



**เอกสารรายงานประกอบ Project**  
**Petverse WebApp**

โดย

663380390-4	นาย ปาณวัฒน์ จันทรทองกลาง
663380616-4	นาย เกรียงไกร ประเสริฐ
663380587-5	นาย กิตติกร เสวกวิหารี
663380045-1	นาย ปิติ มุลเทพพิชัย
663380604-1	นาย บุญยศศักดิ์ โพธิ์อ่อน

อาจารย์ประจำวิชา: ผศ. ดร.ปัญญาพล หอระตะ

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา CP353002 Principle of Software

ภาคเรียนที่ 1 ปีการศึกษา 2568

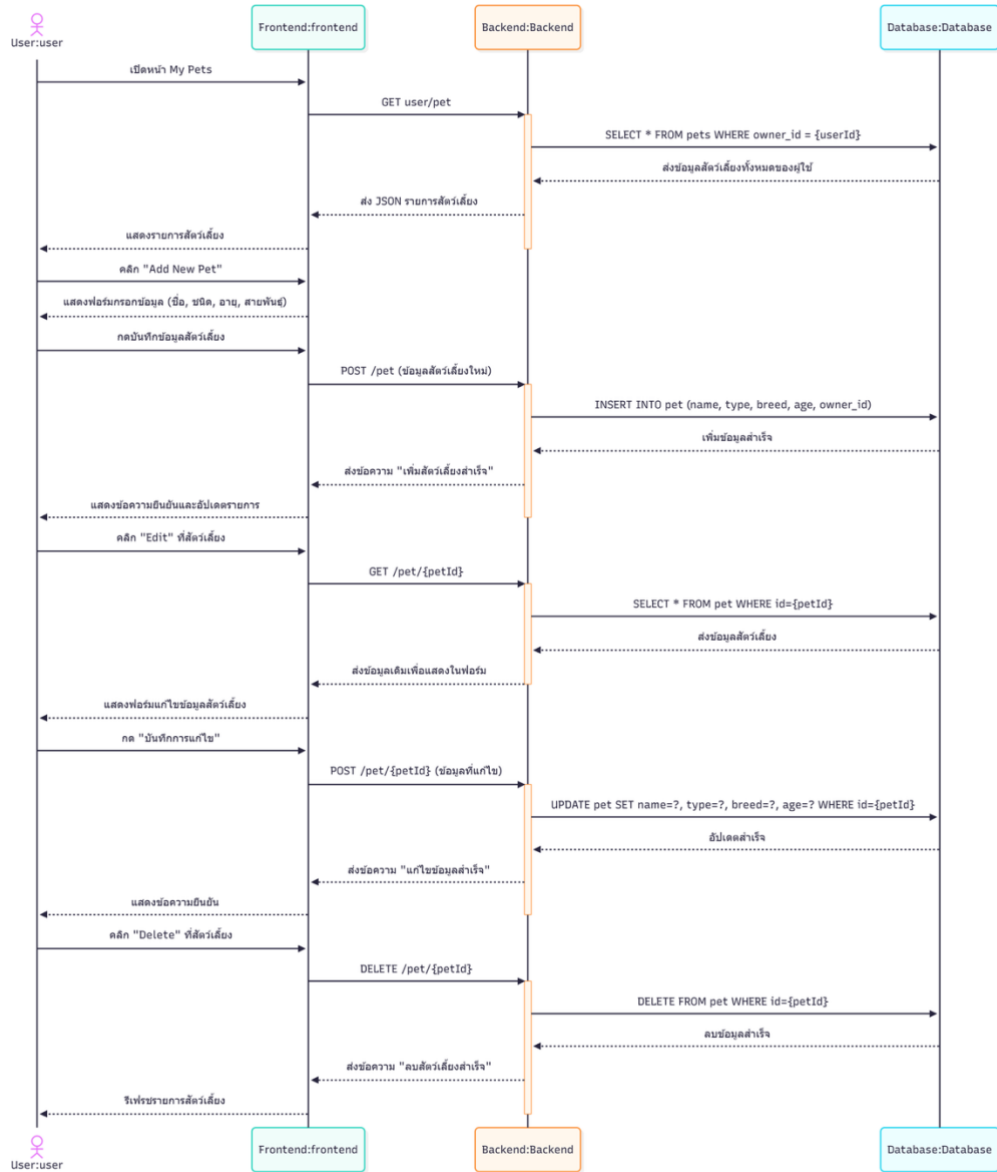
สาขาวิชาวิทยาการคอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์

มหาวิทยาลัยขอนแก่น

(ตุลาคมพ.ศ. 2568)

