**Logo

Description automatically generated**

**โครงงาน**

**เรื่อง Tramer - Tram Tracking Application**

**โดย**

กลุ่ม Epsilon

นายณัฐวีฬ์ เกิงฝาก 6213125

นายพสวัต แตงอ่อน 6213129

นายกันตพัฒน์ เทียนธนวินท์ 6213194

นายชุติวัต วัตรเอก 6213195

**เสนอ**

อาจารย์ ฆนัท พูลสวัสดิ์

โครงงานนี้เป็นส่วนหนึ่งของวิชา

EGCO321 Database System และ EGCO 343 Software Design

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยมหิดล

ภาคเรียนที่ 1 ปีการศึกษา 2564

# คำนำ

รายงานฉบับนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของวิชา Software Designs (EGCO343) เพื่ออธิบายเกี่ยวกับ Software Architecture, การจัดเก็บข้อมูลใน Database, Business Process, Services ต่าง ๆ ที่ใช้ ตลอดไปจนถึง UX/UI Design ของ Tramer Platform ทั้งนี้ก็เพื่อให้ เกิดความเข้าใจในกระบวนการ ขั้นตอนต่าง ๆ ในการออกแบบ Software

ผู้จัดทำหวังว่า รายงานเล่มนี้จะเป็นประโยชน์กับผู้อ่าน หรือนักเรียน นักศึกษา ที่กำลังหา ข้อมูลตัวอย่างเกี่ยวกับการออกแบบ Software

**คณะผู้จัดทำ**

| สารบัญ |
| --- |
| [คำนำ ก](#_Toc90667530)  [สารบัญ ข](#_Toc90667531)  [บทนำ ง](#_Toc90667532)  [ที่มาและความสำคัญ 1](#_Toc90667533)  [วัตถุประสงค์ 1](#_Toc90667534)  [User Story 2](#_Toc90667535)  [เครื่องมือที่ใช้งาน 3](#_Toc90667536)  [Google Maps APIs 3](#_Toc90667537)  [Firebase 3](#_Toc90667538)  [Software Analysis 4](#_Toc90667539)  [Main layer 4](#_Toc90667540)  [Layer 0 5](#_Toc90667541)  [Layer 1 6](#_Toc90667542)  [Layer 5 7](#_Toc90667543)  [Layer 5.2 8](#_Toc90667544)  [Layer 5.3 9](#_Toc90667545)  [Layer 6 10](#_Toc90667546)  [Software Architecture 11](#_Toc90667547)  [Front-End (UX/UI Design) 12](#_Toc90667548)  [ส่วนของ User 12](#_Toc90667549)  [1. หน้าหลัก[User] 13](#_Toc90667550)  [2. สายการเดินรถ(Fliter) 14](#_Toc90667551)  [3. ป้ายถัดไป(Next Stop) 15](#_Toc90667552)  [4. คันถัดไป(Next Tram) 16](#_Toc90667553)  [5. ค้นหาเส้นทาง(Search) 17](#_Toc90667554)  [6. แจ้งปัญหา(Feedback) 18](#_Toc90667555)  [7. เข้าสู่ระบบ(Login) 19](#_Toc90667556)  [ส่วนของ Driver 20](#_Toc90667557)  [1. หน้าหลัก[Driver] 21](#_Toc90667558)  [2. QR Code Scanner 22](#_Toc90667559)  [3. ขับรถ(Driving) 23](#_Toc90667560)  [ส่วนของ Admin 24](#_Toc90667561)  [1. หน้า Sign in 24](#_Toc90667562)  [2. หน้า Home Page 25](#_Toc90667563)  [3. แถบ Menu 26](#_Toc90667564)  [4. หน้า Personal information 26](#_Toc90667565)  [5. หน้า Edit Tram information 27](#_Toc90667566)  [6. หน้า Edit bus stop information 29](#_Toc90667567)  [7. หน้า Database 31](#_Toc90667568)  [Back-End (Services) 34](#_Toc90667569)  [ส่วนของ User 34](#_Toc90667570)  [ส่วนของ Driver 40](#_Toc90667571)  [ส่วนของ Admin 42](#_Toc90667572)  [Database 44](#_Toc90667573)  [Database : service\_provider 44](#_Toc90667574)  [Database : tram 45](#_Toc90667575)  [Database : feedback 47](#_Toc90667576)  [Database : one\_day 48](#_Toc90667577)  [Database : log\_service 49](#_Toc90667578)  [Database : log\_user 50](#_Toc90667579)  [ข้อมูลที่เกี่ยวข้อง 53](#_Toc90667580)  [บรรณานุกรม 54](#_Toc90667581) |

# บทนำ

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

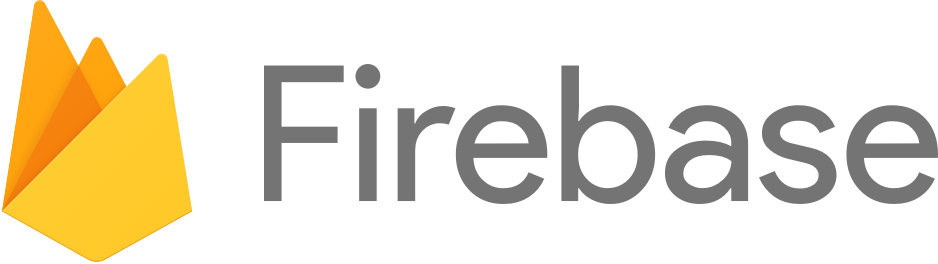
* ผู้โดยสารสามารถทราบตำแหน่งของรถแทรม เพื่อช่วยในการตัดสินใจขึ้นรถ
* ผู้โดยสารสามารถทราบตารางเวลาของรถแทรม เพื่อคำนวณเวลาที่ใช้ในการรอรถแทรมเพื่อเดินทาง
* ผู้โดยสารสามารถทราบสายรถที่ต้องขึ้นจากต้นทางไปปลายทาง
* ผู้โดยสารสามารถทราบแผนที่การเดินทางของรถทุกสาย เพื่อวางแผนการเดินทาง
* ผู้โดยสารสามารถทราบตำแหน่งของป้ายรถแทรมที่ใกล้ที่สุด 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



## Google Maps APIs

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



## Firebase

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

สาเหตุที่ใช้

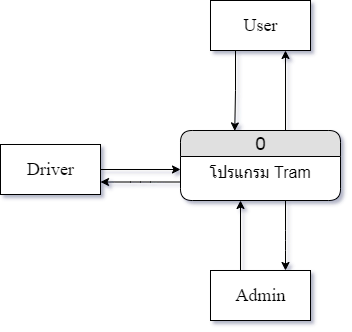
* Hosting ฟรี
* ใช้งานง่าย
* มี documentation ชัดเจน
* Realtime Database
* มี google analytics และ Crashlytics

# Software Analysis

โดยจะทำการอธิบายส่วน Software analysisในรูปของ Dataflow โดยจะมีรูปแบบ ดังนี้

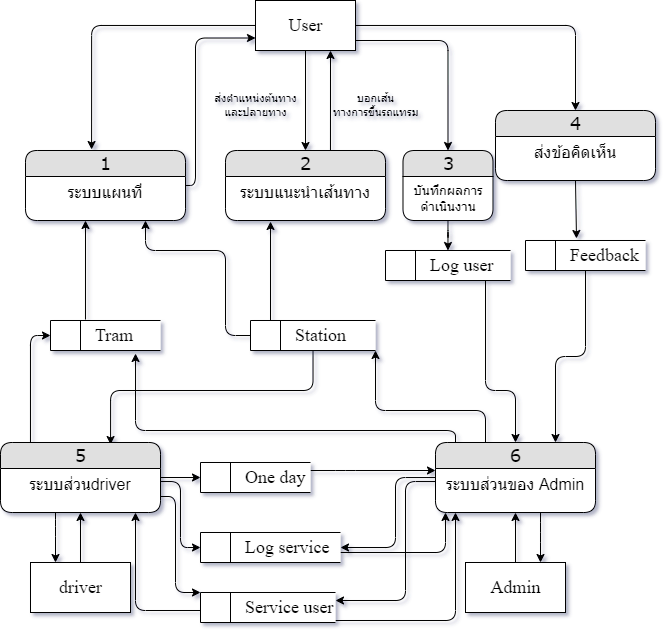
## Main layer

โดยจะแสดงถึงตัวละครทั้งหมดที่ใช้โปรแกรมและการเคลื่อนย้ายของข้อมูล โดยจะมี 3 ตัวละครคือ User, Driver และ Admin



## Layer 0

จะแสดงถึงภาพรวมทั้งหมดของการเคลื่อนไหวของข้อมูลทั้งระบบ โดยมีรูปแบบดังรูป

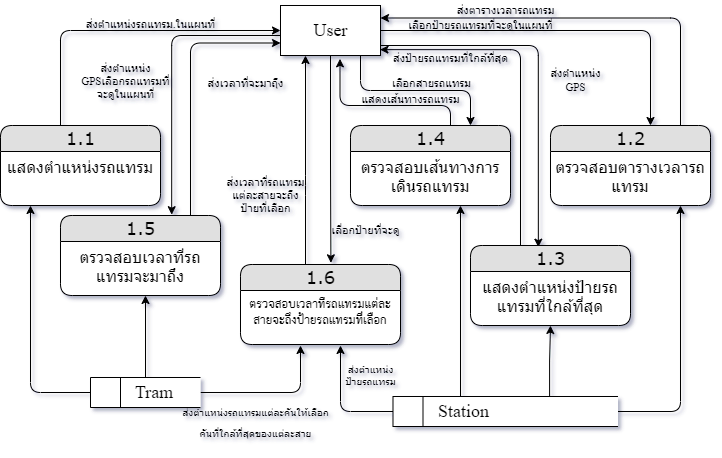


อธิบายองค์ประกอบย่อย

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

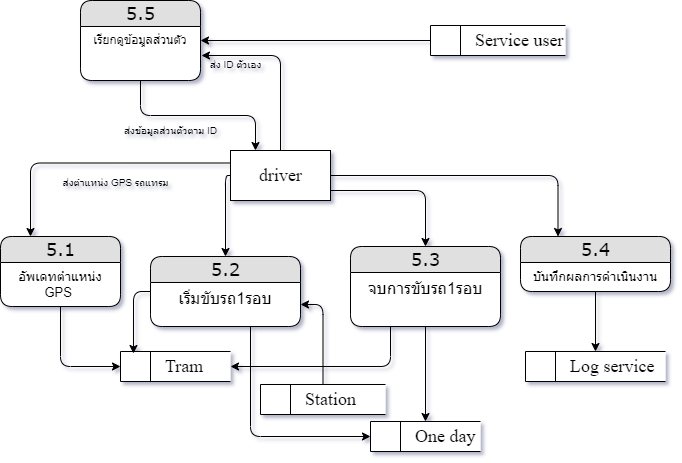


อธิบายองค์ประกอบย่อย

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

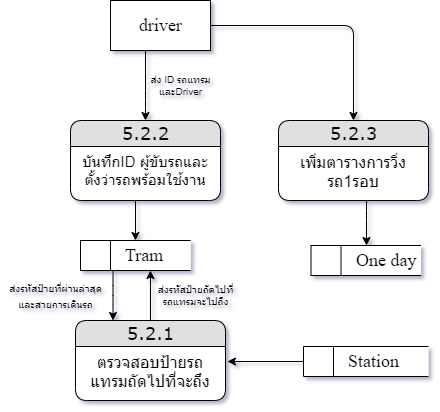


อธิบายองค์ประกอบย่อย

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

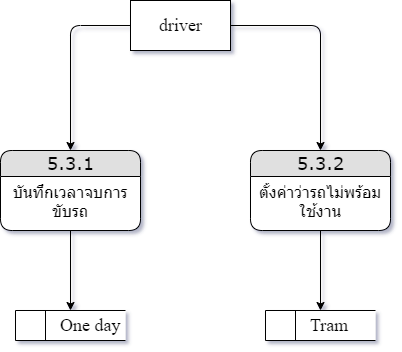


อธิบายองค์ประกอบย่อย

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

ในส่วนของ Software Architecture จะมีการใช้ architecture แบบ monolithic architecture กล่าวคือ component ต่าง ๆ ของแต่ละระบบที่มาจาก environment เดียวกัน จะมีการใช้ database ร่วมกัน ซึ่งภายใน Web Server จะประกอบไปด้วย 4 Sub-Systems ได้แก่

