

# ระบบบริหารงานขนส่ง คิดเงิน 2 ฝั่ง & กระทบยอด

## SPDO Delivery Ops & Billing System

ระบบทดแทนไฟล์ Excel รายเดือน สำหรับผู้รับเหมาขนส่ง last-mile (Shopee SPX / Lazada LEX)  
— แม่นยำ ตรวจสอบได้ และขยายตามจำนวนคนรถได้โดยไม่ต้องสร้างซีดีใหม่

▶ คับ error สูตรที่ต้นต่อ

▶ คิดเงิน SPX / SPDO อัตโนมัติ

▶ กระทบยอดกับเงินที่โอนจริง

▶ Scale = เพิ่ม 1 แถว

จัดทำเพื่อ

SPDO — ผู้รับเหมาขนส่งคนกลาง

จัดทำโดย

Pollaphat (ทีมพัฒนาระบบ)

วันที่

25 มิถุนายน 2026

สถานะ

ข้อเสนอเบื้องต้น (Draft v1)

ปัจจุบันงานทั้งหมดคุมด้วย Excel 1 ไฟล์ต่อเดือน 33 ซิต (ซิตสรุป “ใบคุมเที่ยว” + ซิตเช็คชื่อ + ซิตรายคน 31 ซิต) จากการตรวจไฟล์จริง พบว่าปัญหาไม่ใช่แค่ “สูตรเยอะ” แต่เป็นปัญหาเชิงโครงสร้างที่ทำให้ตัวเลขเชื่อถือไม่ได้และขยายงานต่อไม่ได้

| # | อาการที่พบจริง (ยืนยันจากไฟล์)                                                                                 | ต้นตอที่แท้จริง                                                  |
|---|----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| 1 | สูตร error ~25 จุด (#VALUE!, #REF!) ลามถึงซิตสรุป จน % มาร์จิ้นรวมทั้งบริษัทคำนวณไม่ได้                        | เคาะ space ในเซลล์ตัวเลข + ลบคอลัมน์ + ฝังสูตรคำนวณในเซลล์ข้อมูล |
| 2 | ต้อง map คอลัมน์ด้วยมือทีละแถว เพราะยอดรวมอยู่คนละตำแหน่ง (4W, 2W, รายวัน วางไม่เหมือนกัน)                     | Layout coupling — ค่าตรรกะเดียวกันอยู่คนละเซลล์ต่อคน             |
| 3 | จ่ายเงินผิดจริง โดยไม่มี error เตือน — คอลัมน์ “จ่ายรถ” ของคนรถบางคนชี้ไปช่อง “ค่าฐานเงินเดือน” แทนยอดจ่ายจริง | map พิกัดด้วยมือ + ไม่มีการตรวจสอบ (validation)                  |
| 4 | วันที่ในซิตเช็คชื่อเพี้ยนเป็น “ปี 1969”                                                                        | กรอกปี พ.ศ. 2 หลัก ไม่มีชนิดข้อมูลวันที่                         |
| 5 | ชื่อคน/ชื่อซิตพิมพ์ไม่ตรงกัน (เช่น สกนซ์/สกนธ, รัฐเซีย/รัฐเซีย)                                                | เชื่อมข้อมูลด้วย “ชื่อ” ที่เป็นทั้งคีย์และสูตร                   |
| 6 | เพิ่มคน/Hub ต้องคัดลอกซิต + แก้สูตรด้วยมือ (มี 4 ซิต “ว่าง” ค้างไว้)                                           | ไม่มีฐานข้อมูลกลาง — “คน” ถูกทำเป็นเลย์เอาต์ ไม่ใช่ข้อมูล        |
| 7 | ไม่เคยกระทบยอดกับเงินจริงที่แพลตฟอร์มโอนมา — กำไรที่เห็นเป็น “กำไรบนกระดาษ”                                    | คิดจากเรทที่บริษัทคิดเอง ไม่มีการเทียบ settlement                |

