหน้าจอ	1. MENU_ADD – เพิ่มเมนูใหม่ 2. MENU_UPDATE – อัปเดตหรือแก้ไขเมนู 3. MENU_DELETE – ลบเมนู		
คำอธิบาย	ทดสอบการแสดงผลเมนูใน Create Menu		
เงื่อนไขการทำงาน	ผู้ที่เข้าถึงหน้า Create Menu จะต้องมีตำแหน่งเป็น Admin เท่านั้น		
ขั้นตอนการทำงาน	1. เข้าสู่ระบบในตำแหน่ง Admin 2. กดที่แถบ View Menu 3. เมื่อต้องการเพิ่มเมนู ให้กดที่ปุ่ม  หรือแถบ Create Menu 4.. กรอกรายละเอียดเมนู และแทรกรูป 5. กด Create 6. หากจะแก้ไขเมนูให้กดปุ่ม  เมื่อแก้ไขเสร็จให้กด Update 7. หากต้องการลบเมนูให้กดปุ่ม 		
ผลที่คาดว่าจะได้รับ	1. เมนูที่เพิ่มจะปรากฏอยู่ด้านล่างสุด 2. เมื่อแก้ไขข้อมูล ข้อมูลอัปเดตทันที 3. สามารถลบเมนูได้	ผลการทดสอบ:	
		ผ่าน	
		ผ่าน	
		ผ่าน	
การทดสอบการทำงานแบบไม่สมบูรณ์			
การกระทำ	ผลการทำงานที่คาดว่าจะได้รับ	ผลการทดสอบ	หมายเหตุ
1. กดปุ่ม Add new menu ใน Account Employee	ขึ้นหน้า 	ผ่าน	-



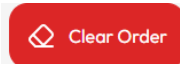
2. กดลบเมนู ใน Account Employee	แสดงแถบด้านบนขวาว่า “Not Authorized Login Again”	ผ่าน	-
3. ไม่กรอกชื่อหรือรายละเอียดเมนู ในระหว่าง Create Menu	จะแสดงแถบ “โปรดกรอกฟิลด์นี้”	ผ่าน	-
4. กรอกชื่อกับรายละเอียดเมนู แต่ไม่กรอกข้อมูลที่เป็นตัวเลขที่เหลือ	แสดงแถบด้านบนขวาว่า “Product validation failed”	ผ่าน	-

## Test Case 5: Table Management (Staff)




รหัสทดสอบ	STAFF_TABLE_01	วันที่ทดสอบ	28/3/2025
ระบบ/ส่วนงาน	การแสดงผลหน้าเว็บ > หน้าออเดอร์		
ชื่อการทำงาน	การแสดงผลแถบออเดอร์ของ QuickBites		
รหัสหน้าจอ-ชื่อหน้าจอ	1. TABLE_ADD – ปุ่มเพิ่มโต๊ะ 2. TABLE_DELETE - ลบโต๊ะ 3. TABLE_OPEN – เปิดการอนุญาตเข้าถึงหน้าเว็บสั่งอาหาร 4. TABLE_CLOSE – ปิดการอนุญาตเข้าถึงหน้าเว็บสั่งอาหาร		
คำอธิบาย	ทดสอบการจัดการโต๊ะ และรายการอาหารที่แต่ละโต๊ะสั่ง		
เงื่อนไขการทำงาน	ผู้ที่เข้าถึงหน้า Order จะต้องมียตำแหน่งเป็น Admin หรือ Employee เท่านั้น		
ขั้นตอนการทำงาน	1. เข้าสู่ระบบในตำแหน่ง Admin หรือ Employee 2. ไปที่แถบ Order 3. กดปุ่ม  เพื่อเปิดสิทธิ์การอนุญาตเข้าถึงลิงก์ของโต๊ะนั้น 4. กดปุ่ม  เพื่อปิดการอนุญาตเข้าถึงลิงก์ของโต๊ะนั้น 5. กดปุ่ม  เพื่อเพิ่มจำนวนโต๊ะ 6. พิมพ์หรือเลื่อนเลือกเลขของโต๊ะ จากนั้นกด Add table 7. หากต้องการลบโต๊ะให้กดปุ่ม  จากนั้นกด Confirm		
ผลที่คาดว่าจะได้รับ	1. เปิด-ปิด สิทธิ์การอนุญาตเข้าถึงลิงก์ได้ 2. เพิ่มจำนวนโต๊ะได้ 3. ลบโต๊ะได้	ผลการทดสอบ:	
		ผ่าน	
		ผ่าน	
		ผ่าน	
การทดสอบการทำงานแบบไม่สมบูรณ์			
การกระทำ	ผลการทำงานที่คาดว่าจะได้รับ	ผลการทดสอบ	หมายเหตุ
1. เพิ่มจำนวนโต๊ะ โดยใส่เลขโต๊ะที่มีอยู่แล้ว	แสดงแถบด้านบนขวาวว่า “Table already exists.”	ผ่าน	-