1. ระบบ feedback
2. ระบบแนะนำเส้นทาง
3. ระบบแผนที่
4. ระบบเก็บ log

โดยจะเห็นได้ว่า Admin จะแก้ไขข้อมูลจาก web develop และส่งข้อมูลไปเก็บใน Database ส่วนรถแทรมจะดึงข้อมูลจาก GPS และส่งข้อมูลเข้าไปใน มือถือของ Driver และ user หรือผู้ใช้ปกติจะใช้จาก Application ซึ่งในตัว App จะดึงข้อมูลจากระบบย่อยต่างๆซึ่งระบบย่อยต่างๆก็จะดึงข้อมูลมาจาก Database อีกที และสุดท้ายข้อมูลการกระทำทั้งหมดของผู้ใช้ต่างๆทั้ง Admin, Driver และ User จะถูกระบบเก็บ log จะดเก็บข้อมูลและส่งเข้า Database

Diagram

Description automatically generated

# Front-End (UX/UI Design)

ในการออกแบบ User Interface จะเน้นการออกแบบให้แทบควบคุมและปุ่มต่างๆ อยู่บริเวณด้านล่างของแอพพลิเคชั่น เพราะว่าเพื่อให้สะดวกต่อการใช้ด้วยมือเดียว และไม่ต้องเอื้อมนิ้วขึ้นไปบริเวณด้านบนของหน้าจอ และเลือกใช้โทนสีหลักคือ ดำ/เหลือง เพื่อให้โทนสีทั้งแอพพลิเคชั่นไม่สว่างจนเกินไป

ในโหมดการใช้งานจะแบ่งเป็นส่วนของ User และ Driver โดยการจะใช้โหมด Driver จะต้องเข้าสู่ระบบโดยใช้ driverID และรหัสผ่าน ที่ได้รับจาก Admin ของระบบโดยตรงเท่านั้น ไม่สามารถสมัครสมาชิกได้

## Text Description automatically generated with low confidenceส่วนของ User

หน้าแรกของแอพพลิเคชั่นที่แสดงโลโก้ของแอพพลิเคชั่นส่วนของ User โดยทั่วไปแล้ว ผู้ใช้จะสามารถใช้ฟังก์ชันทั้งหมดได้โดยไม่ต้องล็อคอินเข้าสู่ระบบ

### A picture containing timeline Description automatically generatedGraphical user interface, text, application, chat or text message Description automatically generated1. หน้าหลัก[User]

จะแสดงรถแทรมและป้ายทั้งหมดที่บริเวณนั้น

จะมีฟังก์ชัน 3 อย่างบริเวณแทบด้านล่าง

* สายการเดินรถ(Fliter) : เลือกแสดงเฉพาะสายที่กำหนด
* Nearby : แสดงรถแทรมคันถัดไป(Next Tram)จากป้ายที่ใกล้ที่สุด
* ค้นหาเส้นทาง(Search) : ค้นหาเส้นทาง

จะมีฟังก์ชัน 2 อย่างที่แผนที่

* กดที่รถแทรม : แสดงป้ายถัดไป(Next Stop)ที่รถแทรมจะผ่าน
* กดที่ป้ายแทรม : แสดงรถแทรมคันถัดไป(Next Tram)ที่จะมาถึง

และมี Overlay แถบเมนูเพิ่มเติมเมื่อกดปุ่มด้านขวาบน

* แจ้งปัญหา(Feedback) : แจ้งปัญหาการและคำติชม
* เข้าสู่ระบบ(Login) : เข้าสู่ระบบเพื่อใช้งาน Tramer[Driver]

**Service/API ที่ใช้**

* Service ตรวจสอบตำแหน่งรถแทรม
* Service แสดงป้ายรถแทรมที่ใกล้ที่สุดจากตำแหน่งผู้ใช้

### Diagram Description automatically generatedDiagram Description automatically generated2. สายการเดินรถ(Fliter)

แสดงเส้นทางการเดินรถทั้งหมดของสายสีต่างๆ

จะมีฟังก์ชัน 2 อย่างบริเวณแทบด้านล่าง

* ลูกศรซ้าย : เลื่อนไปยังสายก่อนหน้า
* ลูกศรขวา : เลื่อนไปยังสายถัดไป

และมีฟังก์ชัน 1 อย่างที่บริเวณแผนที่

* กดที่รถแทรม : แสดงป้ายถัดไป(Next Stop)ที่รถแทรมจะผ่าน

**Service/API ที่ใช้**

* Service ตรวจสอบตำแหน่งรถแทรม
* Service ตรวจสอบเส้นทางการเดินรถแทรมพร้อมแสดงรถแทรมในสาย

### Graphical user interface, application Description automatically generatedGraphical user interface, application Description automatically generated3. ป้ายถัดไป(Next Stop)

แสดงป้ายต่างๆที่รถแทรมคันนั้นจะผ่าน

จะมีฟังก์ชัน 2 อย่างบริเวณแทบด้านล่าง

* ลูกศรซ้าย : เลื่อนไปยังคันก่อนหน้า
* ลูกศรขวา : เลื่อนไปยังคันถัดไป

**Service/API ที่ใช้**

* Service ตรวจสอบตำแหน่งรถแทรม
* Service ตรวจสอบเวลาที่รถแทรมจะมีถึงป้ายสถานีที่รอ

### Graphical user interface Description automatically generated with medium confidenceGraphical user interface Description automatically generated with low confidence4. คันถัดไป(Next Tram)

แสดงเวลาที่รถแทรมสายต่างๆที่ผ่านป้ายนั้นๆจะมาถึง

จะมีฟังก์ชัน 2 อย่างบริเวณแทบด้านล่าง

* ลูกศรซ้าย : เลื่อนไปยังสายก่อนหน้า
* ลูกศรขวา : เลื่อนไปยังสายถัดไป

**Service/API ที่ใช้**

* Service ตรวจสอบตำแหน่งรถแทรม
* Service ตรวจสอบเวลาที่รถแทรมจะถึงป้ายรถแทรมที่เลือก
* Service ตรวจสอบตารางเวลารถแทรม

### Timeline Description automatically generatedDiagram Description automatically generatedGraphical user interface, text, application Description automatically generated5. ค้นหาเส้นทาง(Search)

ค้นหาเส้นทางการเดินทางไปที่หมาย โดยเมื่อกดเข้ามาจะกรอกต้นทางเป็นสถานีที่ใกล้ที่สุดโดยอัตโนมัติ

จะมีฟังก์ชัน 2 อย่างบริเวณแทบด้านล่าง

* ลูกศรซ้าย : เลื่อนไปยังสายก่อนหน้า
* ลูกศรขวา : เลื่อนไปยังสายถัดไป

**Service/API ที่ใช้**

* Service ตรวจสอบตำแหน่งรถแทรม
* Service ค้นหาวิธีการเดินทางจากจุดเริ่มต้นไปปลายทาง

### Text Description automatically generatedA picture containing square Description automatically generatedGraphical user interface, text, application Description automatically generated6. แจ้งปัญหา(Feedback)

แจ้งปัญหาหรือให้คำแนะนำติชมการใช้งานแอพพลิเคชั่น โดยจะมีให้ติ๊กเลือกประเภท และมีช่องให้กรอกรายละเอียด เมื่อกรอกเสร็จและกดส่งแล้ว จะมีหน้าต่างแสดงคำขอบคุณและล้างข้อมูลทั้งหมดเพื่อกรอกใหม่

### Graphical user interface, text, application, chat or text message Description automatically generatedGraphical user interface, text, application, chat or text message Description automatically generated7. เข้าสู่ระบบ(Login)

เข้าสู่ระบบเพื่อใช้งานระบบ Tramer[Driver] โดย username และ password จะได้รับจาก Admin ระบบเท่านั้น ไม่สามารถสมัคร username และกำหนด password เองได้

## Text Description automatically generated with low confidenceส่วนของ Driver

หน้าแรกของแอพพลิเคชั่นที่แสดงโลโก้ของแอพพลิเคชั่นส่วนของ Driver โดยฟังก์ชันทั้งหมดของแอพพลิเคชั่นใช้ได้เหมือนกับผู้ใช้ปกติทั้งหมด ยกเว้นแค่ฟังก์ชัน Nearby ที่ถูกแทนที่ด้วยฟังก์ชัน QR Code ที่ใช้สแกน QR Code ที่รถแทรมเพื่อเข้าใจงานฟังก์ชันข้อมูลรถแทรมและฟังก์ชันบันทึกประวัติการขับรถ

### A picture containing timeline Description automatically generatedGraphical user interface, text, application, chat or text message Description automatically generated1. หน้าหลัก[Driver]

จะมีฟังก์ชันการทำงานเหมือนแบบ User ทุกอย่างยกเว้น

ฟังก์ชัน 1 อย่างบริเวณแทบด้านล่างที่เปลี่ยนจาก Nearby เป็น

* QR Code Scanner : เปิดกล้องเพื่อสแกน QR Code เพื่อดูข้อมูลของรถแทรมคันนั้นๆ

และมี Overlay แถบเมนูเพิ่มเติมที่เปลี่ยนไปคือ

* ออกจากระบบ(Logout) : ออกจากระบบเพื่อกลับไปใช้งาน Tramer[User]

**Service/API ที่ใช้**

* Service เข้าทำงานและออกงาน

### Icon, qr code Description automatically generatedGraphical user interface, text, application Description automatically generated2. QR Code Scanner

เมื่อสแกน QR Code ที่รถแล้ว จะแสดงหน้าข้อมูลรถแทรม(Tram info) ขึ้นมาเพื่อให้ตรวจสอบข้อมูลชื่อผู้ขับ และเวลาเริ่มขับรถแทรม เมื่อกดปุ่มเริ่มขับรถ ระบบจะบันทึกเวลาเริ่มขับไปที่ฐานข้อมูล ก่อนที่จะไปที่หน้าขับรถ(Driving) ต่อไป

**Service/API ที่ใช้**

* Service เริ่มการขับรถ 1 รอบ

### Graphical user interface, text, application Description automatically generatedDiagram Description automatically generated3. ขับรถ(Driving)

หน้าขับรถจะแสดงเส้นทางทั้งหมดที่รถต้องขับไป และแสดงป้ายต่อๆไปที่จะต้องจอด เมื่อผู้ขับต้องการจะสิ้นสุดการขับ ก็ให้กดที่ปุ่มสิ้นสุดการขับ ระบบจะให้กดยืนยันก่อนอีกครั้ง เมื่อกดยืนยัน ระบบจะบันทึกเวลาสิ้นสุดการขับเข้าไปที่ฐานข้อมูล และกลับไปที่หน้าหลัก

**Service/API ที่ใช้**

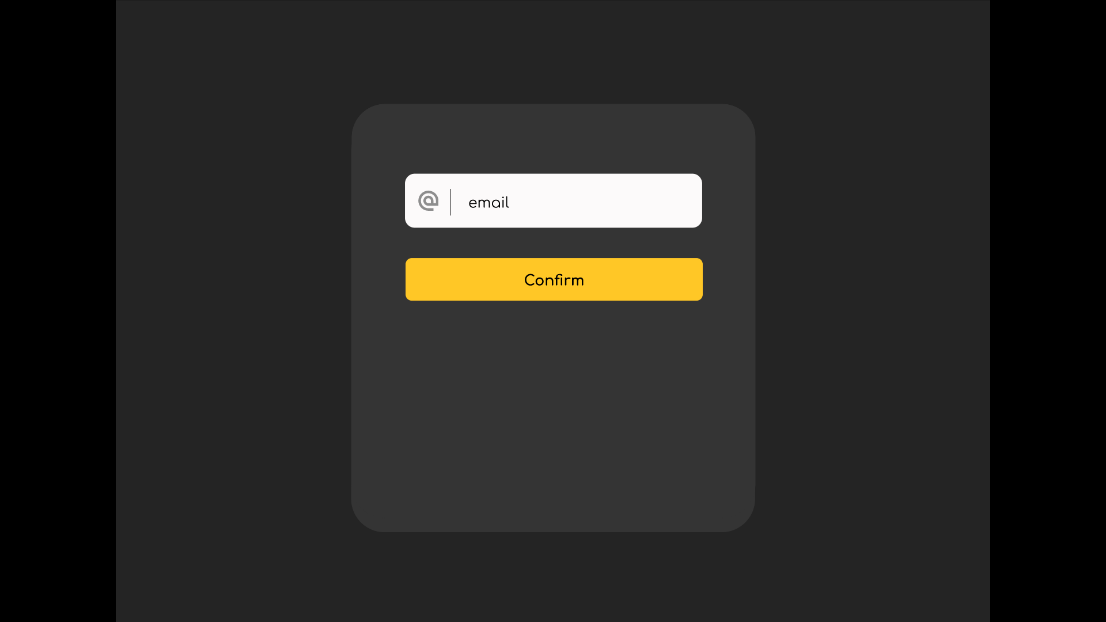
* Service เริ่มการขับรถ 1 รอบ
* Service จบการขับรถ 1 รอบ

