



แอปพลิเคชันและระบบบริหารจัดการรถไฟฟ้าอัจฉริยะ

PSRU Bus Application

Smart Electric Vehicle Management System

กรทักษ์ คงปิ่นนา*, ณัฐวุฒิ บัวแก้ว , นนทกร รณที , นนทัช ทับแก้ว , พงศธร เสือคล้าย ,
พีรวิชญ์ กันทะกะ

Korathak Kongpanna*, Nuthwooth Bauwkeaw , Nonthakon Ronnathi ,

Nontouch Tupkaew , Pongsathon Sueaklay , Peeravit Kanthaka

คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏพิบูลสงคราม อ.เมือง จ.พิษณุโลก ประเทศไทย 65000

Faculty of Industrial Technology, Pibulsongkram Rajabhat University, Muang, Phitsanulok, Thailand, 65000

*Corresponding author E-mail: korathank.k@psru.ac.th

1. บทนำ และ ทบทวนวรรณกรรม

ในยุคดิจิทัล เทคโนโลยีแอปพลิเคชันบนอุปกรณ์พกพาได้เข้ามามีบทบาทสำคัญในชีวิตประจำวันอย่างกว้างขวาง การพัฒนาแอปพลิเคชันที่สามารถอำนวยความสะดวกและแก้ปัญหาเฉพาะด้านได้อย่างมีประสิทธิภาพกลายเป็นเป้าหมายหลักในหลายองค์กร หนึ่งในนั้นคือระบบการเดินทางและขนส่งภายในมหาวิทยาลัยซึ่งยังคงประสบปัญหาความล่าช้า การจัดการทรัพยากรที่ไม่เหมาะสม และความไม่สะดวกของผู้ใช้งาน

แอปพลิเคชัน PSRU Bus Application ถูกออกแบบมาเพื่อตอบสนองต่อการจัดการการเดินทางภายในมหาวิทยาลัย โดยให้ผู้ใช้งานสามารถเรียกรถไฟฟ้าจากจุดเริ่มต้นถึงจุดปลายทางได้อย่างง่ายดายผ่านโทรศัพท์มือถือ นอกจากนี้ แอปพลิเคชันยังมีการแสดงข้อมูลแบบเรียลไทม์ เช่น การแสดงตำแหน่งของรถ สถานะการเดินรถ และการแจ้งเตือนเมื่อรถมาถึงสถานี ซึ่งช่วยเพิ่มความสะดวกและประสิทธิภาพให้กับนักศึกษาและบุคลากร พร้อมทั้งลดภาระของระบบการจัดการรถไฟฟ้าต่างๆ

ทบทวนวรรณกรรม

1. การพัฒนาแอปพลิเคชันสำหรับติดตามรถ

แอปพลิเคชันติดตามรถได้ถูกพัฒนาเพื่อตอบสนองในด้านการจัดการการขนส่งและความปลอดภัยของผู้ใช้ โดยการพัฒนาแอปพลิเคชันในปัจจุบันมักใช้เทคโนโลยี เช่น Google Maps API สำหรับแสดงตำแหน่ง GPS และ Firebase สำหรับจัดเก็บข้อมูลแบบเรียลไทม์ ตัวอย่างเช่น งานวิจัยของจุฑามณี รุ่งแก้ว และ อารียา ตะเคียนราม (2566) ที่พัฒนาระบบสำหรับติดตามรถรับส่งนักเรียนผ่านสมาร์ทโฟน

นอกจากนี้ งานวิจัยของเนติมา อุดร (2562) ได้นำเทคโนโลยี IoT และ RFID มาใช้ในแอปพลิเคชันติดตามรถรับส่งนักเรียน โดยมีฟังก์ชันแจ้งเตือนผ่านแอปพลิเคชัน Line และการติดตามตำแหน่งผ่านระบบ GPS ซึ่งช่วยเพิ่มความมั่นใจให้แก่ผู้ปกครองและช่วยเพิ่มมาตรฐานความปลอดภัยในการเดินทางของนักเรียน

2. การพัฒนาระบบในบริบทการศึกษา

ในงานวิจัยของสุทธิพงศ์ ประทุม และคณะ (2566) ได้นำเสนอระบบต้นแบบสำหรับการตรวจสอบการขึ้น-ลงรถของนักเรียนโดยใช้ IoT และ RFID ซึ่งมุ่งเน้นเพิ่มความปลอดภัยและลดข้อผิดพลาดที่อาจเกิดจากมนุษย์ ระบบดังกล่าวใช้เทคโนโลยี Bluetooth Low Energy (BLE) และการแจ้งเตือนแบบเรียลไทม์ไปยังผู้ปกครองผ่านแอปพลิเคชันมือถือ

3. ข้อดีของแอปพลิเคชันสำหรับติดตามรถ

แอปพลิเคชันเหล่านี้ช่วยลดเวลาการรอคอยและเพิ่มประสิทธิภาพการเดินทางของผู้ใช้งาน ตัวอย่างเช่น แอปพลิเคชัน NSTDA TRAM ที่ถูกพัฒนาขึ้นเพื่อแสดงตำแหน่งรถภายในมหาวิทยาลัยศิลปากร โดยใช้เซ็นเซอร์ GPS/3G และการแสดงข้อมูลแบบเรียลไทม์ผ่านมือถือ ช่วยให้ผู้ใช้สามารถวางแผนการเดินทางได้สะดวกยิ่งขึ้น

2. วัตถุประสงค์ของงานวิจัย

2.1 เพื่อพัฒนาแอปพลิเคชันบนอุปกรณ์พกพาที่ช่วยให้นักศึกษาสามารถเรียกรถไฟฟ้าผ่านจุดให้บริการ (Station) ภายในมหาวิทยาลัยได้อย่างสะดวก รวดเร็ว และง่ายดาย

2.2 เพื่อออกแบบและพัฒนาระบบบริหารจัดการรถไฟฟ้าอัจฉริยะที่สามารถกำหนดเส้นทางการจัดสรรจำนวนรถ ตลอดจนวางแผนและควบคุมการให้บริการได้อย่างมีประสิทธิภาพ

2.3 เพื่อเพิ่มประสิทธิภาพการสัญจรภายในมหาวิทยาลัย โดยมุ่งลดระยะเวลาการรอคอยและเพิ่มความสะดวกสบายในการเดินทางของนักศึกษาและบุคลากร

2.4 เพื่อศึกษาความพึงพอใจและปัจจัยที่ส่งผลต่อการยอมรับการใช้งานแอปพลิเคชันและระบบบริหารจัดการรถไฟฟ้าอัจฉริยะ โดยเน้นการประยุกต์ใช้งานจริงในบริบทของสถาบันการศึกษา