## Test Case 6: Order Management (Staff)

รหัสทดสอบ	STAFF_ORDER_01	วันที่ทดสอบ	28/3/2025
ระบบ/ส่วนงาน	การแสดงผลหน้าเว็บ > หน้าออเดอร์		
ชื่อการทำงาน	จัดการรายการอาหารที่สั่ง		
รหัสหน้าจอ-ชื่อหน้าจอ	1. ORDER_SHOW – แสดงออเดอร์ที่สั่ง 2. ORDER_UPDATE – เปลี่ยนสถานะของออเดอร์ 3. ORDER_CLEARED – ลบประวัติออเดอร์ทั้งหมด		
คำอธิบาย	อัปเดตสถานะของออเดอร์ที่สั่ง		
เงื่อนไขการทำงาน	1. ผู้ที่เข้าถึงหน้า Order จะต้องมีตำแหน่งเป็น Admin หรือ Employee เท่านั้น 2. ต้องมีลูกค้าสั่งอาหารเข้ามาก่อน		
ขั้นตอนการทำงาน	1. เข้าสู่ระบบในตำแหน่ง Admin หรือ Employee 2. ไปยังแถบ Order กดไปที่โต๊ะที่มีการสั่งอาหาร 3. กดไปที่  เพื่อดูว่ามีรายการที่สั่งอะไรบ้าง 4. กดปุ่ม  เพื่อเลือกเปลี่ยนสถานะออเดอร์ 5. เมื่อลูกค้าทานอาหารเสร็จและเช็คเอาท์เรียบร้อยแล้ว ให้กด 		
ผลที่คาดว่าจะได้รับ	1. แสดงเมนูที่สั่งในหน้า Table และ หน้า Home 2. อัปเดตสถานะออเดอร์ได้ 3. รีเซ็ตสถานะโต๊ะได้	ผลการทดสอบ:	
		ผ่าน	
		ผ่าน	
		ผ่าน	
การทดสอบการทำงานแบบไม่สมบูรณ์			
การกระทำ	ผลการทำงานที่คาดว่าจะได้รับ	ผลการทดสอบ	หมายเหตุ
1. กด Clear Order ตอน ที่ไม่มีออเดอร์ใน รายการ	แสดงแถบด้านบนขวาว่า “There are no orders to clear.”	ผ่าน	-







