

Meeting Room Booking System

(ระบบจองห้องประชุมภายในองค์กร)

1. Product Vision (วิสัยทัศน์ผลิตภัณฑ์)

"สร้างระบบบริหารจัดการพื้นที่ประชุมที่เร็วอยู่ต่อและแม่นยำ เพื่อให้พนักงานสามารถจัดสรรเวลาประชุมได้อย่างมั่นใจ โดยขจัดปัญหาความขัดแย้งเรื่องสถานที่ และช่วยให้การทำงานร่วมกันภายในองค์กรลื่นไหลที่สุด"

2. Product Goal (เป้าหมายหลัก)

"ภายในระยะเวลาการพัฒนา ระบบต้องรองรับการจองห้องประชุมของพนักงานทั้งองค์กรได้โดยมีอัตราการจองซ้ำซ้อนเป็น 0% (Zero Double Booking) ผู้ใช้สามารถจองเสร็จสิ้นภายใน 4 คลิก และระบบมีการแจ้งเตือนที่แม่นยำทุกครั้ง"

3. Current Pain Points (ปัญหาปัจจุบันที่ต้องแก้ไข)

จองห้องซ้ำซ้อน: ข้อมูลไม่ถูกอัปเดต ทำให้มีคนไปใช้ห้องพร้อมกัน

ไม่รู้สถานะห้องว่าง: ต้องเดินไปดูหน้าห้อง หรือโทรเช็ค

ไม่มีการแจ้งเตือน: ลืมเวลาประชุม หรือไม่รู้ว่าการจองสำเร็จหรือไม่

4. System Objectives (สิ่งที่ระบบต้องทำได้)

4.1. แสดงสถานะห้องว่างแบบ Real-time

4.2. จองห้องและยกเลิกการจองได้ด้วยตนเอง

4.3. แจ้งเตือนทางอีเมล/แอป เมื่อจองสำเร็จและเมื่อใกล้ถึงเวลาประชุม

4.4. UX/UI ใช้งานง่าย ไม่ซับซ้อน ลด Learning curve ของพนักงาน

5. Anticipated Problems & Solutions (ปัญหาที่คาดว่าจะเกิดและแนวทางป้องกัน)

นอกเหนือจากฟังก์ชันพื้นฐาน ทีม Dev ต้องพิจารณา Logic เพื่อรับกรณีเหล่านี้ (Added Requirements):

ปัญหาที่คาดว่าจะเกิด	แนวทางแก้ไข/fix/feature ที่ต้องมีในระบบ
1. การจองก็อก (Ghost Meeting) จะไปไว้แต่ไม่มาใช้ห้อง ทำให้คนอื่นเสียโอกาส	ระบบ Check-in / Auto-release: ผู้จองต้องกด Check-in (ผ่าน QR หรือ App) ภายใน 15 นาที หากไม่มา ระบบจะตัดสิทธิ์และปล่อยห้องไว้สำหรับคนอื่น
2. การจองห้องใหญ่เกินความจำเป็น ประมาณ 2 คน แต่จองห้องใหญ่ 20 ที่นั่ง	Smart Recommendation: ระบบแนะนำห้องที่เหมาะสมกับจำนวนคน หรือแจ้งเตือนหากจำนวนคนน้อยกว่าเกณฑ์ที่ห้องกำหนด
3. อุปกรณ์ในห้องชำรุด จะนำไปแล้วแต่ใช้งานไม่ได้ (เช่น แอร์เสีย, โปรเจคเตอร์พัง)	Facility Reporting: เพิ่มปุ่ม "แจ้งอุปกรณ์ชำรุด" ในระบบ และระบบทะ Block ห้องนั้นอัตโนมัติ หรือแจ้งเตือนผู้จองคนถัดไป
4. การจองผูกขาด (Recurring Booking) จองเช่ายาวทั้งปี แต่ไม่ได้ใช้จริงทุกครั้ง	Recurring Limits: จำกัดการจองล่วงหน้าแบบซ้ำ (Recurring) ได้สูงสุดไม่เกิน 4 สัปดาห์ เพื่อบังคับให้มีการ Review ความจำเป็นใหม่
5. สิทธิ์การเข้าถึงห้อง (Access Control) ห้องผู้บริหาร หรือห้อง VIP ถูกจองโดยไม่ได้รับอนุญาต	Approval Workflow: สำหรับห้องพิเศษ (VIP) การจองจะสมบูรณ์ก็ต่อเมื่อได้รับการ "กดอนุมัติ" จาก Admin หรือเลขานุการ และเท่านั้น

6. Product Backlog (ลำดับความสำคัญงาน)

ลำดับ	รายการ (Feature / User Story)	Priority	จุดประสงค์ / ปัญหาที่แก้ไข
1	ระบบตรวจสอบสถานะห้องว่าง (Real-time Availability) <i>User Story:</i> ในฐานะผู้ใช้ ฉันต้องการเห็นว่าห้องไหนว่าง/ไม่ว่าง ในวันและเวลาที่ต้องการ	High	เพื่อให้ผู้ใช้รู้ข้อมูลห้องที่ไม่ต้องเดินไปดูเอง (แก้ Pain Point เดิม)
2	ระบบจองห้องประชุม (Create Booking) <i>User Story:</i> ในฐานะผู้ใช้ ฉันต้องการจองห้องโดยระบุเวลา, ห้อง, และหัวข้อการประชุม	High	ฟังก์ชันหลักของระบบ

