



## Software Architecture (207)

ระบุ Architectural Drivers สำหรับโครงการทีม  
(LAB 2)

สมาชิกกลุ่มที่ 8

รหัสนักศึกษา 67543210017-9 ชื่อนายไตรภพ กองใจ

รหัสนักศึกษา 67543210020-3 ชื่อนายนัฐพงศ์ รอดอินทร์

รหัสนักศึกษา 67543210016-1 ชื่อนายชยานันท์ เพชรรักษ์

รหัสนักศึกษา 67543210045-0 ชื่อนายวีราวัชร์ สุประดิษฐ์พงศ์

## 1. System Description

Food Delivery System เป็นระบบเว็บไซต์ (และสามารถต่อยอดเป็น Web/Mobile App) ที่ช่วยให้ผู้ใช้งานสามารถค้นหาร้านอาหาร เมนูอาหาร เปรียบเทียบราคา ค่าจัดส่ง เวลาในการจัดส่ง และทำการสั่งอาหารแบบ Real-time

ระบบรวมการแสดงผลเมนูที่พร้อมขาย การคำนวนราคา/ค่าจัดส่ง การชำระเงินออนไลน์ การติดตามสถานะออเดอร์ และการออกเอกสารยืนยันการสั่งซื้อไว้ในระบบเดียว

ระบบถูกพัฒนาขึ้นเพื่อแก้ปัญหาการสั่งอาหารแบบเดิมที่ต้องโทรศัพท์ร้านเองหลายร้าน ตรวจสอบเมนู/ราคา/เวลาเปิด–ปิดทีละที่ รวมถึงช่วยลด Manual Work ของร้านอาหาร เช่น การรับออเดอร์ทางโทรศัพท์ จดด้วยกระดาษ อัปเดตราคา/เมนู/สต็อกเอง และการเช็คสถานะออเดอร์ด้วยมือ กลุ่มผู้ใช้หลัก

- Customer / End User – ผู้ค้นหาและสั่งอาหาร (เดลิเวอรี่หรือรับเองที่ร้าน)
- Restaurant Manager / Owner – ผู้ดูแลข้อมูลร้าน เมนู ราคา โปรโมชัน และออเดอร์
- Restaurant Staff – พนักงานหน้าร้านที่รับออเดอร์จากระบบ เตรียมอาหาร และอัปเดตสถานะ
- Delivery Rider – ผู้รับงานสั่งอาหารและอัปเดตสถานะการจัดส่ง
- System Administrator (Admin) – ผู้ดูแลระบบกลาง สิทธิ์การใช้งาน การตั้งค่า และ Monitoring

## 2. Target Users

### 1. Customer

- ผู้ค้นหาและสั่งอาหารจากร้านต่าง ๆ
- ต้องการความสะดวก ความรวดเร็ว และการติดตามสถานะออเดอร์แบบ Real-time

### 2. Restaurant Manager/Owner

- ผู้ดูแลข้อมูลร้าน เมนู ราคา เวลาเปิด–ปิด โปรโมชัน
- ต้องการระบบจัดการออเดอร์ที่ชัดเจน ลดความผิดพลาด และดูยอดขายได้ง่าย

### 3. Restaurant Staff

- ผู้ที่รับผิดชอบการรับออเดอร์จากระบบ เตรียมอาหาร และแจ้งสถานะอาหาร
- ต้องการหน้า Dashboard ที่ใช้งานง่ายและเห็นออเดอร์ใหม่แบบ Real-time

### 4. Delivery Rider (ห้ามระบบร่วมฝั่งไรเดอร์ด้วย)

- ผู้รับงานสั่งอาหาร ดูรายละเอียดออเดอร์ สั่นทางจัดส่ง และอัปเดตว่า Delivered แล้ว

### 5. System Administrator

- ดูและการตั้งค่าระบบกลาง, Roles/Permissions, Monitoring, Reporting, Security Policy