## Test Case 7: Employee Management (Admin)

รหัสทดสอบ	EMP_MANAGE_01	วันที่ทดสอบ	28/3/2025
ระบบ/ส่วนงาน	การจัดการข้อมูลผู้ใช้งาน > แก้ไขข้อมูลผู้ใช้งาน		
ชื่อการทำงาน	ทดสอบการจัดการพนักงานในร้าน		
รหัสหน้าจอ-ชื่อหน้าจอ	1. EMP_ADD – เพิ่มพนักงาน 2. EMP_UPDATE – แก้ไขพนักงาน 3.EMP_DELETE – ลบพนักงาน		
คำอธิบาย	จัดการพนักงานในร้าน		
เงื่อนไขการทำงาน	ผู้ที่เข้าถึงหน้า Employee list จะต้องมีตำแหน่งเป็น Admin เท่านั้น		
ขั้นตอนการทำงาน	1. เข้าสู่ระบบในตำแหน่ง Admin 2. ไปที่แถบ Employee list 3. กดปุ่ม  หรือแถบ Add employee เพื่อเพิ่มพนักงาน 4. กรอกรายละเอียด และแทรกรูป จากนั้นกด “Add Employee” 5. หากต้องการแก้ไขพนักงานให้กดปุ่ม  6. หากต้องการลบพนักงานให้กดปุ่ม 		
ผลที่คาดว่าจะได้รับ	1. เพิ่มพนักงานได้สำเร็จ 2. แก้ไขข้อมูลพนักงานได้ 3. ลบพนักงานได้	ผลการทดสอบ:	
		ผ่าน	
		ผ่าน	
		ผ่าน	
การทดสอบการทำงานแบบไม่สมบูรณ์			
การกระทำ	ผลการทำงานที่คาดว่าจะได้รับ	ผลการทดสอบ	หมายเหตุ
1. กรอกข้อมูลพนักงานไม่ครบถ้วน	จะแสดงแถบ “โปรดกรอกฟิลด์นี้”	ผ่าน	-

## Test Case 8: Customer Login


รหัสทดสอบ	CUS_LOG_01	วันที่ทดสอบ	29/3/2025
ระบบ/ส่วนงาน	การจัดการข้อมูลผู้ใช้งาน > ข้อมูลผู้ใช้งาน		
ชื่อการทำงาน	ลูกค้าล็อกอิน		
รหัสหน้าจอ-ชื่อหน้าจอ	CUS_LOG – ล็อกอินใช้งานแบบลูกค้า		
คำอธิบาย	ลูกค้าแสกน QR Code เพื่อเข้าเว็บสั่งอาหาร		
เงื่อนไขการทำงาน	ลูกค้าจะเข้าถึงได้ เมื่อพนักงานเปิดอนุญาตให้เข้าถึง		
ขั้นตอนการทำงาน	ลูกค้าต้องแสกน QR code สั่งอาหารของโต๊ะที่นั่ง		
ผลที่คาดว่าจะได้รับ	1. สามารถเข้าเว็บสั่งอาหารได้อย่างไม่มีปัญหา 2. แสดงลำดับ UserID ได้ถูกต้อง	ผลการทดสอบ:	
		ผ่าน	
		ผ่าน	
การทดสอบการทำงานแบบไม่สมบูรณ์			
การกระทำ	ผลการทำงานที่คาดว่าจะได้รับ	ผลการทดสอบ	หมายเหตุ
1. ลูกค้าแสกนเข้าถึงตอนที่พนักงานยังไม่เปิดสิทธิ์	หน้าเว็บจะแสดงว่า “The table is currently unavailable. Please try again later”	ผ่าน	-

## Test Case 9: Customer Ordering





รหัสทดสอบ	CUS_ORDER_01	วันที่ทดสอบ	29/3/2025
ระบบ/ส่วนงาน	สั่งอาหาร > มุมมองลูกค้า		
ชื่อการทำงาน	Order ลูกค้า		
รหัสหน้าจอ-ชื่อหน้าจอ	1. CUS_MENU_SHOW – แสดงข้อมูลเมนู 2. CUS_MENU_SEARCH – ค้นหาเมนู 3. CUS_MENU_TYPE – เลือกประเภทเมนู 4. CUS_ORDER_ADD – เพิ่มการสั่งเมนู 5. CUS_ORDER_DELETE – ลบเมนูที่สั่ง 6. CUS_ORDER_CONFIRM – ยืนยันการสั่งอาหาร		
คำอธิบาย	ลูกค้าสั่งอาหารในหน้าเว็บ		
เงื่อนไขการทำงาน	ลูกค้าจะสั่งอาหารในลิงก์ได้ เมื่อพนักงานเปิดอนุญาตให้เข้าถึง		
ขั้นตอนการทำงาน	1. ลูกค้าต้องสแกน QR code สั่งอาหารของโต๊ะที่นั่ง 2. กดดูเมนู 3. กดเพิ่มหรือลดปุ่ม   จากนั้นกด Add to cart 4. จะปรากฏ แถบ Cart ด้านล่าง เมื่อกดไปดูจะแสดงอาหารที่สั่ง 5. ยังสามารถแก้ไขได้โดยกด   ปุ่ม  +,- 6. เมื่อยืนยันจะสั่ง ให้กดปุ่ม 7. ถ้าต้องการลบเมนูที่สั่งให้กดปุ่ม 		
ผลที่คาดว่าจะได้รับ	1. สามารถเพิ่มเมนูที่จะสั่งได้ 2. สามารถลบเมนูที่สั่งได้ 3. สามารถยืนยันแสดงบันทึกเมนูที่สั่งได้		ผลการทดสอบ:
			ผ่าน
			ผ่าน
			ผ่าน
การทดสอบการทำงานแบบไม่สมบูรณ์			

