Tramer เป็นแพลตฟอร์ม สำหรับอำนวยความสะดวกในการเดินทางและวางแผนการเดินทางด้วยรถแทรมในมหาวิทยาลัยมหิดล ผู้ใช้จะสามารถ ตรวจสอบตารางเวลา และ เส้นทางการเดินรถแทรม อีกทั้งยัง สามารถ ค้นหาป้ายสถานีรถแทรมใกล้ ๆ สถานที่ต่าง ๆ ค้นหาเส้นทางในการเดินทาง คำนวณเวลาการเดินทางของรถแทรม โดยการดึงข้อมูลจาก google map api เพื่อช่วยเหลือผู้ที่ต้องการวางแผนการเดินทาง และ ผู้ที่ไม่ชำนาญเส้นทางและสถานที่ต่าง ๆ ในมหาวิทยาลัยมหิดลนั้น สามารถ เดินทางได้ง่ายและสะดวกมากขึ้น

**ที่มาและความสำคัญ**

ปัจจุบันการเดินทางด้วยรถแทรมในมหาวิทยาลัยยังอำนวยความสะดวกไม่ได้มากนักสำหรับนักศึกษาที่ไม่ทราบรายละเอียดสถานที่ต่าง ๆ ในมหาวิทยาลัยมหิดล การขึ้นและลงรถแทรมในจุดต่าง ๆ และผู้ที่ต้องการคำนวณเวลาในการเดินทางเพื่อวางแผนการเดินทาง ทั้งนี้ก็เพื่ออำนวยความสะดวกให้กับการเดินทางด้วยรถแทรมในมหาวิทยาลัยมหิดล ผู้จัดทำจึงคิดค้นแพลตฟอร์ม Tramer ที่จะมาอำนวยความสะดวกสำหรับการเดินทางด้วยรถแทรมในมหาวิทยาลัยมหิดล ช่วยค้นหาป้ายสถานีรถแทรมใกล้ ๆ สถานที่ต่าง ๆ ช่วยค้นหาเส้นทางกับสายที่ขึ้นจุดที่ลง พร้อมทั้งคำนวณเวลาการมาถึงและไปยังจุดหมาย ต่าง ๆ ของเส้นทางการเดินรถ

**วัตถุประสงค์**

เป็นตัวช่วยสำหรับการเดินทางด้วยรถแทรมในมหาวิทยาลัยมหิดล เพื่อให้สามารถตรวจสอบเส้นทางและตารางเวลาของรถแทรม เพื่อให้สามารถค้นหาป้ายรถแทรมที่ใกล้สถานที่ต่าง ๆ เพื่อให้สามารถค้นหาเส้นทางการนั่งรถและคำนวณเวลาที่ใช้ในการเดินทาง

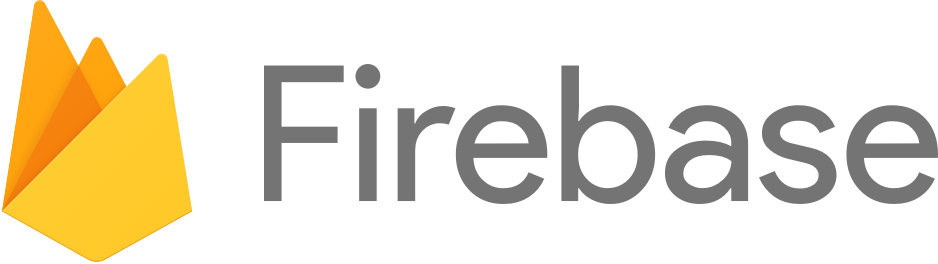
User Story

* ผู้โดยสารสามารถทราบตำแหน่งของรถแทรม เพื่อช่วยในการตัดสินใจขึ้นรถ
* ผู้โดยสารสามารถทราบตารางเวลาของรถแทรม เพื่อคำนวณเวลาที่ใช้ในการรอรถแทรมเพื่อเดินทาง
* ผู้โดยสารสามารถทราบสายรถที่ต้องขึ้นจากต้นทางไปปลายทาง
* ผู้โดยสารสามารถทราบแผนที่การเดินทางของรถทุกสาย เพื่อวางแผนการเดินทาง
* ผู้โดยสารสามารถทราบตำแหน่งของป้ายรถแทรมที่ใกล้ที่สุด เพื่อที่จะสามารถทราบสถานที่ ๆ จะไปรอรถ
* ผู้โดยสารสามารถทราบว่ารถแทรมจะมาถึงและไปถึงป้ายที่เลือกในอีกกี่นาที เพื่อที่จะสามารถเตรียมตัวขึ้นและลงรถได้
* ผู้โดยสารสามารถทราบวิธีการใช้แอพได้ เพื่อที่จะสามารถใช้แอพได้อย่างง่ายดาย
* ผู้โดยสารสามารถแจ้งปัญหาการใช้แอพได้ เพื่อที่ผู้ดูแลจะได้นำไปปรับปรุงแอพต่อไป
* คนขับรถแทรมสามารถส่งตำแหน่ง GPS เพื่อใช้ในฟังก์ชันต่าง ๆ ของแอพ
* คนขับรถแทรมสามารถบันทึกเวลาเดินรถเพื่อเก็ยไปเป็นประวัติการเดินรถ
* คนขับรถแทรมสามารถดูข้อมูลส่วนตัวของคนขับได้
* คนขับรถแทรมสามารถอัพเดทป้ายรถแทรมที่ไปถึงได้
* คนขับรถแทรมและผู้ดูแลระบบสามารถเข้าและออกงานได้ เพื่อบันทึกเวลาการทำงาน
* ผู้ดูแลระบบสามารถเรียกดูและแก้ไขข้อมูลในระบบได้ เพื่อติดตามและทำให้ข้อมูลถูกต้อง เป็นปัจจุบัน
* ผู้ดูแลระบบสามารถบันทึกและเรียกดู ประวัติใช้งานแอพและการแก้ไขข้อมูลต่าง ๆ ได้

**เครื่องมือที่ใช้งาน**



1. ในส่วนของแผนที่จะแก้ชนิดของแผนที่ จะใช้จัดการศูนย์กลางและกำหนดขีดจำกัดของการย่อขยายแผนที่
2. ใช้กำหนดไอคอนมาร์คเกอร์สำหรับสถานที่และสถานีต่าง ๆ
3. ใช้คำนวณและแสดง เส้นทางกับระยะเวลาในการเดินทางของรถ
4. ใช้เพื่อให้สามารถค้นหาสถานที่ใด ๆ ในม.มหิดล ในการค้นหาป้ายสถานีรถแทรม ใกล้ ๆ



Firebase คือฐานข้อมูลแบบ NoSQL โดยจะไม่ใช้ภาษา SQL ในการจัดการข้อมูล แต่ออกแบบให้มีความยืดหยุ่นและเน้นความเร็วในการใช้งาน ซึ่งมีการเก็บข้อมูลแบบ JSON โดยที่มีตารางเหมือนกับ SQL แต่ไม่มีคอลัมน์ ในหนึ่งแถวสามารถเก็บข้อมูลได้ทั้งข้อความ (String) ตัวเลข (Number) รวมไปถึง Object อื่น ๆ

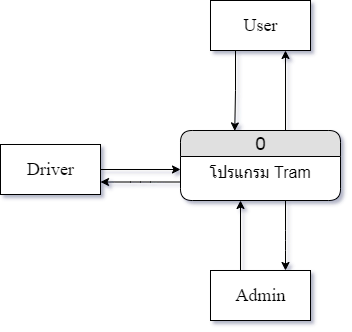
สาเหตุที่ใช้

* Hosting ฟรี
* ใช้งานง่าย
* มี documentation ชัดเจน
* Realtime Database
* มี google analytics และ Crashlytics

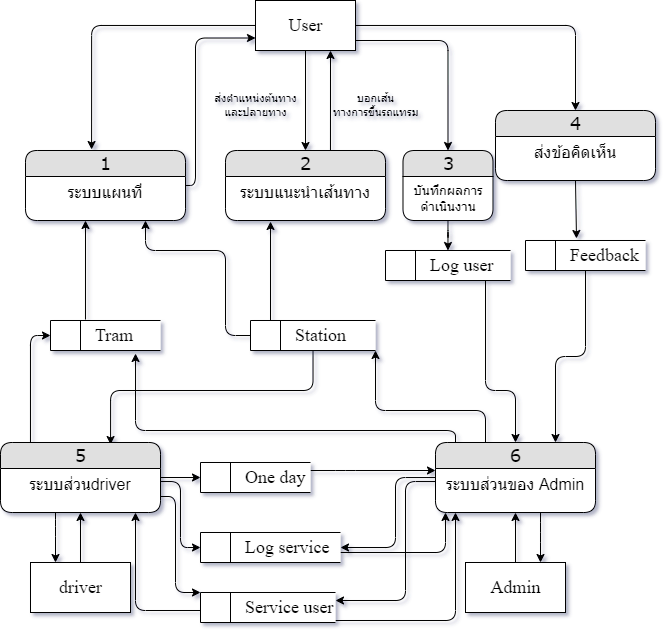
**Software analysis**

โดยจะทำการอธิบายส่วน Software analysisในรูปของ Dataflow โดยจะมีรูปแบบ ดังนี้

**Main layer :** โดยจะแสดงถึงตัวละครทั้งหมดที่ใช้โปรแกรมและการเคลื่อนย้ายของข้อมูล โดยจะมี 3 ตัวละครคือ User, Driver และ Admin



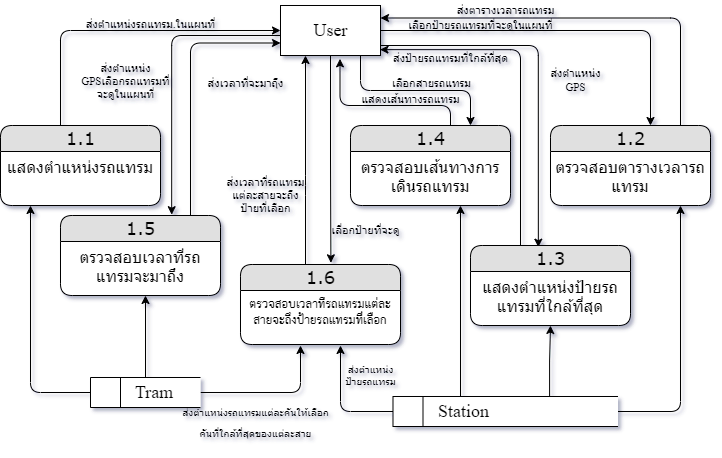
**Layer 0 :** จะแสดงถึงภาพรวมทั้งหมดของการเคลื่อนไหวของข้อมูลทั้งระบบ โดยมีรูปแบบดังรูป