3. วิธีดำเนินงานวิจัย

การวิจัยนี้มุ่งเน้นพัฒนาระบบขนส่งอัจฉริยะสำหรับรถโดยสารพลังงานไฟฟ้าภายในมหาวิทยาลัยราชภัฏพิบูลสงคราม โดยใช้แนวทางพัฒนาระบบที่เป็นมาตรฐานเพื่อเพิ่มความสะดวกและประสิทธิภาพการเดินทางในพื้นที่มหาวิทยาลัย โดยรายละเอียดของการดำเนินงานมีดังนี้

3.1 ศึกษาและวิเคราะห์ข้อมูลที่เกี่ยวข้อง

3.1.1 ศึกษาแบบขนส่งปัจจุบัน

- วิเคราะห์รูปแบบการเดินทางของระบบขนส่งแบบเดิมที่ใช้การโทรเรียกรถ หรือการรอรถแบบไม่รู้วารถอยู่ตำแหน่งไหน รวมถึงปัญหา เช่น ความไม่ยืดหยุ่นของเส้นทาง การเสียเวลารอรถ และการใช้พลังงานที่ไม่คุ้มค่า
- เก็บข้อมูลปัญหาและวิเคราะห์ความต้องการของผู้ใช้งาน จากผู้ใช้งาน ได้แก่
 - การสำรวจความพึงพอใจ



- ความต้องการในการเดินทาง และความคาดหวังจากระบบใหม่
- จัดทำแบบสอบถามเพื่อสำรวจความต้องการ เช่น สถานีรอร์ทที่เหมาะสม เส้นทางที่ต้องการ และการใช้งานผ่านสมาร์ทโฟน
- สัมภาษณ์กลุ่มเป้าหมาย (Focus Group) เช่น นักศึกษา เจ้าหน้าที่มหาวิทยาลัย และคนขับรถให้บริการ

3.1.2 วิเคราะห์แนวทางการเปลี่ยนผ่านสู่ระบบขนส่งแบบออนดีมานด์

- ศึกษาโมเดลระบบขนส่งแบบออนดีมานด์ที่ประสบความสำเร็จในมหาวิทยาลัยหรือพื้นที่ที่คล้ายคลึงกัน
- วิเคราะห์ปัจจัยที่จำเป็น เช่น เทคโนโลยีสนับสนุน ข้อกำหนดด้านการใช้งาน และโครงสร้างพื้นฐานในพื้นที่มหาวิทยาลัย

3.1.3 ศึกษาเทคโนโลยีที่เกี่ยวข้อง

วิเคราะห์เครื่องมือสำคัญ ได้แก่

- GraphHopper เครื่องมือสำหรับการคำนวณเส้นทางที่เหมาะสม
- OpenStreetMap (OSM) สำหรับการแสดงผลแผนที่และตำแหน่งรถแบบเรียลไทม์
- FlutterFlow แพลตฟอร์มพัฒนาแอปพลิเคชันที่ใช้สร้างอินเทอร์เฟซที่ตอบสนองต่อความต้องการของผู้ใช้

3.2 ออกแบบระบบ

3.2.1 กำหนดฟีเจอร์และคุณสมบัติหลักของระบบ

ฟีเจอร์ที่ออกแบบ ได้แก่

- การจองรถผ่านแอปพลิเคชันแบบเรียลไทม์
- แสดงตำแหน่งรถและเส้นทางแบบเรียลไทม์
- คำนวณเส้นทางที่เหมาะสมที่สุดตามคำขอของผู้โดยสาร

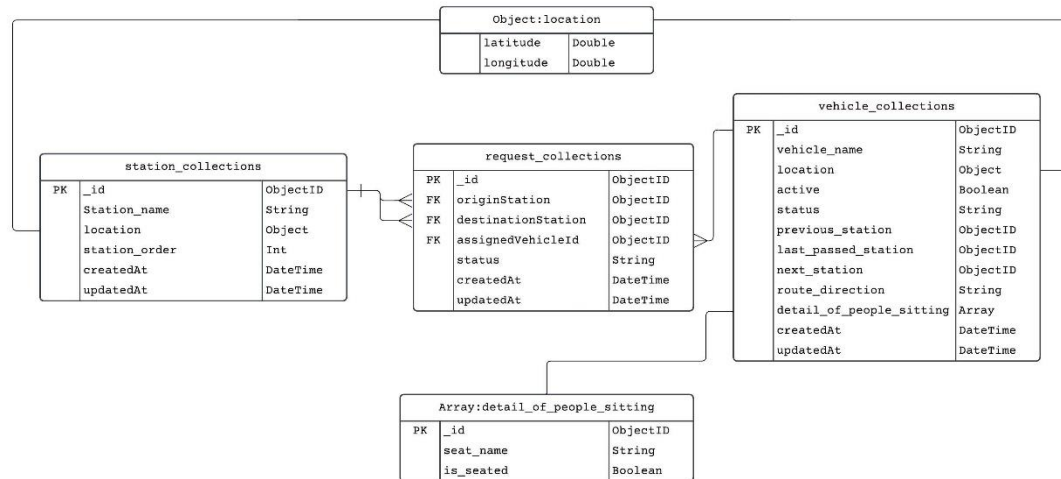


- แจ้งเตือนสถานะการเดินทาง เช่น รถใกล้ถึงจุดรับ

3.2.2 ออกแบบฐานข้อมูล

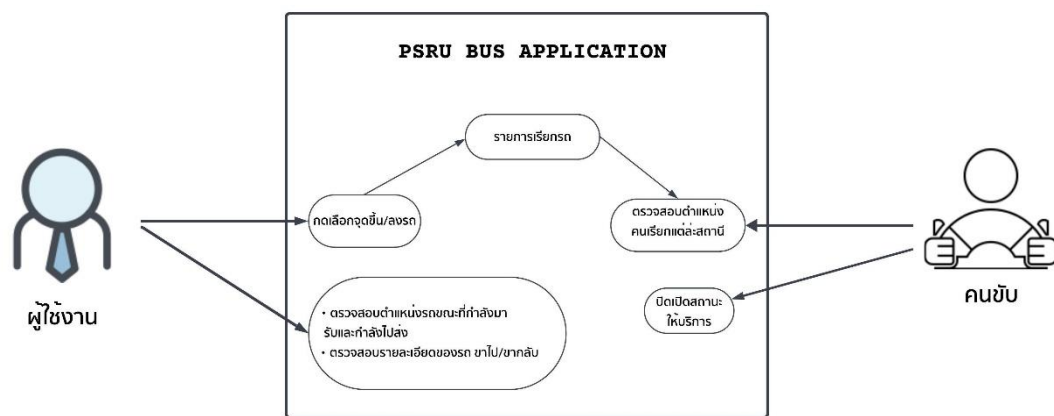
พัฒนาระบบฐานข้อมูลด้วย ER Diagram เพื่อกำหนดความสัมพันธ์ระหว่างข้อมูล