## Sequence diagram

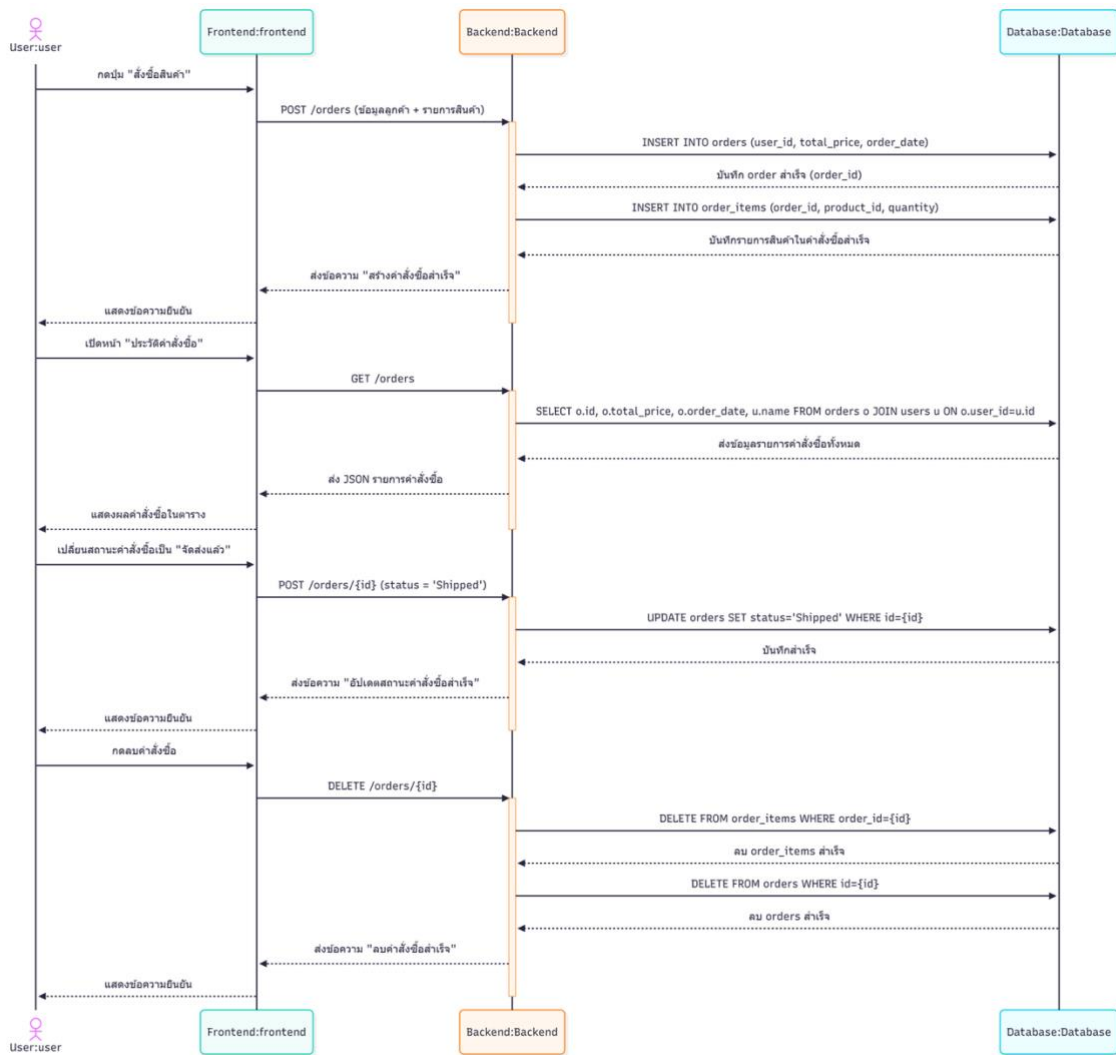
(User PetManagement)



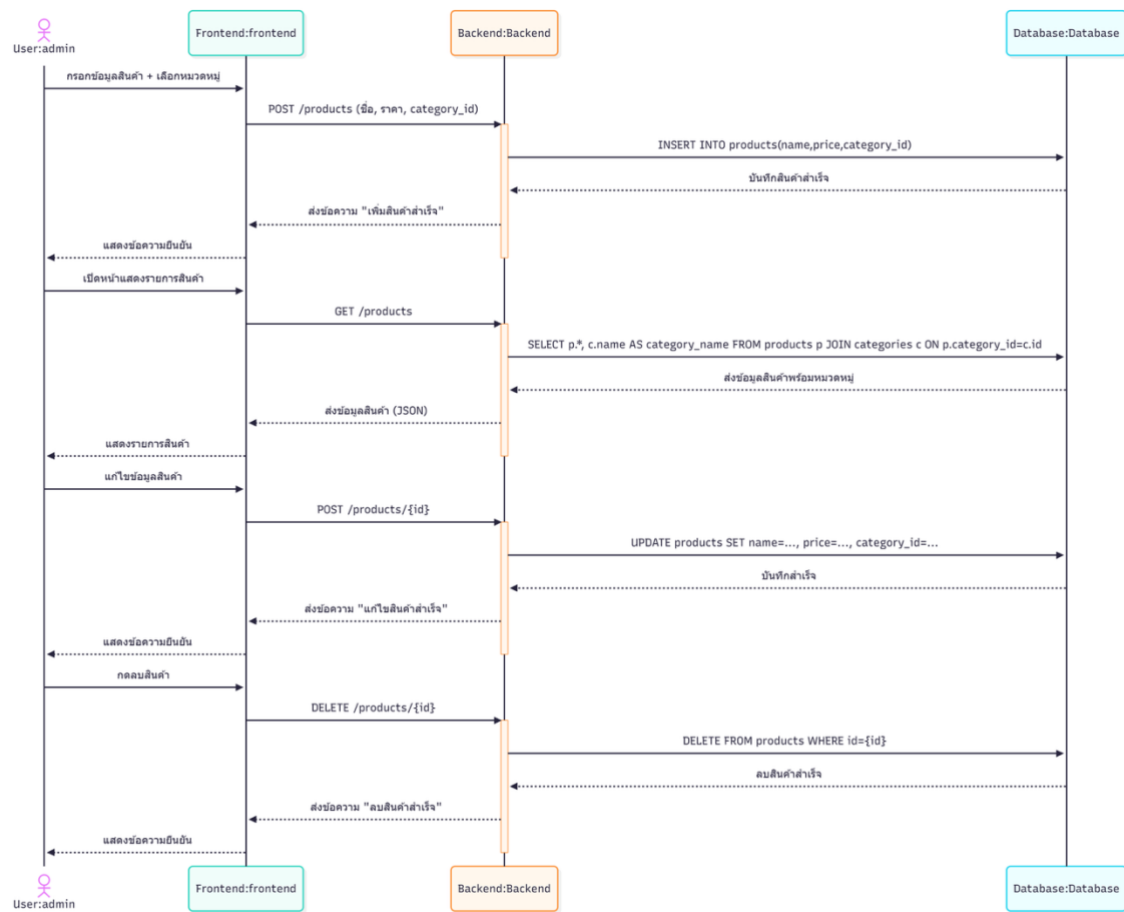
(User PetCommunity)



