



Architectural Drivers Document

จัดทำโดย

นาย พิชณน์ สิ้นธรรสวัสดิ์ 67543210061-7

รายงานนี้เป็นส่วนหนึ่งของวิชา สถาปัตยกรรมซอฟต์แวร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา

System Overview

Hotel Booking System - ระบบจองโรงแรม

1. System Description

Hotel Booking System เป็นระบบออนไลน์สำหรับค้นหา เลือก และจองโรงแรมอย่างสะดวก ผู้ใช้สามารถดูห้องว่าง ราคาตามช่วงเวลา และทำการจองพร้อมชำระเงินได้ทันที ส่วนโรงแรมสามารถอัปเดตข้อมูลห้องพัก จัดการราคา และตรวจสอบยอดการจองได้แบบเรียลไทม์

ระบบนี้ช่วยลดความผิดพลาดจากการจองแบบเดิม เพิ่มความรวดเร็ว โปร่งใส และทำให้ทั้งผู้เข้าพักและผู้ให้บริการจัดการการจองได้ง่ายขึ้นในแพลตฟอร์มเดียว

2. Target Users

- 2.1 นักท่องเที่ยวต่างไทยและประเทศ
- 2.2 บริษัททัวร์
- 2.3 ไกด์นำเที่ยว
- 2.4 ผู้ประกอบการ/บริษัท ทั้งภาครัฐและเอกชน
- 2.5 เจ้าของกิจการโรงแรม ห้องพักรายวัน

3. Key Features

- 3.1 ค้นหาโรงแรมและห้องพัก ตามสถานที่ วันที่ และงบประมาณ
- 3.2 แสดงจำนวนห้องว่างแบบเรียลไทม์ (Real-time Availability)
- 3.3 รายละเอียดโรงแรมครบถ้วน เช่น ราคา สิ่งอำนวยความสะดวก รูปภาพ รีวิว

- 3.4 ระบบจองและชำระเงินออนไลน์ รองรับหลายช่องทาง
- 3.5 การยืนยันการจองอัตโนมัติ ผ่านอีเมลหรือ SMS
- 3.6 จัดการห้องพักและราคา สำหรับโรงแรม เช่น เปิด-ปิดห้อง ปรับราคา
- 3.7 ระบบรีวิวและให้คะแนน จากผู้เข้าพัก
- 3.8 แดชบอร์ดสำหรับผู้ดูแลระบบ เพื่อจัดการข้อมูลโรงแรม ผู้ใช้ และรายการจอง
- 3.9 ฟิลเตอร์ค้นหาขั้นสูง เช่น ระดับดาว สิ่งอำนวยความสะดวก ประเภทเตียง
- 3.10 ระบบรายงานและสถิติ เช่น อัตราการเข้าพัก รายได้ต่อวัน/เดือน

Functional Requirements

4. Functional Requirements

4.1 User Management

- FR-01: ผู้ใช้สามารถ Register และ Login ได้
- FR-02: ผู้ใช้สามารถแก้ไขโปรไฟล์และเปลี่ยนรหัสผ่านได้
- FR-03: ผู้ใช้สามารถกู้รหัสผ่านผ่านอีเมลได้
- FR-04: ผู้ดูแลระบบสามารถจัดการผู้ใช้งาน (ดูข้อมูล/ระงับบัญชี) ได้
- FR-05: ระบบต้องตรวจสอบสิทธิ์การใช้งานตามบทบาท (User / Hotel Owner / Admin)

4.2 Core Features

- FR-06: ผู้ใช้สามารถค้นหาโรงแรมตามสถานที่ วันที่ และจำนวนคนได้
- FR-07: ระบบต้องแสดงจำนวนห้องว่างแบบเรียลไทม์
- FR-08: ผู้ใช้สามารถเลือกห้อง จอง และชำระเงินออนไลน์ได้
- FR-09: ผู้ใช้สามารถยกเลิกการจองตามเงื่อนไขได้
- FR-10: ผู้ใช้สามารถดูประวัติการจองได้
- FR-11: โรงแรมสามารถเพิ่ม/แก้ไข/ลบข้อมูลห้องพักและราคาได้
- FR-12: ระบบต้องป้องกันการจองซ้ำซ้อน (Double Booking)
- FR-13: ผู้ใช้สามารถรีวิวและให้คะแนนโรงแรมได้

FR-14: ระบบต้องสร้างรหัสการจอง (Booking ID) ให้ทุกการจอง

4.3 Notification

FR-15: ระบบส่งอีเมลยืนยันการสมัครสมาชิก

FR-16: ระบบส่งอีเมลยืนยันการจองและการชำระเงิน

FR-17: ระบบส่งแจ้งเตือนให้โรงแรมเมื่อมีการจองใหม่

FR-18: ระบบแจ้งผู้ใช้หากชำระเงินไม่สำเร็จหรือมีการเปลี่ยนแปลงการจอง

4.4 Reporting

FR-19: โรงแรมสามารถดูรายงานยอดการจองและรายได้ได้

FR-20: ระบบสามารถส่งออกรายงานเป็นไฟล์ PDF หรือ Excel ได้

Quality Attributes & Scenarios

5. Quality Attributes

QA-1: Availability

Scenario:

| ส่วน | รายละเอียด |
|------------------|---|
| Source | Users |
| Stimulus | มีการร้องขอสิทธิ์การใช้งานมากเกินไป |
| Artifact | Server/ Database |
| Environment | Peak time |
| Response | Session Timeout แจ้งเตือน ลองใหม่อีกครั้ง |
| Response Measure | Response Time 1m |