- ตาราง station_collections (ข้อมูลเกี่ยวกับสถานี)
 - Station_name - ชื่อสถานี
 - location - พิกัดของสถานี (latitude, longitude)
 - station_order - ลำดับของสถานี (ถ้ามี)
- ตาราง request_collections (ข้อมูลคำขอการเรียกรถ)
 - originStation - สถานีต้นทาง (เชื่อมกับ station_collections)
 - destinationStation - สถานีปลายทาง (เชื่อมกับ station_collections)
 - assignedVehicleId - รถที่ได้รับมอบหมายให้บริการ (เชื่อมกับ vehicle_collections)
 - status - สถานะของคำขอ เช่น picking_up_passengers, completed, cancelled
 - suggestion - ข้อเสนอแนะ (ถ้ามี)
- ตาราง vehicle_collections (ข้อมูลเกี่ยวกับยานพาหนะ)
 - vehicle_name - ชื่อรถ
 - detail_of_people_sitting - รายละเอียดที่นั่ง (จำนวน 13 ที่นั่งใน 1 ยานพาหนะ)
 - location - พิกัดของรถ (latitude, longitude)
 - active - สถานะการใช้งานของรถ
 - previous_station, last_passed_station, next_station - การอ้างอิงสถานีที่เกี่ยวข้องในเส้นทาง
 - route_direction - ทิศทางของเส้นทาง (go หรือ back)



ER Diagram

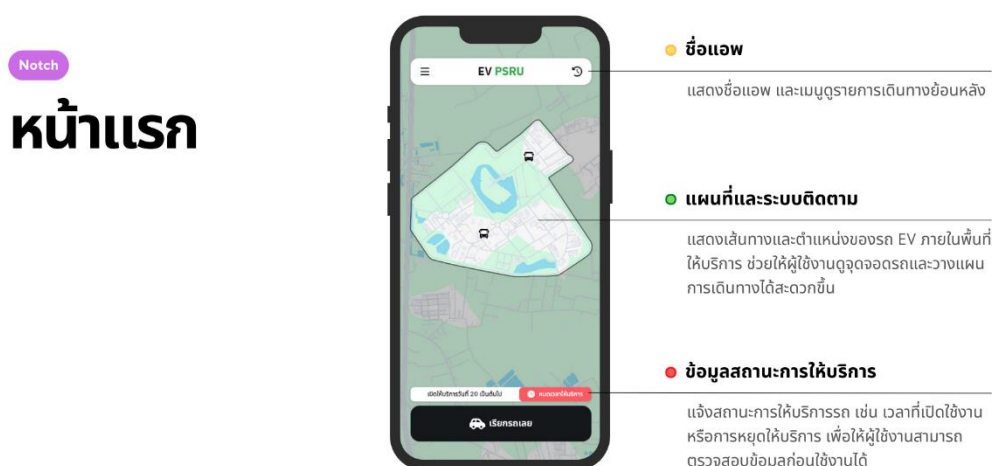
3.2.3 ออกแบบแผนภาพกรณีการใช้งาน (Use Case Diagram)



Use Case Diagram

3.2.4 ออกแบบอินเทอร์เฟซ (UI/UX)

สร้าง Mockup สำหรับหน้าจอการใช้งาน



หน้าแรก (Main Page)

- ชื่อแอปพลิเคชัน

ส่วนบนของหน้าแสดงชื่อแอป (EV PSRU) และเมนูการใช้งาน โดยช่วยให้ผู้ใช้เข้าถึงฟังก์ชันหลักได้ง่ายดาย

- แผนที่และระบบติดตาม

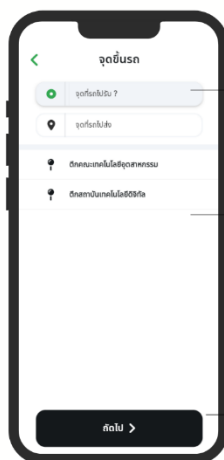
แสดงแผนที่ตำแหน่งของรถ EV ในพื้นที่ที่ให้บริการ พร้อมเส้นทางเพื่อช่วยผู้ขับขี่ดูจุดจอดรถและวางแผนการเดินทาง

- ข้อมูลสถานะการให้บริการ

ระบุสถานะการให้บริการ เช่น ช่วงเวลาที่เปิด-ปิด เพื่อให้ผู้ใช้สามารถตรวจสอบข้อมูลได้ชัดเจน

Notch

จุดขึ้นรถ



● เลือกจุดรับและจุดส่ง

ผู้ใช้งานสามารถระบุจุดสถานที่ที่ต้องการให้รถ EV มารับ (จุดขึ้นรถ) และจุดหมายปลายทางได้อย่างสะดวกผ่านช่องกรอกข้อมูล

● ตัวเลือกสถานที่แนะนำ

แสดงรายชื่อสถานที่ใกล้เคียง เช่น

- ตึกคณะเทคโนโลยีอุตสาหกรรม
- ตึกสถาบันเทคโนโลยีดิจิทัล
- ช่วยให้ผู้ใช้งานเลือกสถานที่ได้ง่ายและรวดเร็วขึ้น

● ปุ่มไปต่อ

เมื่อเลือกจุดรับ-ส่งเสร็จแล้ว ผู้ใช้งานสามารถกด "ถัดไป" เพื่อยืนยันและดำเนินการเรียกรถ EV ต่อไป

จุดขึ้นรถ (Pickup Point Selection)

● เลือกจุดรับและจุดส่ง

ผู้ใช้งานสามารถระบุจุดที่ต้องการให้รถ EV มารับและส่งได้สะดวกผ่านช่องกรอกข้อมูล

● ตัวเลือกสถานที่แนะนำ

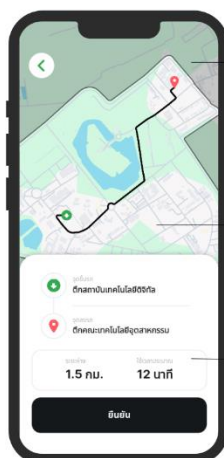
มีรายการสถานที่ใกล้เคียง เช่น ตึกคณะเทคโนโลยีอุตสาหกรรม ตึกสถาบันเทคโนโลยีดิจิทัล เพื่อช่วยให้เลือกได้เร็วขึ้น

● ปุ่มไปต่อ

เมื่อกรอกข้อมูลครบ ผู้ใช้งานสามารถกดปุ่ม "ถัดไป" เพื่อดำเนินการเรียกรถ EV

Notch

รายละเอียดการเดินทาง



วางแผนการเดินทางอัตโนมัติ

แสดงเส้นทางการเดินทางจากจุดเริ่มต้นไปยังจุดหมายปลายทาง พร้อมข้อมูลระยะทางและเวลาที่ใช้โดยประมาณ เพื่อช่วยให้ผู้ใช้งานวางแผนการเดินทางได้สะดวกยิ่งขึ้น