## ส่วนของ Admin

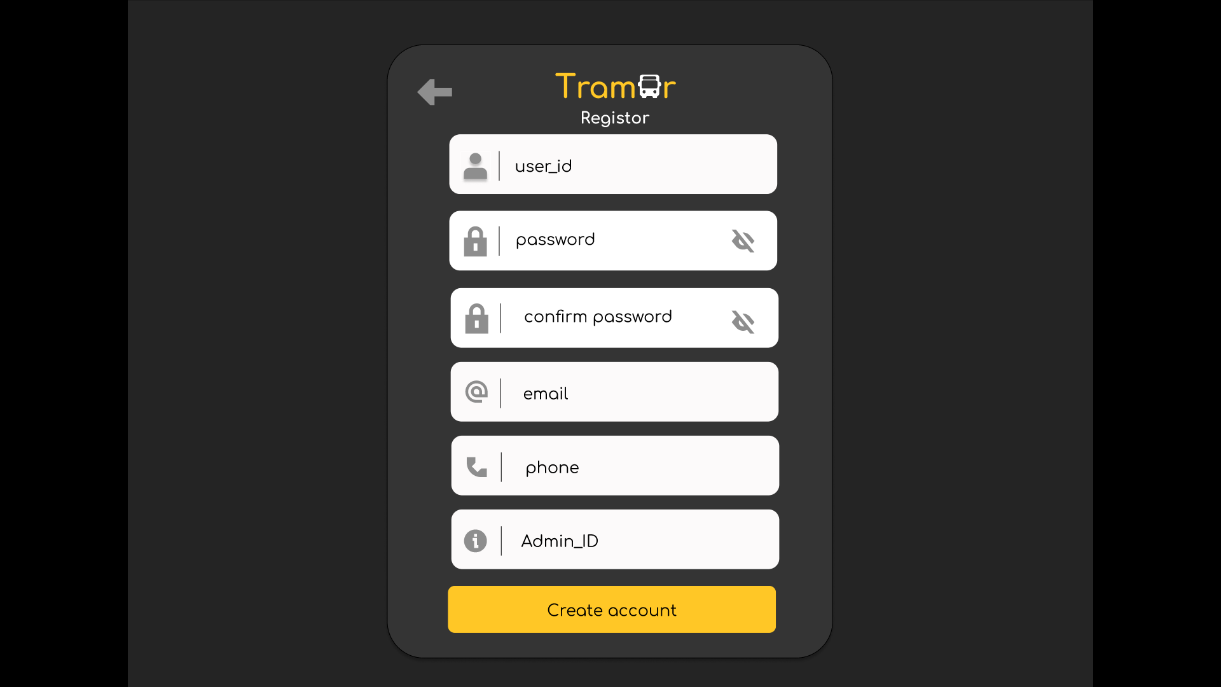
ในส่วน Admin จะต้องเข้าใช้งานผ่านเว็บไซต์โดยใช้ Username และ Password ที่กำหนด เพื่อใช้ระบบหลังบ้านที่คอยควบคุมการทำงานของแอพพลิเคชั่น เช่นการเปิดปิดใช้งานรถ เป็นต้น

### 1. หน้า Sign in

รูปภาพประกอบด้วย ข้อความ, อุปกรณ์อิเล็กทรอนิกส์

คำอธิบายที่สร้างโดยอัตโนมัติ ส่วนแรกที่จะได้พบเมื่อเข้าเว็ปไซท์มาคือหน้า sign in โดยในส่วนนี้จะแบ่งระบบออกเป็น 3 ส่วนใหญ่ๆได้แก่ ส่วน sign in ปกติดังภาพ

ส่วนต่อมาจะเป็นส่วนของการลืม password โดยจะให้ทำการยืนยันจาก e-mail และหลังจากยืนยันเสร็จจะให้เปลี่ยนรหัสผ่านแล้วจะให้กลับไปหน้า sign in อีกครั้ง



ส่วนสุดท้ายจะเป็นหน้าลงทะเบียนบัญชีใหม่โดยผู้ลงทะเบียนจำเป็นจะต้องใส่ Admin\_ID เพื่อเป็นการยืนยันว่าไม่ใช่ผู้ใช้งานทั่วไป

### 2. หน้า Home Page

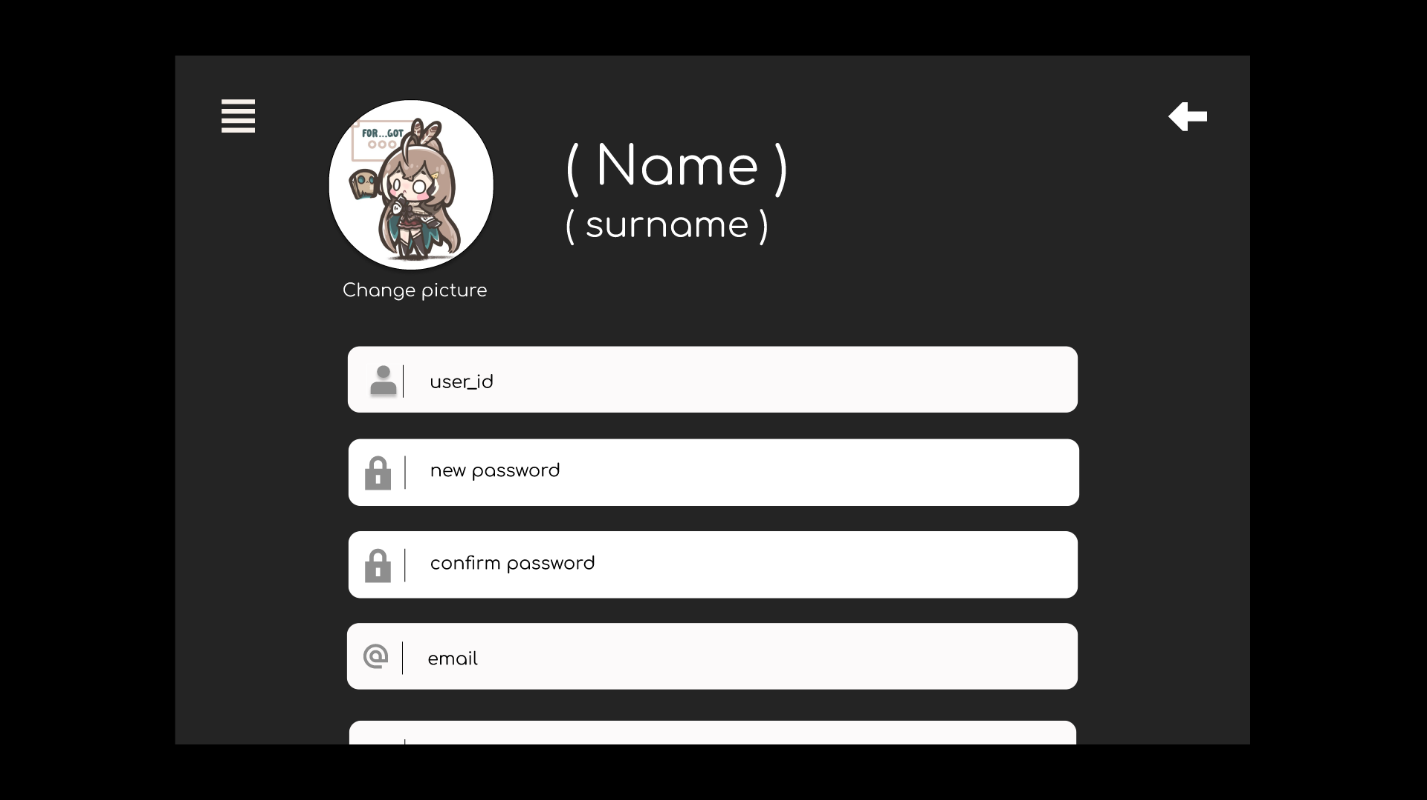
ในส่วนของ Home Page จะแบ่งออกเป็น 3 ส่วนได้แก่ส่วนแก้ไขข้อมูลรถแทรม (edit tram information) ส่วนแก้ไขข้อมูลป้ายรถแทรม (edit bus stop information) และส่วนของการเก็บข้อมูลทั้งหมด(Database)

### 3. แถบ Menu

เมื่อกดแถบเมนูทางด้านบนซ้ายระบบจะขึ้น Overlay menu ขึ้นมา ซึ่งจะแสดงเมนูออกมา 3 ส่วนได้แก่ ปุ่ม Home ( หากเลือกปุ่มนี้จะพากลับมาหน้า Home Page ) ปุ่ม Personal information และ ปุ่ม log out ที่จะพากลับไปที่หน้า Sign in

### รูปภาพประกอบด้วย ข้อความ คำอธิบายที่สร้างโดยอัตโนมัติ4. หน้า Personal information

หน้านี้จะเป็นหน้าแสดงข้อมูลของบัญชีนี้โดยจะแสดงชื่อ, นามสกุล, user\_ID, email, เบอร์โทรศัพท์ และ Admin\_ID โดยจะสามารถแก้ไขข้อมูลหรือทำการลบบัญชีได้

