

Lab 2: Architectural Drivers Document

Project Name: QuickBite Delivery (ระบบจัดส่งอาหาร)

ส่วนที่ 1: System Overview

1. System Description

QuickBite Delivery คือแพลตฟอร์มตัวกลาง (Intermediary Platform) ที่เชื่อมโยงระหว่าง 3 पार्टีหลัก ได้แก่ ลูกค้า (Customers), ร้านอาหาร (Restaurants), และผู้ส่งอาหาร (Riders)

เพื่ออำนวยความสะดวกในการสั่งซื้อและจัดส่งอาหาร

ระบบนี้แก้ปัญหาความยุ่งยากในการเดินทางไปซื้อหน้าร้าน ช่วยให้ร้านอาหารเข้าถึงฐานลูกค้าที่กว้างขึ้น และสร้างอาชีพให้กับไรเดอร์ โดยระบบจะครอบคลุมตั้งแต่การค้นหาร้าน, การสั่งอาหาร, การชำระเงิน, การติดตามสถานะการจัดส่งแบบ Real-time, และระบบจัดการหลังบ้านสำหรับร้านค้าและ Admin

2. Target Users

- Customers (ลูกค้าทั่วไป): ผู้ใช้งานที่ต้องการสั่งอาหารผ่านแอปพลิเคชันบนมือถือ
- Restaurant Partners (ร้านอาหารพาร์ทเนอร์):
เจ้าของร้านอาหารที่ต้องการรับออเดอร์และจัดการเมนูอาหาร
- Riders (ผู้ส่งอาหาร): พนักงานขับรถที่รับงานจัดส่งอาหารจากร้านไปสู่ลูกค้า
- System Admins (ผู้ดูแลระบบ): เจ้าหน้าที่ดูแลความเรียบร้อย จัดการข้อร้องเรียน และอนุมัติร้านค้า/ไรเดอร์

3. Key Features (Top 10)

- Search & Filter: ค้นหาร้านอาหารตามประเภท, เรตติ้ง, หรือระยะทาง
- Order Placement: ระบบสั่งอาหาร ระบุตัวเลือกพิเศษ (Toppings) และคำนวณราคา
- Real-time Tracking: ติดตามตำแหน่งของไรเดอร์และสถานะออเดอร์บนแผนที่
- In-App Payment: ชำระเงินผ่านบัตรเครดิต, Wallet หรือ QR Code
- Rider Dispatching: ระบบจับคู่ไรเดอร์กับออเดอร์อัตโนมัติ (Auto-matching)
- Menu Management: ร้านค้าสามารถเปิด/ปิดเมนู และแก้ไขราคาได้เอง
- Chat System: ระบบแชทระหว่างลูกค้าและไรเดอร์
- Review & Rating: ให้คะแนนร้านค้าและไรเดอร์หลังจบงาน
- Push Notifications: แจ้งเตือนสถานะต่างๆ (รับออเดอร์, อาหารเสร็จ, ใกล้ถึง)
- Promotion/Coupon: ระบบใช้โค้ดส่วนลดและโปรโมชั่นค่าส่ง

ส่วนที่ 2: Functional Requirements

Functional Requirements (Total: 16 items)

User Authentication & Profile

- FR-01: Users (Customer, Rider, Merchant)
ต้องสามารถลงทะเบียนและเข้าสู่ระบบผ่านเบอร์โทรศัพท์ (OTP) หรือ Email ได้
- FR-02: Users สามารถแก้ไขข้อมูลส่วนตัว เช่น ชื่อ, รูปโปรไฟล์, และรหัสผ่านได้
- FR-03: Customers สามารถเพิ่ม, ลบ, และแก้ไขที่อยู่สำหรับการจัดส่ง (Saved Addresses) ได้

Ordering & Browsing (Customer)

- FR-04: Customers สามารถค้นหาร้านอาหารโดยใช้ Keyword หรือเลือกตาม Category ได้
- FR-05: Customers สามารถเพิ่มรายการอาหารลงในตะกร้า (Cart) และแก้ไขจำนวนได้ก่อนยืนยันสั่ง
- FR-06: Customers ต้องสามารถยกเลิกออเดอร์ได้ หากร้านค้ายังไม่กดรับออเดอร์ (ภายในเวลาที่กำหนด)

Restaurant Operations

- FR-07: Restaurants ต้องสามารถกด "รับออเดอร์" (Accept) หรือ "ปฏิเสธ" (Reject) พร้อมระบุเหตุผลได้
- FR-08: Restaurants สามารถแก้ไขสถานะของร้าน (Open/Close) ได้แบบ Real-time
- FR-09: Restaurants สามารถดูสรุปยอดขายรายวัน (Daily Report) ได้