จุดรับ-ส่งที่แม่นยำ

ระบบช่วยระบุจุดขึ้นรถและจุดปลายทางอย่างชัดเจน พร้อมตัวเลือกสถานที่ที่แนะนำใกล้เคียง เพิ่มความรวดเร็วในการเลือกสถานที่

ยืนยันการเดินทางได้ง่าย

ผู้ใช้งานสามารถตรวจสอบข้อมูลการเดินทางและยืนยันการเรียกรถได้อย่างรวดเร็ว เพียงกดปุ่ม "ยืนยัน" ในหน้าจอเดียว

รายละเอียดการเดินทาง (Travel Details)

- วางแผนการเดินทางอัตโนมัติ
แสดงเส้นทางการเดินทางจากจุดเริ่มต้นไปยังจุดหมายปลายทาง พร้อมข้อมูลระยะทางและเวลาที่ใช้โดยประมาณ เพื่อช่วยให้ผู้ใช้งานวางแผนการเดินทางได้สะดวกยิ่งขึ้น
- จุดรับ-ส่งที่แม่นยำ
ระบบช่วยระบุจุดขึ้นรถและจุดปลายทางอย่างชัดเจน พร้อมตัวเลือกสถานที่ที่แนะนำใกล้เคียง เพิ่มความรวดเร็วในการเลือกสถานที่
- ยืนยันการเดินทางได้ง่าย
ผู้ใช้งานสามารถตรวจสอบข้อมูลการเดินทางและยืนยันการเรียกรถได้อย่างรวดเร็ว เพียงกดปุ่ม "ยืนยัน" ในหน้าจอเดียว

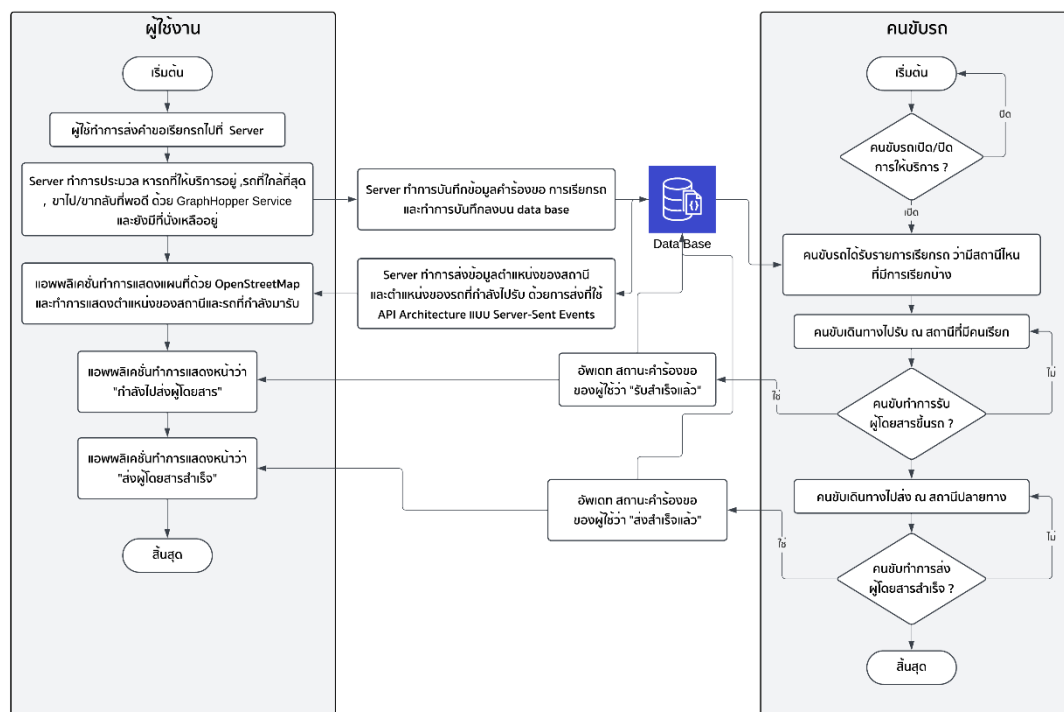


รายละเอียดการเดินทาง (Travel Details)

- ติดตามการเดินทางแบบเรียลไทม์
แสดงข้อมูลเส้นทางและสถานะของรถ EV ที่กำลังเดินทางมาหาคุณ พร้อมตำแหน่งที่อัปเดตแบบเรียลไทม์บนแผนที่
- รายละเอียดรถและสถานะที่นั่ง
ให้ข้อมูลที่ให้บริการ เช่น หมายเลขรถ จำนวนผู้โดยสารปัจจุบัน และความพร้อมของที่นั่ง เพื่อให้ผู้ใช้งานใช้งานได้ทราบ

3.2.5 วางโครงสร้างระบบ (System Architecture)

- ใช้ Flow Diagram อธิบายการทำงานของระบบตั้งแต่ผู้ใช้ส่งคำขอไปยังเซิร์ฟเวอร์ และการประมวลผลเส้นทางใน GraphHopper
- แสดงการเชื่อมต่อระหว่างผู้ใช้งาน, ระบบ API, และเซิร์ฟเวอร์จัดการข้อมูล



Flow Chart การทำงานของแอปพลิเคชัน

ฝั่งผู้ใช้ (User Side):

1. เริ่มต้น
 - ผู้ใช้เข้าสู่แอปพลิเคชันและทำการส่งคำขอเรียกไปยังเซิร์ฟเวอร์ (Request).
2. การประมวลผลคำขอโดยเซิร์ฟเวอร์
 - เซิร์ฟเวอร์ค้นหาหาที่ให้บริการใกล้ที่สุดจากตำแหน่งของผู้ใช้และจุดหมายปลายทาง โดยใช้ GraphHopper Service และฐานข้อมูล
 - เซิร์ฟเวอร์บันทึกคำขอในฐานข้อมูลและเริ่มส่งข้อมูลตำแหน่งแบบเรียลไทม์ (ผ่าน Server-Sent Events)
3. แสดงผลในแอปพลิเคชันของผู้ใช้
 - แอปพลิเคชันแสดงแผนที่ OpenStreetMap พร้อมตำแหน่งของรถที่กำลังมา. แสดงสถานะ "กำลังไปรับผู้โดยสาร"

4. อัปเดตสถานะเมื่อรถมาถึง
 - เมื่อรถมาถึงจุดรับ เซิร์ฟเวอร์อัปเดตสถานะในฐานข้อมูล.
 - ผู้ใช้เห็นข้อความ "กำลังเดินทางไปยังจุดหมาย"
5. สิ้นสุด
 - เมื่อถึงจุดหมายปลายทาง เซิร์ฟเวอร์อัปเดตสถานะคำขอเป็น "สำเร็จ" และแอปพลิเคชันแสดงข้อความ "ส่งผู้โดยสารสำเร็จ"