การกระทำ	ผลการทำงานที่คาดว่าจะได้รับ	ผลการทดสอบ	หมายเหตุ
1. พิมพ์ตัวอักษรในแถบ Search	แสดงเมนูที่ประกอบด้วยตัวอักษรนั้น	ผ่าน	-
2. ทำให้จำนวนที่สั่งเป็น 0 แล้วกด Add to cart	ไม่ปรากฏแถบออเดอร์ที่ด้านล่าง	ผ่าน	-
3. กดที่แถบประเภทเมนู	แสดงประเภทเมนูได้ถูกต้อง	ผ่าน	-
4. กรอกข้อมูลในแถบ Special Instructions	ข้อมูลจะปรากฏใน Additional :	ผ่าน	-

## Test Case 10: Customer Payment

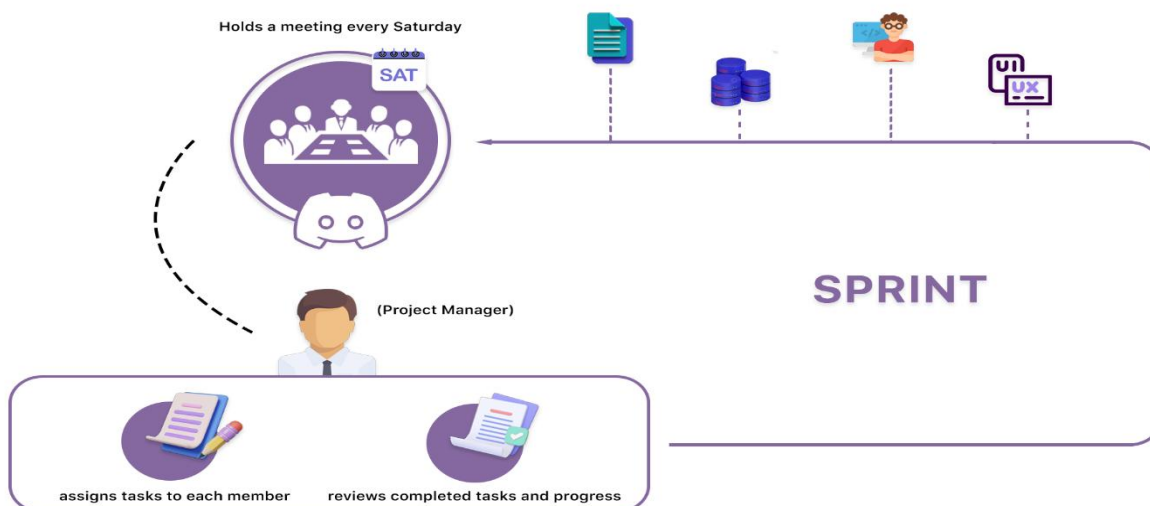
รหัสทดสอบ	CUS_PAY_01	วันที่ทดสอบ	29/3/2025
ระบบ/ส่วนงาน	ระบบชำระเงิน > มุมมองลูกค้า		
ชื่อการทำงาน	ลูกค้าจ่ายเงิน		
รหัสหน้าจอ-ชื่อหน้าจอ	1. CUS_ORDER_SUM – แสดงบันทึกเมนูที่สั่ง 2. CUS_CHECK – เข้าสู่หน้าสแกนจ่ายเงิน 3. CUS_CHECK_CONFIRM – แสดงหน้าเว็บขอบคุณ		
คำอธิบาย	ลูกค้าสแกนจ่ายเงินในหน้าเว็บ		
เงื่อนไขการทำงาน	ลูกค้าจะชำระเงินได้ เมื่อพนักงานเปิดอนุญาตให้เข้าถึงเว็บสั่งอาหาร และได้มีรายการที่สั่งไว้แล้ว		
ขั้นตอนการทำงาน	1. ลูกค้าต้องอยู่ในหน้า Order Summary แล้วกดที่แถบ <div><div>1 Order</div><div>Check Bill</div><div>฿ 210.00</div></div> แต่ไม่อยู่ให้กดปุ่ม  2. จะแสดง Payment Details และ QR Code 3. กด <div>Confirm Payment</div>		
ผลที่คาดว่าจะได้รับ	1. แสดงยอดรวมรายการถูกต้อง 2. ไปยังหน้าเว็บขอบคุณ 3. ออกจากระบบอัตโนมัติ	ผลการทดสอบ:	
		ผ่าน	
		ผ่าน	
		ผ่าน	
การทดสอบการทำงานแบบไม่สมบูรณ์			
การกระทำ	ผลการทำงานที่คาดว่าจะได้รับ	ผลการทดสอบ	หมายเหตุ
1. ลูกค้าใช้แอปธนาคาร สแกนเข้า QR Code ในหน้า Payment Details	เข้าไปหน้าโอนเงินและแสดงราคา ถูกต้อง	ผ่าน	-