Rider Operations

- FR-10: Riders ต้องสามารถมองเห็นงานที่เข้ามา (Incoming Job) และกดรับงานได้
- FR-11: Riders ต้องสามารถอัปเดตสถานะงานได้ (Arrived at Restaurant, Picked Up, Delivered)

Payment & Wallet

- FR-12: ระบบต้องรองรับการชำระเงินผ่าน Credit/Debit Card Gateway ได้
- FR-13: ระบบต้องมี e-Wallet สำหรับ Rider เพื่อใช้หักค่าคอมมิชชั่น หรือรับเงินค่าจ้าง

Location & Tracking

- FR-14: ระบบต้องแสดงพิกัด GPS ปัจจุบันของ Rider บนแผนที่ให้ Customer เห็น
- FR-15: ระบบต้องคำนวณค่าจัดส่ง (Delivery Fee) ตามระยะทางจริงจาก Google Maps API

Notification

- FR-16: ระบบต้องส่ง Notification ไปยัง Customer ทันทีเมื่อสถานะออเดอร์มีการเปลี่ยนแปลง

ส่วนที่ 3: Quality Attributes & Scenarios

Quality Attributes Scenarios

QA-1: Performance (Latency)

Scenario: ลูกค้ายืนยันการสั่งซื้อในช่วงเวลาปกติ

ส่วน	รายละเอียด
Source	Customer (1,000 concurrent users)
Stimulus	กดปุ่ม "Place Order" (ยืนยันคำสั่งซื้อ)
Artifact	Order Processing Service
Environment	Normal Operation
Response	ระบบบันทึกออเดอร์, ตัดสต็อก, และส่งกลับหน้า "Waiting for confirmation"
Response Measure	Latency ในการตอบกลับ < 2 วินาที (95th percentile)

QA-2: Scalability

Scenario: ปริมาณการใช้งานพุ่งสูงในช่วงมือเที่ยง (Lunch Rush Hour)

ส่วน	รายละเอียด
Source	Customers จำนวนมาก (10,000+ users)
Stimulus	การใช้งานพุ่งสูงขึ้น 5 เท่าจากปกติ ภายใน 10 นาที
Artifact	System Backend / Cloud Infrastructure
Environment	High Load Condition (11:30 - 13:00)
Response	ระบบ Scale out จำนวน Server instance อัตโนมัติเพื่อรองรับ Load
Response Measure	ระบบยังทำงานได้โดยไม่มี Error และ Response Time เฉลี่ย < 3 วินาที

QA-3: Availability

Scenario: ส่วนประกอบหนึ่งของระบบ (เช่น Payment Service) ล่ม

ส่วน	รายละเอียด
Source	Internal Failure / Network Issue
Stimulus	Payment Gateway ไม่ตอบสนอง (Downtime)
Artifact	Payment Service / Checkout Process
Environment	Normal Operation
Response	ระบบแจ้งเตือนลูกค้าให้ใช้ Cash on Delivery (COD) แทนชั่วคราว ไม่ให้แอป Crash
Response Measure	System Uptime โดยรวมต้องอยู่ที่ 99.9% ต่อเดือน

QA-4: Security

Scenario: การเข้าถึงข้อมูลส่วนตัวของผู้ใช้

ส่วน	รายละเอียด
Source	Unauthorized User / Hacker
Stimulus	พยายามดึงข้อมูลเบอร์โทรศัพท์และที่อยู่ลูกค้าผ่าน API
Artifact	User Data Database / API Gateway
Environment	Connected to Internet
Response	ระบบตรวจสอบ Token (Authentication) และปฏิเสธการเข้าถึง (Access Denied)
Response Measure	Data Leakage = 0% และบันทึก Security Log ครบถ้วน

QA-5: Usability

Scenario: ผู้ใช้ใหม่สั่งอาหารครั้งแรก

ส่วน	รายละเอียด
Source	New Customer
Stimulus	ต้องการสั่งข้าวมันไก่จากร้านใกล้บ้าน
Artifact	Customer App Interface (UI/UX)
Environment	First time use
Response	ผู้ใช้สามารถทำรายการจนจบ (Checkout) ได้อย่างราบรื่น
Response Measure	ใช้เวลาเรียนรู้และสั่งเสร็จสิ้นภายใน < 3 นาที หรือ < 5 คลิก (หลังจากเลือกร้าน)

QA-6: Interoperability

Scenario: การดึงข้อมูลแผนที่และเส้นทาง

ส่วน	รายละเอียด
Source	Rider App
Stimulus	Rider กดดูเส้นทางไปบ้านลูกค้า
Artifact	Map Service Module
Environment	Normal Operation
Response	ระบบส่งพิกัดให้ Google Maps/Apple Maps เพื่อนำทางได้ถูกต้อง
Response Measure	ความถูกต้องของพิกัด 100% ตามที่ลูกค้าป้อน