ฝั่งคนขับ (Driver Side)

1. เริ่มต้น
 - คนขับเปิดแอปพลิเคชันและเข้าสู่สถานะ "เปิดให้บริการ"
2. รับคำขอเรียกรถ
 - เซิร์ฟเวอร์แจ้งเตือนคำขอเรียกรถที่เข้ามาในระบบ
 - คนขับสามารถดูข้อมูลคำขอและยืนยันการรับงาน
3. เดินทางไปยังจุดรับผู้โดยสาร
 - คนขับนำทางไปยังตำแหน่งที่ผู้โดยสารอยู่ โดยใช้ระบบแผนที่ที่บูรณาการกับแอป
4. ตรวจสอบสถานะการรับผู้โดยสาร
 - หากรับผู้โดยสารสำเร็จ : เซิร์ฟเวอร์อัปเดตสถานะเป็น "รับสำเร็จและกำลังไปส่ง"
 - หากไม่สำเร็จ : คนขับสามารถแจ้งปัญหาและเซิร์ฟเวอร์อัปเดตสถานะ
5. เดินทางไปยังจุดหมายปลายทาง
 - คนขับเดินทางตามเส้นทางที่แนะนำไปยังจุดหมาย
6. ตรวจสอบสถานะการส่งผู้โดยสาร
 - หากส่งสำเร็จ : เซิร์ฟเวอร์อัปเดตสถานะเป็น "ส่งสำเร็จ"
 - หากไม่สำเร็จ : คนขับสามารถแจ้งปัญหาและเซิร์ฟเวอร์อัปเดตสถานะ
7. สิ้นสุด
 - กลับเข้าสู่สถานะพร้อมรับคำขอใหม่

กระบวนการในฝั่ง Server และ Database

1. Server-Sent Events (SSE)

- Server ใช้เทคโนโลยี SSE (Server-Sent Events) ในการส่งข้อมูลแบบเรียลไทม์ไปยังทั้งฝั่งคนขับและผู้ใช้งาน เช่น การอัปเดตตำแหน่งของรถและสถานะคำขอ

2. การอัปเดตสถานะคำขอใน Database

- Server ทำการบันทึกและอัปเดตสถานะคำขอ เช่น จาก "รอดำเนินการ" เป็น "รับสำเร็จ" เพื่อให้ข้อมูลล่าสุดถูกต้องและใช้งานได้ทันที

3.3 พัฒนาระบบต้นแบบ

3.3.1 พัฒนาแอปพลิเคชัน

- ใช้ FlutterFlow สำหรับสร้างแอปพลิเคชันและปรับแต่ง Custom Widget เพื่อรองรับการแสดงผลตำแหน่งรถแบบเรียลไทม์
- พัฒนา API สำหรับเชื่อมต่อกับ GraphHopper และ OpenStreetMap

3.3.2 จัดการเส้นทางการเดินทาง

- เชื่อมต่อกับ GraphHopper เพื่อจัดการเส้นทางและคำนวณเส้นทางที่เหมาะสม
- ทดสอบการรับส่งข้อมูลระหว่างระบบและรถโดยสาร

3.3.3 การแสดงผลแผนที่

- ใช้ OpenStreetMap แสดงตำแหน่งรถในแผนที่แบบเรียลไทม์
- ทดสอบความแม่นยำในการแสดงตำแหน่งรถและข้อมูลที่เกี่ยวข้อง

3.3.4 ทดสอบระบบต้นแบบ

- ทดสอบการใช้งานระบบในสภาพแวดล้อมจำลอง โดยให้ผู้ใช้ทดลองใช้งานและเก็บข้อมูลการทำงาน



3.4 การทดสอบและปรับปรุงระบบ

3.4.1 การทดสอบระบบในสถานการณ์จริง

- ทดลองใช้งานระบบในมหาวิทยาลัย โดยเริ่มจากกลุ่มตัวอย่างเล็กๆ
- วัดประสิทธิภาพ เช่น ระยะเวลาการรอ ความถูกต้องของตำแหน่ง และความพึงพอใจของผู้ใช้งาน

3.4.2 ปรับปรุงระบบตามผลการทดสอบ

- ปรับปรุงการทำงาน เช่น การตอบสนองของแอป การเพิ่มฟีเจอร์ที่จำเป็น
- พัฒนาประสิทธิภาพการประมวลผลเส้นทางและแสดงผลแบบเรียลไทม์

3.5 การนำระบบไปใช้งานจริง

3.5.1 ติดตั้งระบบ

- ติดตั้งระบบเต็มรูปแบบในมหาวิทยาลัย
- ประสานงานกับหน่วยงานที่เกี่ยวข้องเพื่อจัดการดูแลโดยสภาระบบและการใช้งานระบบ

3.5.2 ฝึกอบรมและสนับสนุน

- ให้คำแนะนำผู้ใช้งานแอปพลิเคชัน เช่น วิดีโอสอนการใช้งาน โดยจะมีการโพสต์วิดีโอผ่านเพจต่างๆ ภายในมหาวิทยาลัย

3.5.3 เก็บข้อมูลการใช้งาน

- บันทึกข้อมูลการจราจร เส้นทาง และข้อคิดเห็นจากผู้ใช้งาน
- นำข้อมูลมาวิเคราะห์เพื่อปรับปรุงระบบในอนาคต

4. ผลการวิจัย

4.1 ผลลัพธ์ของการพัฒนาตัวต้นแบบ (Prototype Result)

4.1.1 การพัฒนาส่วนของแอปพลิเคชัน (Mobile Application)

