

## Overview

### 2.1 ภาพรวมระบบเดิม

- “ไปนำแท่น” เป็นเว็บแอปสำหรับ การเดินทางร่วมกัน (carpool) เน้น “ความปลอดภัย + ความสะดวก + ความมั่นใจ” มีฟังก์ชันค้นหา/จับคู่เส้นทาง พร้อมระบบบืนยันตัวตนและໂປຣີຟັ້ນໃຊ້
- ระบบแบ่งผู้ใช้หลักเป็น 4 บทบาท: Guest / Passenger / Driver / Admin

### Stakeholders

#### 2.2 Stakeholders (ผู้มีส่วนได้ส่วนเสีย) + สิ่งที่แต่ละฝ่ายทำได้

ให้คุณเขียนเป็นหัวข้ออย่างแบบนี้ (คัดจากรายงานได้ตรง ๆ)

##### 1. Guest (ผู้เยี่ยมชม)

- ดูข้อมูลเบื้องต้นของเว็บได้ แต่ เข้า feature หลักไม่ได้ เช่น ค้นหาเส้นทาง/จอง/ໂປຣີ/ແຈ້ງ เพื่อป้องกันข้อมูลรั่วไหลและความคุ้มความปลอดภัย

##### 2. Passenger (ผู้โดยสาร)

- ค้นหาเส้นทาง + ดูรายละเอียดผู้ขับขี่
- จอง/ยกเลิกการเดินทาง
- แก้ไขໂປຣີ และดูประวัติการเดินทาง

##### 3. Driver (ผู้ขับขี่)

- สร้าง/โพสต์เส้นทาง (ต้นทาง-ปลายทาง เวลา จำนวนที่นั่ง ฯลฯ)
- บืนยัน/ปฏิเสธคำขอจองจากผู้โดยสาร
- ໂປຣີผู้โดยสารเพื่อประเมินความเห็นชอบ

##### 4. Admin (ผู้ดูแลระบบ)

- สิทธิสูงสุด: เพิ่ม/แก้ไข/ลบบัญชีผู้ใช้ทุกประเภท
- ตรวจสอบกิจกรรมทั้งหมด และจัดการระบบหลังบ้าน/เส้นทางผิดปกติ
- กำหนดสิทธิ์และดูแลความปลอดภัยโดยรวม

## 5. External stakeholders (ระบบ/บริการภายนอกที่พึ่งพา)

- มีการเรียกใช้ Google Maps Directions API เพื่อประมวลผลเส้นทาง และมีตระรากจับคู่เส้นทาง (เช่นคำนวนระยะทาง)
- มีการอัปโหลดรูปไปยัง Cloudinary ในงานฝั่งหลังบ้านบางส่วน (อย่างน้อยใน flow การจัดการyanpathanของ Admin)

## Requirements Summary

### 2.3 Requirements ของ “ระบบเดิม” (สรุปจากรายงาน+คู่มือ)

แบ่งเป็น Functional + Business rules + Non-Functional

#### A) Functional Requirements (ฟังก์ชันหลักที่ระบบ “ต้องทำได้”)

##### ผู้โดยสาร Passenger

- จองเส้นทาง: กด “จองที่นั่ง” ระบบต้องตรวจสอบว่าเส้นทางยังจองได้และมีที่นั่งว่าง แล้วบันทึกการจองพร้อมข้อมูลจุดรับ-ส่งที่ผู้โดยสารเลือก
- ระบบต้อง “ล็อกที่นั่ง” โดย หักจำนวนที่นั่งทันที และตั้งสถานะคำขอเป็น “รอการยืนยัน” เพื่อรอดriver ตอบรับ
- ผู้โดยสารกรอกรายละเอียดในหน้าต่างยืนยันการจอง เช่น จำนวนที่นั่ง จุดรับ จุดส่ง และยืนยันเพื่อบันทึกข้อมูล
- ผู้โดยสารดู “การเดินทางของฉัน” และสามารถเลิกได้ถ้ายังไม่เริ่มเดินทาง (มีขั้นตอนให้เลือกเหตุผล)

##### ผู้driver

- สร้างการเดินทาง: กรอกข้อมูลเส้นทาง/รายละเอียด/ข้อมูลรถ แล้วกด “สร้างการเดินทาง” เพื่อเผยแพร่ให้ค้นหาได้ และถ้ายังไม่ยืนยันตัวตน ระบบจะพาไปทำ Driver verification ก่อน
- จัดการคำขอจอง: ดูคำขอใน “คำขอจองเส้นทางของฉัน” และกดยืนยัน/ปฏิเสธได้
- Driver verification: อัปโหลดเอกสาร (รูปใบขับขี่/รูปเซลฟี่/ข้อมูลบัตร) เพื่อนุมัติสิทธิ์การเป็นผู้ขับขี่
- จัดการข้อมูลรถยนต์: เพิ่ม/แก้ไข/ลบรถได้

ผู้ Admin

- จัดการข้อมูลผู้ใช้/สั่นทาง/การจองได้ทั้งหมด (ตามนิยามบทบาท)

#### B) Business Rules (กติกาสำคัญ)

- ห้าม Driver จองเส้นทางของตนเอง (มีการแจ้ง error ชัดเจนในกรณีทดสอบ)
- การยกเลิกมี “มาตรการควบคุมพุทธิกรรมการยกเลิก” เช่น ต้องเลือกเหตุผลก่อนยกเลิก และมีแนวคิดลงโทษเพื่อความน่าเชื่อถือของระบบ

#### C) Non-Functional Requirement (คุณภาพระบบ)

- ความปลอดภัยบัญชี: มีการตรวจสอบ Token / ตรวจสอบ schema และการแฮชรหัสผ่านด้วย bcrypt ในบาง flow (เช่น เปลี่ยนรหัสผ่าน)
- ความถูกต้อง/กันจองซ้ำ: มีการ “ล็อกที่นั่ง” โดยหักที่นั่งทันทีเมื่อสร้างคำขอของ

### Problems of current system

#### 2.4 ปัญหา/ข้อจำกัดของระบบเดิม (Pain points / Opportunities)

จากรายงานมีข้อเสนอแนะชัดว่าระบบปัจจุบัน “ยังขาดอะไร” ซึ่งถือเป็น ปัญหา/ช่องว่างของระบบเดิม ได้แก่ เช่น:

- ยังไม่มี ติดตามตำแหน่งแบบ real-time ระหว่างผู้ขับขี่และผู้โดยสาร
- ความเพิ่ม ระบบแซท เพื่อสื่อสารระหว่างขึ้น
- ความชำนาญ/ข้อมูลต่าง ๆ ให้เป็น real-time หากก็
- ต้องวางแผนการใช้ API ให้ดี เพราะมีเรื่อง งบ/จำนวนครั้งเรียกใช้ ในช่วงทดสอบ
- ปัญหาด้านความน่าเชื่อถือจาก “การยกเลิก” จึงต้องมีมาตรการควบคุมพุทธิกรรมการยกเลิกของผู้ใช้