อธิบายองค์ประกอบย่อย

1. **ระบบแผนที่** เป็นระบบฝั่งของ User ที่จะแสดงการเคลื่อนไหวของข้อมูลในฟังก์ชันที่ใช้ในฟังก์ชันแผนที่ โดยจะใช้ข้อมูลจาก Database Tram และ Station ในการทำงานของฟังก์ชัน
2. **ระบบแนะนำเส้นทาง** เป็นระบบฝั่ง User ที่จะแนะนำเส้นทางการเดินทางจากจุดเริ่มต้นไปจุดปลายทาง โดยจะให้ User ส่งตำแหน่งเริ่มต้นและปลายทาง ระบบจะทำการดึงข้อมูลจาก Station เพื่อทำการประมวลผลหาเส้นทางที่ดีที่สุดแล้วส่งแสดงผลให้ User ต่อไป
3. **บันทึกผลการดำเนินงาน** เป็นระบบฝั่ง User ที่จะเก็บการใช้งานทั้งหมดของฝั่ง User แล้วไปเก็บใน Log user
4. **ส่งข้อคิดเห็น** เป็นระบบฝั่ง User ที่จะให้ User ส่งข้อคิดเห็นการใช้งานแอปแล้วนำไปเก็บไว้ใน Feedback
5. **ระบบส่วน Driver** เป็นระบบโดยรวมของฝั่ง Driver ที่จะรวมการทำงานต่างๆของฝั่ง Driver
6. **ระบบส่วนของ Admin** เป็นระบบโดยรวมของฝั่ง Admin ที่จะรวมการทำงานต่างๆของฝั่ง Admin

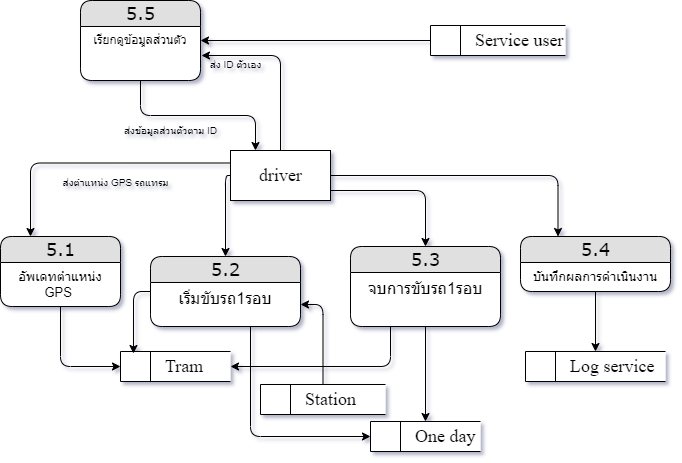
**Layer 1 :** อธิบายการทำงานทั้งหมดในส่วนระบบแผนที่จากใน Layer 0 โดยมีรูปแบบดังรูป



อธิบายองค์ประกอบย่อย

1. **แสดงตำแหน่งรถแทรม** โดยจะแสดงตำแหน่งรถแทรมในแผนที่เป็นจุดๆ โดยจะขอตำแหน่งรถแทรมจาก Tram เพื่อนำมาแสดงผล
2. **ตรวจสอบตารางเวลารถแทรม** โดย User จะเลือกดูตารางเวลาจากการกดที่ป้ายรถแทรมและนำข้อมูลตารางเวลาจาก Station มาแสดงผล
3. **แสดงตำแหน่งรถแทรมที่ใกล้ที่สุด** เป็นการแนะนำให้ User รู้ว่าป้ายรถแทรมที่ใกล้ที่สุดคืออะไร โดยจะทำการนำตำแหน่ง GPS ของผู้ใช้ และตำแหน่งของป้ายรถแทรมมาประมวงผลหาป้ายที่ใกล้ที่สุดแล้วแสดงผลในแผนที่
4. **ตรวจสอบเส้นทางการเดินรถแทรม** เป็นการแสดงเส้นทางการเดินรถแทรมแต่ละสายโดยให้ User เลือกสายที่จะดูและใช้ข้อมูล line จาก Station ที่ต่อ Station เป็นสายและทำมาแสดงผลในแผนที่
5. **ตรวจสอบเวลาการมาถึงของรถแทรม** จะตรวจสอบว่ารถแทรมจะมาถึงป้ายที่ User รอในกี่นาทีโดยดังข้อมูลตำแหน่ง GPS ของ User และรถแทรมที่เลือกแล้วนำมาคำนวณเวลา
6. **ตรวจสอบเวลาที่รถแทรมแต่ละสายจะถึงป้ายรถแทรมที่เลือก** จะตรวจสอบว่ารถแทรมแต่ละสายจะมาถึงป้ายรถแทรมที่เลือกในกี่นาทีโดยดังข้อมูลตำแหน่ง GPS ของ Tram คันถัดไปที่จะผ่านของแต่ละสายและ Station ที่เลือกแล้วนำมาคำนวณเวลา

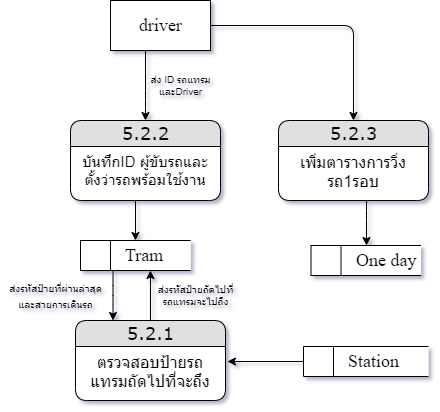
**Layer 5 :** อธิบายการทำงานทั้งหมดในส่วนระบบของ Driver จากใน Layer 0 โดยมีรูปแบบดังรูป



อธิบายองค์ประกอบย่อย

1. **อัพเดทตำแหน่ง GPS** โดยจะส่งตำแหน่ง GPS ของ Driver ที่รถแทรมที่ขับไปเก็บไว้ใน position ของ Tram
2. **เริ่มขับรถ 1 รอบ** จะเป็นกระบวนการทำงานเมื่อรถแทรมเริ่มวิ่ง 1 รอบ
3. **จบการขับรถ 1 รอบ** จะเป็นกระบวนการทำงานเมื่อรถแทรมจบการวิ่ง 1 รอบ
4. **บันทึกผลการดำเนินงาน** เป็นระบบฝั่ง Service Providerในส่วนของ Driver ที่จะเก็บการใช้งานทั้งหมดของ Driver แล้วไปเก็บใน Log service
5. **เรียกดูข้อมูลส่วนตัว** โดยจะส่ง ID ของ Driver เพื่อเรียกดูข้อมูลส่วนตัวของ Driver เอง

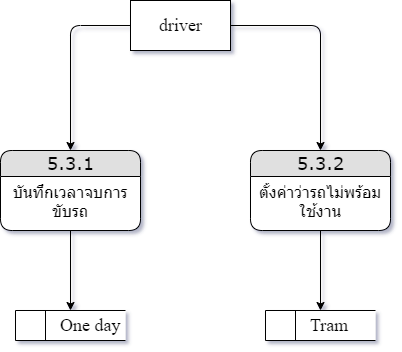
**Layer 5.2 :** อธิบายการทำงานส่วนของ Driver ในส่วนกระบวนการเมื่อเริ่มขับรถ 1 รอบ จากใน Layer 5 โดยมีรูปแบบดังรูป



อธิบายองค์ประกอบย่อย

1. **ตรวจสอบป้ายรถแทรมถัดไปที่จะไปถึง** โดยเมื่อรถแทรมผ่านป้ายไหน Tram จะทำการส่ง ป้ายล่าสุดและสายเดินรถที่เลือกไปให้ Station เพื่ออัพเดทป้ายถัดไปที่จะถึง
2. **บันทึก ID ผู้ขับรถและตั้งค่าว่ารถพร้อมใช้งาน** เมื่อรถเริ่มวิ่งก็จะทำการไปอัพเดทว่าใครขับ รถคันนั้น และให้ Status เป็น 1 ที่แสดงถึงว่ารถคันนี้พร้อมใช้งานและกำลังวิ่งอยู่ในสาย
3. **เพิ่มตารางการวิ่งรถ 1 รอบ** เป็นการบันทึกข้อมูลการวิ่งใน 1 รอบ ในวันนั้นๆอัตโนมัติโดย ส่งเวลาเริ่มวิ่ง รหัสรถแทรม รหัส Driver และสายที่วิ่ง โดยจะจัดเก็บไว้ใน One day

**Layer 5.3 :** อธิบายการทำงานส่วนของ Driver ในส่วนกระบวนการเมื่อจบการขับรถ 1 รอบ จากใน Layer 5 โดยมีรูปแบบดังรูป



อธิบายองค์ประกอบย่อย

1. **บันทึกเวลาจบการขับรถ** เป็นการบันทึกเวลาวิ่งรอบเสร็จใน 1 รอบ โดยส่งเวลาจบไปเก็บใน One day ส่วนของเวลาจบ
2. **ตั้งค่าว่ารถไม่พร้อมใช้งาน** เป็นการอัพเดท Status ให้เป็น 0 แสดงว่ารถไม่พร้อมใช้งาน