## Test Case 11: Notification

รหัสทดสอบ	NOTI_01	วันที่ทดสอบ	29/3/2025
ระบบ/ส่วนงาน	การจัดการแจ้งเตือน		
ชื่อการทำงาน	แจ้งเตือนระบบ		
รหัสหน้าจอ-ชื่อหน้าจอ	1. NOTI_CUS – ลูกค้ากดเรียกพนักงาน 2. NOTI_STAFF – แจ้งเตือนที่ส่งไปยังพนักงาน		
คำอธิบาย	แสดงการแจ้งเตือนของระบบ		
เงื่อนไขการทำงาน	ต้องมีลูกค้ามีกดปุ่มเรียกพนักงาน และ ลิงก์สั่งอาหารต้องเปิดอนุญาต		
ขั้นตอนการทำงาน	1. ลูกค้าสแกน QR code 2. ให้ลูกค้ากดที่ปุ่ม 		
ผลที่คาดว่าจะได้รับ	1. แสดงแถบด้านบนว่า “A waiter has been notified for table.....”  2. มีการแจ้งเตือนส่งไปยังหน้าเว็บของพนักงาน	ผลการทดสอบ:	
		ผ่าน	
		ผ่าน	
การทดสอบการทำงานแบบไม่สมบูรณ์			
การกระทำ	ผลการทำงานที่คาดว่าจะได้รับ	ผลการทดสอบ	หมายเหตุ
1. ในหน้าเว็บพนักงานให้กดปุ่ม 	แสดงโต๊ะที่แจ้งเตือนถูกต้อง	ผ่าน	-
2. ในหน้าเว็บพนักงานให้เข้าไปที่แถบ Order	จะแสดงรูป  บนโต๊ะที่มีการแจ้งเตือน	ผ่าน	-
3. พนักงานกดที่ 	แสดงแถบด้านบนขวาว่า “The waiter has attended to the call for table.....”	ผ่าน	-

## Process Model

เราจะใช้ Agile Process Model ในการเก็บข้อมูลและFeedback นำกลับมาพัฒนาและปรับปรุงเป็นรอบๆสั้นๆ สามารถปรับเปลี่ยนได้เมื่อต้องการโดยไม่ต้องทำการทดสอบซอฟต์แวร์ทุกครั้ง



โดยสำหรับ Quickbites จะมีขั้นตอนใน Agile Model ดังนี้

### 1. Sprint Planning (วางแผน Sprint)

- กำหนดเป้าหมายของ Sprint
- แบ่งงานเป็น Task ย่อย ๆ

### 2. Weekly Scrum (สแตนด์อัปทุกสัปดาห์)

- ใช้เวลาประมาณ 15 นาทีต่อสัปดาห์ (Weekly Standup)
- สมาชิกแต่ละคนตอบคำถาม 3 ข้อ 1. ทำอะไรไปแล้วเมื่อสัปดาห์ก่อน? 2. วันนี้จะทำอะไรต่อ ? 3. มีปัญหาอะไรที่ต้องการความช่วยเหลือ?