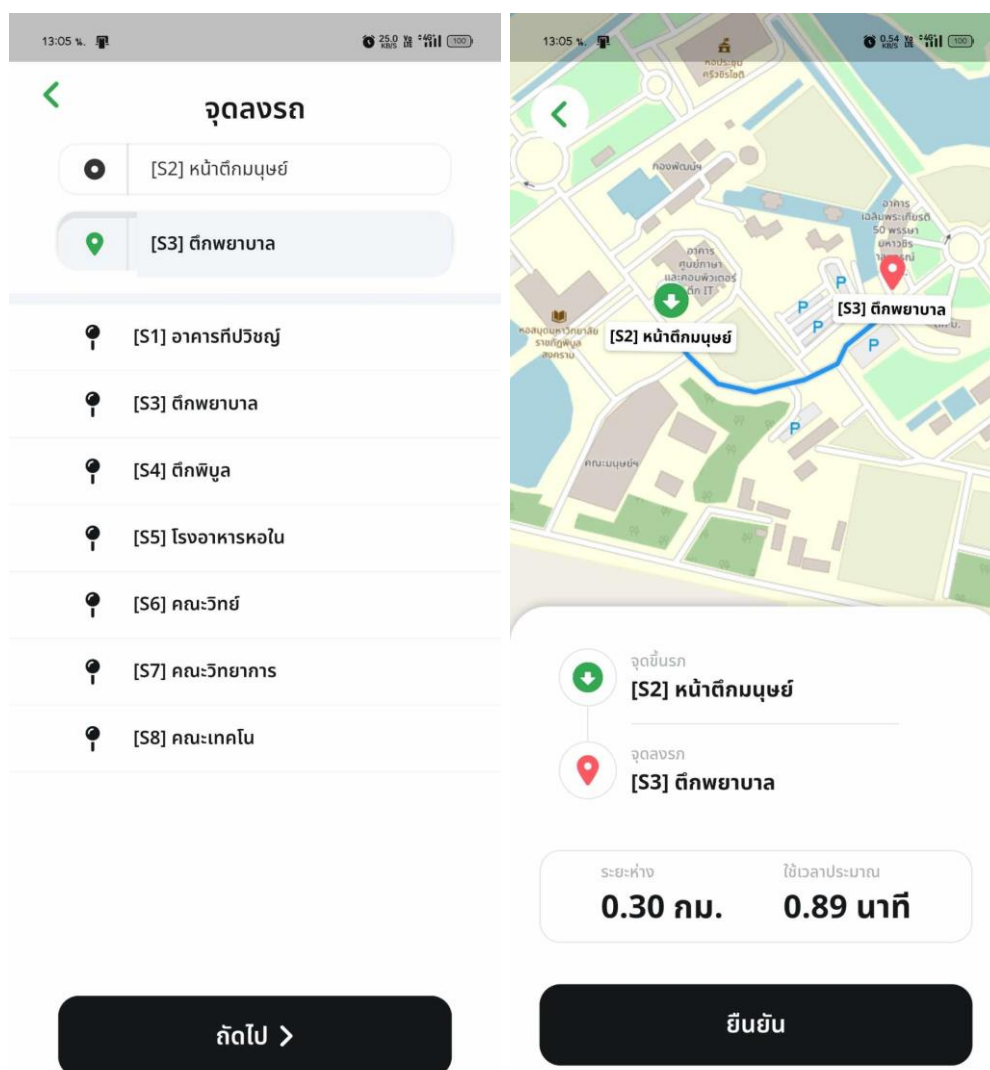
4.1.1.1 ผลลัพธ์ของการพัฒนาส่วนของหน้า Home Page



- สามารถตรวจสอบข้อมูลของรถที่กำลังวิ่งอยู่บนแผนที่
- สามารถตรวจสอบข้อมูลของสถานที่ทำการเชื่อมกันเป็นลำดับ
- สามารถตรวจสอบว่า รถผ่านสถานีไหนมาแล้ว และกำลังจะไปสถานีไหน
- สามารถตรวจสอบว่าใน ณ ขณะนั้นรถกำลังเป็น ขาไป หรือ ขากลับ
- สามารถตรวจสอบสถานะการให้บริการของรถได้ว่า รถคันนั้นให้บริการอยู่ไหม

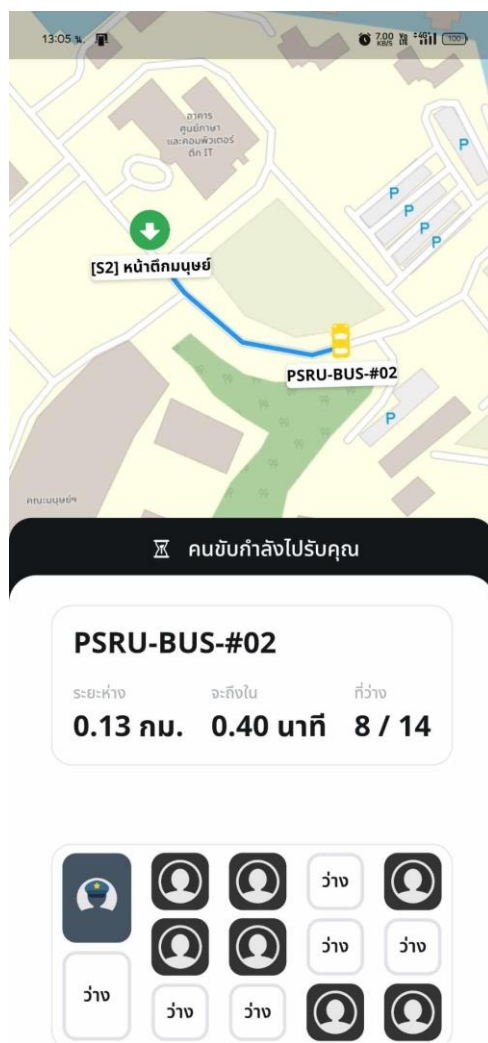
- สามารถตรวจสอบจำนวนคนที่นั่งอยู่ในรถได้ว่ามีจำนวนคนนั่งอยู่เท่าไร

4.1.1.2 ผลลัพธ์ของการพัฒนาส่วนของหน้า Select Station



- ผู้ใช้สามารถทำการเลือกสถานีต้นและสถานีปลายทางที่ต้องการได้
- หลังจากที่ได้เลือกสถานีต้นทางและปลายทางผู้ใช้สามารถทราบถึง ระยะทาง โดยประมาณและระยะเวลาโดยประมาณที่ใช้ในการเดินทาง