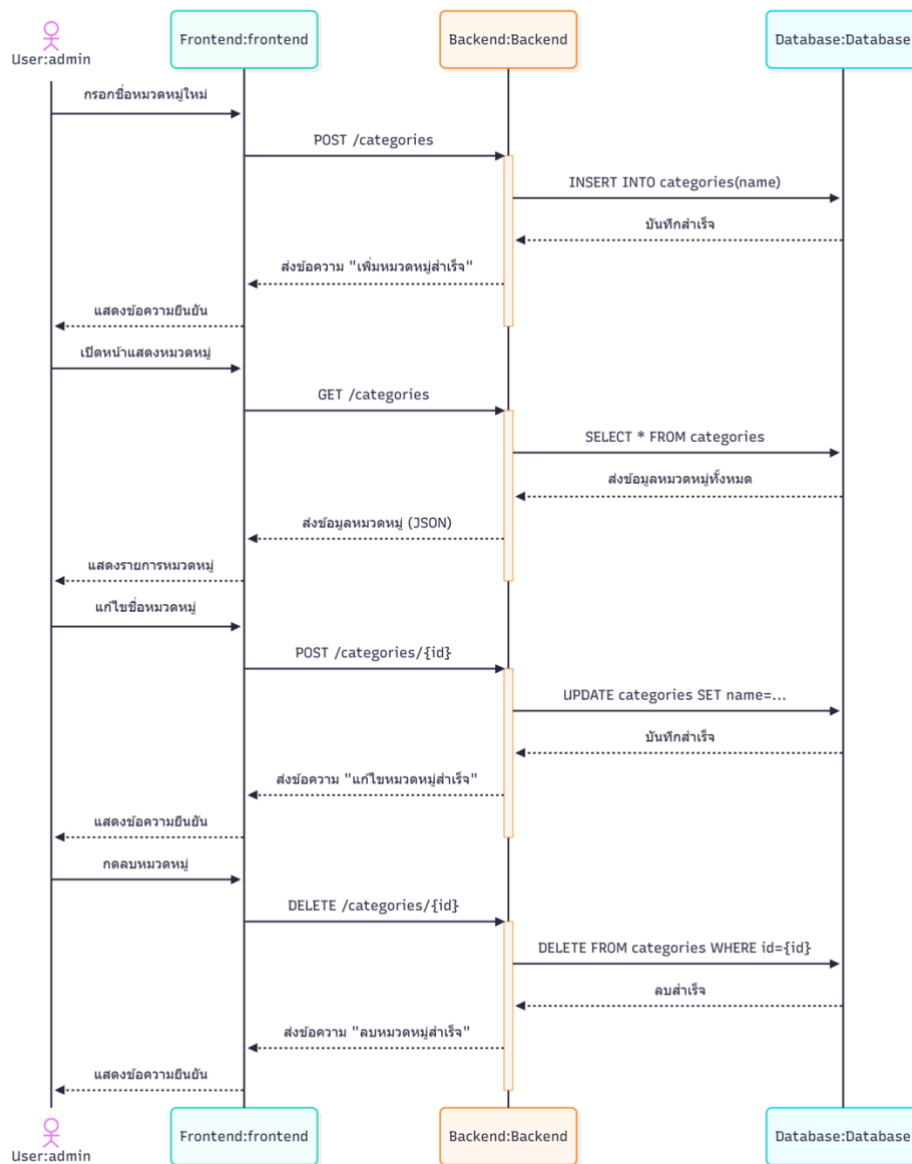
(User Order)