คำอธิบายแบบเต็ม:

เมื่อผู้ใช้เข้าใช้งานมากเกินไป ระบบจะระงับการเข้าใช้งานของผู้ใช้บางส่วนเป็นการชั่วคราว เพื่อแบ่งพื้นที่ให้ผู้ใช้งานอีกส่วนหนึ่งได้ใช้งาน จนกว่าผู้ใช้งานที่ใช้อยู่จะลดลง จากนั้นค่อยให้ผู้ใช้งานที่ถูกระบบระงับไว้ใช้งานได้

QA-2: Scalability

Scenario:

| ส่วน | รายละเอียด |
|------------------|--|
| Source | Back-end team |
| Stimulus | ต้องการระบบที่มีความยืดหยุ่นเพื่อรองรับการใช้งานที่อาจเพิ่มขึ้นในอนาคต |
| Artifact | ระบบทั้งหมด |
| Environment | Normal time |
| Response | ออกแบบระบบให้ง่ายต่อการดัดแปลง |
| Response Measure | - |

คำอธิบายแบบเต็ม:

ทีมนักพัฒนาต้องการระบบที่มีความยืดหยุ่นเพื่อนำมารองรับการใช้งานที่อาจจะเพิ่มขึ้นในอนาคต โดยการออกแบบระบบให้ง่ายต่อการพัฒนาแบบต่อเนื่อง โดยไม่ให้เกิด Down-time

QA-3: Performance

Scenario:

| ส่วน | รายละเอียด |
|-------------|---|
| Source | Users |
| Stimulus | ต้องการใช้งานจากอุปกรณ์ที่มีสมรรถภาพต่ำ |
| Artifact | Web App/ Application |
| Environment | Normal-time |
| Response | เปลี่ยนไปใช้โหมดคุณภาพต่ำ |

| | |
|------------------|-------------------|
| Response Measure | Response Time 2ms |
|------------------|-------------------|

คำอธิบายแบบเต็ม:

โหลดคุณภาพต่ำ สำหรับผู้ใช้งานเข้าใช้งานด้วยอุปกรณ์รุ่นเก่า หรือมีสมรรถภาพที่ต่ำกว่าระบบออกแบบไว้ เพื่อให้การใช้งานตอบสนองไวในทุกอุปกรณ์ โดยอาจจะลดการโหลดของ Element ลด Animation และอื่นๆ

QA-4: Security

Scenario:

| ส่วน | รายละเอียด |
|------------------|---|
| Source | ผู้ใช้งานต้องสงสัย |
| Stimulus | เขาใช้งานหลายครั้งมากเกินไปในเวลาเดียวกัน |
| Artifact | Web App/ Application |
| Environment | Normal-time |
| Response | ระบบการใช้งานจนกว่าเจ้าหน้าที่จะตรวจสอบ |
| Response Measure | Response Time 1s |

คำอธิบายแบบเต็ม:

เมื่อมีผู้ใช้งานต้องสงสัยเข้าใช้งานจาก IP เดียวกันหลายๆครั้งในเวลาเดียวกัน ระบบจะทำการระงับการใช้งาน โดยให้ติดต่อเจ้าหน้าที่ เพื่อให้เจ้าหน้าที่พิจารณาต่อไป

QA-5: Modifiability

Scenario:

| ส่วน | รายละเอียด |
|------------------|--|
| Source | Users |
| Stimulus | ต้องการปรับแต่งหน้าการใช้งานเช่น เปลี่ยนเป็น Darkmode เพิ่มขนาด ตัวหนังสือ หรือต้องการปรับเปลี่ยนข้อมูล ความเป็นส่วนตัว |
| Artifact | Web App/ Application |
| Environment | Normal-time |
| Response | สามารถปรับแต่งได้ในเมนู Settings |
| Response Measure | Response Time 2ms |

คำอธิบายแบบเต็ม:

เมนู Settings สามารถทำได้หลายอย่าง ทั้งปรับแต่ง UI การกำหนดชื่อผู้ใช้งาน
ปรับเปลี่ยนข้อมูลความเป็นส่วนตัว เพิ่มความปลอดภัยในการใช้งาน

QA-6: Usability

Scenario:

| ส่วน | รายละเอียด |
|-------------|-------------------|
| Source | Users |
| Stimulus | เข้าใช้งานตามปกติ |
| Artifact | UX/UI |
| Environment | Normal-time |

| | |
|------------------|--|
| Response | UI มีความสวยงาม หน้าใช้ มีขนาดตัวหนังสือ ที่เหมาะสมสำหรับขนาดหน้าจอต่างๆ สีมี Contrast ที่สามารถอ่านได้สบายตา สามารถ ใช้งานจากหน้าจอแบบไหนก็ได้ |
| Response Measure | Response Time 2s |

