



Software Architecture(207)  
ระบุ Architectural Drivers สำหรับโปรเจกต์ของทีม  
(LAB 2)

สมาชิก

รหัสนักศึกษา 67543210024-5 ชื่อนายสิริ รัตนรินทร์

รหัสนักศึกษา 67543210003-9 ชื่อนายปรานต์ มีเดช

รหัสนักศึกษา 67543210007-0 ชื่อฉัตรดนัย มณีนาวล

# System Overview - Hotel Booking System

## 1. System Description

Hotel Booking System คือเว็บไซต์ที่ให้ผู้ใช้หา เปรียบเทียบ และจองห้องพักได้ทันที ระบบเชื่อมผู้ใช้กับโรงแรมหลายแห่ง แสดงห้องว่างแบบ Real-time จองและจ่ายเงินจบในหน้าเดียว

ระบบถูกพัฒนาขึ้นเพื่อแก้ปัญหาการจองแบบเดิมที่ต้องติดต่อโรงแรมหลายแห่งทีละที ใช้เวลานาน และเปรียบเทียบราคาได้ยาก นอกจากนี้ โรงแรมยังสามารถจัดการห้องพัก อัปเดตราคา และ ติดตาม สถานะการจองได้อย่างมีประสิทธิภาพ ลดภาระงาน Manual ลงอย่างมาก

กลุ่มผู้ใช้หลักคือนักท่องเที่ยว นักธุรกิจผู้เดินทางที่ต้องการที่พักตลอดจนผู้จัดการโรงแรมและพนักงานที่ต้อง การ ระบบจัดการจองที่ชัดเจน

## 2. Target Users

- Guest/Customer - ผู้หาและจองห้องพัก
- Hotel Manager - ผู้ดูแลข้อมูลห้อง ราคา และนโยบาย
- Hotel Staff - ผู้รับผิดชอบ Check-in / Check-out
- System Administrator - ผู้ดูแลระบบและตั้งค่าต่างๆ

## 3. Key Features Room Search & Filtering

- Real-time Availability Check
- Online Booking
- Payment Integration
- Booking Management (View / Modify / Cancel)
- User Profile & Booking History
- Hotel Management Dashboard
- Review & Rating System
- Email Notifications
- Reporting & Analytics

# Functional Requirements

## User Management

- FR-01: ระบบรองรับ register via email & password
- FR-02: รองรับ Login/Logout สำหรับ Guest, Hotel Stagg & Admin
- FR-03: จัดการ User Roles และ Permissions ตามประเภทผู้ใช้
- FR-04: ผู้ใช้สามารถแก้ไขข้อมูลโปรไฟล์ (ชื่อ, เบอร์โทร, ที่อยู่)
- FR-05: รองรับการกู้ email & password

## Room Search & Booking

- FR-06: ค้นหาห้องพักตามวันที่ check-in/check-out จำนวนผู้เข้าพัก และสถานที่
- FR-07: แสดงเฉพาะห้องว่างแบบ Real-time
- FR-08: Filter ผลการค้นหาตามราคา คะแนนรีวิว สิ่งอำนวยความสะดวก และระยะทาง
- FR-09: แสดงรายละเอียดห้องพัก (รูปภาพ ราคา สิ่งอำนวยความสะดวก รีวิว)
- FR-10: เลือกห้องและทำการจอง พร้อมกรอกข้อมูลผู้เข้าพักและคำร้องพิเศษ

## Payment & Confirmation

- FR-11: รองรับการชำระเงินออนไลน์ผ่านบัตรเครดิต/เดบิต และ E-wallet
- FR-12: ส่งอีเมลยืนยันการจองพร้อม Booking ID หลังชำระเงินสำเร็จ
- FR-13: สร้าง QR Code หรือ Booking Voucher สำหรับเช็คอิน

## Booking Management

- FR-14: ดูประวัติการจองทั้งหมด ทั้งที่เคยเข้าพักและที่กำลังจะถึง
- FR-15: ยกเลิกการจองตามนโยบายของโรงแรม
- FR-16: แก้ไขการจอง เช่น เปลี่ยนวันหรืออัปเดตห้อง ตามเงื่อนไขที่โรงแรมกำหนด