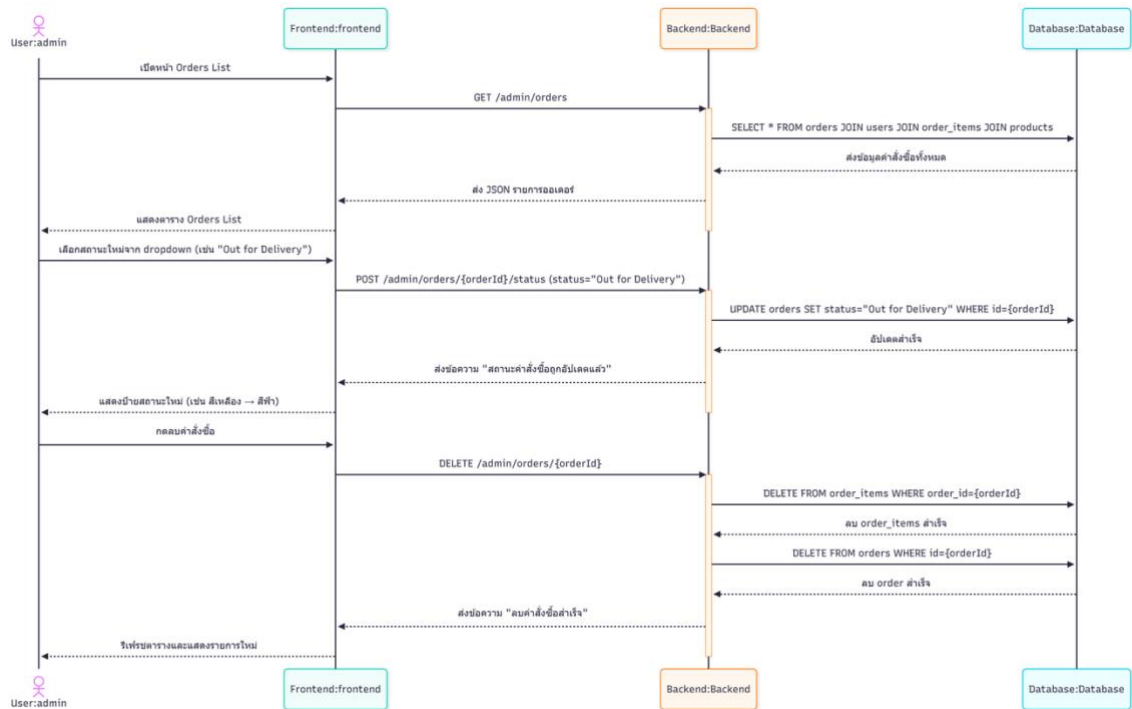
(Admin ProductManagement)



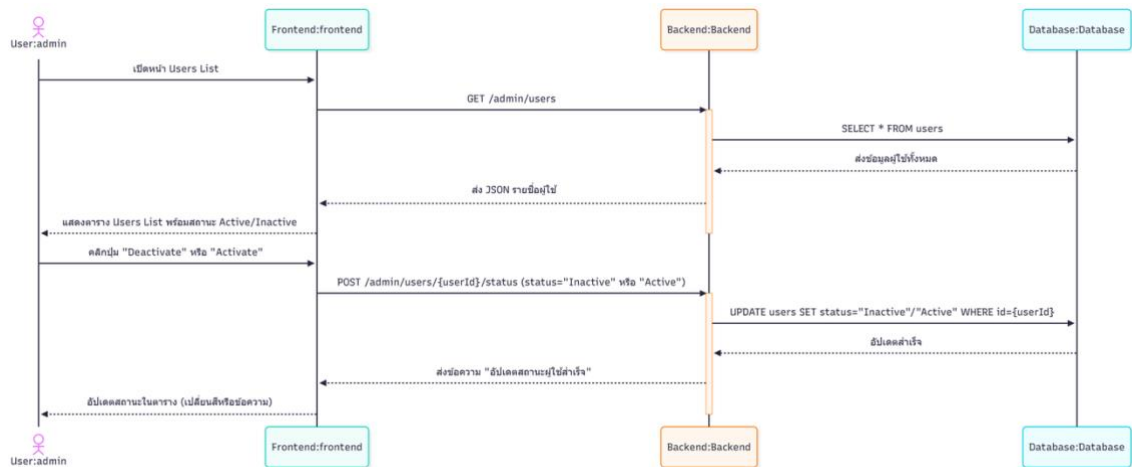
(Admin CategoryManagement)