#### ⚠️ หลักฐานชิ้นสำคัญ: ไฟล์ปัจจุบัน “จ่ายเงินผิด” โดยไม่มีสัญญาณเตือน

ในซิตสรุป คอลัมน์ “จ่ายรถ (SPDO 1-30)” ของคนรถ 2 คน ถูกตั้งสูตรชี้ไปที่ช่อง “ค่าฐานเงินเดือน” แทนที่จะเป็นยอดจ่ายจริง ทำให้ยอดหัวบิล ไม่ตรงกับผลรวมรอบ 1-15 บวก รอบ 16-30 ของคนคนนั้นเอง — เป็นความผิดพลาดที่ตาเปล่ามองไม่เห็น เพราะไม่ขึ้นเป็น error สีแดง นี่คือเหตุผลที่ระบบต้อง “คำนวณด้วยตรรกะเดียว + ตรวจสอบอัตโนมัติ”

#### เป้าหมายของระบบ

0

error สูตรลามข้ามซิต  
(แยกข้อมูลออกจากการคำนวณ)

2 ผัง

SPX / SPDO คิดถูก  
อัตโนมัติด้วยตรรกะเดียว

กำไรจริง

กระทบยอดกับเงิน  
ที่แพลตฟอร์มโอนจริง

+1 แถว

เพิ่มคน/Hub โดยไม่  
สร้างซิต/แก้สูตร

หัวใจของการออกแบบคือ แยกระบบเป็น 3 ชั้น เพื่อตัดต้นตอปัญหา — ชั้นข้อมูลเก็บเฉพาะข้อมูลดิบ, ชั้นคำนวณคิดเงินตอนอ่าน (ไม่ฝังในเซลล์), ชั้นแสดงผลดึงไปแสดง

### PRESENTATION · ชั้นแสดงผล

ใบคุมเที่ยว · สลิป/ใบจ่ายคนรถ · dashboard มาร์จิ้น · LINE

### CALCULATION · ชั้นคำนวณ

Rate Engine (เรทแบบมีเวอร์ชัน) + Reconciliation — คิดเงิน “ตอนอ่าน” ไม่ฝังสูตรในเซลล์ข้อมูล

### DATA · ชั้นข้อมูล

โครงสร้าง Normalized: 1 แถว/คน/วัน + ฐานข้อมูลกลางชุดเดียว — เก็บข้อมูลดิบเท่านั้น ไม่มีสูตรปน

## โมดูลหลัก 10 ตัว

| โมดูล                                      | หน้าที่                                                           | แทนอะไรในไฟล์เดิม                  |
|--------------------------------------------|-------------------------------------------------------------------|------------------------------------|
| M1 Master Data                             | ทะเบียนคนรถ / Hub / รถ — ตำแหน่งว่าง = สถานะ vacant               | หัวข้อรายชื่อคน 31 ชิต             |
| M2 Rate Card                               | ตารางเรทกลาง แบบมีเวอร์ชัน แยกฝั่ง SPX/SPDO ต่อประเภทรถ           | เรทที่ฝังในหัวข้อลัมน์             |
| M3 Daily Entry                             | บันทึก 1 แถว/คน/วัน เก็บข้อมูลดิบ + validation ที่ต้นทาง          | ตารางวันที่ 1-31 ในชิตรายชื่อคน    |
| M4 Inbound + Dispatch<br><span>ใหม่</span> | รับสื่อตงานจาก SPX/LEX เป็นยอดควบคุม + จ่ายงานให้คนรถ             | — (ไม่เคยมี)                       |
| M5 Attendance                              | เช็คชื่อ มา/ลา/หยุด เป็นเวลาจริง → ผูกฐานประกันรายเดือน           | ชิตเช็คชื่อ (ที่วันที่เพี้ยน)      |
| M6 Pay Engine <span>MVP</span>             | คิด SPX / SPDO / margin / incentive 10% ตามรอบบิล                 | สูตรคิดเงินที่ฝังเซลล์             |
| M7 Auto Summary<br><span>MVP</span>        | ใบคุมเที่ยวรุ่นใหม่ดึงจากฐานข้อมูล (QUERY/SUMIFS)                 | ชิตสรุปที่อ้างอิงข้ามชิตที่ละเซลล์ |
| M8 Reconciliation<br><span>ใหม่</span>     | เทียบยอดที่ระบบคิด vs เงินจริงที่โอน → หาส่วนต่าง/งานถูกหัก       | — (หัวใจของคุณค่าใหม่)             |
| M9 Payslip                                 | สร้างใบจ่าย PDF ต่อคนต่อรอบ ส่งเข้า LINE                          | สลิปทำมือ                          |
| M10 RBAC + Audit Log                       | สิทธิ์ตามบทบาท + บันทึกทุกการแก้ค่าเงิน (ใคร/เมื่อไร/เท่าไร-ใหม่) | — (ไม่เคยมี)                       |