## Hotel Management

- FR-17: เพิ่ม แก้ไข หรือลบข้อมูลห้องพัก (ประเภท ราคา รูปภาพ รายละเอียด)
- FR-18: ตั้งค่า Room Availability และบล็อกวันที่ไม่เปิดให้จอง
- FR-19: ดูและจัดการคำจองแบบ Real-time
- FR-20: ตั้งค่า Pricing Strategy เช่น Dynamic Pricing, โปรโมชั่น หรือส่วนลด

## REview & Rating

- FR-21: ผู้ที่เข้าพักแล้วสามารถเขียนรีวิวและให้คะแนนหลังเช็คอิน
- FR-22: ระบบแสดงคะแนนเฉลี่ยและรีวิวบนหน้ารายละเอียดโรงแรม

Notification

- FR-23: ส่งอีเมลแจ้งเตือนเมื่อมีการจอง ยืนยัน หรือยกเลิก
- FR-24: ส่ง Reminder Email ก่อนวันเช็คอิน 2-3 วัน
- FR-25: แจ้งเตือน Hotel Staff เมื่อมีการจองใหม่หรือมีการยกเลิก

Reporting & Analytic

- FR-26: System Admin เข้าดูรายงานรวม เช่น จำนวนการจอง รายได้ และอัตราการเข้าพัก
- FR-27: Hotel Manager สามารถ Export รายงานการจองและรายได้เป็นไฟล์ Excel/PDF

Quality Attributes & Scenarios

QA-1: Performance

Scenario:

ส่วน	รายละเอียด
Source	ผู้ใช้ทั่วไป (Guest User)
Stimulus	ผู้ใช้กดปุ่มค้นหาห้องพักโดยระบุวันที่ Check-in/Check-out และสถานที่
Artifact	Search Module และ Database Query System
Environment	สภาวะปกติ (Normal Load) - มีผู้ใช้พร้อมกัน 1,000 คน
Response	ระบบทำการ Query ฐานข้อมูล Filter ห้องว่าง และแสดงผลลัพท์การค้นหา
Response Measure	ระบบต้องแสดงผลการค้นหาภายใน 2 วินาที ในอัตราความสำเร็จ 95% ของ Requests

ในการใช้งานจริง ผู้ใช้คาดหวังว่าเมื่อทำการค้นหาห้องพัก ระบบจะแสดงผลได้อย่างรวดเร็ว การที่ผู้ใช้ต้องรอนานกว่า 2 วินาทีจะทำให้เกิด Frustration และอาจออกจากเว็บไซต์ไป Performance ที่ดีจะช่วยเพิ่ม Conversion Rate ในการจอง โดยเฉพาะในช่วง Peak Season ที่มีผู้ใช้เข้ามาค้นหาพร้อมกันจำนวนมาก ระบบต้องสามารถรับมือกับ Concurrent Requests ได้โดยไม่มี Performance Degradation

## QA-2: Scalability

Scenario:

ส่วน	รายละเอียด
Source	ผู้ใช้หลายพันคนพร้อมกัน (Spike in Traffic)
Stimulus	จำนวนผู้ใช้เพิ่มขึ้นจาก 1,000 เป็น 10,000 คนพร้อมกันในช่วง Holiday Season
Artifact	Application Servers, Database, และ Load Balancer
Environment	Peak Load - ช่วงเทศกาลหรือ Special Promotion
Response	ระบบทำการ Auto-scale เพิ่ม Server Instances และกระจาย Load
Response Measure	ระบบต้องรักษา Response Time ไม่เกิน 3 วินาที และ Uptime ไม่ต่ำกว่า 99.5%

