



ใบงาน

LAB2

Architectural Drivers Document

เสนอ

อาจารย์ธนิศ เกตุแก้ว

โดย

นายพิชิตกร ชชาติปุระ 67543210004-7

นายธนมินทร์ เปลียนพร้อม 67543210032-8

นายธนพล ตริรัตน์นภาพ 67543210070-8

นางสาวรัฐฉิกาภรณ์ กวงคำ 67543210063-3

ใบงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา

ENGSE207 SOFTWARE ARCHITECTURE

ภาคเรียนที่ 2 ปีการศึกษา 2568

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา เชียงใหม่ (วิทยาเขตดอยสะเก็ด)

ส่วนที่ 1: System Overview

(Food Delivery System)

1. System Description

ระบบจัดส่งอาหารเป็นแพลตฟอร์มดิจิทัลที่เชื่อมต่อผู้บริโภค ร้านอาหาร และไรเดอร์เข้าด้วยกัน ผู้ใช้สามารถดูเมนู สั่งอาหาร และชำระเงินผ่านแอปได้อย่างสะดวกโดยไม่ต้องเดินทางไปร้าน ร้านอาหารสามารถขยายช่องทางการขายและเข้าถึงลูกค้าได้มากขึ้น ส่วนไรเดอร์สามารถสร้างรายได้จากการรับงานจัดส่ง ระบบนี้ช่วยแก้ปัญหาความไม่สะดวก ประหยัดเวลา และเพิ่มทางเลือกในการสั่งอาหาร นอกจากนี้ยังช่วยเพิ่มยอดขายให้ร้านอาหารและสร้างโอกาสในการทำงานให้กับไรเดอร์

2. Target Users

- Customers – กลุ่มผู้ที่ต้องการสั่งอาหารผ่านระบบ
- Restaurants – ร้านที่เข้าร่วมเพื่อขายอาหารผ่านแพลตฟอร์ม
- Riders – ผู้ที่ทำหน้าที่รับอาหารจากร้านและส่งถึงลูกค้า
- Admins – บริหารจัดการ และควบคุมระบบทั้งหมด
- Customer Support – ตอบคำถาม แจ้งปัญหา และช่วยแก้ไขข้อผิดพลาด

3. Key Features (Top 10)

- ชำระเงินออนไลน์
- สั่งอาหาร
- filter สำหรับประเภทอาหาร
- ระบบติดตามตำแหน่ง
- ประวัติการสั่ง
- ระบบบัญชี
- ค้นหาร้านอาหารใกล้เคียง/ตามชื่อ
- ระบบตะกร้า
- สะสมแต้ม
- รีวิวและให้คะแนนร้านอาหาร/คนขับ

ส่วนที่ 2: Functional Requirements

User Management

- FR-UM-1 : ผู้ใช้ (ลูกค้า, ร้านอาหาร, คนขับ) ต้องสามารถลงทะเบียนและเข้าสู่ระบบได้
- FR-UM-2 : ผู้ใช้ต้องสามารถแก้ไขข้อมูลส่วนตัว (Profile) ได้
- FR-UM-3 : ระบบต้องรองรับการกู้คืนรหัสผ่าน (Password Recovery)

Core Features (Ordering)

- FR-OR-1 : ผู้ใช้สามารถเลือกอาหารจากร้านและทำการสั่งซื้อได้
- FR-OR-2 : ระบบต้องรองรับการส่งแบบ Delivery
- FR-OR-3 : ต้องแสดงรายการอาหาร, ราคา และรายละเอียด
- FR-OR-4 : ผู้ใช้ต้องสามารถเพิ่ม/ลดเมนูในตะกร้าก่อนยืนยันคำสั่งซื้อได้

Notification

- FR-NT-1 : ระบบต้องส่งแจ้งเตือน (Push Notification/SMS/Email) เมื่อสถานะการสั่งซื้อมีการเปลี่ยนแปลง (เช่น ยืนยันคำสั่งซื้อ, กำลังเตรียมอาหาร, คนขับมารับแล้ว, กำลังจัดส่ง, จัดส่งสำเร็จ)
- FR-NT-2 : ร้านอาหารต้องได้รับการแจ้งเตือนทันทีเมื่อมีคำสั่งซื้อใหม่