### ขอบเขต MVP

โมดูลที่ให้คุณค่าสูงสุด (M4 จ่ายงาน, M8 กระทบยอด) **ไม่อยู่ใน MVP** — แต่ออกแบบโครงสร้างข้อมูลให้รองรับตั้งแต่แรก แล้วทยอยเปิดในเฟสถัดไป เพื่อให้เริ่มใช้ได้เร็วและไม่ต้องรื้อภายหลัง

```
# ความสัมพันธ์ของข้อมูล (< = หนึ่ง-ต่อ-หลาย) hubs <- drivers <- daily_entries (work_date DATE, pieces_in/out, size_s,
size_l0..l3, rst, pooh, lost, fuel, same, advance, attendance) <- attendance rate_cards (vehicle × employment × side ×
size_tier, min_pieces, base, rate, effective_from/to) billing_periods <- payouts (spx_total, spdo_total, margin,
incentive) <- derived settlement_lines -> reconciliation (variance, actual_margin) <- derived audit_log
```

### ทำไมโครงสร้างนี้ทดแทน “1 ชิตต่อคน” ได้ และขยายได้

| คุณสมบัติของโครงสร้างใหม่                                                 | ผลที่ได้                                                                                                         |
|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| driver_id เป็นคีย์ — ทุกธุรกรรมอ้างรหัส ไม่ใช่ชื่อ                        | ดับปัญหาชื่อพิมพ์ไม่ตรง (สกนซ์/สกนธ์) — เชื่อมข้อมูลไม่พลาด                                                      |
| daily_entries เป็นตารางเดี่ยว (long-format) ทุกประเภทรถใช้คอลัมน์เดียวกัน | ความต่าง 4W/2W/รายวัน ย้ายไปอยู่ที่ค่าในคอลัมน์ ไม่ใช่ตำแหน่งคอลัมน์ → ไม่มีคอลัมน์รายวันให้ลบ = ไม่มี #REF! อีก |
| work_date เป็นชนิดวันที่จริง + รอบบิลคำนวณจากวันที่                       | ดับบัก “ปี 1969” ที่รบกวน แสดงเป็น พ.ศ. ที่หน้าจอล                                                               |
| เพิ่มคน/Hub = เพิ่ม 1 แถว                                                 | ตำแหน่งว่าง = สถานะ vacant ไม่ใช่ชิตเปล่าไม่แต่ละสูตร                                                            |
| payouts เป็นค่าที่คำนวณ (derived) คิดตอนอ่าน                              | แก้เรทแล้วคำนวณใหม่ทั้งระบบพร้อมกัน ไม่ต้องไล่แก้ 31 ชิต                                                         |
| rate_cards มีเวอร์ชัน (effective_from/to)                                 | ขึ้นเรทใหม่ = เพิ่มเวอร์ชัน — บิลย้อนหลังยังคิดด้วยเรทที่ถูกตามวันงาน                                            |

#### หลักการเดียวที่แก้ได้ทุกปัญหา

“ใช้คอลัมน์เดียวกันทุกคน” + “เก็บข้อมูลดิบ ไม่เก็บผลลัพธ์ที่คำนวณแล้ว” — เมื่อข้อมูลถูกแยกออกจากการคำนวณ error ทั้งคลาส (#REF!, #VALUE!, map ผิด) ก็หายไปพร้อมกัน