### 3. Key Features

- Restaurant & Menu Search
- Real-time Availability (ร้านเปิด/ปิด เมนูที่พร้อมขาย/หมดสต็อก)
- Online Food Ordering (Delivery & Pickup)
- Payment Integration (บัตร/โอน/E-Wallet/COD ตามที่กำหนด)
- Order Management (View / Track / Modify / Cancel ตามเงื่อนไข)
- Real-time Order Tracking (เตรียมอาหาร – กำลังจัดส่ง – ส่งสำเร็จ)
- User Profile & Order History
- Restaurant Management Dashboard
- Review & Rating System (รีวิวอาหาร/ร้าน/ไรเดอร์)
- Email / In-app Notifications
- Reporting & Analytics (ยอดขาย ยอดออเดอร์ ช่วงเวลา Peak และฯ)

#### Functional Requirements

##### User Management

- FR-01: ระบบรองรับการลงทะเบียน (Register) ผ่าน Email & Password (และสามารถต่อโยด Social Login ได้ในอนาคต)
- FR-02: รองรับ Login/Logout สำหรับ Customer, Restaurant Staff/Manager, Rider และ Admin
- FR-03: จัดการ User Roles และ Permissions ตามประเภทผู้ใช้ (Customer / Restaurant / Rider / Admin)
- FR-04: ผู้ใช้สามารถแก้ไขข้อมูลโปรไฟล์ (ชื่อ, เบอร์โทร, ที่อยู่จัดส่งหลัก, รูปโปรไฟล์)
- FR-05: รองรับการรีเซ็ต Email & Password (Forgot Password / Reset Link ทาง Email)

##### Restaurant & Food Search / Ordering

- FR-06: ค้นหาร้านอาหาร/เมนู ตาม:  
พิกัด/พื้นที่จัดส่ง (Location / Delivery Area)
  - ประเภทอาหาร (Cuisine) เช่น ไทย จีน ญี่ปุ่น อาหารตามสั่ง
  - ช่วงราคา และเวลาเปิด–ปิด
- FR-07: แสดงเฉพาะร้านที่พร้อมให้บริการ ในช่วงเวลาปัจจุบัน (ร้านเปิด + มีบริการส่งในพื้นที่)
- FR-08: Filter ผลการค้นหา ตาม:

- ราคาอาหาร
  - ค่าจัดส่ง
  - เวลาโดยประมาณในการจัดส่ง
  - คะแนนรีวิว
  - โปรโมชัน / คูปองส่วนลด
- 
- FR-09: แสดงรายละเอียดร้าน/เมนูอาหาร เช่น:
    - รูปภาพเมนู
    - รายละเอียดส่วนประกอบเบื้องต้น (Ingredient/Description)
    - ราคา
    - สถานะเมนู (พร้อมขาย / หมด)
    - รีวิวและคะแนน
  - FR-10: ลูกค้าสามารถ:
    - เลือกเมนูเพิ่มลงใน Cart
    - ระบุจำนวน / ตัวเลือก (เช่น ไม่ใส่เผ็ด เพิ่มไข่ดาว ฯลฯ)
    - กรอกข้อมูลที่อยู่จัดส่ง เบอร์ติดต่อ คำสั่งพิเศษ (Notes)
    - ทำการยืนยันสั่งอาหาร (Place Order)

### 3. Payment & Confirmation

- FR-11: รองรับการชำระเงินออนไลน์ผ่าน:
  - บัตรเครดิต/เดบิต
  - E-wallet
  - ช่องทางอื่น ๆ ตามที่ Payment Gateway รองรับ  
(อาจรองรับ Cash on Delivery / จ่ายปลายทาง เป็นตัวเลือกเพิ่มเติม)
- FR-12: ส่ง Email Confirmation และ/หรือ In-app Notification หลังการชำระเงินสำเร็จ พร้อม Order ID และรายละเอียดออเดอร์
- FR-13: ระบบสร้าง QR Code หรือ Order Code สำหรับ:
  - ยืนยันอodeอร์กรณีลูกค้ารับเงื่อนที่ร้าน (Pickup)
  - ให้ Rider แสดง/ยืนยันกับลูกค้าในตอนส่งมอบอาหาร