### 3. Development & Testing (พัฒนาและทดสอบ)

- ออกแบบและพัฒนาระบบของซอฟต์แวร์ตามความต้องการที่ได้รับมาจากขั้นตอนก่อนๆ
- ทดสอบโค้ดให้แน่ใจว่าไม่มีบั๊ก

### 4. Sprint Review (รีวิวและประเมิน)

- รีวิวและประเมินสิ่งที่ทำใน Sprint ครั้งนี้ แล้วกำหนดเป้าหมายที่จะทำและปรับปรุงใน Sprint ของสัปดาห์ถัดไป



## Risk Management

หมายเลข	ประเภทความเสี่ยง	รายละเอียดของความเสี่ยง	ความน่าจะเป็น	ระดับผลกระทบ
1	Project	การสื่อสารไม่ชัดเจน	ค่อนข้างสูง	ต่ำ
2	Project	สมาชิกขาดการเข้าประชุม	ปานกลาง	ต่ำ
3	Project	การบริหารเวลาไม่ดี	ค่อนข้างต่ำ	ปานกลาง
4	Product	เครือข่ายขัดข้อง	ค่อนข้างต่ำ	ปานกลาง
5	Product	ความรู้ด้านการพัฒนาซอฟต์แวร์ ไม่เพียงพอ	ค่อนข้างสูง	สูง
6	Business	ความคล้อยคลึงของระบบภายใน ตลาด	ปานกลาง	ปานกลาง
7	Product	เซิร์ฟเวอร์ไม่ให้บริการ	ค่อนข้างต่ำ	สูง
8	Project	ภัยพิบัติ เช่น แผ่นดินไหว	ค่อนข้างต่ำ	สูง

## Risk Planning

ปัจจัยกระตุ้นความเสี่ยง	แผนการรับมือ
การสื่อสารไม่ชัดเจน ทำให้เกิดความเข้าใจผิด ขณะทำงาน	กำหนดช่องทางการสื่อสารที่ชัดเจน เช่น ใช้แอป พลิเคชันสำหรับการสื่อสารกลุ่ม สรุปประเด็น สำคัญในที่ประชุมทุกครั้ง
สมาชิกขาดการเข้าประชุม ทำให้โครงการไม่ คืบหน้า	กำหนดกฎระเบียบที่ชัดเจน ส่งการแจ้งเตือน และเตือนความสำคัญ จัดประชุมสั้นๆ หรือ ยืดหยุ่น
การบริหารเวลาไม่ดี ทำให้เกิดการล่าช้าหรือ ไม่ได้ทำงานให้เสร็จตามกำหนด	การกำหนดกรอบเวลาที่ชัดเจนสำหรับแต่ละงาน และการติดตามความคืบหน้าเป็นระยะ
ความรู้ด้านการพัฒนาซอฟต์แวร์ไม่เพียงพอ ทำ ให้ซอฟต์แวร์มีข้อผิดพลาด ประสิทธิภาพต่ำและ ความช้าในการทำงาน	ฝึกอบรมและพัฒนาทักษะ ทำการทดสอบอย่าง สม่ำเสมอ แจกจ่ายงานให้เหมาะสมกับความ ถนัดแต่ละบุคคล
ความคล้ายคลึงของระบบภายในตลาด ทำให้ การแข่งขันสูงขึ้น ซอฟต์แวร์ขาดความแตกต่าง	สร้างความแตกต่าง ตอบสนองและปรับปรุงตาม Feedback เน้นการตลาดที่เจาะกลุ่มเป้าหมาย ออกแบบ UX/UI ที่ดีที่สุด
เซิร์ฟเวอร์ไม่ให้บริการ เกิดการหยุดชะงักของการ ให้บริการระบบและอาจเกิดการสูญเสียข้อมูล	มีแผนสำรอง ทดสอบระบบสำรองและแผนฟื้นฟู ให้แน่ใจว่าเซิร์ฟเวอร์สำรองสามารถใช้งานได้ใน กรณีที่เซิร์ฟเวอร์หลักหยุดทำงาน
ภัยพิบัติ เช่น แผ่นดินไหว อาจทำให้เกิด ความไม่ต่อเนื่องของการทำงาน อุปกรณ์ เสียหาย	Backup ข้อมูลไว้เสมอ ติดตามข่าวสารการแจ้ง เตือนภัยพิบัติอย่างบ่อยครั้ง และปฏิบัติตาม กฎระเบียบอย่างเคร่งครัด