4.1.1.3 ผลลัพธ์ของการพัฒนาส่วนหน้าจอ หน้า Real-Time Trip



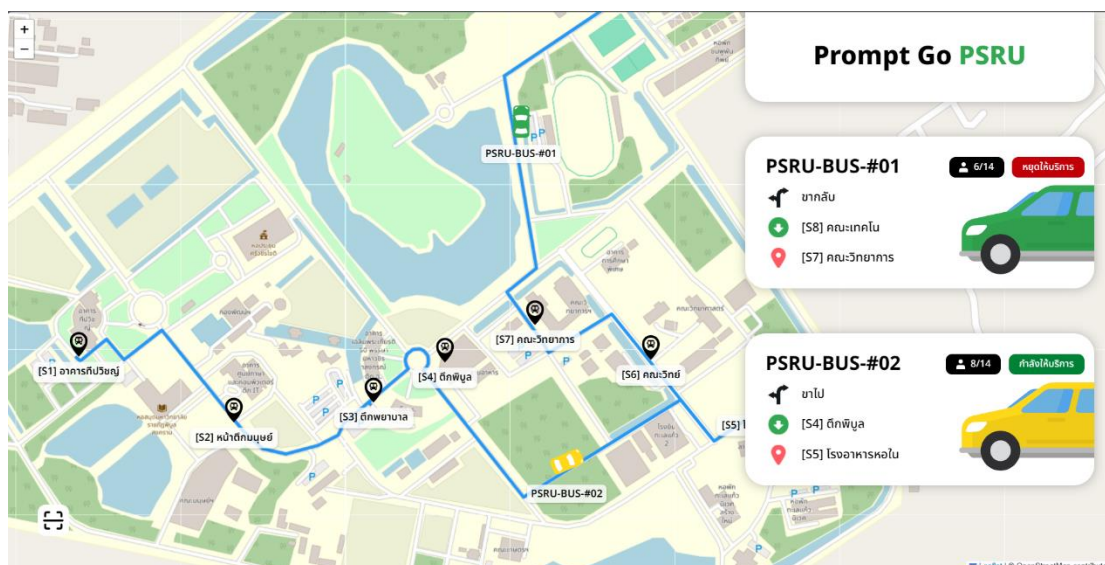
- ผู้ใช้สามารถทราบถึงข้อมูลชื่อของรถที่กำลังมารับผู้ใช้หลังจาก ผู้ใช้ทำการกดเรียกรถ
- ผู้ใช้สามารถทราบถึงสถานะว่า ขณะนี้รถกำลังไปรับหรือไปส่งผู้ใช้
- ผู้ใช้สามารถทราบถึงระยะทางโดยประมาณที่ใช้ในการเดินทาง ณ ขณะนั้น
- ผู้ใช้สามารถทราบถึงระยะเวลาโดยประมาณที่ใช้ในการเดินทาง ณ ขณะนั้น
- ผู้ใช้สามารถทราบถึงสถานะที่นั่งที่ว่างและไม่ว่างของบนรถแบบละเอียด ณ ขณะนั้น

4.1.2 การพัฒนาส่วนของเว็บแอปพลิเคชัน (Web Application)

การพัฒนามนเว็บแอปพลิเคชันแบ่งเป็นหลักๆ สองส่วน โดยประกอบไปด้วย ส่วนของผู้ใช้งานทั่วไป และ ส่วนของการใช้งานระบบของคณบดี ดังนี้

4.1.2.1 การพัฒนาส่วนของผู้ใช้งานทั่วไป (Public Website)

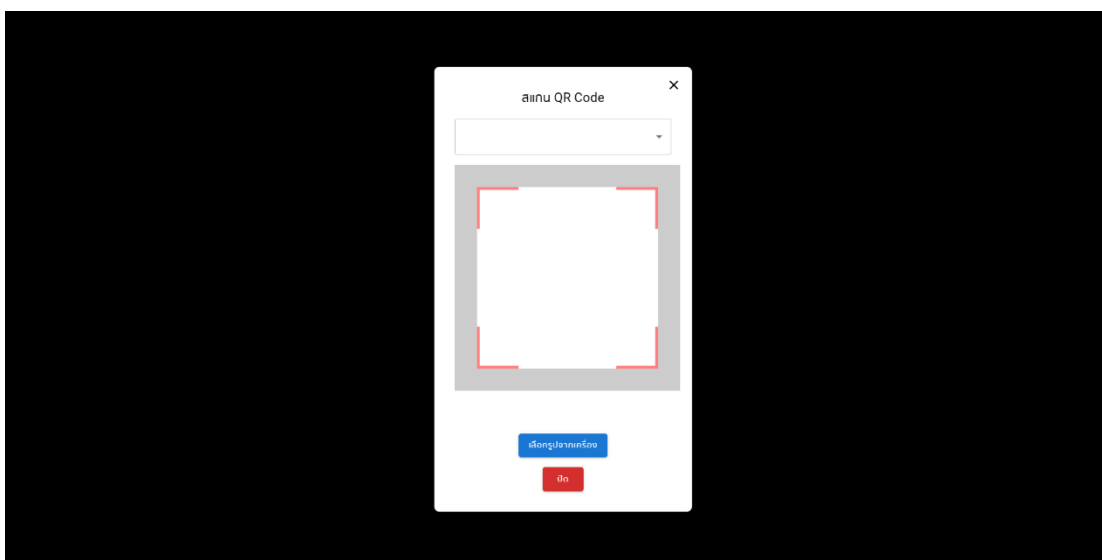
4.1.2.1.1 ผลลัพธ์ของการพัฒนาส่วนของหน้าผู้ใช้งานทั่วไป (Public Website)



- สามารถตรวจสอบข้อมูลของรถที่กำลังวิ่งอยู่บนแผนที่
- สามารถตรวจสอบข้อมูลของสถานีที่ทำการเชื่อมกันเป็นลำดับ
- สามารถตรวจสอบว่า รถผ่านสถานีไหนมาแล้ว และกำลังจะไปสถานีไหน
- สามารถตรวจสอบว่าใน ณ ขณะนั้นรถกำลังเป็น ขาไป หรือ ขากลับ
- สามารถตรวจสอบสถานะการให้บริการของรถได้ว่า รถคันนั้นให้บริการอยู่ไหม

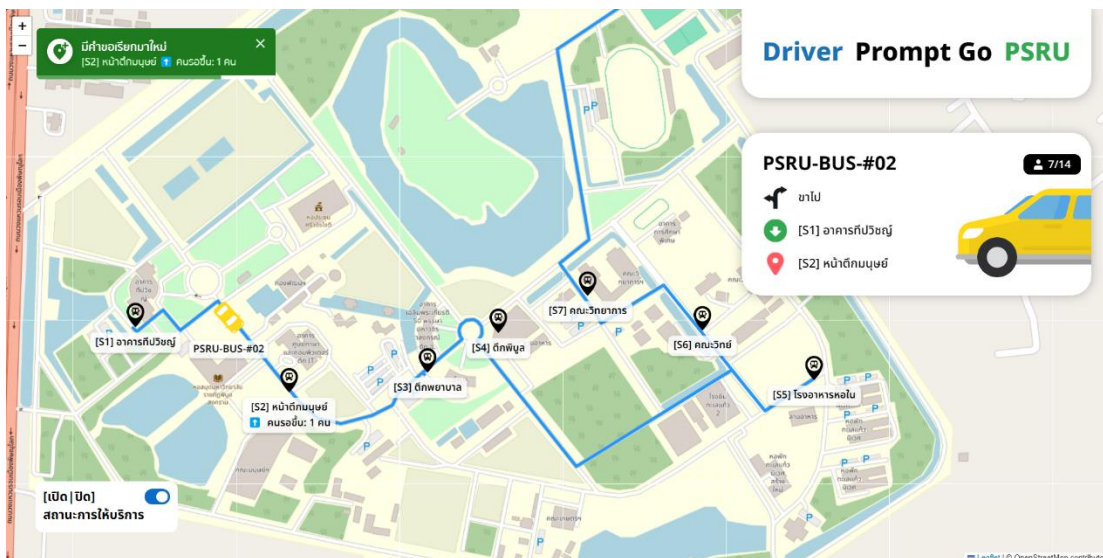
4.1.2.2 การพัฒนาส่วนของระบบขับเคลื่อน (Driver Management System)