Reporting

- FR-RP-1 : ผู้ดูแลระบบและร้านอาหารต้องสามารถดูรายงานยอดขายรายวัน/รายสัปดาห์/รายเดือนได้
- FR-RP-2 : ผู้ใช้ (ลูกค้า) ต้องสามารถดูประวัติการสั่งซื้อ และใบเสร็จได้

ส่วนที่ 3: Quality Attributes & Scenarios

QA-1: Performance

Scenario:

ส่วน	รายละเอียด
Source	ผู้ใช้ทั่วไปที่ต้องการสั่งอาหาร
Stimulus	ผู้ใช้ทำการค้นหาร้านอาหารพร้อมกันจำนวนมาก
Artifact	Backend Server, Database, API
Environment	มีผู้ใช้งานจำนวนมาก
Response	ประมวลผลได้รวดเร็วและไม่ล่ม
Response Measure	Error ไม่เกิน 1%

คำอธิบายแบบเต็ม : ระบบต้องสามารถรองรับการค้นหาร้านอาหารจากผู้ใช้จำนวนมากในเวลาเดียวกันโดยไม่เกิดความล่าช้าหรือระบบล่มโดย Backend, Database และ API ต้องประมวลผลคำขอได้อย่างมีประสิทธิภาพในสภาพแวดล้อมที่มีโหลดสูง ระบบถือว่าทำงานได้ตามเกณฑ์หากอัตรา Error ไม่เกิน 1% ในช่วงการใช้งานหนาแน่น

QA-2: Scalability

Scenario:

ส่วน	รายละเอียด
Source	การตลาด/การเติบโตของธุรกิจ
Stimulus	จำนวนผู้ใช้งานเพิ่มขึ้น 10 เท่า (จาก 10,000 เป็น 100,000 คนต่อวัน)
Artifact	Application servers, Database cluster
Environment	การเพิ่มทรัพยากรของระบบเพียงเล็กน้อย (เช่น เพิ่ม Server 25%)
Response	ระบบต้องแสดงรายการร้านอาหาร และเมนูที่เกี่ยวข้อง
Response Measure	เวลาที่ใช้ในการโหลดข้อมูลร้านอาหารและเมนูต้องไม่เกิน 3 วินาที

คำอธิบายแบบเต็ม: ระบบต้องตอบสนองต่อการค้นหา การสั่งซื้อ และการชำระเงินได้อย่างรวดเร็ว โดยเฉพาะในช่วงเวลาที่มีการใช้งานสูง (Peak Hours) เพื่อให้ผู้ใช้ได้รับประสบการณ์ที่ดี

QA-3: Availability

Scenario:

ส่วน	รายละเอียด
Source	Failure condition
Stimulus	Server หลั้ก ของระบบล้มเหลว (เช่น ไฟดับ, Hardware failure)
Artifact	Application layer, Database layer
Environment	ระบบอยู่ในสภาพการทำงานปกติ
Response	เปลี่ยนไปใช้ Server สำรอง (Failover) และให้บริการต่อ
Response Measure	เวลาที่ระบบหยุดชะงัก (Downtime) ต้องไม่เกิน3นาที (RTO: Recovery Time Objective)

คำอธิบายแบบเต็ม: ระบบต้องพร้อมใช้งานอยู่เสมอ (uptime สูง) โดยเฉพาะช่วงเวลา
ที่คนนิยมสั่งอาหาร ระบบต้องมีกลไกสำรอง/กู้คืนอัตโนมัติ (Failover/Redundancy)
เพื่อลดเวลาที่ระบบล่ม

QA-4: Security

Scenario:

ส่วน	รายละเอียด
Source	External Attacker
Stimulus	ผู้โจมตีพยายามดักจับข้อมูลบัตรเครดิตระหว่างการทำธุรกรรม
Artifact	Payment Processing Module, API Gateway
Environment	การสื่อสารระหว่าง Client (App) และ Server
Response	ข้อมูลบัตรเครดิตต้องถูกเข้ารหัส (Encryption) และการทำธุรกรรมต้องใช้ โปรโตคอลที่ปลอดภัย (เช่น TLS/SSL)
Response Measure	ไม่มีการรั่วไหลของข้อมูลการชำระเงินที่ละเอียดอ่อน

คำอธิบายแบบเต็ม: ข้อมูลส่วนตัวของผู้ใช้, ข้อมูลการชำระเงิน, และข้อมูลการติดตามตำแหน่งต้องได้รับการป้องกันการเข้าถึงโดยไม่ได้รับอนุญาต (Authentication, Authorization, Encryption)

QA-5: Modifiability

Scenario:

ส่วน	รายละเอียด
Source	ทีมพัฒนา (Development Team)
Stimulus	ต้องการเพิ่มวิธีการชำระเงินใหม่ (เช่น QR Code Payment)
Artifact	Payment Processing Module
Environment	ระบบถูกออกแบบตามหลัก Microservices หรือ Modular Architecture
Response	การเพิ่มฟีเจอร์ใหม่ต้องไม่กระทบต่อฟังก์ชันการทำงานอื่นๆ ของระบบหลัก
Response Measure	เวลาที่ใช้ในการพัฒนา, ทดสอบ, และนำขึ้นใช้งาน (Deploy) ของ Payment Module ใหม่ไม่เกิน5วันทำการ

คำอธิบายแบบเต็ม: ระบบควรถูกออกแบบให้เป็น Module แยกส่วน (Decoupled) เพื่อให้ง่ายต่อการแก้ไข เพิ่มฟีเจอร์ใหม่ หรืออัปเดตเทคโนโลยีโดยไม่กระทบกับส่วนอื่นๆ ของระบบ

QA-6: Usability

Scenario:

ส่วน	รายละเอียด
Source	ผู้ใช้งานใหม่ (New Customer)
Stimulus	ผู้ใช้ต้องการ สั่งอาหารเป็นครั้งแรก
Artifact	User Interface (UI) และ User Experience (UX)
Environment	ผู้ใช้เข้าถึงผ่าน Mobile Application
Response	ผู้ใช้ต้องสามารถ ค้นหา, เลือก, และยืนยันคำสั่งซื้อได้อย่างง่าย
Response Measure	ผู้ใช้ใหม่สามารถทำรายการสั่งซื้อสำเร็จได้โดยใช้เวลาไม่เกิน 2 นาที หรือ คลิกไม่เกิน 6 ครั้ง นับตั้งแต่เปิดแอป

คำอธิบายแบบเต็ม: แอปพลิเคชันต้องใช้งานง่าย มีหน้าต่างที่เข้าใจง่าย และขั้นตอนการสั่งซื้อที่กระชับ

ส่วนที่ 4: Constraints

Technical Constraints

- C-TC-1: ระบบต้องใช้ Google Maps Platform API ในการแสดงแผนที่และระบบติดตามตำแหน่ง (GPS Tracking)
- C-TC-2: ระบบต้องถูกพัฒนาให้รองรับทั้ง iOS และ Android Mobile Application
- C-TC-3: การประมวลผลการชำระเงินต้องเป็นไปตามมาตรฐาน PCI DSS (Payment Card Industry Data Security Standard)

Time Constraints

- C-T-1: ต้องสามารถปล่อย เวอร์ชันแรก (MVP - Minimum Viable Product) ได้ภายใน 6 เดือน

Budget Constraints

- C-B-1: ต้นทุนในการดำเนินงาน (Operation Cost) ในปีแรกต้องไม่เกิน 1,100,000 บาท (ตัวเลขสมมติ)

Legal/Policy Constraints

- C-LP-1: ต้องปฏิบัติตาม พระราชบัญญัติคุ้มครองข้อมูลส่วนบุคคล (PDPA) ของประเทศไทยในการจัดเก็บและใช้ข้อมูลผู้ใช้

ส่วนที่ 5: Assumptions

1. ร้านอาหาร:

ร้านอาหารคู่ค้าส่วนใหญ่มีความพร้อมในการรับคำสั่งซื้อผ่านระบบดิจิทัล (เช่น มีแท็บเล็ต/อุปกรณ์รับออเดอร์)

2. การชำระเงิน:

ผู้ใช้ส่วนใหญ่มีความเชื่อมั่นในการทำธุรกรรมชำระเงินออนไลน์

3. คนขับ/ไรเดอร์:

มีจำนวนคนขับเพียงพอในพื้นที่ให้บริการเพื่อรองรับปริมาณคำสั่งซื้อ

4. Internet:

ผู้ใช้และคนขับมีสัญญาณอินเทอร์เน็ตที่เสถียรและเพียงพอต่อการใช้งาน

5. Location Data:

ข้อมูลตำแหน่งที่ตั้ง (GPS) จากอุปกรณ์ของผู้ใช้และคนขับมีความแม่นยำในระดับที่ยอมรับได้

ส่วนที่ 6: Priority & Trade-offs

Quality Attributes Priority

Rank	Quality Attribute	เหตุผล
1	Availability (ความพร้อมใช้งาน)	ระบบล่มหมายถึงการสูญเสียยอดขายทันทีและสร้างความไม่พอใจอย่างรุนแรงแก่ลูกค้าและร้านอาหาร ระบบต้องพร้อมใช้งานอยู่เสมอเพื่อให้เกิดการทำธุรกรรม
2	Security (ความปลอดภัย)	เนื่องจากการจัดการข้อมูลส่วนตัวและการชำระเงิน (บัตรเครดิต) การรั่วไหลของข้อมูลเป็นความเสี่ยงทางธุรกิจและทางกฎหมายที่ร้ายแรงที่สุด ต้องให้ความสำคัญสูงสุด
3	Performance (ประสิทธิภาพ)	ความเร็วเป็นสิ่งสำคัญใน Food Delivery หากแอปช้า ผู้ใช้อาจเปลี่ยนไปใช้คู่แข่งการสั่งซื้อและการติดตามต้องรวดเร็ว
4	Scalability (ความสามารถในการขยายตัว)	ระบบถูกสร้างขึ้นเพื่อเติบโต หากไม่สามารถรองรับจำนวนผู้ใช้ที่เพิ่มขึ้นได้ จะส่งผลกระทบต่อ Performance และ Availability ในที่สุด
5	Usability (ความสามารถในการใช้งาน)	หากใช้งานยาก ลูกค้าจะไม่ใช้ และส่งผลต่อยอดขาย แต่ถือว่ามีความสำคัญน้อยกว่าการที่ระบบล่มหรือข้อมูลรั่วไหล
6	Modifiability (ความสามารถในการปรับเปลี่ยน)	มีความสำคัญในการพัฒนาต่อยอดในอนาคต แต่ในระยะเริ่มต้น ความเสถียรและความปลอดภัยมีความสำคัญเร่งด่วนกว่า

Trade-offs Analysis

Trade-off #1: Availability vs Performance

อธิบาย : การออกแบบระบบให้มี Availability สูงมักเกี่ยวข้องกับการใช้ Redundancy (การทำซ้ำ), Failover, และการกระจายข้อมูลไปยังหลาย Data Center ซึ่งอาจเพิ่มความซับซ้อนในการจัดการ และบางครั้งอาจเพิ่มความหน่วงเล็กน้อยเมื่อเทียบกับการใช้ Server เดียวที่ประมวลผลเร็วที่สุด (ซึ่งทำให้ Performance ดีเยี่ยมแต่ Availability ต่ำ) ในทางกลับกัน การออกแบบที่มุ่งเน้น Performance สูงสุดอาจทำให้มี Single Point of Failure และส่งผลให้ Availability ตกต่ำเมื่อเกิดปัญหา

Decision: เราเลือกให้ความสำคัญกับ Availability (ความพร้อมใช้งาน)