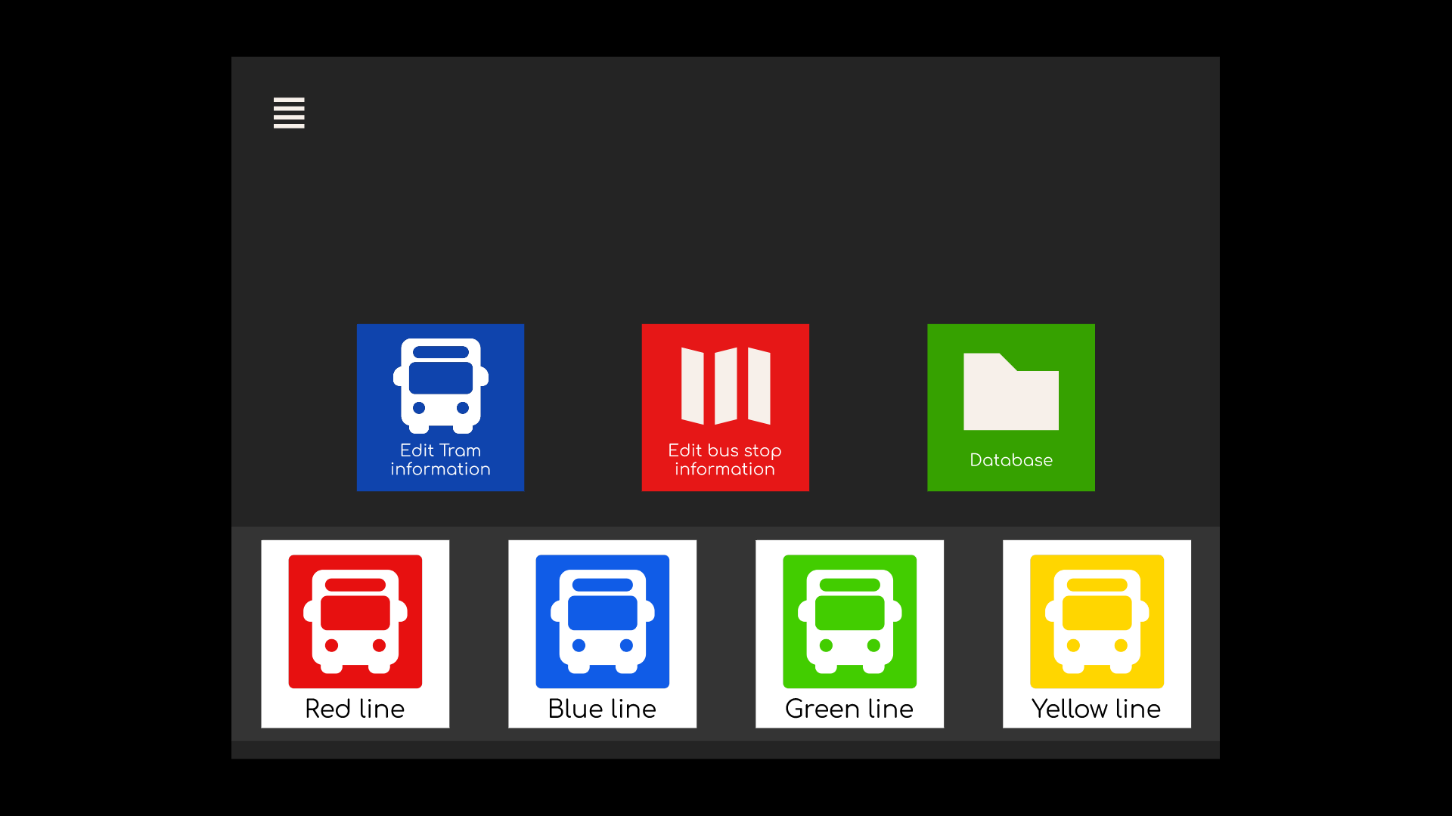


ช่องเปลี่ยน password และเปลี่ยนภาพโปรไฟล์ จะขึ้นมาให้ใส่เฉพะหน้าแก้ไขข้อมูล

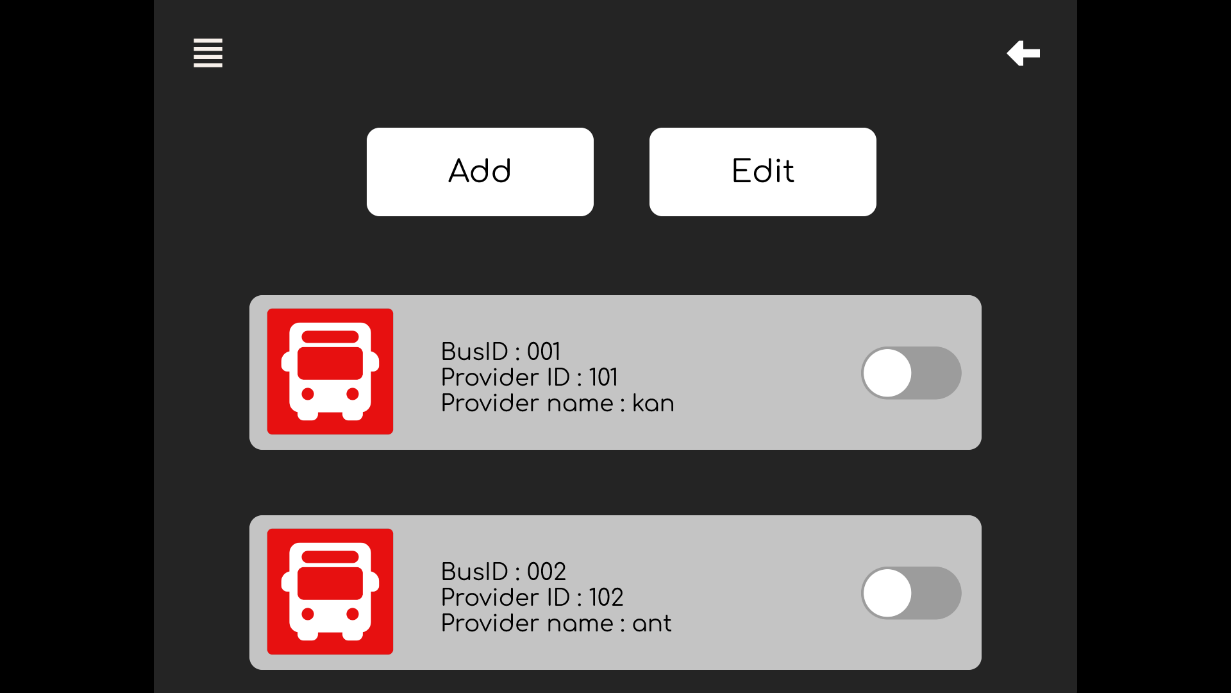
**Service/API ที่ใช้**

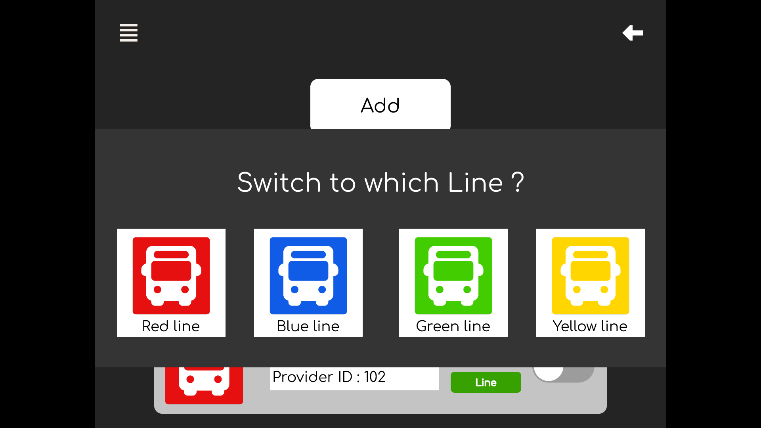
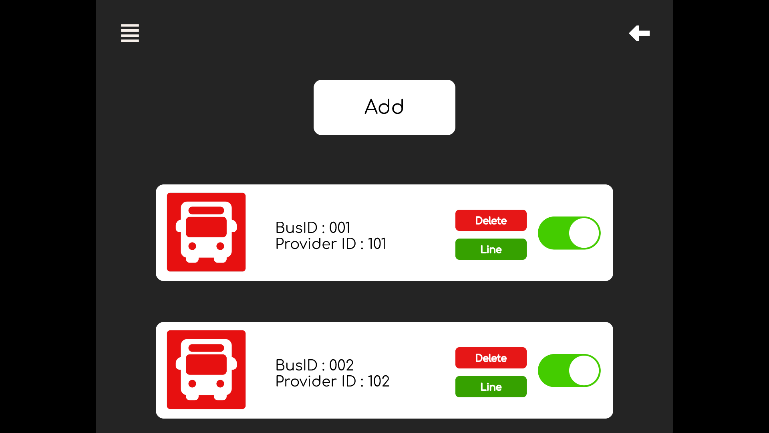
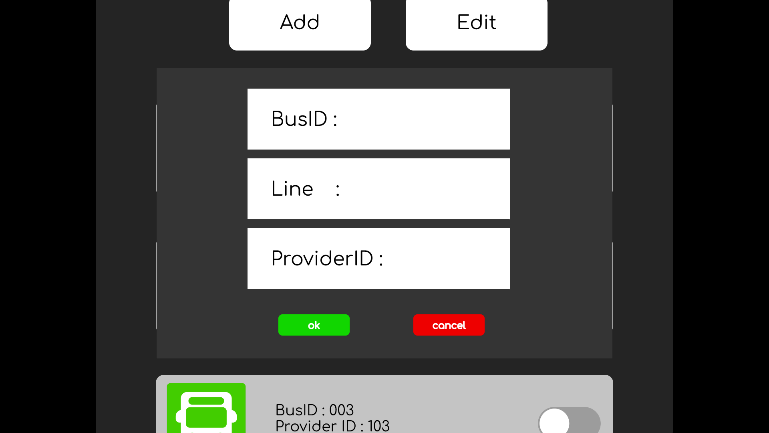
* Service แก้ไขข้อมูลส่วนตัว

### 5. หน้า Edit Tram information



เมื่อเรากดปุ่ม Edit Tram information ระบบจะแสดง overlay ขึ้นมาให้เราเลือกว่าจะดูข้อมูลของ รถแทรมสีอะไร



 เมื่อเลือกสีของรถแทรมได้ระบบจะพามาหน้าดูข้อมูลและเราจะสามารถเลือกเปิดใช้งานรถแทรม เพิ่มรถแทรมหรือแก้ไขข้อมูลได้

ในหน้า edit จะสามารถย้ายสีของแทรมหรือลบข้อมูลได้และในการ Add จะต้องกรอกข้อมูลดังภาพ

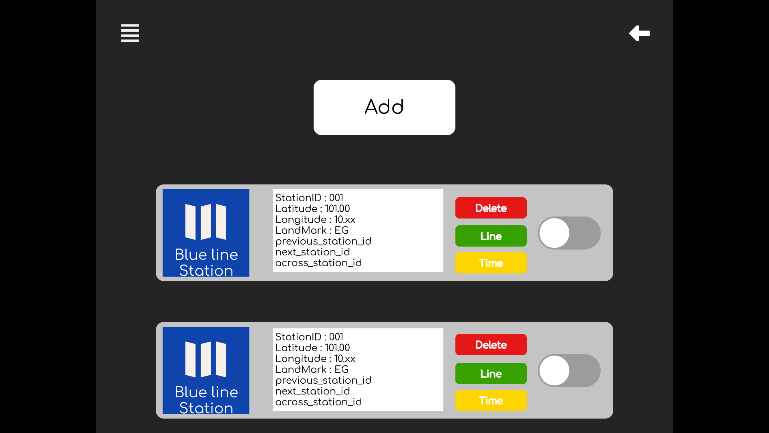
**Service/API ที่ใช้**

* Service แก้ไขข้อมูลของรถแทรม

### 6. หน้า Edit bus stop information

 หน้า Edit bus stop information จะคล้ายๆกับหน้า Edit Tram informationคือจะแสดง overlay ขึ้นมาให้เลือกดูสถานีของแต่ละสาย

โดยจะต่างกับ Edit Tram information ตรงที่จะเพิ่มปุ่ม Time ที่จะแสดงตารางเวลาที่รถแทรมจะมาถึงได้และสามารถแก้ไขข้อมูลตารางเวลาในหน้า edit ได้

รูปภาพประกอบด้วย โต๊ะ

คำอธิบายที่สร้างโดยอัตโนมัติ

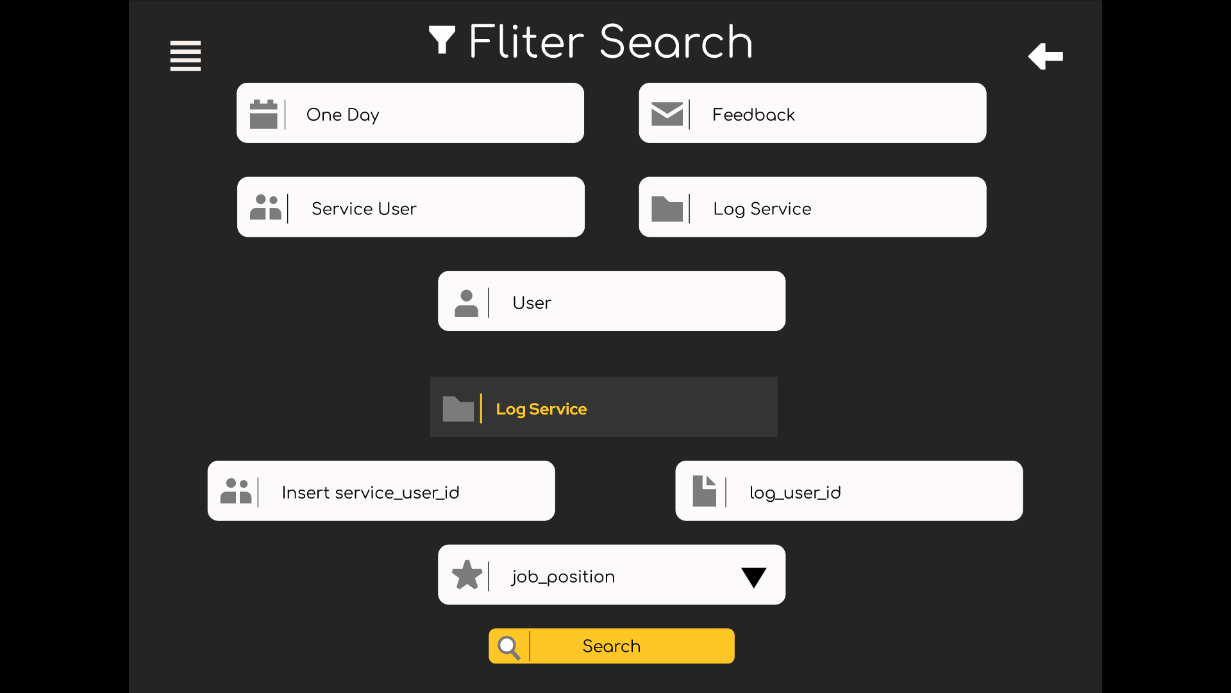
หน้า edit จะสามารถย้ายสีของป้ายเพิ่มหรือลบข้อมูลได้และยังสามารถแก้ไขตารางเวลาได้อีกด้วย