4.1.2.2.1 ผลลัพธ์ของการพัฒนาส่วนของระบบขับเคลื่อนหน้า Login Page (Driver Management System)



- คนขับรถสามารถเข้าสู่ระบบหน้าของ Login Page ของคนขับได้ ที่มุมซ้ายล่างของหน้าในหัวข้อ 4.1.2.1.1 ผลลัพธ์ของการพัฒนา ส่วนของหน้าผู้ใช้งานทั่วไป (Public Website)
- คนขับสามารถเลือก input ของกล้องสำหรับการสแกน QR Code เพื่อทำการระบุตัวตนตามที่ได้พัฒนาได้ให้ไป
- คนขับสามารถกดปุ่ม “เลือกรูปภาพจากเครื่อง” เพื่อทำการเลือก QR Code ในเครื่องเพื่อทำการสแกนระบุตัวตนได้เช่นกัน
- คนขับสามารถทำการกดปุ่ม “ปิด” หากไม่ต้องการที่จะทำการเข้าสู่ระบบ

4.1.2.2.2 ผลลัพธ์ของการพัฒนาส่วนของระบบขับเคลื่อนหน้า Driver Page (Driver Management System)



- คนขับสามารถที่จะทำการเปิด/ปิด สถานะการให้บริการได้
- คนขับสามารถทราบถึงจำนวนของ “คนรอขึ้น” และ “คนรอลง” ของแต่ละสถานีได้
- คนขับสามารถรับการแจ้งเตือนที่มุมซ้ายบน เมื่อมีรายการเรียก ขึ้นมาใหม่ได้
- คนขับสามารถตรวจสอบข้อมูลของรถที่กำลังวิ่งอยู่บนแผนที่
- คนขับสามารถตรวจสอบข้อมูลของสถานีที่ทำการเชื่อมกันเป็น ลำดับ
- คนขับสามารถตรวจสอบว่า รถผ่านสถานีไหนมาแล้ว และกำลัง จะไปสถานีไหน
- คนขับสามารถตรวจสอบว่าใน ณ ขณะนั้นรถกำลังเป็น ขาไป หรือ ขากลับ
- คนขับสามารถทราบถึงจำนวนของคนที่นั่งอยู่บนรถ ณ ปัจจุบัน ได้



4.2 ผลลัพธ์ของทดสอบระบบ (System Testing Results)

4.2.1 ผลลัพธ์ปัญหาและข้อสังเกตของการทดสอบส่วนของแอปพลิเคชัน (Mobile Application)

- การเปลี่ยนสถานะรถมารับและมาส่งของสถานีบางสถานียังไม่ค่อยสัมพันธ์กับความเป็นจริง (ปัจจุบันใช้การตรวจสอบว่ารถถึงสถานีในระยะห่างระหว่างรถและสถานีเป็น 50 เมตร)
- มีบางรายการที่ทำการกดเรียกรถแล้วไม่แสดงในส่วนของชื่อรถ ระยะทาง เวลา จำนวนของที่นั่งและรถพร้อมเส้นทางบนแผนที่ (มีโอกาสดังขึ้นน้อยๆ มากๆ อาจเกิดจากปัญหาของ server)
- มีบางครั้งที่ในหน้าที่ติดตามรถแบบ real time ในขณะที่มารับหรือไปส่งไม่อัปเดต อาจเกิดจากการที่ connect ของ SSE (Server Sent Event) ที่ทำการเชื่อมต่ออยู่นั้นเกิดจากหลุด (มีโอกาสดังขึ้นน้อยๆ มากๆ วิธีแก้ปัญหาคือคิดไว้คือ อาจจะทำการเพิ่มเพื่อทำการกดเพื่อ refresh การเชื่อมต่อกับ SSE (Server Sent Event) ใหม่อีกครั้ง)

4.2.2 ผลลัพธ์ปัญหาและข้อสังเกตของการทดสอบส่วนของเว็บแอปพลิเคชันส่วนของคนขับ (Driver Web Application)

- สถานะเปิด-ปิด มีบางจังหวะที่กดเปิดหรือปิดแล้ว ดังกลับมาที่เดิมอาจเป็นปัญหาเนื่องมาจากค่าเริ่มต้นที่นำมาใช้ในการ set ค่าให้ปุ่มเปิด/ปิดนั้นที่มาจากกรดึงข้อมูลแบบ real time มาแทนที่ ก่อนที่จะทำการ request เพื่อทำการอัปเดตสถานะการเปิด/ปิด การให้บริการก่อน
- การแสดงจำนวนคนที่รอขึ้นหรือลง ณ สถานีนั้นๆมีบางจังหวะรถนั้นไปถึงสถานีและจำนวนของของสถานีลดลงแต่ได้หายไปทั้งหมด
- ในหน้าของการสแกนเพื่อทำการเข้าสู่ระบบนั้นหากเปิดบน Browser ที่เป็น Chrome จะเกิดหน้าจอกระพริบระหว่างเปิดกล้อง

7. เอกสารอ้างอิง

- จุฬามณี รุ่งแก้ว, และ อาริยา ตะเคียนราม. (2566). การพัฒนาแอปพลิเคชันสำหรับติดตามรถรับส่งนักเรียน. วารสารวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏศรีสะเกษ, 3(2), 39-49
- สุทธิพงศ์ ประทุม, วิภาวรรณ บัวทอง, และ พิธา จารุพนผล. (2566). ระบบต้นแบบในการตรวจสอบการขึ้น-ลงโดยสารของนักเรียนโดยใช้อุปกรณ์ IoT. การประชุมวิชาการระดับชาติ, 1(1), 10-15
- เนติมา อุดร. (2562). การพัฒนาต้นแบบระบบติดตามรถรับส่งนักเรียนอัจฉริยะบนโมบายแอปพลิเคชันเพื่อเพิ่มความปลอดภัยของเด็กด้วยเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่ง RFID และซอฟต์แวร์รหัสเปิด. วารสารเทคโนโลยีดิจิทัล, 2(1), 55-62
- เปรม ینگเวชชากุล, จิรวดี โยรัมย์, และ กิตติคุณ บุญเกตุ. (2566). การพัฒนาระบบติดตามดูแลนักเรียนที่มีประสิทธิภาพในการโดยสารรถรับ-ส่งประจำโรงเรียน โดยใช้เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งและ RFID. วารสารวิศวกรรมและเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยกาฬสินธุ์, 1(3), 20-25
- มหาวิทยาลัยศิลปากร. (2566). รายงานความร่วมมือในการทดสอบผลงานวิจัยระบบติดตามตำแหน่งรถบริการด้วยโทรศัพท์มือถือ. โครงการพัฒนาระบบติดตามและให้ข้อมูลรถบริการภายในมหาวิทยาลัยศิลปากร, 15-20