#### 4. Order Management

- FR-14: ลูกค้าสามารถดูประวัติการสั่งอาหารทั้งหมด:
  - ออเดอร์ที่ส่งไปแล้ว
  - ออเดอร์ที่กำลังดำเนินการอยู่
  - ออเดอร์ที่สำเร็จ/ยกเลิกแล้ว
- FR-15: ลูกค้าสามารถยกออเดอร์ได้ ภายใต้เงื่อนไขของระบบ/ร้าน (เช่น ก่อนร้านกด “กำลังปั่นอาหาร”)
- FR-16: ลูกค้าสามารถแก้ไขออเดอร์บางส่วนได้ เช่น:
  - เปลี่ยนที่อยู่จัดส่ง (ก่อนเข้าสู่ขั้นตอนจัดส่ง)
  - เพิ่มหมายเหตุเพิ่มเติมทั้งนี้ขึ้นอยู่กับสถานะออเดอร์และนโยบายของระบบ

#### 5. Restaurant Management

- FR-17: ร้านอาหารสามารถเพิ่ม/แก้ไข/ลบข้อมูลเมนูอาหาร:
  - ชื่อเมนู
  - ประเภท
  - ราคา
  - รูปภาพ
  - รายละเอียด
- FR-18: ตั้งค่า Availability:
  - เวลาเปิด–ปิดของร้าน
  - เมนูที่หมด (Out of Stock)
  - วันที่ร้านปิดพิเศษ (วันหยุดร้าน)
- FR-19: ร้านอาหารสามารถดูและจัดการออเดอร์แบบ Real-time:
  - ออเดอร์ใหม่
  - ออเดอร์ที่กำลังปั่น
  - ออเดอร์ที่รอ Rider มารับ
  - ออเดอร์ที่ส่งแล้ว
- FR-20: ตั้งค่า Pricing & Promotion:
  - ส่วนลดเมนู
  - คูปองส่วนลด
  - ราคายอดพิเศษตามช่วงเวลา (Happy Hour / Lunch Promotion)

## 6. Review & Rating

- FR-21: ลูกค้าที่ได้รับอาหารแล้วสามารถเขียนรีวิวและให้คะแนน:
  - ร้านอาหาร
  - คุณภาพอาหารประสบการณ์การจัดส่ง
- FR-22: ระบบแสดงคะแนนเฉลี่ยและรีวิวนหน้ารายละเอียดร้าน/เมนู เพื่อใช้ในการตัดสินใจซื้ออาหาร

## 7. Notification

- FR-23: ส่ง Email / In-app Notification เมื่อ:
  - ลูกค้าสั่งออเดอร์สำเร็จ
  - ร้านยืนยันรับออเดอร์
  - Rider รับงาน
  - ออเดอร์ถูกยกเลิก
- FR-24: ส่ง Reminder หรือ Update Status เช่น:
  - แจ้งเตือนเมื่ออาหารใกล้ถึง
  - แจ้งเตือนเมื่อ Rider ถึงหน้าบ้าน
- FR-25: แจ้งเตือน Restaurant Staff และ Rider เมื่อ:
  - มีออเดอร์ใหม่
  - มีการแก้ไข/ยกเลิกออเดอร์

## 8. Reporting & Analytics

- FR-26: System Admin สามารถดูรายงานรวม เช่น:
  - จำนวนออเดอร์รวมในระบบ
  - รายได้รวม
  - จำนวนผู้ใช้ใหม่/เดิม
  - Peak Time ของการสั่งอาหาร
- FR-27: Restaurant Manager สามารถ Export รายงาน:
  - ยอดขายรายวัน/สัปดาห์/เดือน
  - รายงานเมนูขายดี
  - รายงานค่าคอมมิชชัน/ค่าธรรมเนียมเป็นไฟล์ Excel/PDF

## Quality Attributes & Scenarios

QA-1: Performance

Scenario:

ส่วน	รายละเอียด
Source	ผู้ใช้ทั่วไป (Customer)
Stimulus	ผู้ใช้กดค้นหาร้านอาหาร/เมนู โดยระบุ Location และเงื่อนไขต่าง ๆ
Artifact	Search Module และ Database Query System
Environment	สภาพปกติ (Normal Load) – มีผู้ใช้พร้อมกัน 1,000 คน
Response	ระบบ Query ฐานข้อมูล Filter ร้าน/เมนูที่พร้อมส่ง และแสดงผลลัพธ์
Response Measure	แสดงผลการค้นหาภายใน 2 วินาที ในอัตราความสำเร็จ 95% ของ Requests