ข้อมูลที่ต้องกรอกเมื่อนทำการเพิ่มป้ายรถแทรมป้ายใหม่

**Service/API ที่ใช้**

* Service แก้ไขข้อมูลของป้ายรถแทรม
* Service เรียกดูข้อมูลการเดินรถใน 1 วัน

### รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จอภาพ, หน้าจอ คำอธิบายที่สร้างโดยอัตโนมัติ7. หน้า Database

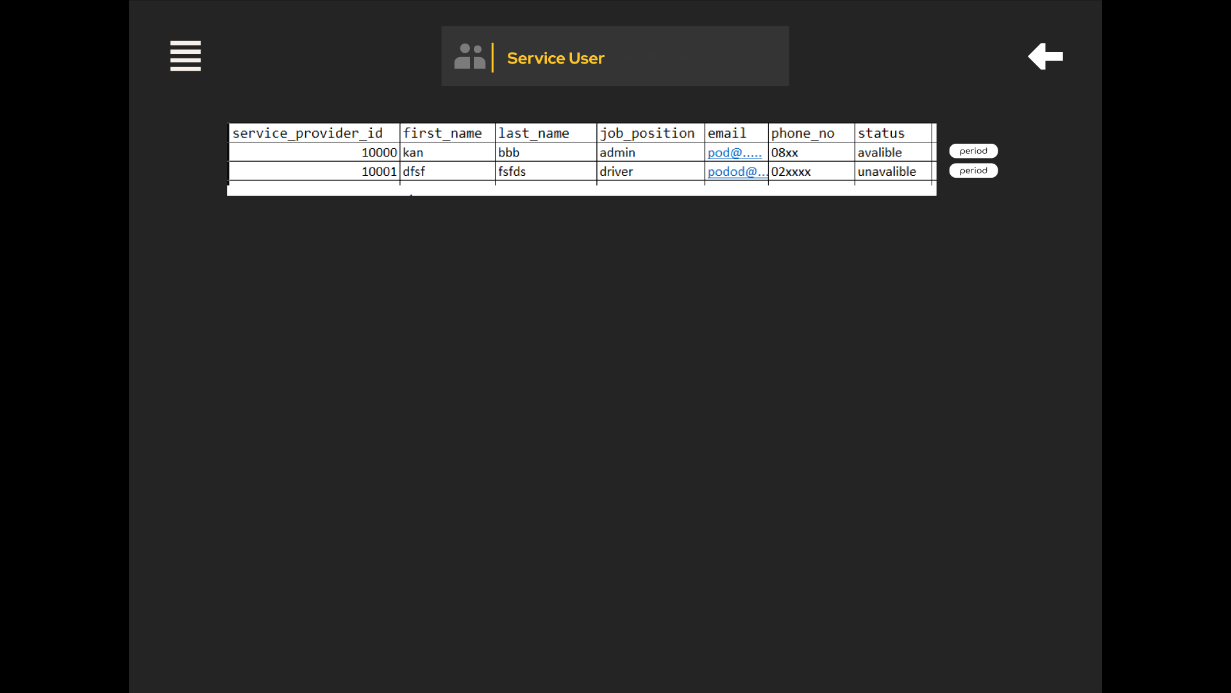
เมื่อเลือกปุ่ม Database ระบบจะพามาหน้า filter ซึ่งผู้ใช้จะต้องเลือกว่าจะดึงข้อมูลเกี่ยวกับเรื่องอะไรแล้วระบบจะขึ้น Overlay มาให้กรอกข้อมูลที่เกี่ยวข้องซึ่งเราไม่จะเป็นต้องกรอกข้อมูลทั้งหมดระบบจะทำการประมวลผลดึงเฉพาะข้อมูลที่เกี่ยวข้องกับข้อมูลที่เรากรอกออกมาให้

ตัวอย่างการเลือก log service ระบบจะเพิ่มข้อมูลที่เกี่ยวข้องมาให้กรอกและดึเฉพะข้อมูลที่กรอกออกมาให้หากไม่กรอกก็จะดึงข้อมูลออกมาทั้งหมด

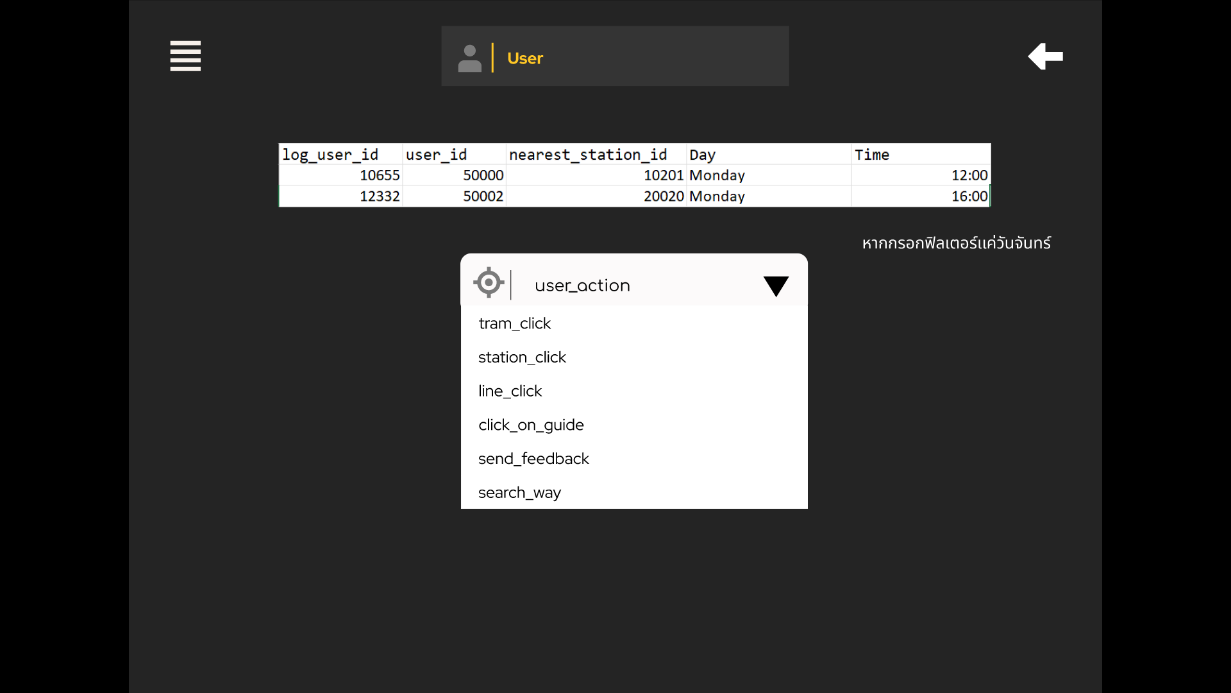
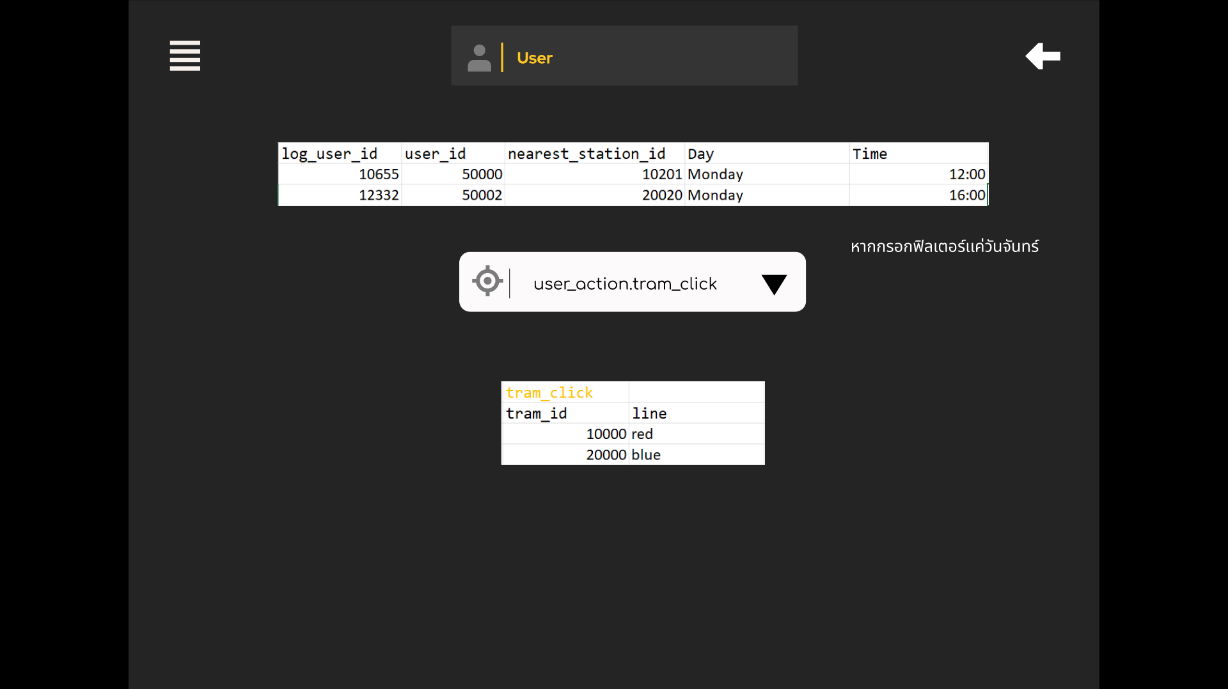
รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติรูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

ภายใน log service สามารถเลือก action type ที่เราต้องการออกมาดูได้จะเห็นว่าภายในการ edit จะเก็บข้อมูลก่อนแหละหลังเอาไว้

ในส่วนของ Service user เราสามารเลือกกดปุ่ม period เพื่อดูตารางเวลาการเข้าออกงานของAdmin\_ID นั้นได้

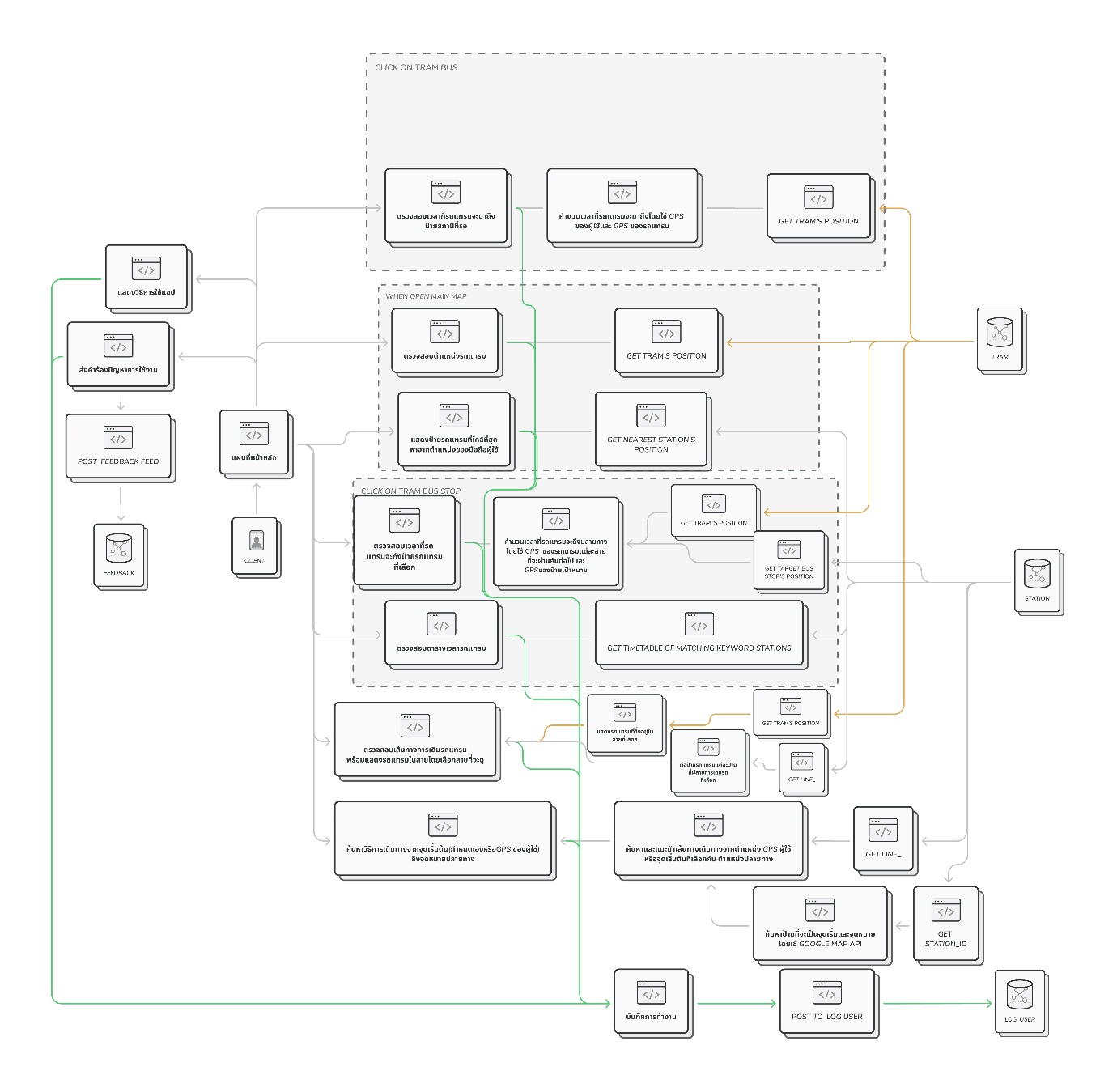
ตัวอย่างการดึง Database ของ User

**Service/API ที่ใช้**

* Service เรียกดูข้อมูล Log user
* Service เรียกดูข้อมูลการเดินรถใน 1 วัน
* Service เรียกดูแบบตอบรับ
* Service เรียกดูข้อมูล Log service
* Service แก้ไขข้อมูลส่วนตัว(เฉพาะส่วน get info)