**Layer 6 :** อธิบายการทำงานทั้งหมดในส่วนระบบของ Admin จากใน Layer 0 โดยมีรูปแบบดังรูป

รูปภาพประกอบด้วย ข้อความ, สีดำ, สัญลักษณ์, หลากหลาย

คำอธิบายที่สร้างโดยอัตโนมัติ

อธิบายองค์ประกอบย่อย

1. **แก้ไขข้อมูลรถแทรม** Admin จะสามารถแก้ข้อมูลส่วนต่างๆของ Tram ส่วนไหนได้เข้าถึงโดย trami\_d
2. **แก้ไขข้อมูลป้ายรถแทรม** Admin จะสามารถแก้ข้อมูลส่วนต่างๆของ Station ส่วนไหนก็ได้ station\_id
3. **เรียกดูข้อมูลการเดินรถ** Admin จะส่งว่าจะดูข้อมูลการเดินรถวันที่เท่าไรแล้ว One day ก็จะส่งข้อมูลการเดินรถพร้อมรายละเอียดในวันนั้นๆมา
4. **เรียกดูข้อมูลใช้งานของ User** Admin จะสามารถดูประวัติการใช้แอปของ User ได้ โดยมี filter คือ log\_service\_id, service\_user\_id, job\_position, service\_action.type
5. **เรียกดูแบบตอนรับ** Admin จะสามารถเข้าไปดูแบบสอบถามการใช้งานแอปที่ได้จาก User ได้โดยมี filter คือ type ของ feedback
6. **บันทึกเวลาเข้าออกงาน** โดยจะให้ Admin ทำการแก้ Status การเข้า-ออกงานโดยการส่ง ID ของ Admin และเวลาเข้า-ออกงาน
7. **แก้ไขข้อมูลส่วนตัว** โดยจะส่ง ID ของ Service provider และข้อมูลที่แก้แล้วนำไปแก้ข้อมูลใน Service user โดยจะแก้ที่ Service Provider คนไหนก็ได้
8. **เรียกดูข้อมูลส่วนตัว** โดยจะส่ง ID ของ Service provider เพื่อเรียกดูข้อมูลส่วนตัวของ Service provider ที่ค้นหา
9. **เรียกดูข้อมูลใช้งานของ service** โดยจะดูประวัติการแก้ไขข้อมูลของฝั่ง Service provider โดยจะมี fliter คือ log\_user\_id, user\_id, user\_action, nearest\_station\_id, log\_time
10. **บันทึกข้อมูลการใช้งาน** เป็นระบบฝั่ง Service Providerในส่วนของ Admin ที่จะเก็บการใช้งานทั้งหมดของ Admin แล้วไปเก็บใน Log service

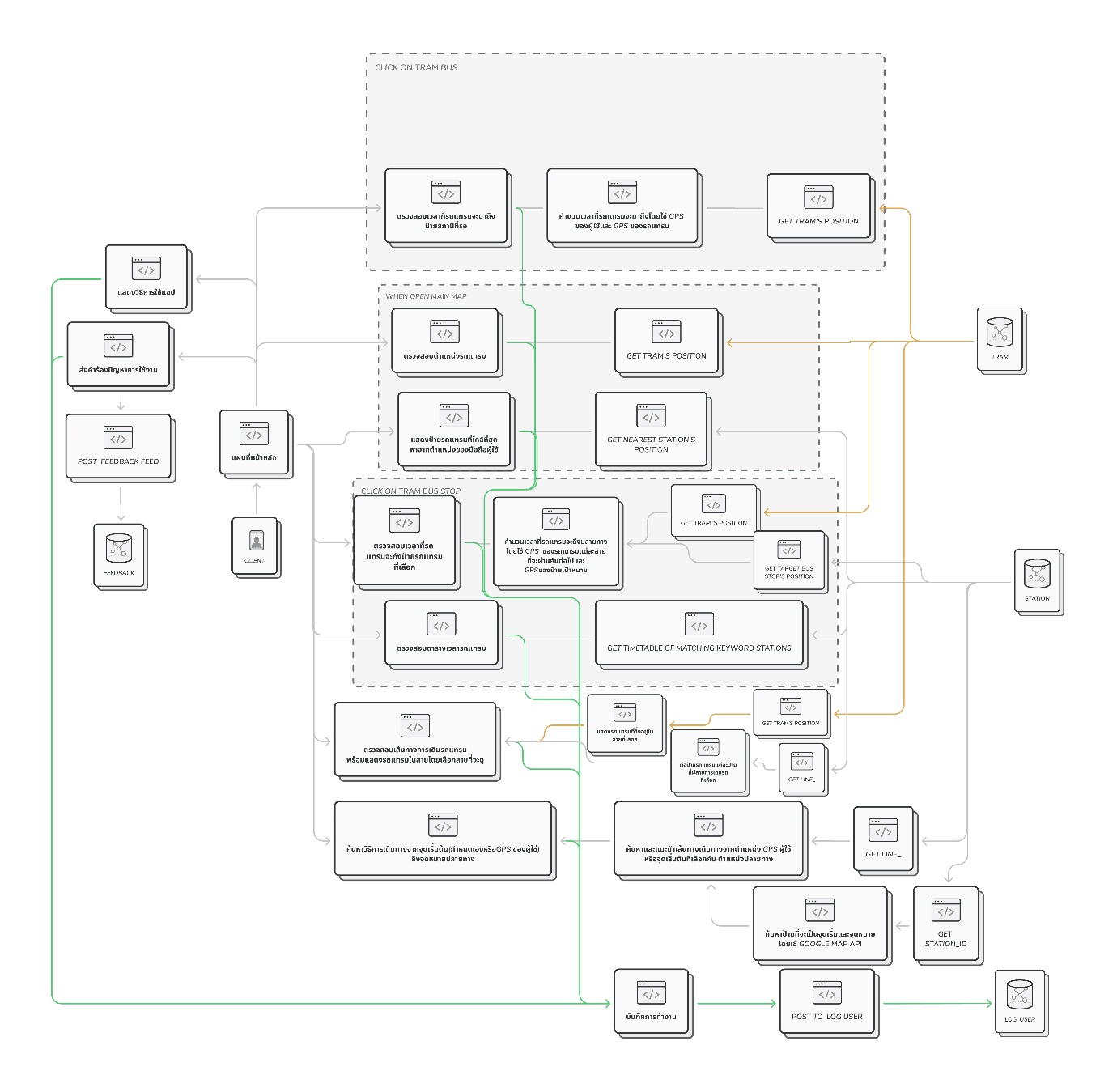
Software Architecture

Front-End (UX/UI Design)

Back-End (Services)

**User**

จะแสดงฟังก์ชันการทำงานของ User โดยจะมีการทำงานโดยรวมดังรูป



การเข้าถึงข้อมูล

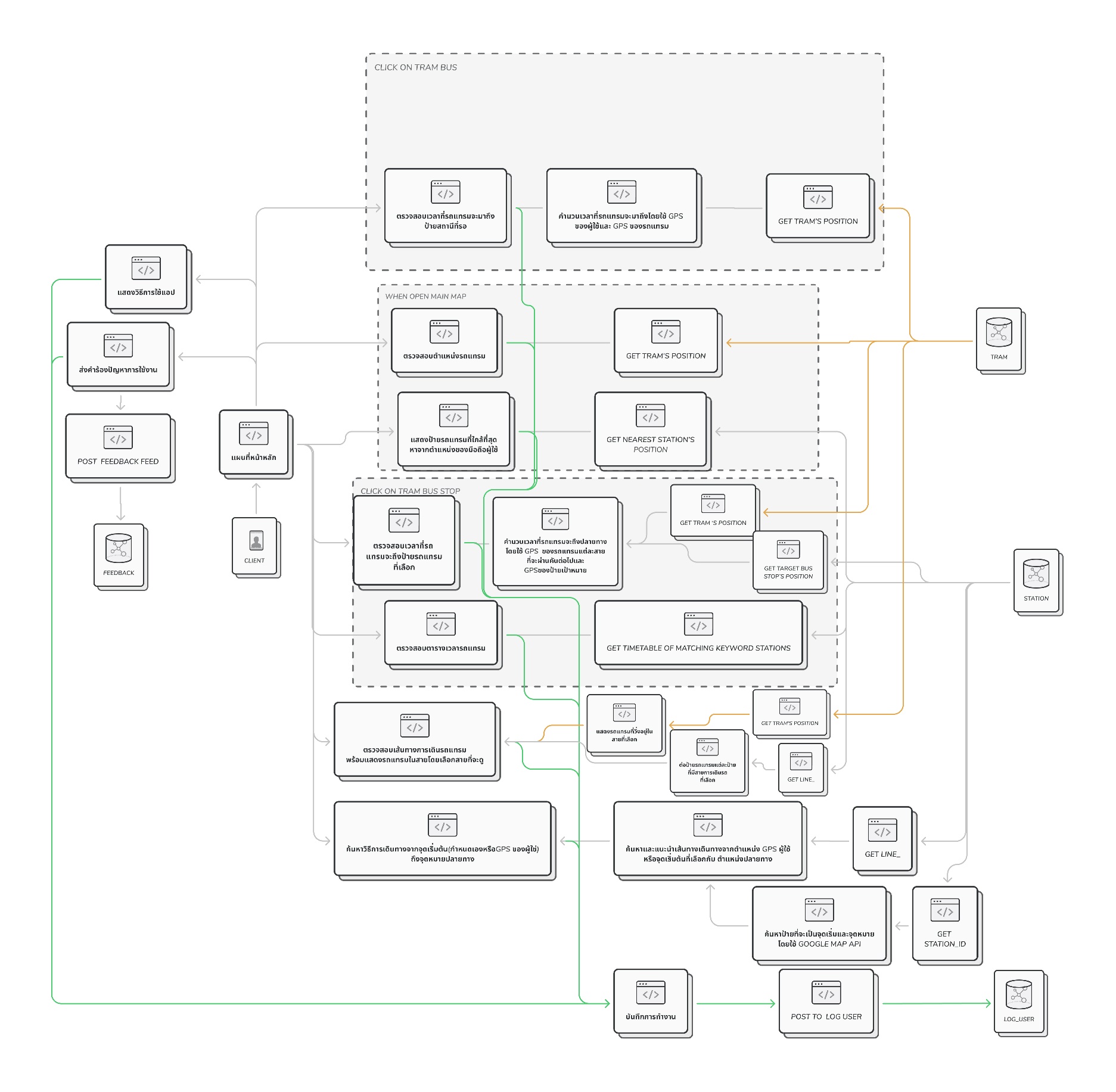
สีเทา Station

สีเขียว Log\_user

สีส้ม Tram

โดยเมื่อเข้าใช้งานฝั่ง user ก็จะเข้าสู่หน้าแผนที่หลักที่แสดงค่าต่างๆในแผนที่ โดยจะมีคำสั่งย่อยตามประเภทการกระทำของ User โดยจะแบ่งการทำงานดังนี้