ทำไม : ในธุรกิจ Food Delivery การสูญเสียคำสั่งซื้อ เนื่องจากการล่มของระบบ (Availability ต่ำ) สร้างความเสียหายและภาพลักษณ์ที่แย่กว่าการตอบสนองที่ช้าลงเล็กน้อย (Performance ลดลงเล็กน้อย) การออกแบบระบบจะเน้นการใช้ Load Balancing และ Database Replication เพื่อให้มั่นใจว่าแม้ Server หลักจะล้ม ก็ยังมี Server สำรองให้บริการได้ทันที แม้ว่าการตอบสนองอาจช้าลง 1-2 วินาทีในช่วงที่มีปัญหา แต่ยังสามารถรับคำสั่งซื้อได้ต่อเนื่อง

Trade-off #2: Security vs Usability

อธิบาย : ความปลอดภัยและความสามารถในการใช้งานมักเป็นสิ่งที่ขัดแย้งกันอย่างหลีกเลี่ยงไม่ได้ในระบบดิจิทัล สำหรับระบบจัดส่งอาหาร:

1. ถ้าเน้น Security สูงสุด: อาจบังคับให้ผู้ใช้ต้องเข้าสู่ระบบใหม่ทุกครั้ง, ต้องใส่รหัสผ่านที่ซับซ้อน, และต้องยืนยันตัวตนด้วยรหัส OTP ก่อนชำระเงินทุกรายการ ซึ่งแม้จะปลอดภัยมาก แต่จะทำให้ขั้นตอนการสั่งอาหารยาวนานและผู้ใช้รู้สึกหงุดหงิด
2. ถ้าเน้น Usability สูงสุด: อาจอนุญาตให้ผู้ใช้เข้าสู่ระบบค้างไว้ตลอดไป, อนุญาตให้ชำระเงินด้วยบัตรที่บันทึกไว้ในคลิกเดียวโดยไม่ต้องยืนยันตัวตนเพิ่มเติม ซึ่งสะดวกมาก แต่หากโทรศัพท์ของผู้ใช้หายหรือถูกขโมย บัญชีและบัตรเครดิตของผู้ใช้อาจถูกนำไปใช้ในทางที่ผิดได้ง่าย

การตัดสินใจที่ดีที่สุดคือการสร้าง สมดุลที่เหมาะสม เพื่อรักษาความเชื่อมั่นของผู้ใช้ และยังคงความสะดวกสบายไว้

Decision : เลือกสมดุลที่เน้น Usability ภายใต้การรักษา Security

ขั้นพื้นฐานที่จำเป็น

ทำไม : ธุรกิจ: ธุรกิจ Food Delivery ขึ้นอยู่กับ ความรวดเร็วและความสะดวกสบายเป็นหลัก ผู้ใช้ต้องการสั่งอาหารให้เสร็จภายในไม่กี่นาที หากขั้นตอนความปลอดภัย

ทำให้กระบวนการสั่งซื้อล่าช้าจนเกินไป ผู้ใช้อาจเปลี่ยนไปใช้คู่แข่งทันที

แนวทาง: จะใช้วิธีการรักษาความปลอดภัยแบบ "ตามบริบท" (Contextual Security)

- Usability Focus: อนุญาตให้ผู้ใช้ บันทึกข้อมูลการเข้าสู่ระบบ ไว้ได้
- (เพื่อลดการล็อกอินซ้ำ) และอนุญาตให้ใช้ การชำระเงินในคลิกเดียว สำหรับคำสั่งซื้อที่มีมูลค่าต่ำ
- Security Safeguard: กำหนดให้ผู้ใช้ต้อง ยืนยันตัวตนเพิ่มเติม (เช่น OTP หรือ Biometrics) เฉพาะ เมื่อมีการทำธุรกรรมที่มี มูลค่าสูง หรือเมื่อมีการ เพิ่ม/แก้ไข ข้อมูลบัตรเครดิต เท่านั้น วิธีนี้ช่วยให้ผู้ใช้ส่วนใหญ่ได้รับประสบการณ์ที่รวดเร็ว ในขณะที่ข้อมูลที่สำคัญที่สุดยังคงปลอดภัย