3	ระบบป้องกันการจองซ้ำ (Conflict Detection Logic) User Story: ในฐานะระบบ ฉันต้องปฏิเสธการจองที่กระทำเวลาทับซ้อนกับ Booking ที่มีอยู่	High	สำคัญที่สุด เพื่อให้เป้าหมาย "Zero Double Booking" เป็นจริง
4	ระบบยกเลิกการจอง (Cancel Booking) User Story: ในฐานะผู้ใช้ ฉันต้องการยกเลิกการจองเพื่อให้คนอื่นจองต่อได้	High	พื้นฐานการจัดการ Resource
5	ระบบแจ้งเตือนทางอีเมล (Email Notification) User Story: ในฐานะผู้ใช้ ฉันต้องการได้รับอีเมลยืนยันเมื่อจองเสร็จ และเตือนล่วงหน้า 15 นาที	Medium	แก้ Pain Point เรื่องการลีบเวลา หรือไม่แน่ใจว่าจะองได้ไหม
6	ระบบเช็คอินและตัดสิทธิ์อัตโนมัติ (Check-in & Auto-release) User Story: ในฐานะระบบ ฉันจะยกเลิกการจองอัตโนมัติถ้าผู้ใช้มีกด Check-in ภายใน 15 นาที	Medium	แก้ปัญหาที่ 1: Ghost Meeting (จองก็กลัวไม่มา)
7	ระบบจำกัดการจองซ้ำ (Limit Recurring Booking) User Story: ในฐานะระบบ ฉันจะไม่อนุญาตให้จองล่วงหน้าแบบซ้ำ (Recurring) เกิน 4 สัปดาห์	Medium	แก้ปัญหาที่ 4: การจองผูกขาด (จองแข่งข้าว)
8	ระบบจัดการห้องประชุม (Admin Management) User Story: ในฐานะ Admin ฉันต้องการเพิ่ม/ลบ/แก้ไข ข้อมูลห้องและสถานะห้อง	Medium	เพื่อการดูแลระบบระยะยาว
9	ระบบอนุมัติการจอง (Approval Workflow) User Story: ในฐานะผู้อนุมัติ ฉันต้องกด Approve ก่อน การจองห้อง VIP จึงจะสมบูรณ์	Low	แก้ปัญหาที่ 5: สิทธิ์การเข้าถึงห้อง(เฉพาะห้อง VIP/ผู้บริหาร)
10	ระบบแจ้งแจ้งอุปกรณ์ชำรุด (Facility Reporting) User Story: ในฐานะผู้ใช้ ฉันต้องการแจ้งปัญหา (เช่น แอร์เสีย) เพื่อให้ระบบ Block ห้องนั้น	Low	แก้ปัญหาที่ 3: อุปกรณ์ชำรุด
11	ระบบแนะนำห้องอัจฉริยะ (Smart Recommendation)	Low	แก้ปัญหาที่ 2: จองห้องใหม่เกินจำเป็น

	User Story: ໃນຮຽນຮູ້ໃຊ້ເປົ້າຮູບຈຳນວນຄົນ ຮະບບຄວາມແນະນຳທີ່ອັນດັບເໜາມະສົມທີ່ສຸດໃຫ້		
12	ປະລາຍງວິທີການຈອງ (Booking History) User Story: ໃນຮຽນຮູ້ໃຊ້ ດັ່ງຕົ້ນທີ່ຕ້ອງການດູຮາຍການທີ່ ເຄຍຈອງຍ້ອນໜັງ	Low	ເພື່ອດູ້ຂໍ້ມູນຢ້ອນໜັງ (Nice to have)

ສໍາໜັກທີ່ມີ Dev (3 คน) ໄທເຮີມໂຟກສົມທີ່ High Priority ກ່ອນ

Phase 1: Core Functionality (High Priority)

- ອອກແບບ Database ເພື່ອຮອງຮັບການຈອງແລະປ້ອງກັນ Time overlapping (ປ້ອງກັນການຈອງໜັງ)
- ໜ້າແສດງປະລາຍງສະຖານະທີ່ວ່າງ (Dashboard)
- ຮະບບຈອງທີ່ປະລາຍງ (Create Booking)
- ຮະບບຍົກເລີກການຈອງ (Cancel Booking)

Phase 2: Management & Notifications (Medium Priority)

- ຮະບບແຈ້ງເຕືອນທາງອື່ນເມລ (Notification Service)
- ຮະບບ Check-in ແລະຕັດສີທີ່ອັດໄວ້ມັກຕີ (ແກ້ປ່າຍຫາ Ghost Meeting)
- ຮະບບຈັດການທີ່ປະລາຍງ (Admin: Add/Edit Rooms & Equipment status)

Phase 3: Advanced Features (Low Priority)

- ຮະບບ Approval ສໍາໜັກທີ່ VIP
- ຮະບບແນະນຳທີ່ອັນດັບເໜາມຈຳນວນຄົນ
- ໜ້າດູປະລາຍງວິທີການຈອງຢ້ອນໜັງ
- ຂໍ້ມູນເບື້ອງຕັນສໍາໜັກທີ່ Dev (Information for Developers)

สิ่งที่ทีมต้องใช้ในการวางแผน Sprint 1:

Tech Stack: ใช้อะไรพัฒนา

Business Rules:

- จองล่วงหน้าได้สูงสุด 30 วัน
- ระยะเวลาจองขั้นต่ำ 30 นาที

Design Assets: ใช้อะไรออกแบบ

Access: Git Repository URL :