**เมื่อเข้าแผนที่หน้าหลัก**



การเข้าถึงข้อมูล

สีเทา Station

สีเขียว Log\_user

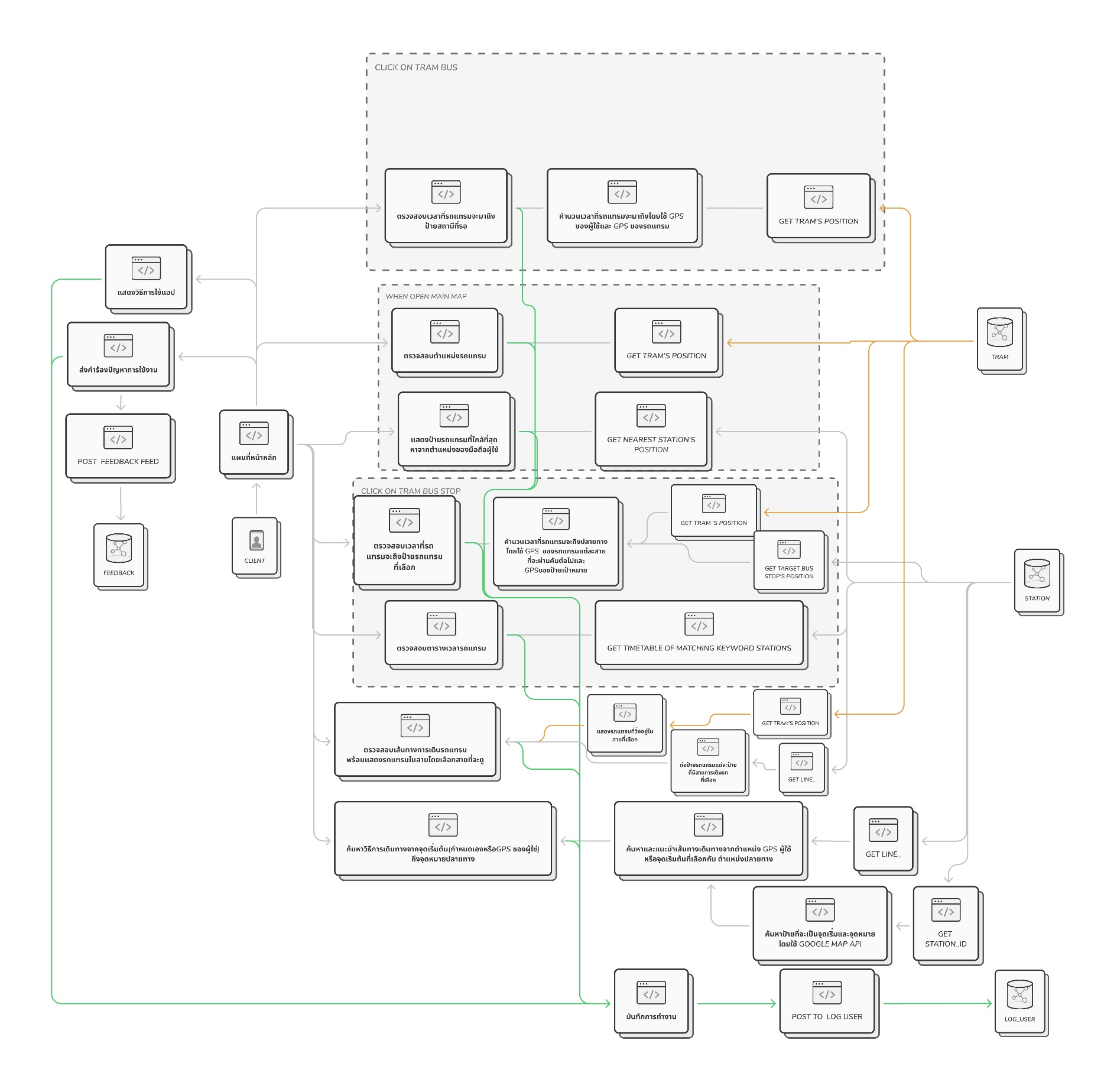
สีส้ม Tram

ตรวจสอบตำแหน่งรถแทรม

เป็นการแสดงตำแหน่งรถแทรมจากในแผนที่โดยใช้ข้อมูลตำแหน่งlatitude และ longitude จากใน Tram database โดยใช้ Google map api ในการแสดงตำแหน่ง

แสดงป้ายรถแทรมที่ใกล้ที่สุดจากตำแหน่งผู้ใช้

เป็นการแสดงป้ายที่มีตำแหน่งใกล้ที่สุดจากตัวผู้ใช้งานโดยเรียกใช้ตำแหน่งป้ายรถแทรมแต่ละป้ายกับตำแหน่งผู้ใช้แล้วมาเปรียบเทียบหาป้ายที่ใกล้ที่สุดแล้วมาแสดงผลทางหน้าจอ

**เมื่อกดที่รถแทรม**

การเข้าถึงข้อมูล

สีเทา Station

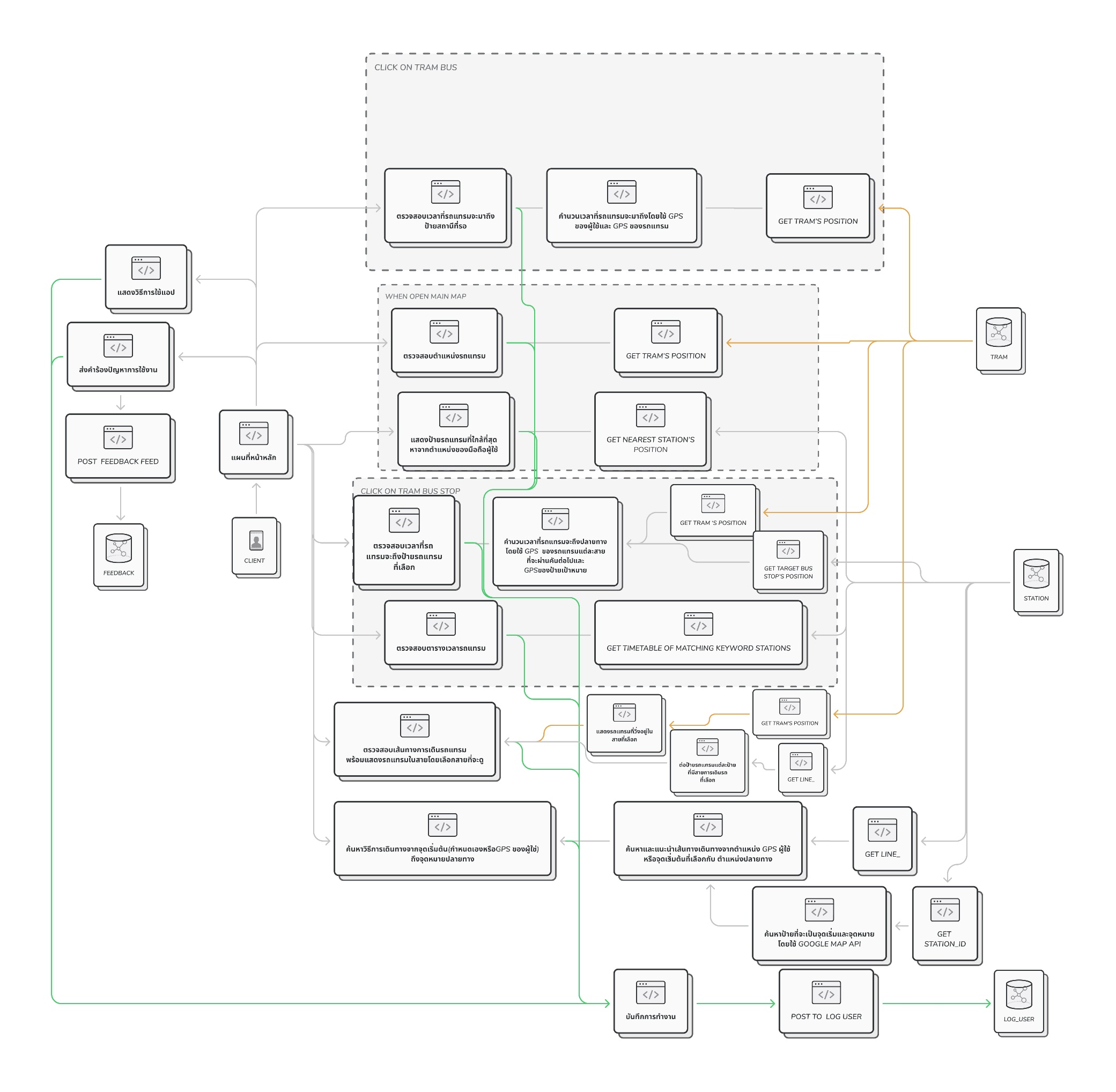
สีเขียว Log\_user

สีส้ม Tram

ตรวจสอบเวลาที่รถแทรมจะมีถึงป้ายสถานีที่รอ

เมื่อกดที่รถแทรม จะทำการคำนวนเวลาที่รถแทรมคันนั้นจะมาถึงที่ป้ายรถแทรมที่รอ โดยจะดึงข้อมูลตำแหน่งของรถแทรมมาจาก Tram และใช้ข้อมูลตำแหน่งของ User มาคำนวณโดยใช้ Google Map API แล้วแสดงผลเวลาที่รถแทรมจะมาถึงป้ายที่รอ