ส่วนที่ 4: Constraints

Constraints

Technical Constraints

- C-01: แอปพลิเคชันฝั่ง Customer และ Rider ต้องรองรับทั้ง iOS และ Android (Cross-platform framework เช่น Flutter หรือ React Native)
- C-02: Backend ต้อง Deploy บน Cloud Provider (AWS หรือ Google Cloud) เท่านั้น เพื่อความยืดหยุ่น

Time Constraints

- C-03: ต้องส่งมอบระบบเวอร์ชัน MVP (Minimum Viable Product) ภายใน 3 เดือน เพื่อให้ทันรอบการระดมทุน

Budget Constraints

- C-04: งบประมาณสำหรับ Cloud Infrastructure ต้องไม่เกิน 30,000 บาท/เดือน ในช่วง 6 เดือนแรก

Legal/Policy Constraints

- C-05: การเก็บข้อมูลส่วนตัวลูกค้า ต้องปฏิบัติตาม พ.ร.บ.คุ้มครองข้อมูลส่วนบุคคล (PDPA) อย่างเคร่งครัด

ส่วนที่ 5: Assumptions

Assumptions (สิ่งที่เชื่อว่าเป็นจริง)

1. A-01: ผู้ใช้งานทุกคน (Customers, Riders, Restaurants) มี Smartphone ที่เชื่อมต่อ Internet (4G/5G) ตลอดเวลาขณะใช้งาน
2. A-02: ร้านค้าพาร์ทเนอร์มีความพร้อมในการใช้อุปกรณ์ Tablet หรือ Smartphone ในการรับออเดอร์
3. A-03: Third-party Map API (เช่น Google Maps) มีความแม่นยำเพียงพอในการระบุตำแหน่งบ้านเลขที่ในพื้นที่ให้บริการ
4. A-04: ระบบ Payment Gateway ภายนอก (ธนาคาร/บัตรเครดิต) มี Uptime อย่างน้อย 99%
5. A-05: ไรเดอร์สามารถอ่านภาษาไทยและใช้แผนที่นำทางได้คล่องแคล่ว

ส่วนที่ 6: Priority & Trade-offs

Quality Attributes Priority

Rank	Quality Attribute	เหตุผล
1	Availability	หากระบบล่ม (Down) จะเกิดการสูญเสียรายได้ทันทีและเสียความเชื่อมั่น (Critical ที่สุด)
2	Performance	ธุรกิจ Food Delivery แข่งกันที่ความเร็ว แอปต้องโหลดไว ออเดอร์ต้องแจ้งทันที
3	Security	มีธุรกรรมการเงินและข้อมูลส่วนตัว แต่ Trade-off กับความสะดวกได้บ้างในบางจุด
4	Scalability	สำคัญมากในช่วง Peak Hour แต่อาจยังไม่ Critical เท่าความเสถียร (Availability)
5	Usability	แอปต้องใช้ง่าย แต่ผู้ใช้ส่วนใหญ่คุ้นเคยกับแอปสั่งอาหารอยู่แล้ว
6	Interoperability	เชื่อมต่อ Maps/Payment เป็นเรื่องพื้นฐานที่ต้องทำได้

Trade-offs Analysis

Trade-off: Security vs. Performance (Convenience)

สถานการณ์: การตรวจสอบความปลอดภัยทุกครั้งที่เปิดแอป vs ความรวดเร็วในการสั่งอาหาร

- **Option A (High Security):** บังคับให้ User ใส่ PIN หรือสแกนนิ้วทุกครั้งที่เกิดสั่งซื้อ หรือ Login ใหม่ทุก 24 ชม.
 - Pros: ปลอดภัยสูง ป้องกันคนแอบกดสั่ง
 - Cons: ลูกค้าน่ารำคาญ ขั้นตอนเยอะ อาจเปลี่ยนไปใช้แอปคู่แข่ง
- **Option B (High Performance/Convenience):** จำ Session ตลอดไป (Remember Me) กดสั่งได้เลยไม่ต้องยืนยันซ้ำ
 - Pros: สั่งงานเร็วมาก (1-Click Order) UX ดีเยี่ยม
 - Cons: ความเสี่ยงสูงหากมือถือหาย

Decision: เลือก Balanced Approach (เน้น Convenience แต่มี Checkpoint)

เราเลือกที่จะให้ระบบ Remember Login Session ไว้ตลอด เพื่อความเร็วในการใช้งาน แต่จะ บังคับ Biometric Scan (FaceID/TouchID) หรือ PIN เฉพาะตอนจะ "Confirm Payment" เท่านั้น

Reason: เพื่อให้ Flow การเลือกร้านและสั่งอาหารลื่นไหลที่สุด (Performance/Usability)

แต่ยังคงมีด้านความปลอดภัยสุดท้ายก่อนเงินออกจากกระเป๋า (Security) ซึ่งเป็นจุดที่ลูกค้ายอมรับได้