## Version Control

ภายในทีม จะมีการแบ่งหน้าที่ให้แต่ละคนพัฒนาคอมโพเนนต์ต่าง ๆ และบางครั้งก็จะมีการช่วยกันพัฒนาคอมโพเนนต์เดียวกัน ซึ่งจะมีการอัปเดตเวอร์ชันของคอมโพเนนต์นั้นเรื่อย ๆ และเขียนรายละเอียดเอาไว้ว่ามีการปรับเปลี่ยน แก้ไข คอมโพเนนต์ดังกล่าว ตรงไหนบ้าง

### ตัวอย่าง













- การพัฒนาคอมโพเนนต์ที่เื้อร่วมกัน

### รายละเอียด

<b>Table V0.6</b>  6604062630251 • 4 days ago	- Added Delete Confirmation Popup Window
<b>Fix show userID</b>  RA9T0R • 16 days ago	
<b>Table V0.5</b>  RA9T0R • 16 days ago	add noti call waiter
<b>Table V0.4</b>  6604062630251 • 16 days ago	- Fix Bugs: <ul style="list-style-type: none"> <li>• Preventing users from accessing unavailable table</li> <li>• Preventing the database to add invalid table</li> </ul>
<b>Table V0.3</b>  6604062630251 • 16 days ago	- Implements Cookies
<b>Table V0.2</b>  RA9T0R • 16 days ago	
<b>Table V0.1</b>  RA9T0R • 16 days ago	

- การพัฒนาระบบ Admin Dashboard

## รายละเอียด

<b>Admin Dashboard 0.08</b>  6604062630251 • 23 days ago	
<b>Analytics</b>  RA9TOR • 23 days ago	
<b>Admin Dashboard V0.07</b>  6604062630251 • 23 days ago	
<b>Home</b>  RA9TOR • 23 days ago	
<b>Analytics</b>  RA9TOR • 24 days ago	
<b>Admin Dashboard V0.06</b>  6604062630251 • last month	
<b>Order manage</b>  RA9TOR • last month	
<b>Admin Dashboard V0.05</b>  6604062630251 • last month	
<b>Admin-Dashboard V0.04</b>  6604062630251 • last month	
<b>Merge branch 'main' of <a href="https://github.com/RA9TOR/QuickBites">https://github.com/RA9TOR/QuickBites</a></b>  6604062630251 • last month	
<b>UPDATE</b>  RA9TOR • last month	
<b>Admin Dashboard V0.03</b>  6604062630251 • last month	

- Login Page New Layout (V2)

- Adjust Edit Menu Page to be as the same as Create Menu Page

- Bug Fixes: Error Handling  
 - Software Testing:  
   • Add Employee Invalid Input  
   • Server Error 400

- Create Menu V1: layout is fully fixed.