# Back-End (Services)

## ส่วนของ User

จะแสดงฟังก์ชันการทำงานของ User โดยจะมีการทำงานโดยรวมดังรูป

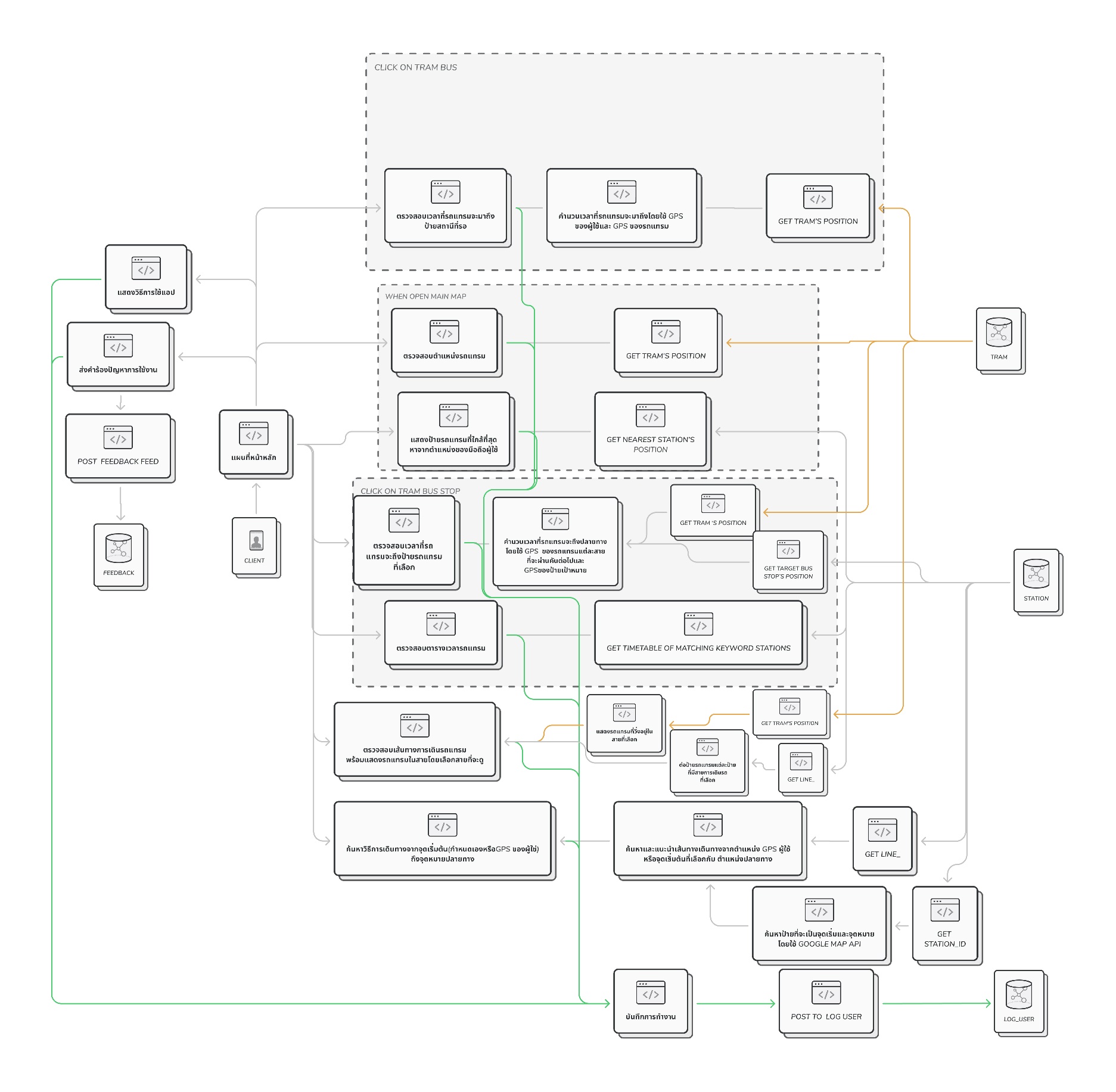
การเข้าถึงข้อมูล

สีเทา Station

สีเขียว Log\_user

สีส้ม Tram

โดยเมื่อเข้าใช้งานฝั่ง user ก็จะเข้าสู่หน้าแผนที่หลักที่แสดงค่าต่างๆในแผนที่ โดยจะมีคำสั่งย่อยตามประเภทการกระทำของ User โดยจะแบ่งการทำงานดังนี้

**1. เมื่อเข้าแผนที่หน้าหลัก**

การเข้าถึงข้อมูล

สีเทา Station

สีเขียว Log\_user

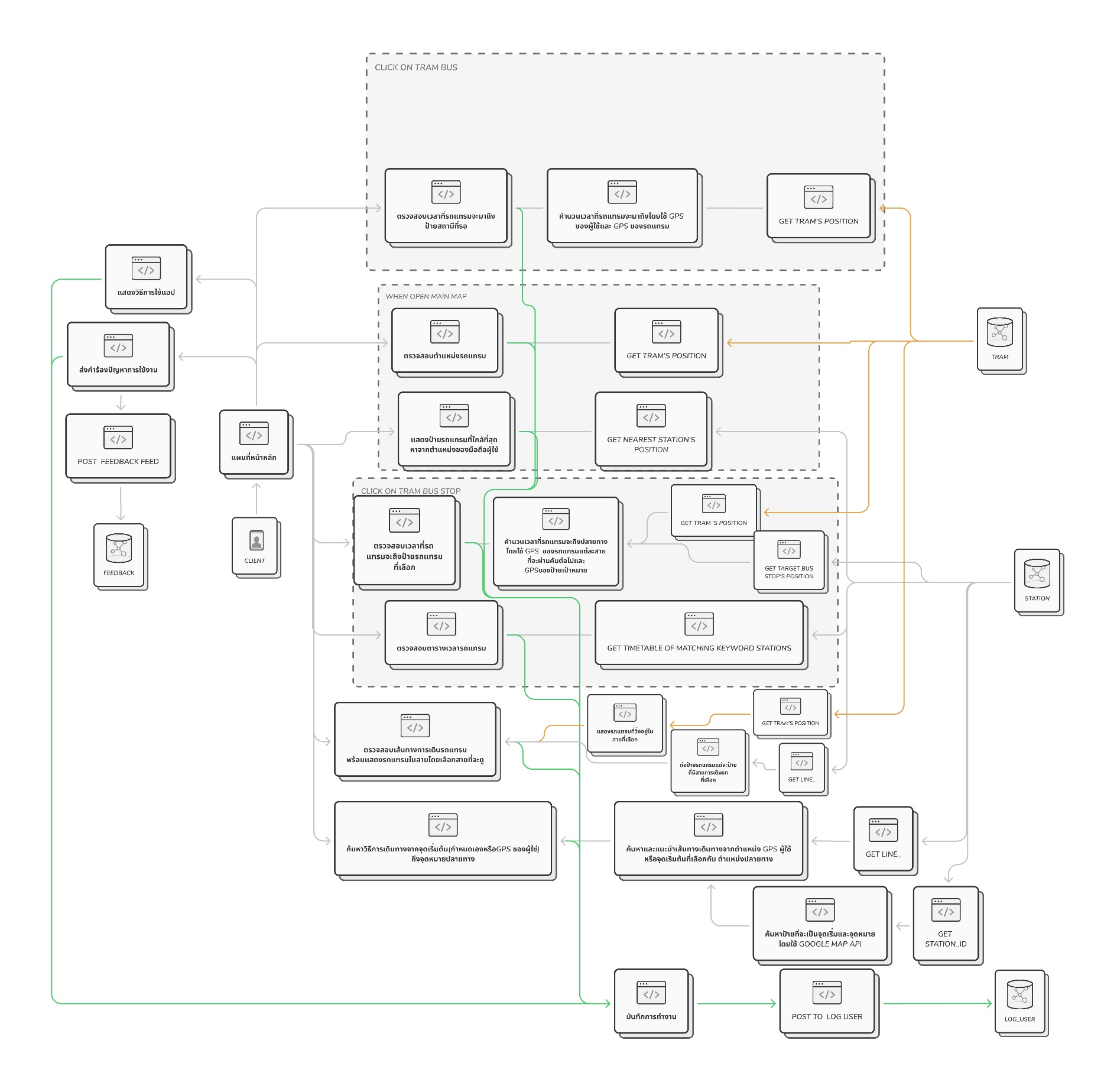
สีส้ม Tram

ตรวจสอบตำแหน่งรถแทรม

เป็นการแสดงตำแหน่งรถแทรมจากในแผนที่โดยใช้ข้อมูลตำแหน่งlatitude และ longitude จากใน Tram database โดยใช้ Google map api ในการแสดงตำแหน่ง

แสดงป้ายรถแทรมที่ใกล้ที่สุดจากตำแหน่งผู้ใช้

เป็นการแสดงป้ายที่มีตำแหน่งใกล้ที่สุดจากตัวผู้ใช้งานโดยเรียกใช้ตำแหน่งป้ายรถแทรมแต่ละป้ายกับตำแหน่งผู้ใช้แล้วมาเปรียบเทียบหาป้ายที่ใกล้ที่สุดแล้วมาแสดงผลทางหน้าจอ