## 4.1 Rate Card แบบมีเวอร์ชัน (ค่าจริงจากไฟล์ พร้อมใส่)

| ประเภท            | ขั้นต่ำ (ขั้น) | ฐาน SPX / SPDO ต่อวัน | เรท S (SPX/SPDO) | เรท L (SPX/SPDO)    |
|-------------------|----------------|-----------------------|------------------|---------------------|
| 4W รายเดือน       | 70             | 615.39 / 533.33       | 8 / 6            | L2 12/10 · L3 13/11 |
| 2W รายเดือน       | 90             | 615.39 / 466.66       | 6 / 5            | L2 6/5 · L3 7/6     |
| รายวัน (เหมา/วัน) | 65             | 1,150 / 1,000         | 11 / 9           | ตามตกลง             |

มิติเรท: platform x vehicle x employment x size\_tier x side(SPX/SPDO) + ช่วงเวลาที่มีผล — เปลี่ยนเรท = เพิ่มเวอร์ชันใหม่ไม่ทับของเก่า

## 4.2 เครื่องคิดเงิน 2 ฝั่งคู่ขนาน (ตรรกะเดียวทุกประเภทรถ)

$$SPX_{net} = base + (\text{ขั้นเกินขั้นต่ำ} \times \text{เรท}_S) + \Sigma(L \text{ tier} \times \text{เรท}_L \text{ ฝั่ง SPX}) - \text{Same} - \text{POOH}$$

$$SPDO_{net} = base + \text{RST} + \text{เบิก} + \Sigma(L \text{ tier} \times \text{เรท}_L \text{ ฝั่ง SPDO}) - \text{Same} - \text{POOH} - \text{Card} - \text{LOST}$$

$$margin = SPX_{net} - SPDO_{net}$$

## ความไม่สมมาตรที่เป็นหัวใจของมาร์จิ้น (ต้องยืนยันกับลูกค้า)

Same / POOH ทั้ง 2 ฝั่ง แต่ RST / Card / LOST ทั้ง 2 ฝั่งเฉพาะฝั่ง SPDO · ขั้น < ขั้นต่ำ → ได้แค่ค่าคงไม่ได้ค่าแรง · รายวันวันหยุด/ลา → ไม่ได้ค่าเหมา · ต้องนิยาม “base คิดอย่างไรต่อรอบบิล” ให้ชัด (พบความไม่สอดคล้องระหว่างรอบ 1 กับรอบ 2 ในไฟล์เดิม)

## 4.3 กระทบยอด 3 ชั้น

| ชั้น         | ตรรกะ                                                                                      | จับอะไร                               |
|--------------|--------------------------------------------------------------------------------------------|---------------------------------------|
| ① ภายในวัน   | รับเข้า = จ่ายออก + RST + LOST · แจ้งเตือน margin ติดลบ, จ่ายออก > รับเข้า                 | error ที่ต้นทาง (Excel ทำไม่ได้)      |
| ② รอบบิล     | รวม SPX/SPDO ต่อคนต่อรอบ + incentive 10%                                                   | สรุปแทนใบคุมเที่ยว                    |
| ③ Settlement | variance = ยอดที่ระบบคิด - ยอดที่แพลตฟอร์มโอนจริง (แยก unpaid / underpaid / size mismatch) | งานจ่ายขาด, ถูกหัก RST/LOST, ปรับไซส์ |

## ★ จุดขายหลัก

ต่อให้ Excel ไม่มี error เลย ตัวเลขถ้าไรก็ยังเชื่อไม่ได้จนกว่าจะเทียบกับเงินที่โอนจริง — “กำไรจริง = settlement ที่ได้รับ - SPDO ที่จ่ายคนรถ” คือสิ่งที่ระบบให้ได้ แต่ Excel ปัจจุบันทำไม่ได้เลย