**เมื่อกดที่ป้ายรถแทรม**



การเข้าถึงข้อมูล

สีเทา Station

สีเขียว Log\_user

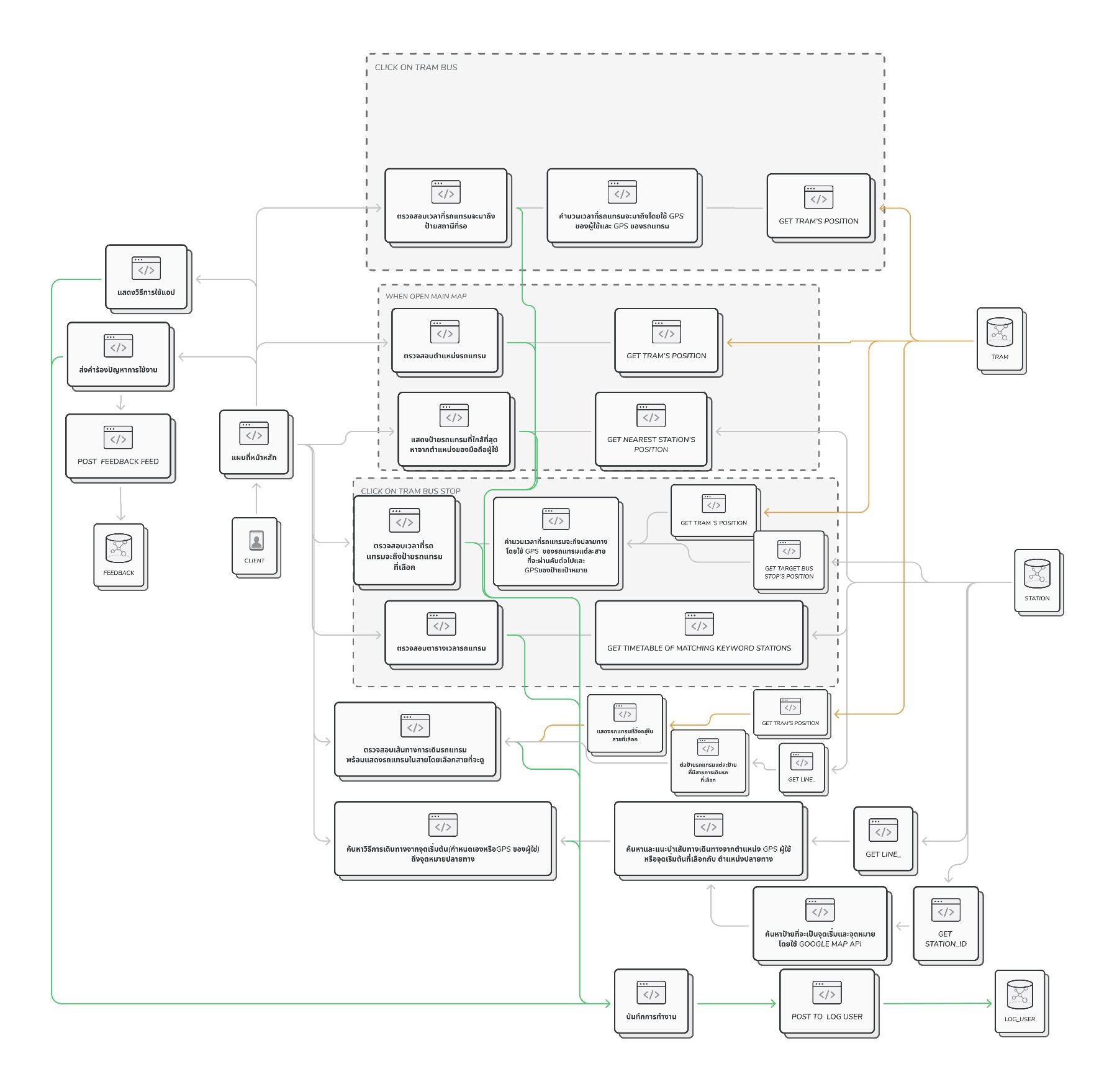
สีส้ม Tram

ตรวจสอบเวลาที่รถแทรมจะถึงป้ายรถแทรมที่เลือก

เมื่อ User กดที่ป้ายรถแทรมนั้น จะแสดงรถแทรมที่จะมาถึงป้ายที่เลือกคันล่าสุดของ แต่ละสาย โดยจะเรียกใช้ข้อมูลตำแหน่ง Tram คันต่อไปที่จะผ่านป้ายรถแทรมที่เลือกในแต่ละสาย และเรียกดูตำแหน่ง Station ที่เลือกแล้วนำมาคำนวณหาเวลาที่จะมีถึงโดยใช้ Google Map API

ตรวจสอบตารางเวลารถแทรม

จะแสดงตารางเวลาการมาถึงของรถแทรมในเวลาต่างๆ โดยจะแบ่งเป็นจันทร์-ศุกร์ และเสาร์-อาทิตย์ โดยจะเรียกดู time\_table ของป้ายรถแทรมที่เลือกแล้วแสดงผล

**อื่นๆ**

การเข้าถึงข้อมูล

สีเทา Station

สีเขียว Log\_user

สีส้ม Tram

ตรวจสอบเส้นทางการเดินรถแทรมพร้อมแสดงรถแทรมในสาย

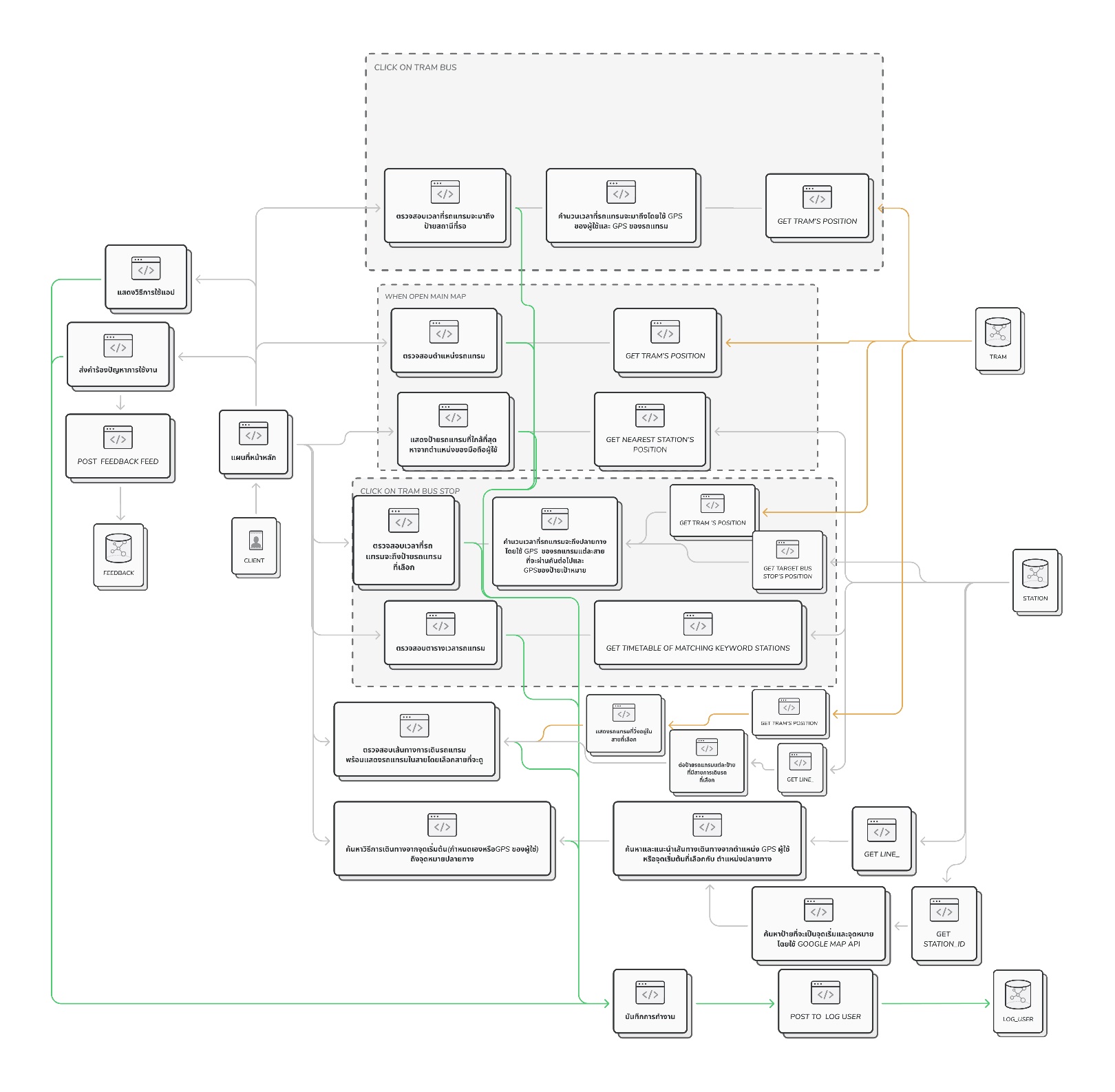
เป็นการให้ User เลือกว่าจะดูสายการเดินรถแทรมของสายไหน โดยพอเลือกสาย

ก็จะไปดึงข้อมูลจาก Station แล้วทำการเชื่อมป้ายรถแทรมแต่ละป้ายที่มีสายที่เลือกโดย

เริ่มจากสถานีต้นทางแล้วเลือกเชื่อมป้ายรถแทรมที่ตัวแปร next\_station\_id ของสายการเดินรถที่เลือก ทำต่อเรื่อยๆ จนวนมาที่สถานีต้นทาง แล้วจะได้เส้นทางการเดินรถที่เลือกที่แสดงผล