ในมุมผู้ใช้ เมื่อค้นหาร้านอาหาร ผู้ใช้คาดหวังว่า “ผลลัพธ์ต้องขึ้นเร็ว” หากต้องรอนานเกิน 2 วินาทีจะเริ่มรู้สึกหงุดหงิด (Frustration) และอาจปิดเว็บไซต์/แอปแล้วไปใช้บริการคู่แข่ง Performance ที่ดีจะช่วยเพิ่ม Conversion Rate ในการสั่งอาหารอย่างมาก โดยเฉพาะช่วงมื้อกลางวัน/เย็น หรือช่วงโปรโมชั่นที่มีผู้ใช้จำนวนมาก ระบบจึงต้องรองรับ Concurrent Requests ได้โดยไม่มีอาการหน่วงชัดเจน

## QA-2: Scalability

Scenario:

ส่วน	รายละเอียด
Source	ผู้ใช้หลายพันคนพร้อมกัน (Spike in Traffic)
Stimulus	จำนวนผู้ใช้เพิ่มจาก 1,000 เป็น 10,000 คนพร้อมกันช่วงโปรดาราคา / เทศกาล
Artifact	Application Servers, Database, Load Balancer
Environment	Peak Load – ช่วงเทศกาลหรือ Flash Sale / Promotion พิเศษ
Response	ระบบทำ Auto-scale เพิ่ม Server Instances และกระจาย Load
Response Measure	รักษา Response Time ไม่เกิน 3 วินาที และ Uptime ไม่ต่ำกว่า 99.5%

Food Delivery มีช่วง Peak ที่ชัดเจน เช่น เที่ยง 11.00–13.00, เย็น 17.00–20.00, ช่วงวันหยุดหรือ Flash Sale เมื่อ Traffic เพิ่มขึ้นอย่างรวดเร็ว ระบบต้อง Scale Out ได้อย่างต่อเนื่องรับ荷重ที่เพิ่มขึ้น โดยยังรักษา User Experience ได้ดี หากระบบล่มหรือช้ามากในช่วงนี้จะเสียรายได้และความเชื่อมั่นของลูกค้าอย่างรุนแรง

### QA-3: Security

Scenario:

ส่วน	รายละเอียด
Source	Malicious User / Hacker
Stimulus	ความพยายามทำ SQL Injection, Brute Force Login หรือเข้าถึงข้อมูลการชำระเงิน/ที่อยู่จัดส่งแบบ Unauthorized
Artifact	Authentication System, Payment Gateway, Database
Environment	Runtime – ระบบให้บริการปกติ
Response	ตรวจจับ Malicious Request, Block การเข้าถึง, Log Security Event และแจ้งเตือน Admin
Response Measure	Block การโจมตีได้ 100% และแจ้งเตือน Admin ภายใน 5 นาที โดยไม่มีข้อมูล Sensitive รั่วไหล

ระบบ Food Delivery เก็บข้อมูลสำคัญทั้งข้อมูลส่วนบุคคล (ชื่อ, ที่อยู่, เบอร์โทรศัพท์) และข้อมูลการชำระเงิน หากเกิดการโจมตีหรือข้อมูลรั่วไหลจะกระทบต่อลูกค้าและธุรกิจ อาจเข้าข่ายผิด PDPA และกฎหมายอื่น ๆ และเสียชื่อเสียงระยะยาว ดังนั้น Security เป็น Priority สูงสุด ต้องมีมาตรการเข้มงวด เช่น:

- ใช้ HTTPS ทุกการสื่อสาร
- ใช้การ Hash รหัสผ่านด้วย bcrypt/argon2
- ป้องกัน Injection / XSS / CSRF
- ใช้ Payment Gateway ที่ผ่านมาตรฐาน PCI DSS

#### QA-4: Availability

Scenario:

ส่วน	รายละเอียด
Source	Hardware Failure หรือ Network Outage
Stimulus	Primary Database Server ล้ม หรือ Network Interruption
Artifact	Database Server, Application Servers, Network Infrastructure
Environment	Runtime – มีอุเดอร์หลายรายการกำลังดำเนินการอยู่
Response	ระบบทำ Automatic Failover ไป Backup Server และคืนบริการ
Response Measure	Downtime ไม่เกิน 5 นาทีต่อเดือน (99.9% Uptime) และไม่มี Transaction สูญหาย

การสั่งอาหารออนไลน์ ผู้ใช้คาดหวังว่าระบบจะใช้งานได้ในช่วงเวลาที่ต้องการอาหาร (ซึ่งมักเป็นช่วง Peak) หากระบบล้มระหว่างที่ลูกค้าชำระเงินอาจเกิดปัญหา Double Charge หรืออุเดอร์หาย ทำให้ลูกค้าไม่พอใจ และต้องจัดการเคลียร์ปัญหาย้อนหลัง ระบบต้องออกแบบให้มี Failover & Backup ที่เหมาะสมเพื่อป้องกัน “Lost Order / Lost Payment”

## QA-5: Maintainability

Scenario:

ส่วน	รายละเอียด
Source	Development Team
Stimulus	ต้องเพิ่ม Feature ใหม่ (เช่น Loyalty Points, Voucher, รองรับ Payment Method ใหม่) หรือแก้ไข Bug
Artifact	Source Code, Database Schema, API Endpoints
Environment	Development & Deployment
Response	Developer สามารถเพิ่ม Feature / แก้ไข Bug โดยไม่กระทบ Module อื่น
Response Measure	สามารถ Deploy การเปลี่ยนแปลงภายใน 2 ชั่วโมง หลัง Code Review & Testing เสร็จ

ธุรกิจ Food Delivery ต้องมีการปรับตัวตลอดเวลา เช่น โปรโมชันใหม่, เมนูพิเศษ, การร่วมมือกับ Partner ใหม่ หรือเพิ่มฟีเจอร์ เช่น ระบบสะสมแต้ม, Subscription สำหรับลูกค้าประจำ เป็นต้น ระบบที่ Maintainability สูงจะช่วยเพิ่มความเร็วในการออกฟีเจอร์ใหม่ ลด Cost การ Maintenance และลดโอกาส Regression

## QA-6: Usability

### Scenario

ส่วน	รายละเอียด
Source	First-time User
Stimulus	ผู้ใช้เข้าเว็บไซต์/แอปครั้งแรกและต้องการสั่งอาหาร
Artifact	User Interface (Search, Menu, Cart, Checkout, Payment)
Environment	ใช้งานผ่าน Desktop Browser หรือ Mobile Device
Response	ระบบมี Flow การสั่งอาหารที่ชัดเจน ใช้งานง่าย และมี Call-to-Action ที่ไม่สับสน
Response Measure	ผู้ใช้ใหม่สามารถสั่งอาหารสำเร็จภายใน 5 นาที โดยไม่ต้องดู Tutorial และมี Success Rate ไม่น้อยกว่า 85%

Usability ส่งผลโดยตรงต่อ Conversion Rate หาก UI ใช้งาน ขึ้นตอน曳והเกินไป หรือสับสน ผู้ใช้มักจะออกจากระบบก่อนจบการสั่งอาหาร ระบบที่ดีต้อง:

- คันหาร้าน/เมนูได้ง่าย
- เพิ่มลงตัวร้าได้ชัดเจน
- Checkout ขั้นตอนน้อยที่สุดเท่าที่เป็นไปได้
- รองรับ Mobile Friendly อย่างดี

## Constraints

### Technical Constraints

- ระบบเป็น **Web-based Application** รองรับ Browsers หลัก (Chrome, Firefox, Edge ฯลฯ)
- รองรับ **Responsive Design** เพื่อใช้งานได้ทั้งบน Desktop และ Mobile
- Database ใช้ **Relational Database** เช่น PostgreSQL หรือ MySQL เพื่อรองรับ Transaction และ Data Integrity
- ใช้ Payment Gateway ที่ผ่านมาตรฐาน PCI DSS
- ต้องใช้ **HTTPS** ในการสื่อสารทั้งหมด
- ควรมี **API Layer** แยก Frontend/Backend เพื่อรองรับ Mobile App ในอนาคต