## คำแนะนำ: เริ่มที่ Google Sheets ที่ออกแบบใหม่ → ยกขึ้นเว็บแอปเมื่อโครงสร้างข้อมูลนิ่ง

สำหรับธุรกิจขนาด ~26 คน งบจำกัด **ไม่ควรเริ่มด้วยเว็บแอปเต็มรูปแบบ** เพราะความเร็วในการได้ใช้งานจริง (time-to-value) และการยอมรับของผู้ใช้สำคัญกว่า

| เฟส                            | เทคโนโลยี                                                                   | เหตุผล                                                                                                  |
|--------------------------------|-----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| Phase 0-1<br>(Bridge + MVP)    | Google Sheets โครงใหม่ (flat table + RateCard + QUERY/SUMIFS) + Apps Script | แอดมิน/บัญชีคุ้นเคยอยู่แล้ว · ไม่มีค่า hosting · ประวัติเวอร์ชัน = audit ฟรี · ได้ใช้จริงใน 1-2 สัปดาห์ |
| Phase 2+<br>(System of record) | Supabase (Postgres + Auth + RLS) + Next.js มือถือ-first + n8n + LINE        | ความถูกต้องระดับธุรกรรมสำหรับงานเงิน · RBAC ที่ระดับฐานข้อมูล · ขยายได้หลายร้อยคนโดยไม่แตะ schema       |

ทางเลือกที่พิจารณาแล้วแต่ไม่เลือกเป็นหลัก: AppSheet/Glide (no-code คุมตรรกะ 2 มุ่งระยะยาว) · Airtable (ค่า seat แพงเมื่อเพิ่มคน) · เว็บแอป custom ตั้งแต่ต้น (ช้า/แพงเกินสำหรับ MVP)

## การเก็บข้อมูลรายวัน — ลด error ที่ต้นทาง

| ช่องทาง             | ใคร            | การป้องกัน error ที่ต้นทาง                                                                                   |
|---------------------|----------------|--------------------------------------------------------------------------------------------------------------|
| ฟอร์มมือถือ / LINE  | คนรถ (เฟสหลัง) | ทุกช่องเป็นตัวเลข + ตัด space อัตโนมัติ → ดับ #VALUE! · บังคับ รับเข้า=จ่ายออก+RST+LOST ก่อนส่ง · แบนรูป POD |
| หน้าจ่ายงาน / Sheet | แอดมิน Hub     | เลือกคนรถจาก dropdown → กันพิมพ์ชื่อผิด · เลือกวันที่จากปฏิทิน → กันบัก 1969                                 |
| นำเข้า CSV / paste  | จาก SPX/LEX    | ตั้งยอดควบคุมของแต่ละวัน                                                                                     |

กลยุทธ์การยอมรับ: เฟสแรกไม่บังคับคนรถรอกเอง (กัน adoption พัง) — แอดมินรอกเหมือนเดิมแต่บนโครงสร้างที่ถูกต้อง แล้วค่อยดันให้คนรถรอกเองในเฟสหลัง

0

1-2 สัปดาห์

Bridge & Migration — Quick Win เริ่มที่นี่

- สกัด master (คนรถ/Hub/เรพ/วันหยุด) จากไฟล์จริง ทำความสะอาดชื่อ/วันที่
- แปลง 31 ซีดรายคน → ตารางเดี่ยว (flat table) + ตั้ง RateCard แบบมีเวอร์ชัน
- กระทบยอดให้ตรงไฟล์เดิม (ยกเว้น 25 จุด error + จุดที่ map ผิด) เพื่อสร้างความเชื่อมั่น

1

1-2 สัปดาห์

MVP — ทดแทน Excel MVP

- เครื่องคิดเงิน 2 ฝั่ง + ใบบุคเที่ยวอัตโนมัติ (QUERY/SUMIFS) + incentive 10%
- validation + dropdown ครบทุกช่องกรอก
- รันคู่ขนานกับ Excel 1 รอบบิล แล้วเทียบผล + อบรมผู้ใช้

2

1.5-2 สัปดาห์

Settlement & Payslip คุณค่าใหม่

- กระทบยอด settlement + คำนวณ variance (เริ่มที่ SPX)
- สร้างใบจ่าย PDF ต่อคนต่อรอบ ส่งเข้า LINE + audit log

3

6-8 สัปดาห์

Core Web App (ตามความจำเป็นจริง) ภายหลัง

- Supabase schema + หน้ากรอกบนมือถือ + RBAC ผ่าน RLS
- ใบบุคเที่ยวสร้างจากฐานข้อมูล + export Excel หน้าตาคุ้นเคย · n8n นำเข้า settlement / ส่งสลิป LINE / ปิดรอบอัตโนมัติ