ค้นหาวิธีการเดินทางจากจุดเริ่มต้นไปปลายทาง

โดยจะให้ User นำเข้าข้อมูลจุดเริ่มต้น(จะพิมพ์ค้นหาหรือใช้ GPS ของ User เอง) และจุดหมายปลายทางที่เลือก โดยการกำหนดจุดเริ่มและจุดหมายจะใช้เป็นป้ายรถแทรมที่ใกล้ที่สุดแล้วทำการทดลองเชื่อมป้ายรถแทรมจากตัวแปร next\_station\_id ใน station และถ้าจุดหมายอยู่ในสายรถแทรมที่อยู่ตรงข้ามถนน ในข้อมูลป้ายรถแทรมใน station จะเก็บตัวแปร across\_station\_id เพื่อเก็บป้ายตรงข้ามของถนนเพื่อทำการข้ามถนนและหาเส้นทางการเดินรถจนกว่าจะถึงปลายทาง โดยที่ช่องค้นหาจะใช้ข้อมูลพื้นที่จาก Google map API ในการบอกว่าเมื่อเราพิมพ์ค้นหาสถานที่ แล้วจะได้ผลลัพธ์เป็นการบอกว่าสถานที่ที่ค้นหาคือที่ต่อไปนี้ แล้วนำสถานที่นั้นไปหาว่าป้ายที่ใกล้ที่สุดคือที่ไหน แล้วกำหนดจุดนั้นว่าเป็นต้นทางและปลายทางและทำการค้นหาเส้นทางต่อไป

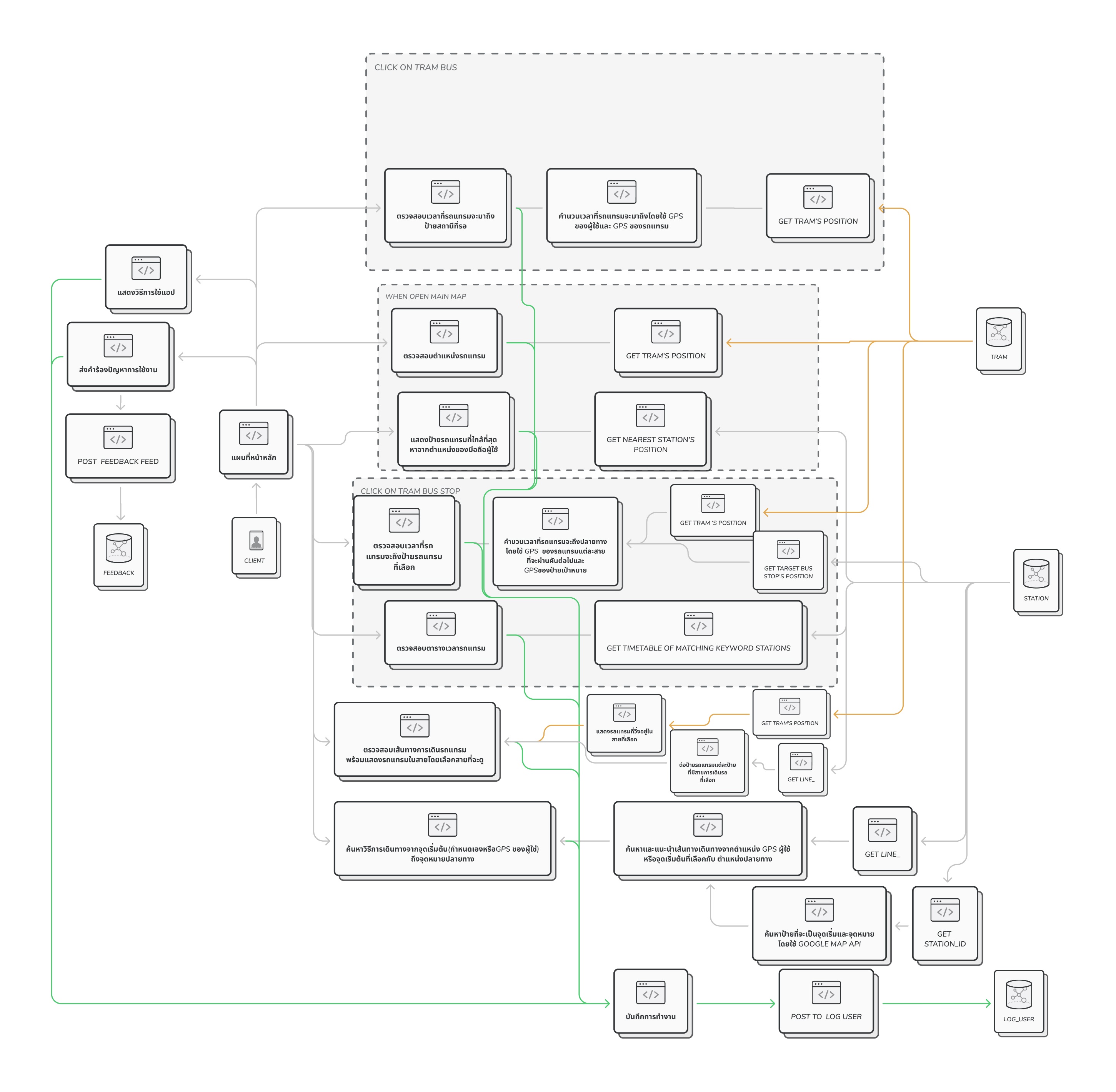
แสดงวิธีการใช้แอป

อธิบายคู่มือในการใช้แอป

ส่งคำร้องปัญหาการใช้แอป

ส่งแบบสอบถามการใช้งานแล้วส่งเขาไปเก็บใน feedback

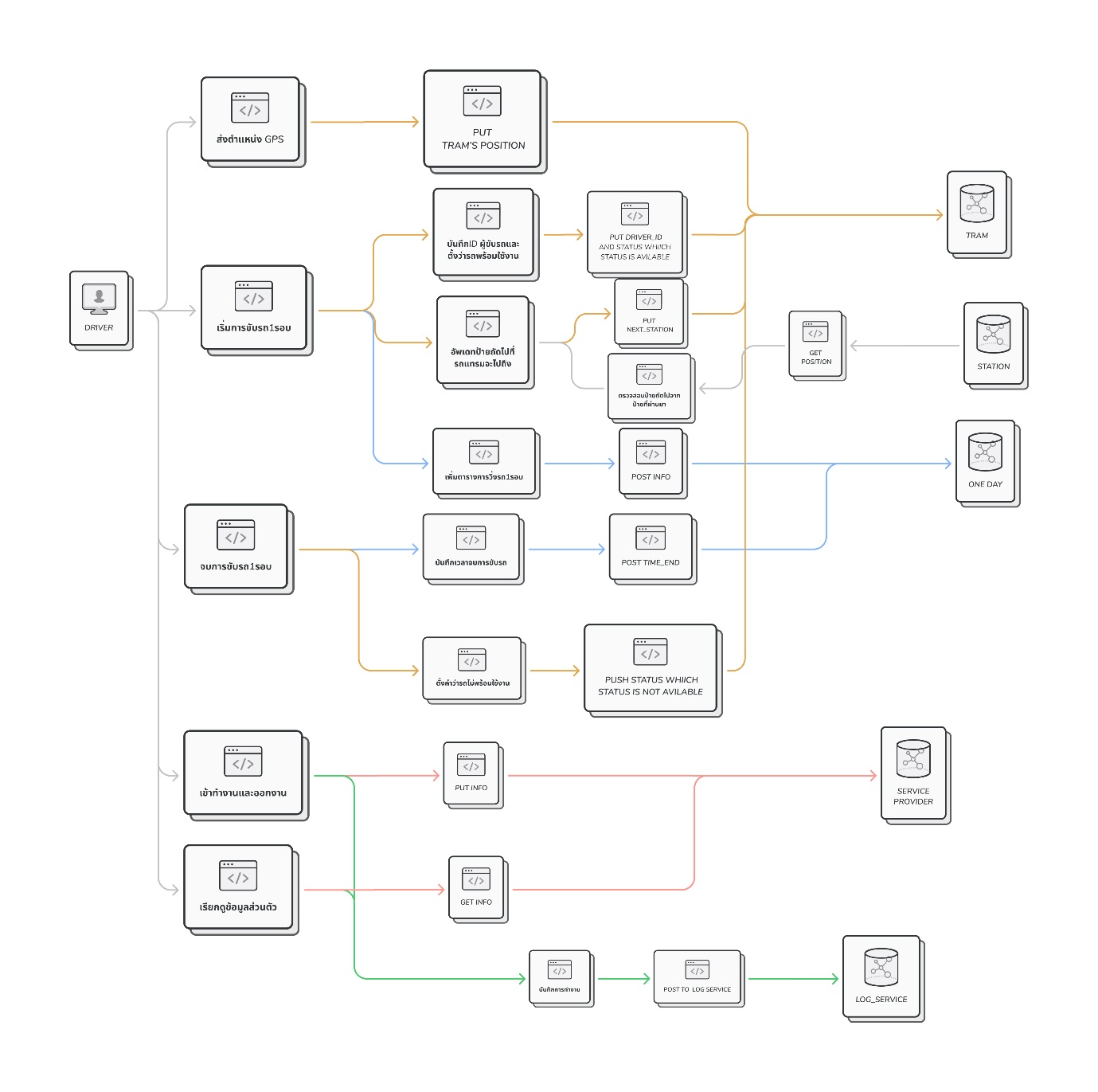
ส่วนบันทึกการใช้งาน



โดยจะเก็บการกระทำทุกอย่างของ User ไว้เป็นประวัติการใช้งาน โดยจะเก็บทุกการกระทำ(สายสีเขียวที่โยงทุกฟังก์ชันการใช้งาน)ไว้ใน Log\_user

**Driver**

จะแสดงฟังก์ชันการทำงานของ Driver โดยจะมีการทำงานโดยรวมดังรูป



**ส่งตำแหน่ง GPS**

การเข้าถึงข้อมูล

สีเทา Station

สีเขียว Log\_service

สีส้ม Tram

สีฟ้า One day

สีแดง Service provider

Driver จะส่งตำแหน่งรถแทรมผ่านทาง GPS แล้วเก็บใน Tram ตาม ID

รถแทรมที่ขับ

**เริ่มการขับรถ 1 รอบ**

เมื่อ Driver สแกน QR code ที่อยู่ที่รถแทรม จะถือว่าเป็นการเริ่มการขับรถ

1 รอบ โดยเมื่อเริ่มการขับรถจะมีกระบวนการย่อยดังนี้

* บันทึก ID ชอง Driver ใน Tram ที่ขับ แล้วตั้งสถานะของรถแทรมว่าพร้อมใช้งาน
* อัพเดทป้ายรถแทรมที่จะถึงถัดไปแล้วใส่ไว้ใน Tram โดยจะตรวจสอบว่าแทรมป้ายถัดไปคือป้ายอะไรจากตำแหน่งใน Station
* บันทึกการเดินรถรอบนั้นๆในวันนั้น ไว้ใน One\_day โดยบันทึกเวลาเริ่ม ID Driver และ ID รถแทรม