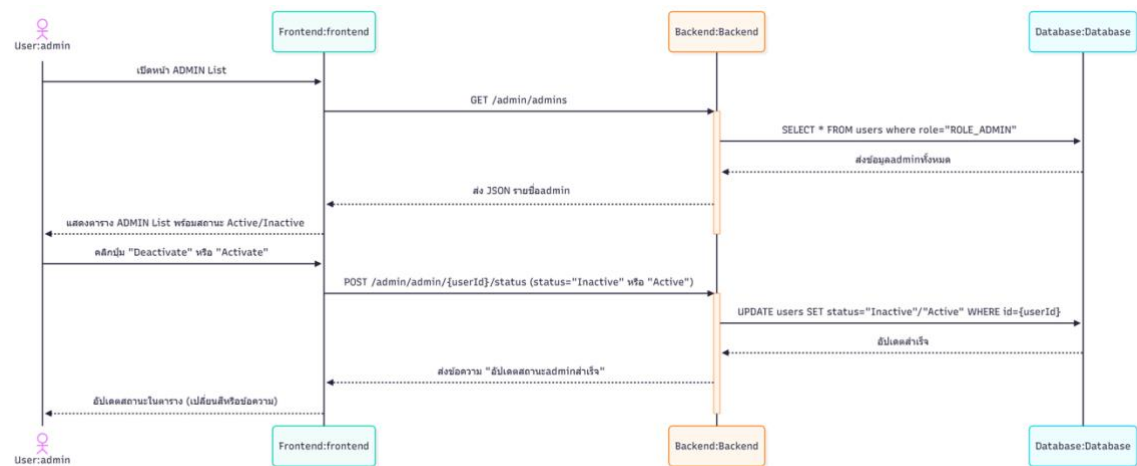
## (Admin OrderManagement)



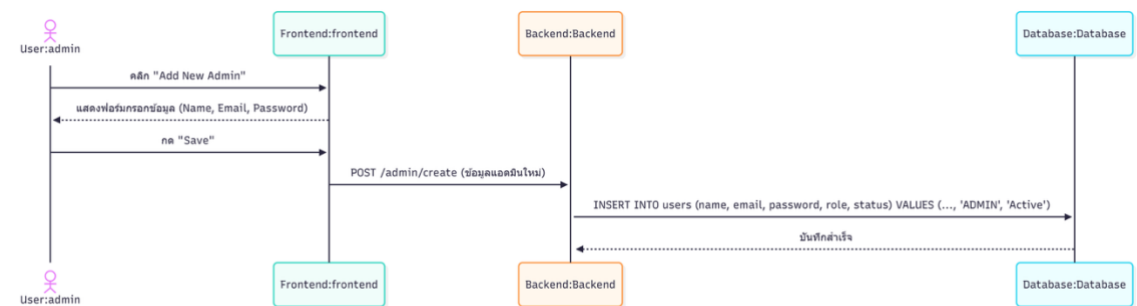
## (Admin UserManagement)



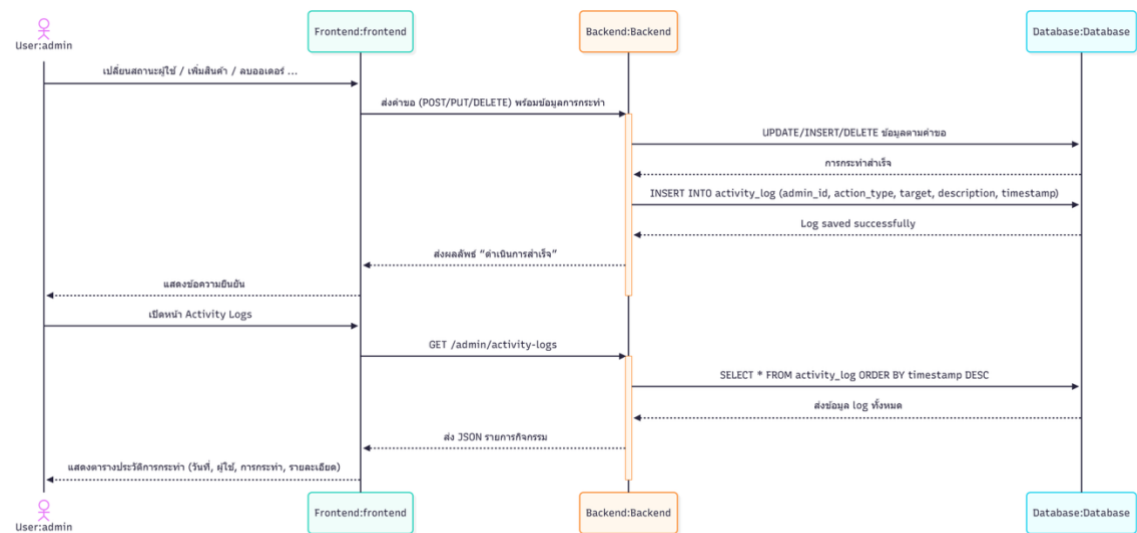
## (Admin Admin management)