ธุรกิจโรงแรมมี Peak Seasons ที่ชัดเจน เช่น ช่วงปีใหม่ สงกรานต์ หรือช่วงที่มี Promotion พิเศษ ในช่วงเวลาดังกล่าว Traffic สามารถเพิ่มขึ้นอย่างรวดเร็วและมีจำนวนมาก ระบบจำเป็นต้อง Scale Out ได้อย่างอัตโนมัติเพื่อรองรับ Load ที่เพิ่มขึ้นโดยไม่กระทบต่อ User Experience การที่ระบบล่มหรือช้าลงในช่วงนี้จะส่งผลให้เสีย Revenue และ Brand Reputation อย่างมาก

## QA-3: Security

Scenario:

ส่วน	รายละเอียด
Source	Malicious User หรือ Hacker
Stimulus	มีความพยายาม SQL Injection หรือ Unauthorized Access เข้าถึงข้อมูลการชำระเงิน
Artifact	Authentication System, Payment Gateway และ Database
Environment	Runtime - ระบบกำลังให้บริการปกติ
Response	ระบบตรวจจับ Malicious Request, Block การเข้าถึง, Log Security Event และแจ้งเตือน Admin
Response Measure	Block Attack ได้ 100% และแจ้งเตือน Admin ภายใน 5 นาที โดยไม่มีข้อมูล Sensitive รั่วไหล

ระบบ Hotel Booking จัดเก็บข้อมูลที่ละเอียดอ่อนมาก รวมถึงข้อมูลส่วนตัวของลูกค้า ข้อมูลบัตรเครดิต และประวัติการจอง การถูกโจมตีหรือข้อมูลรั่วไหลจะส่งผลกระทบร้ายแรงทั้งต่อลูกค้าและธุรกิจ อาจถูกฟ้องร้องตาม PDPA หรือกฎหมายคุ้มครองข้อมูลอื่นๆ และเสียชื่อเสียงอย่างถาวร ดังนั้น Security จึงเป็น Priority สูงสุด

#### QA-4: Availability

Scenario:

ส่วน	รายละเอียด
Source	Hardware Failure หรือ Network Outage
Stimulus	Primary Database Server ล่มหรือเกิด Network Interruption
Artifact	Database Server, Application Servers และ Network Infrastructure
Environment	Runtime - ระบบกำลังให้บริการ มี Active Bookings กำลังดำเนินการ
Response	ระบบทำ Automatic Failover ไปยัง Backup Server และกู้คืนบริการ
Response Measure	Downtime ไม่เกิน 5 นาทีต่อเดือน (99.9% Uptime) และไม่มี Transaction สูญหาย

การจองโรงแรมเป็นบริการที่ลูกค้าคาดหวังให้ใช้งานได้ตลอด 24/7

เนื่องจากลูกค้าอาจจองห้องพักจากทั่วโลกในเขตเวลาต่างๆ กัน

การที่ระบบล่มแม้เพียงชั่วโมเมนต์เดียวอาจทำให้เสียโอกาสในการขายและลูกค้าหันไปใช้คู่แข่ง นอกจากนี้

หากระบบล่มขณะที่ลูกค้ากำลังชำระเงิน อาจทำให้เกิด Double Charge หรือ Lost Transaction

ซึ่งส่งผลให้ลูกค้าไม่พอใจและเกิดปัญหาตามมา

#### QA-5: Maintainability

Scenario:

ส่วน	รายละเอียด
Source	Development Team
Stimulus	ต้องเพิ่ม Feature ใหม่ (เช่น ระบบ Loyalty Points) หรือแก้ไข Bug ที่พบ
Artifact	Source Code, Database Schema และ API Endpoints
Environment	Development และ Deployment Phase
Response	Developer สามารถเพิ่ม Feature หรือแก้ไข Bug โดยไม่กระทบต่อ Module อื่น
Response Measure	สามารถ Deploy การเปลี่ยนแปลงภายใน 2 ชั่วโมงหลังจาก Code Review และ Testing เสร็จสิ้น