**จบการขับรถ 1 รอบ**

เมื่อ Driver ทำการขับรถ 1 รอบเสร็จแล้วกดปุ่มว่าวิ่งรถเสร็จสิ้นแล้ว จะถือว่าจบการขับรถในรอบนั้นๆแล้ว โดยจะมีกระบวนการย่อยดังนี้

* บันทึกเวลาจบการขับรถ โดยจะเก็บเวลาที่ขับรถเสร็จไว้ใน one day ในส่วนที่เพิ่มเข้ามาตอนเริ่มการขับรถ
* ตั้งค่าว่ารถไม่พร้อมใช้งาน โดยไปปรับ status ใน Tram ตาม tram\_id ที่ขับว่ารถไม่พร้อมให้บริการจนกว่าจะมีคนมาขับรถ

**เข้าทำงานและออกงาน**

จะเก็บเมื่อ Driver ได้จำการ login และ logout เข้าสู่ระบบ โดยจะตั้ง status ว่าทำงานและเก็บเวลาเริ่มเมื่อเข้าสู่ระบบ และจะตั้ง status ว่าไม่ทำงานและเก็บเวลาจบเมื่อออกจากระบบ

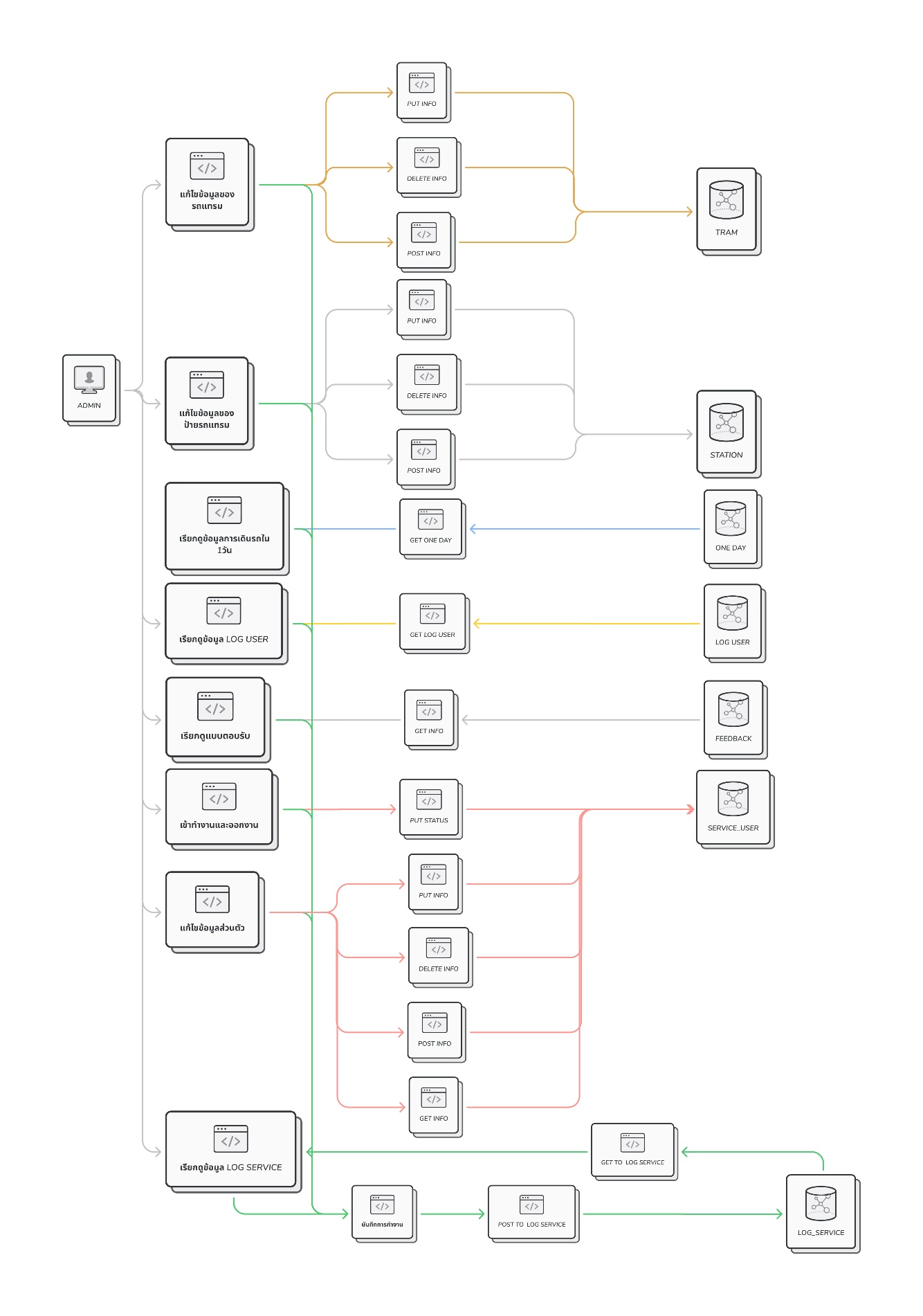
**เรียกดูข้อมูลส่วนตัว**

Driver จะสามารถดูข้อมูลส่วนตัวได้ คือ ID ชื่อ-นามสกุล อีเมล เบอร์โทร และตำแหน่งงาน

**ส่วนบันทึกการใช้งาน**

โดยจะเก็บการกระทำในส่วน เข้าทำงานและออกงาน และ เรียกดูข้อมูลส่วนตัว ไว้เป็นประวัติแล้วเก็บไว้ใน log\_service

**Admin**

จะแสดงฟังก์ชันการทำงานของ Admin โดยจะมีการทำงานโดยรวมดังรูป

การเข้าถึงข้อมูล

สีเทา Station

สีเขียว Log\_service

สีส้ม Tram

สีฟ้า One day

สีแดง Service provider

**แก้ไขข้อมูลส่วนรถแทรม**

โดย Admin จะสามารถเพิ่ม ลบ แก้ ข้อมูลใน Tram ได้ โดยสามารถแก้ tram\_id, status, line ได้

**แก้ไขข้อมูลส่วนป้ายรถแทรม**

โดย Admin จะสามารถเพิ่ม ลบ แก้ ข้อมูลใน Station ได้ โดยสามารถแก้ station\_id, position(latitude, longitude), land\_mark และข้อมูลใน line\_ ได้

**ส่วนการเรียกดูข้อมูล**

Admin นั้น จะสามารถดูข้อมูลส่วนไหนก็ได้ใน Database โดยจะเป็นข้อมูลที่ดูได้เท่านั้น ไม่สามารถแก้ได้ ซึ่งแต่ละข้อมูลก็จะมี Filter ที่ใช้เรียกดูแตกต่างกัน โดยมีดังนี้

* one day filter วันที่ที่ต้องการดู
* feedback filter ประเภทของ feedback ที่ต้องการดู
* service user filter ใช้ service\_user\_id ที่จะดู
* log service filter log\_service\_id, service\_user\_id, job\_position, service\_action.type
* log user filter log\_user\_id, user\_id, user\_action, nearest\_station\_id, log\_time

**ส่วนบันทึกการใช้งาน**

โดยจะเก็บการกระทำทั้งหมดของ Admin ไว้เป็นประวัติแล้วเก็บไว้ใน log\_service

Database

**Database : service\_provider**

ในส่วนของตัวละครที่จะใช้ในโปรเจคนี้คือ 3 ตัวละครคือ ผู้ใช้ทั่วไปหรือ User Driver และ Admin โดย 2 ตัวละครหลังจะสามารถรวมว่าเป็นผู้ให้บริการหรือ Service provider โดย Database นี้จะเป็น Database ไว้เก็บข้อมูลทั่วไปของส่วน Service Provider แต่ส่วนของ User ไม่ได้มีการใช้ Login เข้าสู่ระบบจึงไม่มีการเก็บข้อมูลส่วนนี้จากทาง User โดยมีโครงสร้างดังนี้

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

อธิบายตัวแปรใน Database

**service\_provider\_id :**

รหัสประจำตัวของ Service provider ที่ใช้ในการระบุ Service Provider แต่ละคน

**first\_name & last name :**

เก็บชื่อและนามสกุลของ Service Provider

**job\_position :**

ระบุตำแหน่งงานของ Service Provider แต่ละคนว่าอยู่ตำแหน่งอะไรระหว่าง Driver กับ Admin

**email :**

บอกถึงอีเมลของ Service provider

**phone\_no :**

บอกเบอร์โทรของของ Service provider

**status :**

ระบุว่า Service Provider คนนั้นๆ ได้ทำงานอยู่หรือไม่ โดยกำหนดให้ 0 = ไม่ทำงาน 1 = ทำงาน และอาจจะมีระบุสถานะเพิ่มเติมในอนาคต

**period :**

บันทึกวันเวลาที่ได้เข้าทำงานวันเวลาไหนบ้าง โดยจะสามารถเก็บได้หลายช่วงเวลา โดยจะมีตัวแปรย่อยดังนี้

**date\_ :** ระบุวันที่เข้าระบบ

**time\_in & time\_out :** เก็บเวลาที่เข้า-ออกงาน โดย1วันอาจจะมีการเข้า-ออกงานหลายรอบก็ได้

**Database : tram**

เป็น Database ส่วนของรถแทรมแต่ละคันโดย มีโครงสร้างดังนี้

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

อธิบายตัวแปรใน Database

**tram\_id :**

รหัสประจำตัวของแต่ละรถแทรม ที่ใช้ในการระบุความเป็นรถแทรมแต่ละคัน

**service\_provider\_id(เฉพาะ Driver) :**

รหัสประจำตัวของ Service provider หรือ Driver ที่ใช้ในการระบุคนขับรถแทรมแต่ละคันในขณะนั้น