## (Admin addAdmin)

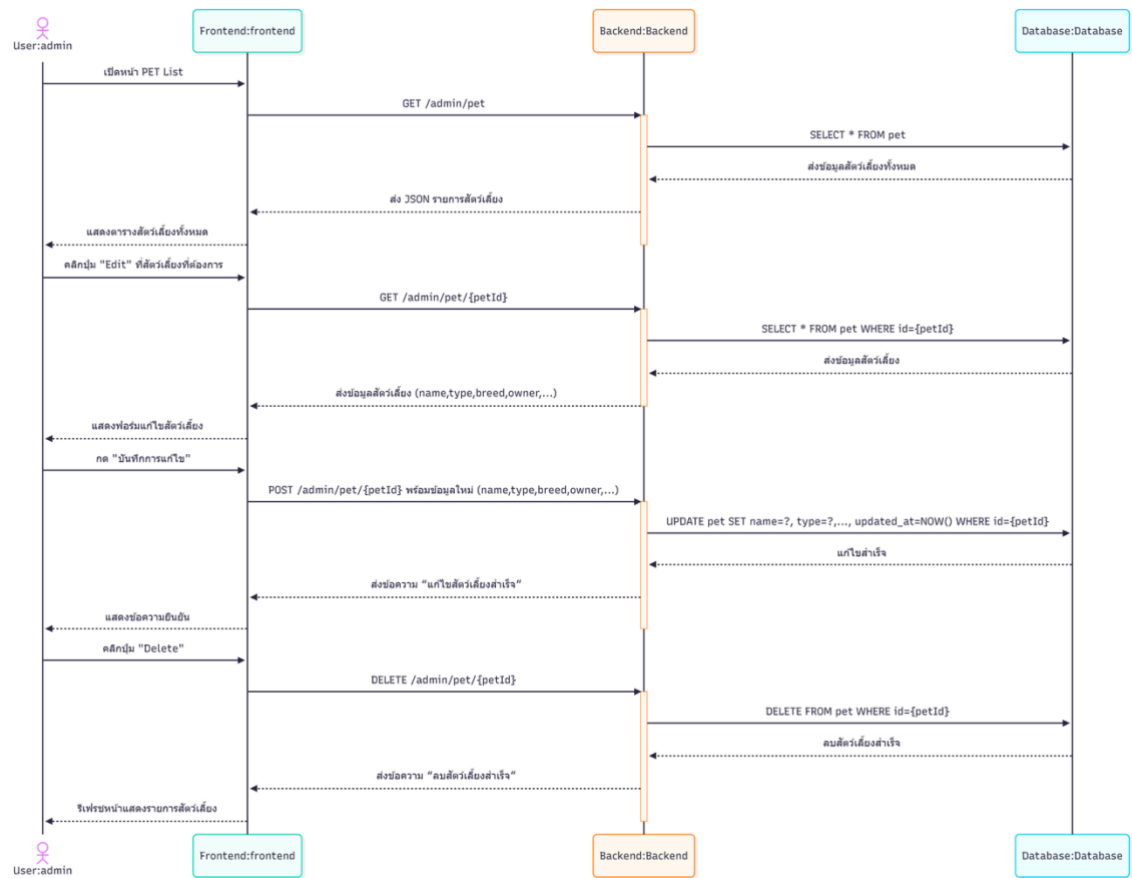


## (Admin Activity Log)





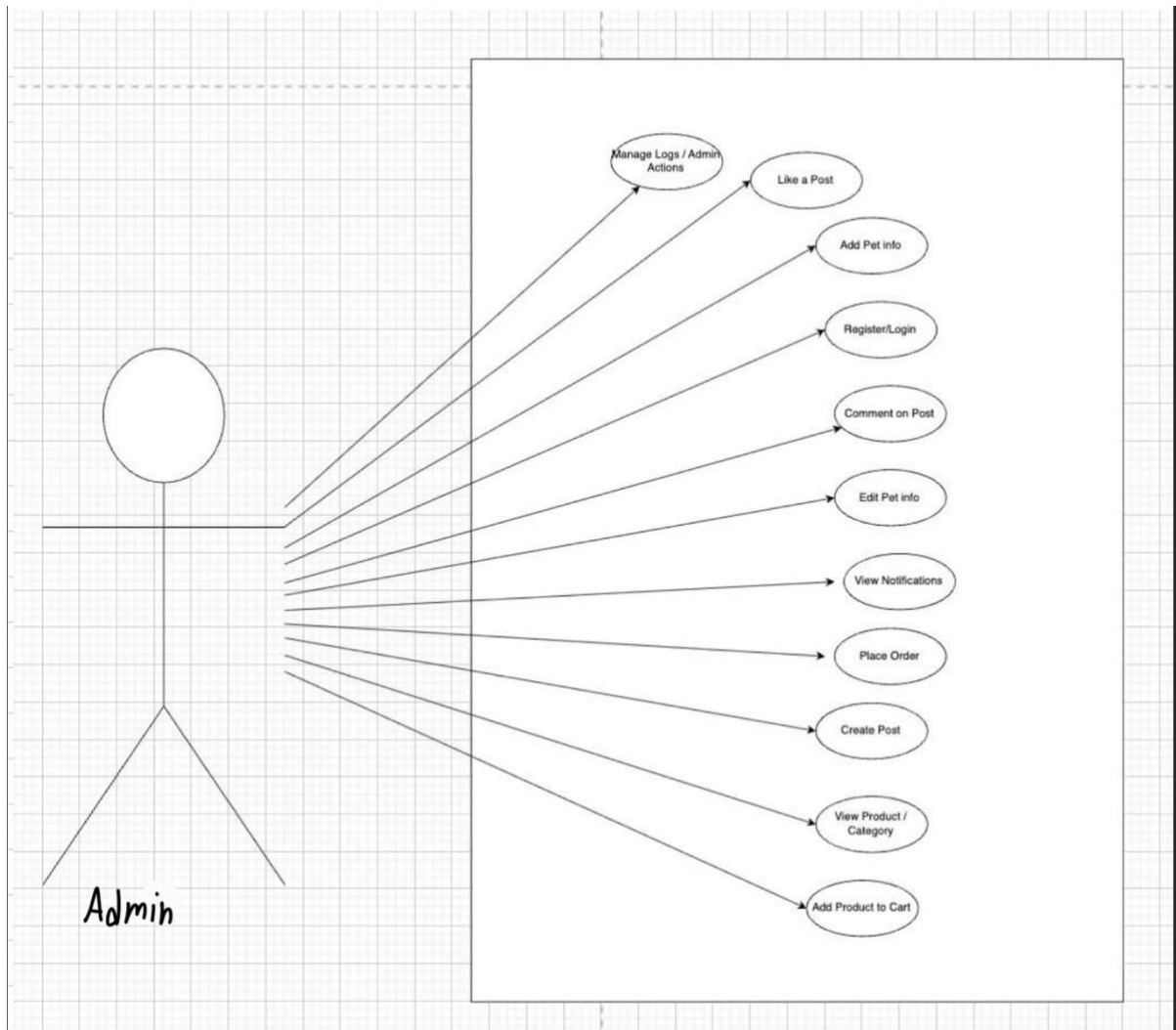
(Admin PetManagement)