**2. เมื่อกดที่รถแทรม**

การเข้าถึงข้อมูล

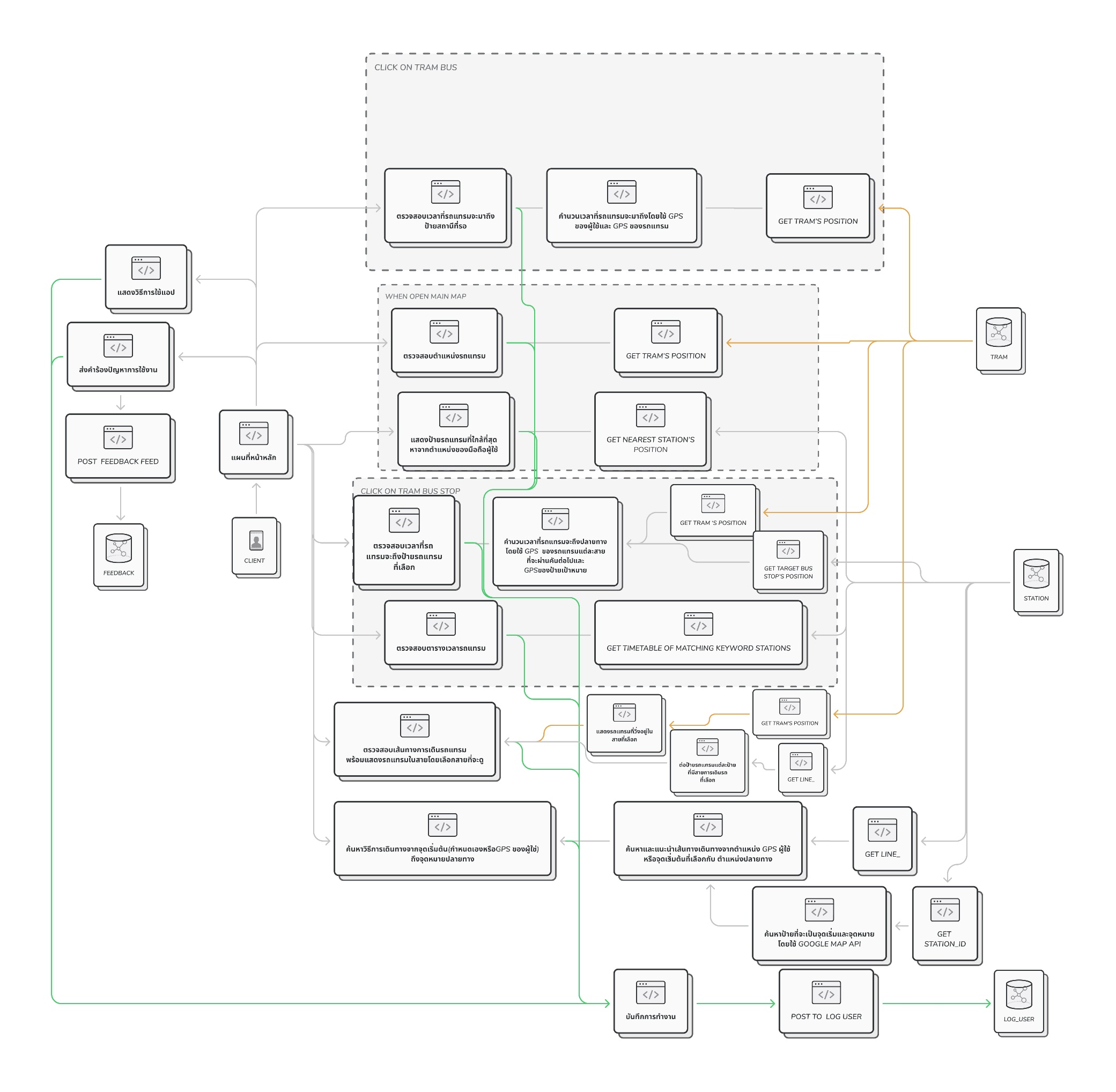
สีเทา Station

สีเขียว Log\_user

สีส้ม Tram

ตรวจสอบเวลาที่รถแทรมจะมีถึงป้ายสถานีที่รอ

เมื่อกดที่รถแทรม จะทำการคำนวนเวลาที่รถแทรมคันนั้นจะมาถึงที่ป้ายรถแทรมที่รอ โดยจะดึงข้อมูลตำแหน่งของรถแทรมมาจาก Tram และใช้ข้อมูลตำแหน่งของ User มาคำนวณโดยใช้ Google Map API แล้วแสดงผลเวลาที่รถแทรมจะมาถึงป้ายที่รอ

**3. เมื่อกดที่ป้ายรถแทรม**

การเข้าถึงข้อมูล

สีเทา Station

สีเขียว Log\_user

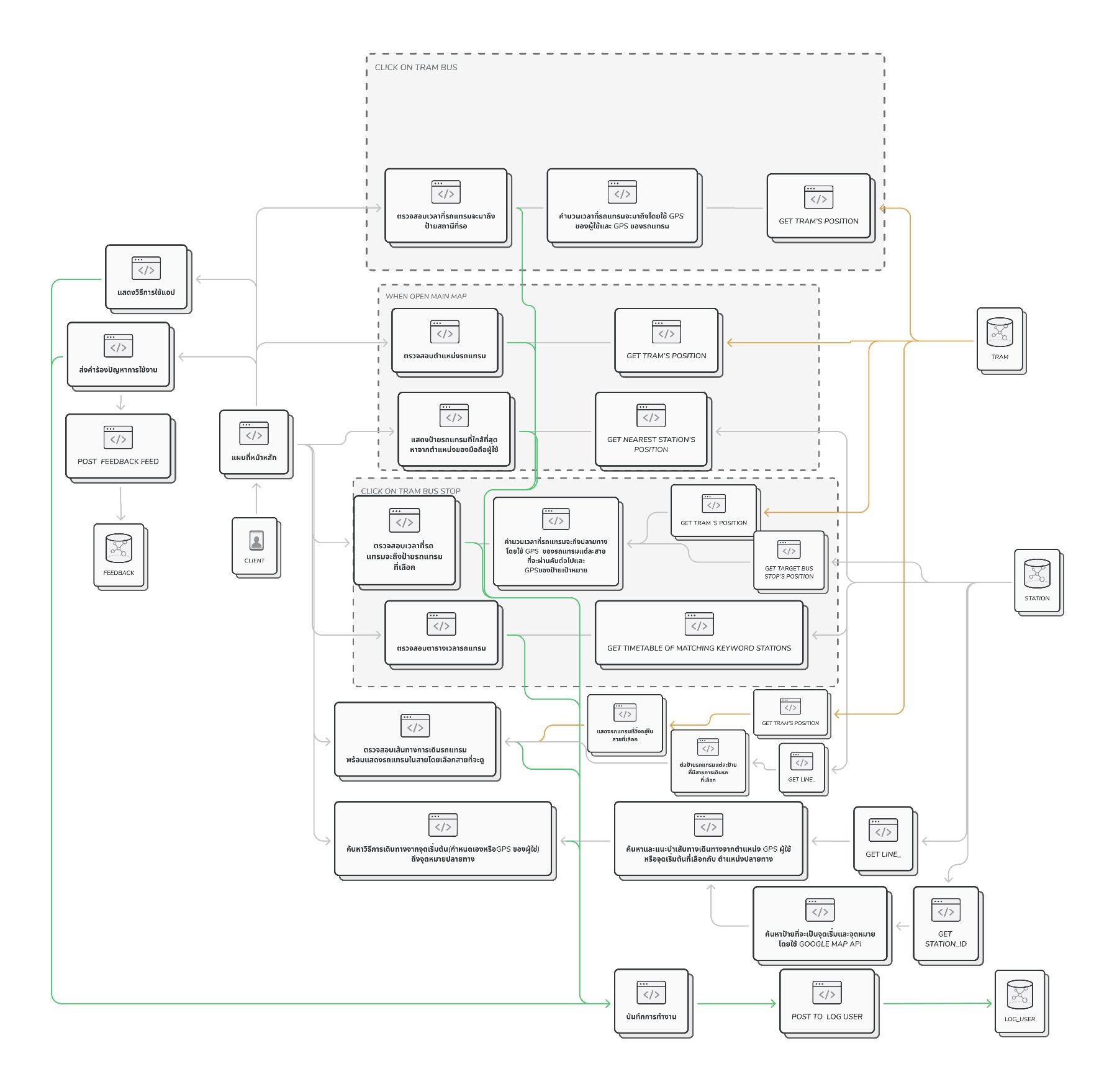
สีส้ม Tram

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

**4. อื่นๆ**

การเข้าถึงข้อมูล

สีเทา Station

สีเขียว Log\_user

สีส้ม Tram

ตรวจสอบเส้นทางการเดินรถแทรมพร้อมแสดงรถแทรมในสาย

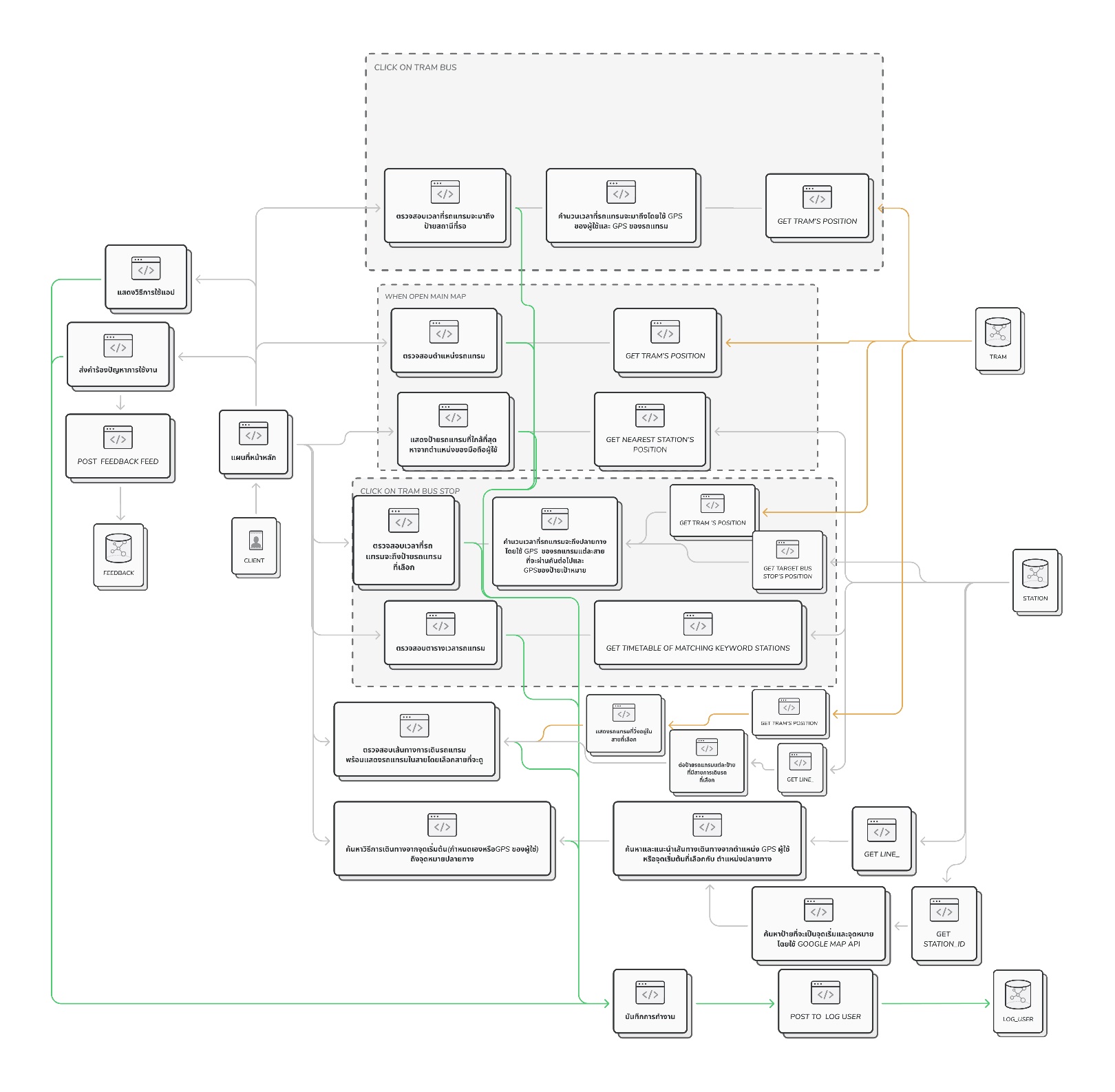
เป็นการให้ User เลือกว่าจะดูสายการเดินรถแทรมของสายไหน โดยพอเลือกสาย