### Time Constraints

- Phase 1 (4 เดือน):
  - ฟีเจอร์หลัก เช่น User Management, Restaurant & Menu Search, Cart & Ordering, Payment, Basic Order Tracking
- Phase 2 (2 เดือนถัดไป):
  - Review & Rating System
  - Analytics Dashboard สำหรับ Admin และ Restaurant
- ใช้ Agile Methodology – แต่ละ Sprint ยาว 2 สัปดาห์
- Testing Period ไม่น้อยกว่า 2 สัปดาห์ รวม UAT ก่อน Production Launch

### Budget Constraints

- งบรวมไม่เกิน 800,000 บาท รวม Development, Infrastructure, Third-party Services
- งบ Cloud Hosting ประมาณ 10,000 บาท/เดือน (เช่น AWS/GCP)
- ไม่สามารถใช้ Premium Analytics Tools ต้องใช้ Open Source/Free Tier เป็นหลัก
- Payment Gateway Fee ต้องไม่เกิน 3% ต่อ Transaction

### Legal/Policy Constraints

- ต้องปฏิบัติตาม PDPA ในการจัดเก็บ/ประมวลผลข้อมูลส่วนบุคคล
- ต้อง Comply กับมาตรฐานความปลอดภัยในการชำระเงิน (PCI DSS ผ่าน Payment Gateway)
- เก็บ Audit Log อย่างน้อย 1 ปี สำหรับ Transaction และ Security Events
- ต้องมีเอกสาร Terms of Service และ Privacy Policy ให้ผู้ใช้ยอมรับก่อนใช้งาน
- รองรับกระบวนการ ขอคืนเงิน (Refund) ตามกฎหมายคุ้มครองผู้บริโภค และนโยบายร้าน/ระบบ

## Assumptions

- ผู้ใช้มือถือที่รองรับ HTML5/CSS3/JavaScript และใช้ Modern Browsers
- ผู้ใช้มี Internet ความเร็วพอสมควร (อย่างน้อยระดับ 3G/4G)
- ร้านอาหารที่เข้าร่วมมีข้อมูลเมนู ราคา และพร้อมอัปเดตสถานะเมนู/เวลาเปิด-ปิด
- Payment Gateway มี Uptime อย่างน้อย 99.9%
- ลูกค้ามี Email และ/หรือเบอร์โทรศัพท์ใช้งานได้จริงสำหรับรับ Notification
- Infrastructure รองรับการ Scale ของ Database ทั้งแนวตั้ง/แนวอนตาม Load
- มี CDN สำหรับ Static Assets (รูปเมนู, รูปร้าน, Static Files) เพื่อเพิ่มความเร็วการโหลด
- Restaurant Staff / Rider มีความรู้พื้นฐานในการใช้งานคอมพิวเตอร์/สมาร์ตโฟน และใช้งาน Dashboard/App ได้หลังการอบรมระยะสั้น

## Quality Attributes Priority

Rank	Quality Attribute	เหตุผล
1	Security	ระบบเกี่ยวข้องกับข้อมูลส่วนบุคคล (ที่อยู่จัดส่ง, เบอร์โทรศัพท์) และข้อมูลการชำระเงิน การละเมิดความปลอดภัยส่งผลกระทบร้ายแรงทั้งต่อผู้ใช้และธุรกิจ เสียงต่อการฟ้องร้องและเสียชื่อเสียงระยะยาว จึงต้องให้ความสำคัญสูงสุด
2	Availability	การสั่งอาหารมักเกิดในช่วงเวลาเฉพาะ หากระบบล่มหรือเข้าไม่ได้ในช่วงมีมือลักษณะจะสูญเสียรายได้ทันทีและผู้ใช้หันไปใช้คู่แข่งได้ง่าย
3	Performance	ผู้ใช้คาดหวัง Response ที่รวดเร็ว ทั้งการค้นหาร้านและการยืนยันสั่งอาหาร หากระบบช้า ผู้ใช้มีแนวโน้มยกเลิกการสั่งและออกจากระบบทันที
4	Usability	Flow การสั่งอาหารที่ง่ายและชัดเจนมีผลต่อ Conversion Rate โดยตรง หากขั้นตอนสับสน ผู้ใช้จะไม่จบอุเดอร์
5	Scalability	แม้เหตุการณ์ Peak Load จะเกิดเป็นช่วง ๆ แต่เมื่อเกิดขึ้น ระบบต้องรองรับได้ไม่เช่นนั้นจะกระทบรายได้อย่างรุนแรงในช่วงโปรโมชั่นหรือเทศกาล
6	Maintainability	แม้สำคัญต่อระยะยาว แต่เมื่อเทียบกับ Security/Availability/Performance ที่กระทบผู้ใช้โดยตรงจึงจัดอันดับรองลงมา อย่างไรก็ตามยังต้องออกแบบให้ดูแลง่าย เพื่อลดต้นทุนในอนาคต