ธุรกิจโรงแรมมีการเปลี่ยนแปลงอยู่ตลอดเวลา ไม่ว่าจะเป็น Promotion ใหม่ การเพิ่ม Payment Method หรือการปรับปรุง UI/UX ตาม User Feedback ระบบที่มี Maintainability สูงจะช่วยให้ทีม Development สามารถตอบสนองต่อการเปลี่ยนแปลงได้รวดเร็ว ลดค่าใช้จ่ายในการ Maintenance และลด Risk ที่การแก้ไขส่วนหนึ่งจะไปทำให้ส่วนอื่นเสีย (Regression) ซึ่งต้องอาศัย Clean Code, Good Architecture และ Comprehensive Testing

QA-6: Usability  
Scenario

ส่วน	รายละเอียด
Source	First-time User (ผู้ใช้ใหม่ที่ไม่คุ้นเคยกับระบบ)
Stimulus	ผู้ใช้เข้ามาในเว็บไซต์และต้องการจองห้องพักครั้งแรก
Artifact	User Interface (Search, Booking Flow, Payment Page)
Environment	Normal Usage - ผู้ใช้เข้าถึงผ่าน Desktop Browser หรือ Mobile Device
Response	ระบบแนะนำขั้นตอนการจองที่ชัดเจนและผู้ใช้สามารถทำตามได้
Response Measure	ผู้ใช้สามารถทำการจองสำเร็จภายใน 5 นาทีโดยไม่ต้องดู Tutorial และมี Success Rate ไม่ต่ำกว่า 85%

Usability เป็นปัจจัยสำคัญที่ส่งผลต่อ Conversion Rate โดยตรง หากกระบวนการจองซับซ้อนหรือสับสน ผู้ใช้จะออกจากเว็บไซต์ก่อนทำการจอง (High Bounce Rate) Interface ที่ดีต้องใช้งานง่าย ชัดเจน มี Clear Call-to-Action และรองรับทั้ง Desktop และ Mobile อย่างลงตัว การลด Friction ในกระบวนการจองจะช่วยเพิ่มยอดขายและความพึงพอใจของลูกค้า โดยเฉพาะผู้ใช้ที่ไม่มีความชำนาญด้านเทคโนโลยีสูง

## Constraints

### Technical Constraints

- ต้องพัฒนาด้วย Web Technologies ซึ่งระบบต้องเป็น Web-based Application รองรับ browsers ต่างๆ (Chrome, Firefox, MS edge)
- รองรับ Mobile Responsive โดยที่ UI ใช้งานได้ทั้งบน Desktop และ Mobile ได้
- Database ต้องเป็น Relational Database โดยใช้ PostgreSQL หรือ MySQL เพื่อรองรับ Transaction และ Data Integrity
- ต้องใช้ Payment Gateway ที่ได้รับการรับรองตามมาตรฐาน PCI DSS
- ต้องใช้ HTTPS ในการสื่อสารทั้งหมด

### Time Constraints

- Phase 1 เสร็จภายใน 4 เดือนโดยประกอบด้วยฟีเจอร์หลักเช่น Search, Booking, Payment
- Phase 2 เสร็จภายใน 2 เดือนถัดไปซึ่งจะเพิ่ม Review System, Analytics Dashboard
- การ Sprint ละ 2 สัปดาห์โดยการ Development จะเป็นแบบ Sprint แต่ละรอบจะใช้เวลา 2 สัปดาห์ตาม Agile Methodology
- Testing Period 2 สัปดาห์โดยต้องมี UAT Testing ก่อน Production Launch

### Budget Constraints

- งบรวมไม่เกิน 800,000 บาทรวมค่า Development, Infrastructure และ Third-party Services
- Cloud Hosting Budget 10,000 บาทต่อเดือนสำหรับ Cloud Infrastructure (AWS/GCP)
- ไม่สามารถใช้ Premium Analytics Tools และต้องใช้ Open Source หรือ Free Tier Services
- Payment Gateway Fee ต้องไม่เกิน 3% ต่อ Transaction