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

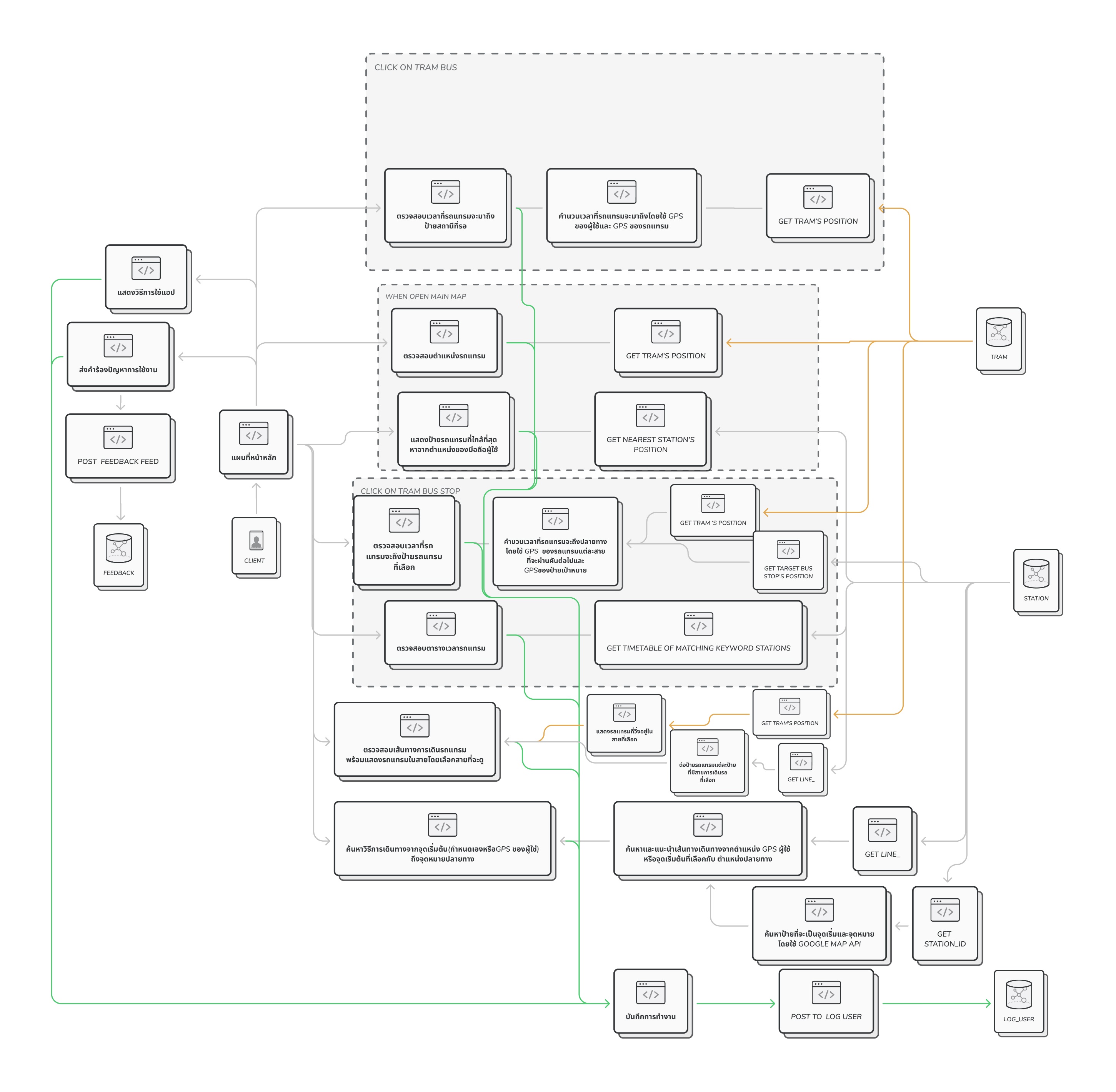
แสดงวิธีการใช้แอป

อธิบายคู่มือในการใช้แอป

ส่งคำร้องปัญหาการใช้แอป

ส่งแบบสอบถามการใช้งานแล้วส่งเขาไปเก็บใน feedback

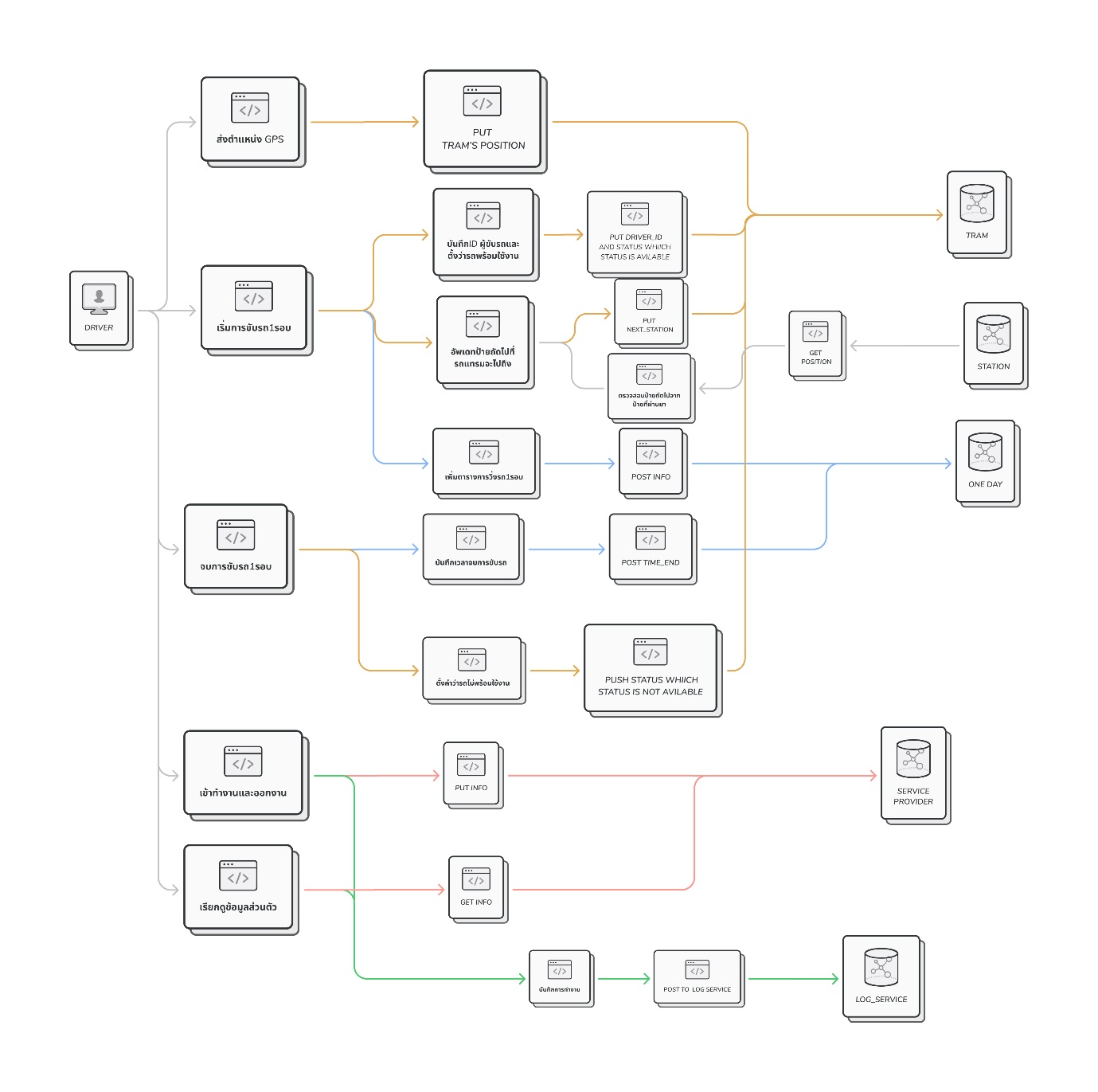
ส่วนบันทึกการใช้งาน



โดยจะเก็บการกระทำทุกอย่างของ User ไว้เป็นประวัติการใช้งาน โดยจะเก็บทุกการกระทำ(สายสีเขียวที่โยงทุกฟังก์ชันการใช้งาน)ไว้ใน Log\_user

## ส่วนของ Driver

จะแสดงฟังก์ชันการทำงานของ Driver โดยจะมีการทำงานโดยรวมดังรูป



**1. ส่งตำแหน่ง GPS**

Driver จะส่งตำแหน่งรถแทรมผ่านทาง GPS แล้วเก็บใน Tram ตาม ID

รถแทรมที่ขับ

การเข้าถึงข้อมูล

สีเทา Station

สีเขียว Log\_service

สีส้ม Tram

สีฟ้า One day

สีแดง Service provider

**2. 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

**3. จบการขับรถ 1 รอบ**

เมื่อ Driver ทำการขับรถ 1 รอบเสร็จแล้วกดปุ่มว่าวิ่งรถเสร็จสิ้นแล้ว จะถือว่าจบการขับรถในรอบนั้นๆแล้ว โดยจะมีกระบวนการย่อยดังนี้

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

**4. เข้าทำงานและออกงาน**

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

**5. เรียกดูข้อมูลส่วนตัว**

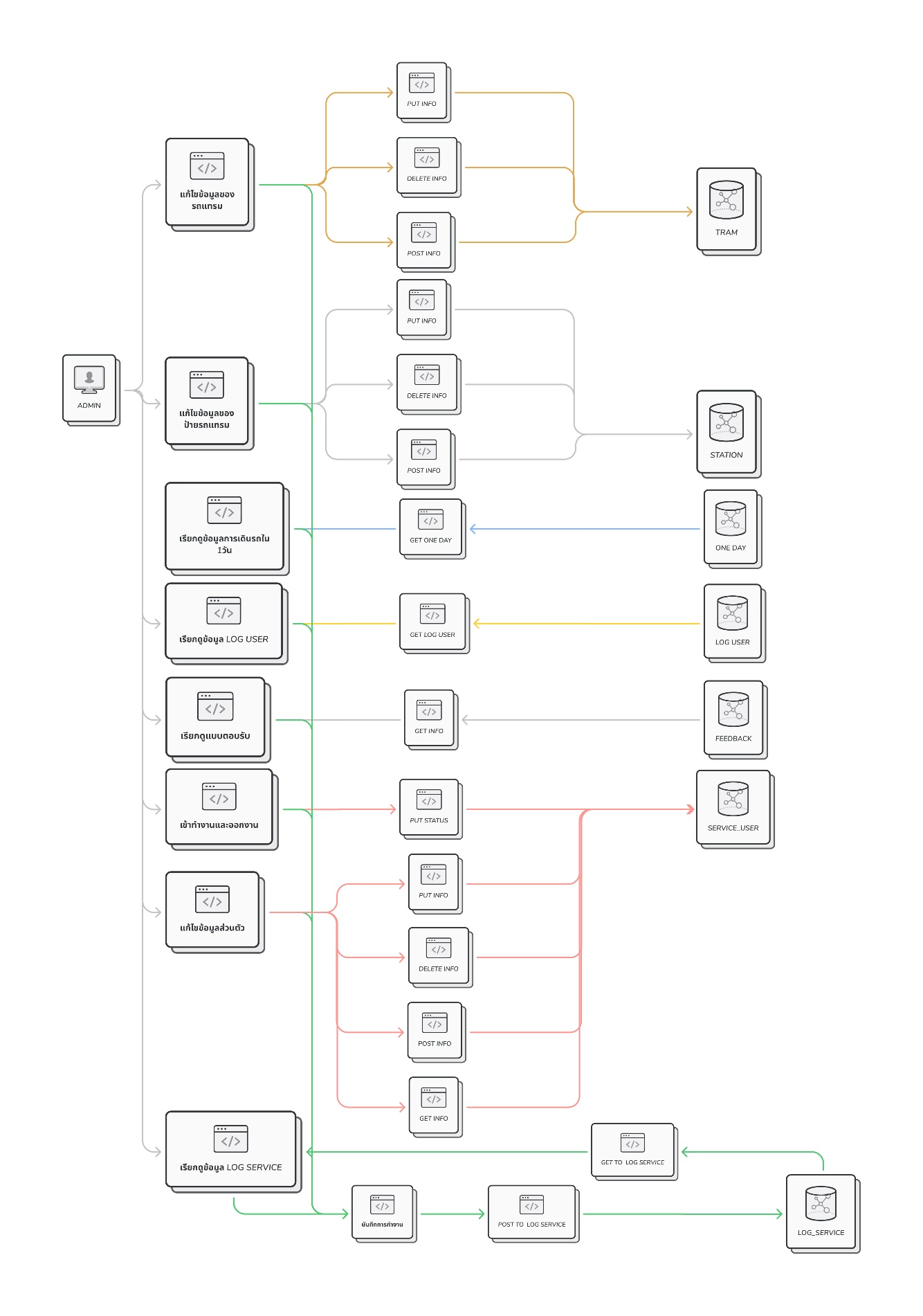
Driver จะสามารถดูข้อมูลส่วนตัวได้ คือ ID ชื่อ-นามสกุล อีเมล เบอร์โทร และตำแหน่งงาน

**6. ส่วนบันทึกการใช้งาน**

โดยจะเก็บการกระทำในส่วน เข้าทำงานและออกงาน และ เรียกดูข้อมูลส่วนตัว ไว้เป็นประวัติแล้วเก็บไว้ใน log\_service

## ส่วนของ Admin

จะแสดงฟังก์ชันการทำงานของ Admin โดยจะมีการทำงานโดยรวมดังรูป



การเข้าถึงข้อมูล

สีเทา Station

สีเขียว Log\_service

สีส้ม Tram

สีฟ้า One day

สีแดง Service provider

**1. แก้ไขข้อมูลส่วนรถแทรม**

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

**2. แก้ไขข้อมูลส่วนป้ายรถแทรม**

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

**3. 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

**4. 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

# ข้อมูลที่เกี่ยวข้อง

Git hub ของโครงงาน

<https://github.com/Domineice/Tram_tracking_draft>

Backend design

<https://www.figma.com/files/project/43364045/Team-project?fuid=1044613070526616251>

UI ส่วน User และ Driver Application

<https://www.figma.com/file/VhyuUDX6ykVqMXXQXrMFaD/Tramer?node-id=0%3A1>

UI ส่วน Admin

<https://www.figma.com/file/9DNdWLhyLUTpgVUu2d8bM1/tram_tracking?node-id=0%3A1>

# บรรณานุกรม

Google. (ม.ป.ป.). *FireBase Product*. เรียกใช้เมื่อ 14 ธันวาคม 2021 จาก FireBase: https://firebase.google.com/products-build

Google. (ม.ป.ป.). *Google Maps Reference*. เรียกใช้เมื่อ 14 ธันวาคม 2021 จาก Google Maps Platform: https://developers.google.com/maps/documentation/javascript/reference