## Trade-offs Analysis

### Trade-off #1: Performance vs Security

#### สถานการณ์:

การเข้ารหัสข้อมูลทุกครั้ง (HTTPS, Encryption at Rest บางส่วน) และการ Hash Password ด้วย bcrypt/argon2 รอบสูง (เช่น 12–14 rounds) จะเพิ่มความปลอดภัย แต่เพิ่มภาระ CPU ทำให้ Login และ Transaction อาจช้าลงเล็กน้อย

#### Decision:

##### เลือก Option A – Prioritize Security

เนื่องจาก Security เป็น Priority อันดับ 1 การเพิ่ม Response Time ถึง 200–500ms ยังอยู่ในเกณฑ์ที่ยอมรับได้ (รวมแล้วยังควรต่ำกว่า 2 วินาทีสำหรับการทำงานหลัก) และสามารถขยายด้วยเทคนิคอื่น เช่น:

- Caching
- CDN สำหรับ Static Content
- Optimization ที่ระดับ Query/Index

### Trade-off #2: Availability vs Cost

#### สถานการณ์:

การทำ High Availability ระดับ 99.99% ต้องใช้ Multi-region Deployment, Advanced Load Balancing, Database Replication แบบซับซ้อน ซึ่งค่าใช้จ่ายสูงกว่าระบบ Single-region ธรรมดาถึง 3–4 เท่า ขณะที่ Budget Cloud Infrastructure จำกัดที่ ~10,000 บาท/เดือน

#### Decision:

##### เลือก Option B – Target 99.5% Uptime

เพราะข้อจำกัดด้านงบประมาณในช่วงเริ่มต้น การมี Downtime รวมประมาณ 3–4 ชั่วโมงต่อปีอยู่ในระดับที่ยอมรับได้สำหรับธุรกิจช่วงเริ่มต้น และสามารถวางแผน:

- เพิ่มระดับ HA และ DR (Disaster Recovery) เมื่อยอดผู้ใช้/Revenue โตขึ้น
- จัดการ Maintenance ช่วง Off-peak เพื่อลดผลกระทบต่อผู้ใช้

### Trade-off #3: Feature Richness vs Time-to-Market

สถานการณ์:

มี Feature Requests จำนวนมาก เช่น:

- Loyalty Program / Points
- Multi-language Support
- Advanced Recommendation System (แนะนำเมนู/ร้านตามพฤติกรรมสั่ง)
- Social Media Integration
- ระบบ Subscription / Corporate Account

หากต้องพัฒนาทั้งหมดก่อน Launch จะใช้เวลานาน (8-10 เดือน) ขณะที่ตลาดแข่งขันสูงและต้องการเข้าตลาดเร็วเพื่อดึงผู้ใช้และร้านอาหารเข้าระบบก่อน

Decision:

เลือก Option B – MVP First

เน้นออกแบบหลักที่จำเป็นต่อการสั่งอาหารให้ครบก่อน (Search, Ordering, Payment, Basic Tracking)  
แล้ว Launch เพื่อ:

- เก็บ Feedback จากผู้ใช้จริง
- ปรับปรุงและเพิ่มฟีเจอร์ที่ “มี Demand จริง”
- ลดความเสี่ยงในการทุ่มพัฒนา Feature ที่อาจไม่ถูกใช้

ใช้แนวคิด Agile over Waterfall – ออกแบบ Increment และ Iteration แทนการรอ Big Bang Release ครั้งเดียว