**line :**

ระบุสายของรถแทรมว่ารถแทรมคันนั้นๆ วิ่งสายไหน โดยจะแบ่งคือ 1: สาย MLC 2: สายสีน้ำเงิน 3: สายสีแดง 4: สายสีเขียว

**status :**

ระบุสถานะของรถแทรม โดยกำหนดให้ 0 = ไม่ทำการวิ่ง 1 = ทำการวิ่ง 2 = ซ่อมบำรุง 3 = ปลดประจำการ และอาจจะมีระบุสถานะเพิ่มเติมในอนาคต

**position :**

ระบุตำแหน่งของรถแทรมที่จะมีการอัพเดทตลอดเวลาที่มีการวิ่งของรถโดยโดยจะเก็บเป็น latitude และ longitude

**Database : station**

ไว้เก็บข้อมูลของป้ายรถแทรมของแต่ละป้าย โดยข้อมูลเส้นทางจะเก็บเป็นให้ป้ายรถแทรมต่อๆกันจนเป็นสาย โดยมีโครงสร้างดังนี้

รูปภาพประกอบด้วย ข้อความ, หน้าจอ

คำอธิบายที่สร้างโดยอัตโนมัติ

อธิบายตัวแปรใน Database

**station\_id :**

รหัสของแต่ละป้ายรถแทรม ที่ใช้ในการระบุป้ายรถแทรมแต่ละป้าย

**latitude& longitude :**

ระบุตำแหน่งของป้ายแทรมที่ใช้บอกตำแหน่งรถแทรม โดยตำแหน่งจะคงที่ตลอด

**landmark :**

ไว้เก็บตึกบริเวณป้ายโดยเมื่อใช้ข้อมูลตึกหรือคณะในการค้นหาจะทำการแนะนำป้ายบริเวณนั้นที่ใกล้ที่สุดให้ โดยอาจจะมา Landmark ได้หลายตึกต่อ 1 ป้าย

**line\_ :**

ใช้เก็บว่าป้ายนั้นๆมีสายเดินรถสายไหนผ่านป้ายนี้บ้าง โดยแต่ละสายก็จะมีวิธีการดินรถแตกต่างกันและเก็บข้อมูลเฉพาะของแต่ละสาย โดยจะมีข้อมูลดังนี้

**line :**

ระบุสายของรถแทรมว่ารถแทรมคันนั้นๆ วิ่งสายไหน โดยจะแบ่งคือ 1: สาย MLC 2: สายสีน้ำเงิน 3: สายสีแดง 4: สายสีเขียว

**previous\_station\_id :**

เก็บข้อมูลว่าป้ายก่อนหน้านี้ของรถแทรมสายนั้นผ่านป้ายอะไรมา ไว้ทำการเชื่อมป้ายแต่ละ ป้ายเป็นเส้นทาง

**next\_station\_id :**

เก็บข้อมูลว่าป้ายต่อไปของรถแทรมสายนั้นคือป้ายไว้ทำการเชื่อมป้ายแต่ละป้ายเป็นเส้นทาง และใช้สำหรับระบบแนะนำเส้นทาง

**across\_station\_id :**

เก็บข้อมูลว่าป้ายที่อยู่ถนนฝั่งตรงข้ามคือป้ายอะไร ไว้สำหรับระบบแนะนำเส้นทางเมี่อการ เดินทางต้องเปลี่ยนฝั่งถนนเพื่อใช้ในการเดินทาง

**time\_table :**

เก็บตารางเวลาที่รถแทรมแต่ละสายจะถึงป้ายแต่ละป้าย โดยจะแบ่งวันเป็น จันทร์-ศุกร์ และ เสาร์-อาทิตย์

**Database : feedback**

ไว้เก็บข้อมูลข้อคิดเห็นการใช้งานของโปรแกรมจาก user โดยจะมีโครงสร้างดังนี้

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

อธิบายตัวแปรใน Database

**station\_id :**

รหัสของแต่ละ feedback ที่ส่งมา

**type :**

ไว้เก็บว่าปัญหาที่ได้รับมานั้นอยู่ในประเภทอะไร อาทิ ความเร็วการใช้งาน ปัญหาทรัพยากร หรือความสะดวกสบายในการใช้

**note :**

ไว้เก็บข้อคิดเห็นเพิ่มเติมของแต่ละ feedback

**Database : one\_day**

ไว้เก็บข้อมูลรายละเอียดประวัติการเดินรถในรอบ 1 วันของแต่ละวัน โดยจะมีโครงสร้างดังนี้

รูปภาพประกอบด้วย ข้อความ, หน้าจอ, จอภาพ, ตั้ง

คำอธิบายที่สร้างโดยอัตโนมัติ

อธิบายตัวแปรใน Database

**date\_id :**

รหัสของแต่ละข้อมูลของ one\_day

**date\_ :**

เก็บข้อมูลวันว่าข้อมูลที่บันทึกนั้นว่าบันทึกของวันไหน

**bus\_history :**

เก็บประวัติการเดินรถในรอบ 1 วันโดยจะมีรายละเอียดย่อย ดังนี้ **service\_provider\_id(เฉพาะ Driver) :**

รหัสประจำตัวของ Service provider หรือ Driver ที่ใช้ในการระบุคนขับรถแทรมในรอบนั้น

**tram\_id :**

รหัสประจำตัวรถแทรมที่ขับในรอบนั้น

**line :**

ระบุว่ารถแทรมคันนี้ได้ขับรถสายไหน

**time\_start&time\_end :**

ระบุเวลาเริ่ม-จบของการเดินรถแทรมในรอบนั้น

**Database : log\_service**

ไว้เก็บประวัติการเข้าถึงข้อมูลของฝั่ง Service provider โดยจะมีวิธีกี่เก็บข้อมูลดังนี้

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

อธิบายตัวแปรใน Database

**log\_service\_id :**

รหัสประจำตัวของ log ที่บ่งบอกว่าคือ log อะไร

**service\_provider\_id :**

รหัสประจำตัวของ Service Provider ว่าเก็บ log มาจากใคร

**job\_position :**

ระบุว่าService Provider คนนั้นอยู่ตำแหน่งอะไร Driver หรือ Admin

**Login\_time&logout\_time :**

ระบุเวลาที่อยู่ในระบบของ Service Provider คนนั้น

**service\_action :**

เก็บรายละเอียดการเข้าถึงข้อมูลของ Service Provider คนนั้น โดยมารายละเอียดย่อย ดังนี้

**type :**

ระบุประเภทของการเข้าถึงข้อมูล ว่าคือ การอ่าน เพิ่ม แก้ หรือลบข้อมูล

**time :**

ระบุเวลาที่เข้าถึงข้อมูลนั้น

**access\_position :**

ระบุตำแหน่งที่เข้าถึงข้อมูลว่าเข้าถึงตำแหน่งไหน โดยมีรายละเอียดคือ database\_name : ระบุชื่อ Database ที่เข้าถึง access\_id : ระบุ ID ใน Databaseที่เข้าถึง position\_command : เก็บตำแหน่งที่เข้าถึงโดยเก็บเป็น command ของ Nosql ที่เข้าถึง เนื่องจากไม่สามารถระบุความลึกของการ query ข้อมูลทุกข้อมูลแบบเจาะจงได้

**edit :**

ถ้าเป็นการแก้ไขข้อมูลจะทำการเก็บข้อมูลเก่าไว้ใน previous\_info และเก็บข้อมูลใหม่ไว้ใน edit\_info

**delete\_info :**

ถ้าเป็นการลบข้อมูลก็จะเก็บข้อมูลที่ลบในตัวแปรนี้

**Database : log\_user**

ไว้เก็บประวัติการใช้งานแอปของฝั่ง Service providerโดย 1 log จะเก็บเพียง 1 การใช้งานเท่านั้น โดยจะมีวิธีกี่เก็บข้อมูลดังนี้

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

อธิบายตัวแปรใน Database

**Log\_user\_id :**

รหัสประจำตัวของ log ที่บ่งบอกว่าคือ log อะไร

**user\_id :**

รหัสประจำตัวของ user ว่าเก็บ log มาจากใคร โดยเนื่องจากแอปไม่ต้อง login เพื่อเข้าใช้งานจึงเก็บเป็น ID ของโทรศัพท์แทน

**log\_time :**

ระบุเวลาในการใช้งานฟังก์ชันของแอปนั้นๆ โดยจะเก็บวันและเวลาใน action\_date กับ action\_time

**log\_position :**

เก็บข้อมูลว่าได้ใช้งานฟังก์ชันของแอปที่บริเวณไหน โดยเก็บเป็น latitude กับ longitude และ nearest\_station\_id ที่บอกป้ายที่ใกล้ที่สุดที่ได้ใช้งานฟังก์ชันของแอป

**user\_action :**

ระบุรายละเอียดว่าได้ใช้งานฟังก์ชั่นอะไร โดย 1 user\_action จะเก็บเพียงการทำงานเดียวเท่านั้นโดยมีประเภทการทำงานดังนี้

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

**tram\_click :**

บอกว่าได้กดดูรถแทรมคันไหน โดยบอกเป็น tram\_id และสายรถแทรมที่วิ่ง

**station\_click :**

บอกว่าได้กดดูป้ายรถแทรมป้ายไหน โดยบอกเป็น station\_id และเช็คว่าได้กดดูส่วน timetable ไหมและดู timetable ของสายอะไร

**line\_click :**

บอกว่าได้เลือกสายที่ดูหรือไม่และดูที่สายไหน

**click\_on\_guide :**

เช็คว่าได้มาการดูวิธีการใช้งานแอปไหม

**feedback\_id :**

เช็คว่ามีการส่ง feedback ไหมโดยเก็บเป็น feedback\_id ที่ส่ง

**search\_way :**

เก็บข้อมูลการใช้งานฟังก์ชันการแนะนำเส้นทางโดยเก็บข้อมูลที่ค้นหา(พิมพ์ค้นหาว่าอะไร) ของจุดเริ่มและจุดสิ้นสุดใน search\_begin และ search\_end และป้ายรถแทรมที่เป็นจุดมาร์คเริ่มต้น และสิ้นสุดที่ station\_start\_id และ station\_end\_id ตามลำดัง