- Create Menu -> Ready for Deployment (V.1)

- Create Menu Layout Fix

## QA (Quality Assurance)

No	Name	Type	Test case	Status
1.	Manage User	Functional	1. สามารถเพิ่มข้อมูลพนักงานได้	ผ่าน
			2. สามารถลบข้อมูลพนักงานได้	ผ่าน
			3. สามารถแก้ไขข้อมูลพนักงานได้	ผ่าน
			4. ระบบสามารถตรวจสอบความถูกต้องใน Text field ก่อนยืนยันได้	ผ่าน
2.	Manage Table	Functional	1. สามารถเพิ่มโต๊ะอาหารได้	ผ่าน
			2. สามารถลบโต๊ะอาหารได้	ผ่าน
			3. สามารถเปิด-ปิดการอนุญาตเข้าถึงลิงก์โต๊ะอาหารได้	ผ่าน
			4. ได้รับการแจ้งเตือนจากโต๊ะที่ลูกค้าเรียก	ผ่าน
3.	Manage Menu	Functional	1. สามารถแสดงแถบรายละเอียดเมนูได้	ผ่าน
			2. สามารถเพิ่มข้อมูลเมนูอาหารได้	ผ่าน
			3. สามารถลบข้อมูลเมนูอาหารได้	ผ่าน
			4. สามารถแก้ไขข้อมูลเมนูอาหารได้	ผ่าน
			5. ระบบสามารถตรวจสอบความถูกต้องใน Text field ก่อนยืนยันได้	ผ่าน
4.	Manage Order	Functional	1. สามารถบันทึกรายการอาหารที่สั่ง	ผ่าน
			2. สามารถอัปเดตสถานะเมนูได้	ผ่าน
5.	Login	Functional	1. สามารถล็อกอินเพื่อใช้งานระบบได้	ผ่าน
			2. ระบบสามารถตรวจสอบถูกต้องใน Text field ก่อนเข้าสู่ระบบ	ผ่าน
			3. แจ้งเตือนเมื่อเข้าสู่ระบบผิดพลาด	ผ่าน
			4. จำแนกตำแหน่ง Admin และ Employee ได้	ผ่าน
6.	Order	Functional	1. สามารถเพิ่มรายการไปยัง Cart ได้	ผ่าน
			2. เพิ่ม-ลดจำนวนเมนูได้	ผ่าน

7.	Cart	Functional	1. สามารถเพิ่ม-ลดจำนวนรายการที่อยู่ใน Cart ได้	ผ่าน
			2. สามารถลบรายการเมนูออกจาก Cart ได้	ผ่าน
			3. สามารถยืนยันการสั่งอาหารได้	ผ่าน

No	Name	Type	Test case	Status
8.	QR Code Scan	Functional	1. สามารถสแกน QR code เพื่อเข้าถึงสั่งอาหารได้	ผ่าน
			2. สามารถสแกน QR code เพื่อชำระเงินได้	ผ่าน
9.	Sales Analytic	Functional	1. สามารถแสดงยอดสั่ง จำนวนลูกค้า และ ยอดขายได้	ผ่าน
			2. สามารถแสดงกราฟยอดขายในรูปแบบ รายวัน, รายเดือน และรายปี	ผ่าน
10.	Notifications	Functional	1. แสดงการแจ้งเตือนเมื่อลูกค้ากดเรียกพนักงาน	ผ่าน
			2. สามารถกดปุ่มตอบรับได้	ผ่าน
11.	Performance	Non-Functional	1. เมื่อมีลูกค้าจำนวนมาก ระบบสามารถให้บริการได้ไม่ลดความเร็ว	ไม่ผ่าน
12.	Security	Non-Functional	1. สามารถตรวจสอบผู้มีสิทธิ์ใช้งานระบบได้	ผ่าน
			2. ป้องกันการเข้าถึงโดยไม่ได้รับอนุญาต	ผ่าน
13.	Usability	Non-Functional	1. UI สามารถใช้งานได้ง่าย	ผ่าน
			2. ใช้งานได้ทั้งในมือถือ และคอมพิวเตอร์	ผ่าน
14.	Maintainability	Non-Functional	1. สามารถปรับปรุงระบบได้โดยไม่ต้องระงับการบริการ	ไม่ผ่าน
15.	Serviceability	Non-Functional	1. ซ่อมแซมระบบได้รวดเร็วเมื่อเกิดปัญหา	ไม่ผ่าน
16.	System efficiency	Non-Functional	1. สามารถจัดการปริมาณข้อมูลได้เป็นอย่างดี เช่น รายการอาหารของ Table, อาหารที่ได้รับความนิยม	ผ่าน