4

ต่อเนื่อง

Scale & Self-service ภายหลัง

- LINE LIFF ให้คนรถยืนยันยอด/เช็คคินเอง · dashboard มารจิ้นต่อ Hub/ประเภท/แพลตฟอร์ม · export เข้าระบบบัญชี (FlowAccount/PEAK)

## กลยุทธ์ลดความเสี่ยงด้านการลงทุน

Phase 0-2 อยู่บน Google Sheets ทั้งหมด (ลงทุนต่ำ ที่สุจริตระการคิดเงินได้จริง) ก่อนตัดสินใจลงทุนเว็บแอป — ลูกค้าได้เห็นของใช้งานได้และผลตอบแทนก่อนจ่ายเงินก้อนใหญ่

## คำถามสำคัญที่ต้องยืนยันกับลูกค้าก่อนเริ่ม (กำหนดขอบเขต/ราคา)

| # | ประเด็น                                                                                                                                                                   |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | กฎคิดเงินที่ “ถูกต้อง” คืออะไรกันแน่ — base คิดอย่างไรต่อรอบบิล, ขึ้น<ชั้นต่ำได้เท่าไร, รายวันวันหยุดได้/ไม่ได้, RST/LOST/Same/POOH หักฝั่งไหนบ้าง (ขอเป็นลายลักษณ์อักษร) |
| 2 | Settlement มาในรูปแบบไหน — ลงรายชั้น/มี order ref หรือเป็นยอดรวม? ขอไฟล์ตัวอย่างจริง 1-2 รอบ (กำหนดว่ากระทบยอดได้ละเอียดแค่ไหน)                                           |
| 3 | เรทต่างกันต่อ Hub หรือเปลี่ยนกลางเดือนไหม และมีแผนขึ้นเรทเมื่อใด                                                                                                          |
| 4 | เฟสแรกให้คนกรอกเอง หรือแอดมินกรอกแทน (กระทบขอบเขตงาน LINE อย่างมาก)                                                                                                       |
| 5 | ใครต้องเห็นอะไรบ้าง / ใครปิดรอบได้ — เจ้าของ / แอดมิน Hub / บัญชี (กำหนด RBAC)                                                                                            |
| 6 | ต้อง export เข้าระบบบัญชีอะไร (FlowAccount/PEAK/Excel) และ Lazada (LEX) อยู่ในเฟสไหน                                                                                      |

## ความเสี่ยง &amp; ข้อควรระวัง

| ประเด็น                                        | การบรรเทา                                                              |
|------------------------------------------------|------------------------------------------------------------------------|
| สูตรเดิมบางจุดผิดอยู่แล้ว → migration จะไม่ตรง | ตกลงกับลูกค้าก่อนว่า “ค่าที่ถูก” คืออะไร ไม่ลอกของเดิมที่ผิด           |
| กฎคิดเงินมี edge case ซ้อนกัน                  | ยืนยันเป็นลายลักษณ์อักษรทุกข้อก่อนเขียน engine — ไม่เตา                |
| ความละเอียดของ settlement                      | ขอไฟล์ตัวอย่างจริงก่อนสัญญาพีเจอร์กระทบยอด                             |
| การยอมรับของผู้ใช้                             | validation เข้ม + ล็อกคอลัมน์สูตร + φόρμสั้นภาษาไทย + รันคู่ขนาน 1 รอบ |

## ก้าวต่อไปที่แนะนำ

นัดประชุม 1 รอบเพื่อ (ก) ยืนยันกฎคิดเงินทุกข้อ (คำถามข้อ 1) และ (ข) รับไฟล์ Excel ปัจจุบัน + ไฟล์ settlement ตัวอย่าง → เริ่ม Phase 0 ทันที โดยใช้ไฟล์จริงพิสูจน์ว่าระบบให้ตัวเลขตรง (และชี้จุดที่ไฟล์เดิมจ่ายเงินผิด) ก่อนลงทุนเฟสถัดไป

— จบเอกสารข้อเสนอ · Pollaphat x SPDO · 25 มิถุนายน 2026 —