คำอธิบายแบบเต็ม:

การออกแบบ UX/UI เป็นเรื่องที่สำคัญ เพื่อให้ใช้งานง่ายสะดวก หน้าใช้งาน

Constraints & Assumptions

6. Constraints

6.1 Technical Constraints

C-01: ระบบจะไม่รองรับอุปกรณ์รุ่นเก่าหรืออุปกรณ์ที่มีประสิทธิภาพต่ำ เนื่องจากข้อจำกัดด้านเทคโนโลยี ทำให้ต้องใช้ hardware และ software ที่ทันสมัยเท่านั้น

C-02: ระบบต้องรองรับช่องทางการชำระเงินผ่านแอปพลิเคชันธนาคารที่มีความปลอดภัยสูง ซึ่งจำกัดการออกแบบระบบให้ต้องทำงานร่วมกับบริการธนาคารที่ได้รับการรับรองเท่านั้น

6.2 Time Constraints

C-03: ระบบไม่จำเป็นต้องจัดอบรมผู้ใช้งานเชิงลึก ทำให้ระยะเวลาในการพัฒนาสามารถเน้นไปที่การออกแบบฟังก์ชันหลักและการทดสอบแทนการทำ training module

TM-04: การรองรับเฉพาะอุปกรณ์และช่องทางการชำระเงินที่จำกัดช่วยลดเวลาในการทดสอบ compatibility และ integration

6.3 Budget Constraints

C-05: การจำกัดการรองรับเฉพาะอุปกรณ์ใหม่และช่องทางการชำระเงินที่ปลอดภัยช่วยลดค่าใช้จ่ายด้านการปรับปรุงระบบให้เข้ากับอุปกรณ์เก่าและหลายช่องทางการจ่ายเงิน

C-06: ไม่ต้องจัดอบรมผู้ใช้งานจำนวนมากช่วยลดค่าใช้จ่ายด้าน training

6.4 Legal/Policy Constraints

C-07: ระบบจะต้องทำตามเงื่อนไขข้อตกลงตาม พระราชบัญญัติคุ้มครองข้อมูลส่วนบุคคล พ.ศ. 2562 หรือ PDPA และทำให้มั่นใจว่าข้อมูลผู้ใช้ถูกจัดการอย่างปลอดภัย

7. Assumptions

7.1 Technical Assumptions

A-01: ระบบไม่มีการกำหนดรองรับการใช้งานผ่านอุปกรณ์รุ่นเก่า หรืออุปกรณ์ที่มีประสิทธิภาพ

A-03: ระบบต้องเชื่อมต่อกับอินเทอร์เน็ตตลอดเวลา เนื่องจากไม่มีฟังก์ชัน Offline

7.2 Operational Assumption

A-02: ผู้ใช้งานระบบจะมีประสบการณ์และความสามารถด้านเทคโนโลยีพอสมควร ดังนั้นจึงไม่ต้องการอบรมการใช้งาน แต่อาจจะมีการแนะนำวิธีใช้เล็กน้อย

7.3 Business Assumption

A-03: ระบบจะใช้ช่องทางการชำระเงินผ่านระบบแอปพลิเคชันธนาคารที่มีความปลอดภัยสูงเท่านั้น เพื่อให้ง่ายต่อการพัฒนาระบบและเพิ่มความปลอดภัย

Priority & Trade-offs

8 Quality Attributes Priority

| Rank | Quality Attribute | เหตุผล |
|------|-------------------|--|
| 1 | Availability | เพื่อไม่ให้เกิด Down-time มากเกินไปอาจนำไปสู่การสูญเสียกำไรไปมาก |
| 2 | Performance | เพื่อให้ผู้ใช้งานใช้งานได้โดยไม่ติดขัดสิ้นไหน |
| 3 | Security | เพื่อไม่ให้เกิดการรั่วไหลของข้อมูลภายใน |
| 4 | Usability | เพื่อให้สะดวกต่อการใช้งาน |
| 5 | Modifiability | ผู้ใช้งานมากมายตามมีความต้องการไม่เหมือนกัน ต้องคอยตอบสนองให้การใช้งานราบรื่นอยู่บ้าง |
| 6 | Scalability | การปรับปรุงการพัฒนาต้องเป็นอย่างค่อยๆ เป็นค่อยๆ ไป รอรับ Feet back จากผู้ใช้งานก่อนจะลงมือทำ |

9. Trade-offs Analysis

T-01: Availability vs Scalability

Availability เป็นปัจจัยสำคัญในการพัฒนาเนื่องจากระบบที่ดีคือระบบที่มีความพร้อมต่อการใช้งานในทุกเวลา ถ้าผู้ใช้งานสามารถใช้งานได้ตลอดเวลา ก็จะไม่เกิดการสูญเสียแต่เพิ่มพูนกำไรให้กับบริษัท

Decision: การปรับเปลี่ยนส่วนต่างๆ ของระบบมากเกินไป อาจจะทำให้เกิด Bug ที่อาจทำให้ผู้ใช้งานลดลง หรือเกิด Down-time