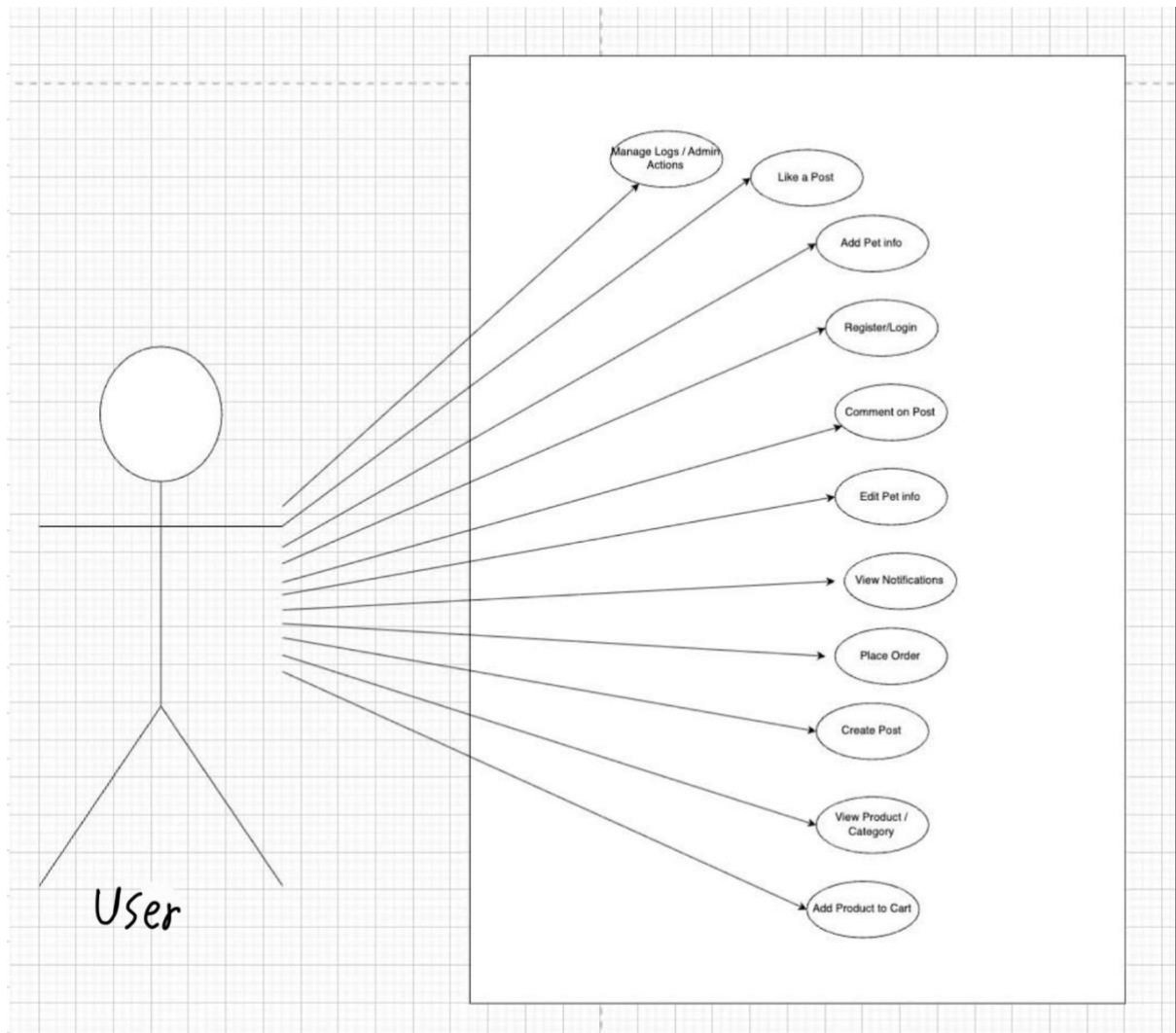


(Admin PetCommunityManagement)