### Legal/Policy Constraints

- ต้องเป็นไปตาม PDPA โดยจะจัดเก็บและใช้ข้อมูลส่วนบุคคลตามกฎหมาย
- ต้อง Comply กับ PCI DSS เพื่อตามมาตรฐานความปลอดภัยสำหรับการรับชำระเงินด้วยบัตร
- เก็บ Audit Log ไว้อย่างน้อย 1 ปีสำหรับ Transaction และ Security Events
- ต้องมี Terms of Service และ Privacy Policy ซึ่งมีเอกสารกฎหมายให้ผู้ใช้อยอมรับ
- ต้องรองรับการคืนเงินตาม Consumer Protection Act  
ระบบต้องรองรับกระบวนการคืนเงินที่ถูกต้องตามกฎหมาย

## Assumptions

1. ผู้ใช้มีอุปกรณ์ที่รองรับ HTML5 โดยสมมติว่าผู้ใช้ส่วนใหญ่ใช้ Modern Browsers และอุปกรณ์ที่รองรับ HTML5, CSS3 และ JavaScript
2. มี Internet Connection  
ที่เสถียรซึ่งผู้มีการเชื่อมต่ออินเทอร์เน็ตที่มีความเร็วเพียงพอในการใช้งานระบบ (อย่างน้อย 3G)
3. โรงแรมมีระบบจัดการห้องพักพื้นฐานโดยโรงแรมที่ร่วมงานมีข้อมูลห้องพักและราคาที่พร้อมใช้งาน



4. Payment Gateway มี Uptime 99.9% ซึ่ง Third-party Payment Service ที่ใช้มีความเสถียรสูงและพร้อมให้บริการตลอด
5. ลูกค้ามี Email Address ที่ Active สำหรับรับ Booking Confirmation
6. Database สามารถ Scale ได้โดย Infrastructure รองรับการ Scale Database แนวนอนหรือแนวตั้งตามความต้องการ
7. มี CDN สำหรับ Static Assets ใช้ Content Delivery Network เพื่อเร่งความเร็วในการโหลดรูปภาพและ Static Files
8. Hotel Staff มีความรู้พื้นฐานด้านคอมพิวเตอร์โดยพนักงานโรงแรมสามารถใช้งานระบบ Dashboard ได้หลังผ่านการอบรมสั้นๆ

### Quality Attributes Priority

Rank	Quality Attribute	เหตุผล
1	Security	เนื่องจากระบบจัดเก็บข้อมูลการชำระเงินและข้อมูลส่วนบุคคลที่ละเอียดอ่อน การละเมิดความปลอดภัยจะส่งผลกระทบร้ายแรงต่อลูกค้าและธุรกิจ อาจนำไปสู่การฟ้องร้องและความเสียหายต่อชื่อเสียงในระยะยาว ดังนั้น การรักษาความปลอดภัยจึงเป็นสิ่งสำคัญสูงสุดที่ไม่อาจประนีประนอมได้
2	Availability	ลูกค้าต้องสามารถจองห้องได้ตลอด 24 ชั่วโมงจากทุกภูมิภาค Downtime แม้เพียงเล็กน้อยอาจทำให้สูญเสียรายได้และลูกค้าไปยังคู่แข่ง ดังนั้น การรักษา Uptime สูงจึงเป็นสิ่งจำเป็นต่อความสำเร็จของธุรกิจ
3	Performance	ผู้ใช้คาดหวังเวลาในการตอบสนองที่รวดเร็ว โดยเฉพาะช่วงค้นหาและจองห้อง ประสิทธิภาพของระบบส่งผลโดยตรงต่ออัตราการจองและความพึงพอใจของลูกค้า หากระบบช้า ผู้ใช้มีแนวโน้มออกจากเว็บไซต์ทันที
4	Usability	อินเทอร์เฟซที่ใช้งานง่ายและชัดเจนช่วยลดอัตราการออกจากหน้าและเพิ่มยอดการจอง หากขั้นตอนจองซับซ้อนหรือทำให้สับสน ผู้ใช้อาจละทิ้งกระบวนการก่อนเสร็จสิ้น ดังนั้น ความสามารถในการใช้งานจึงมีผลโดยตรงต่อความสำเร็จของธุรกิจ

		จ
5	Scalability	ธุรกิจโรงแรมมีช่วง Peak Season ที่ปริมาณผู้ใช้งานเพิ่มขึ้นอย่างมาก ระบบจึงต้องสามารถขยายรองรับโหลดที่สูงขึ้นได้แม้จะเกิดไม่บ่อย หากรองรับไม่ได้จะส่งผลกระทบต่อรายได้และประสบการณ์ของลูกค้าอย่างรุนแรง
6	Maintainability	แม้การบำรุงรักษาระบบจะสำคัญต่อการพัฒนาในระยะยาว แต่เมื่อเทียบกับคุณสมบัติอื่นที่กระทบต่อผู้ใช้และความปลอดภัยโดยตรง จึงถูกจัดลำดับความสำคัญรองลงมา อย่างไรก็ตาม ยังคงต้องให้ความสำคัญเพื่อช่วยลดต้นทุนการดูแลระบบในอนาคต

## Trade-offs Analysis

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

สถานการณ์:

การเข้ารหัสข้อมูลทุกครั้งที่มีการสื่อสารระหว่าง Client-Server และการ Hash Password หลายรอบจะช่วยเพิ่มความปลอดภัย แต่จะทำให้ Response Time ช้าลงเนื่องจากต้องใช้ Computing Power มากขึ้น โดยเฉพาะการเข้ารหัสข้อมูลบัตรเครดิตและการ Hash Password ด้วย bcrypt รอบสูง (12-14 rounds) อาจทำให้ Login และ Payment Process ช้าลง

Decision:

เราเลือก Option A (Prioritize Security) เนื่องจาก Security เป็น Priority อันดับ 1 และไม่สามารถประนีประนอมได้ การเพิ่ม Response Time 200-500ms ยังคงอยู่ในระดับที่ยอมรับได้ (ภายใต้ 2 วินาที) และเราสามารถใช้เทคนิคอื่นๆ เช่น Asynchronous Processing และ CDN เพื่อชดเชย Performance ที่เสียไป

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

สถานการณ์:

การรักษา Availability 99.99% (Downtime ไม่เกิน 52 นาทีต่อปี) จะต้องมี Multi-region Deployment, Load Balancers, Auto-scaling และ Database Replication แบบ Multi-master ซึ่งมีค่าใช้จ่ายสูงมาก (มากกว่าแบบ Single-region ประมาณ 3-4 เท่า) ในขณะที่ Budget มีจำกัดอยู่ที่ 10,000 บาทต่อเดือนสำหรับ Cloud Infrastructure

Decision:

เราเลือก Option B (99.5% Uptime) เนื่องจาก Budget Constraint เป็นข้อจำกัดที่สำคัญในระยะเริ่มต้น 99.5% Uptime (Downtime ประมาณ 3-4 ชั่วโมงต่อปี) ยังคงอยู่ในระดับที่ยอมรับได้สำหรับธุรกิจขนาดกลาง และเราวางแผนที่จะ Upgrade เป็น Option C เมื่อธุรกิจเติบโตและมี Revenue เพียงพอ

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

สถานการณ์:

มี Feature Requests มากมายจากผู้มีส่วนได้ส่วนเสีย เช่น Loyalty Program, Multi-language Support, Virtual Tour, AI Recommendation System และ Social Media Integration หากต้องการพัฒนาครบทุก Feature อาจใช้เวลาจนถึง 8-10 เดือน แต่ตลาดมีการแข่งขันสูงและต้องการ Launch เร็วเพื่อครองส่วนแบ่งตลาด

Decision:

เราเลือก Option B (MVP First) เนื่องจากตลาดมีการแข่งขันสูงและ Time-to-Market เป็นปัจจัยสำคัญ การ Launch เร็วจะช่วยให้เราได้ Feedback จากผู้ใช้งานจริงและสามารถปรับปรุงระบบตามความต้องการที่แท้จริง แทนที่จะพัฒนา Features ที่อาจไม่ตรงกับความต้องการในระยะเวลานาน (Agile over Waterfall)