## Use case





## คู่มือการ Deploy server โดยใช้ Amazon Web Service(AWS)

ขั้นตอนการนำ Spring Boot Application ขึ้นสู่ระบบคลาวด์ของ Amazon Web Services (AWS) โดยใช้ Elastic Beanstalk (EB) เป็นบริการหลักในการจัดการทรัพยากร (EC2, Load Balancer), S3 สำหรับจัดเก็บไฟล์, RDS สำหรับฐานข้อมูล MySQL และ IAM สำหรับการจัดการสิทธิ์การเข้าถึง

### ขั้นตอนที่ 1: การตั้งค่า IAM (Identity and Access Management)

ขั้นตอนนี้จำเป็นสำหรับการสร้างสิทธิ์การเข้าถึงบริการต่างๆ ของ AWS อย่างปลอดภัย

#### 1. สร้าง IAM User (สำหรับ S3 Access):

- ไปที่คอนโซล IAM และสร้าง User ใหม่
- ตั้งค่า Access Type เป็น **Programmatic access**
- แนบนโยบาย (Attach policies) AmazonS3FullAccess
- สร้าง User และจดบันทึก **Access Key ID** และ **Secret Access Key** ไว้ นี่คือข้อมูลสำคัญที่จะใช้ในแอปพลิเคชัน Spring Boot

#### 2. สร้าง IAM Role (สำหรับ EC2 Instance ของ Elastic Beanstalk):

- ไปที่ Roles และ "Create role"
- เลือก Use case เป็น **EC2**
- แนบนโยบายที่จำเป็นสำหรับ Elastic Beanstalk เช่น AWSElasticBeanstalkWebTier และ AWSElasticBeanstalkWorkerTier
- ตั้งชื่อ Role (เช่น eb-ec2-role) และสร้าง Role นี้

---

### ขั้นตอนที่ 2: การสร้าง S3 Buckets (Simple Storage Service)

S3 จะถูกใช้เพื่อจัดเก็บไฟล์ที่มีการอัปโหลด เช่น รูปภาพสินค้า หรือรูปโปรไฟล์

#### 1. สร้าง Buckets:

- ไปที่คอนโซล S3 และสร้าง Buckets ตามที่แอปพลิเคชันต้องการ (เช่น your-app-products, your-app-profiles)

#### 2. ตั้งค่าการเข้าถึง (Permissions):

- ในแท็บ "Permissions" ของแต่ละ Bucket ให้ปิดการใช้งาน **"Block all public access"** และยืนยัน
- แก้ไข **Bucket Policy** เพื่ออนุญาตให้มีการอ่านไฟล์แบบสาธารณะ (Public Read) โดยทั่วไปจะใช้ JSON policy ที่ตั้งค่า Principal เป็น \* (ทุกคน) และ Action เป็น s3:GetObject

---

### ขั้นตอนที่ 3: การสร้างฐานข้อมูล (RDS - Relational Database Service)

เราจะใช้ RDS เพื่อสร้างฐานข้อมูล MySQL ที่มีการจัดการโดย AWS

### 1. Launch RDS Instance:

- ไปที่คอนโซล RDS และ "Create database"
- เลือก "Standard Create" และ Engine type เป็น **MySQL**
- เลือก Template เป็น "**Free tier**" เพื่อการทดสอบ
- ตั้งค่า **DB instance identifier**, **Master username**, และ **Master password**

### 2. ตั้งค่าการเชื่อมต่อ (Connectivity):

- ในส่วน "Connectivity" ตั้งค่า "**Publicly accessible**" เป็น "**Yes**"
- เลือก VPC Security Group หรือสร้างใหม่ และแก้ไข Inbound Rules เพื่ออนุญาตการเชื่อมต่อผ่านพอร์ต **MySQL (3306)** จาก IP ของคุณ (เพื่อทดสอบ) และจาก Security Group ของ Elastic Beanstalk (สำหรับการใช้งานจริง)

### 3. จัดบันทึก Endpoint:

- หลังจากสร้างฐานข้อมูลเสร็จสิ้น (สถานะ "Available") ให้คัดลอก URL ของ **Endpoint** ไว้

---

## ขั้นตอนที่ 4: การกำหนดค่า Spring Boot Application

แอปพลิเคชันต้องได้รับการกำหนดค่าให้เชื่อมต่อกับบริการของ AWS ที่เราสร้างขึ้น

### 1. เพิ่ม Dependencies:

- ในไฟล์ pom.xml เพิ่ม AWS SDK สำหรับ S3 (aws-java-sdk-s3)

### 2. แก้ไข application.properties:

- **S3 Configuration:**
  - aws.access.key = (Access Key จากขั้นตอนที่ 1.1)
  - aws.secret.key = (Secret Key จากขั้นตอนที่ 1.1)
  - aws.region = (Region ของ S3 Bucket เช่น us-east-1)
  - aws.s3.bucket.category = (ชื่อ S3 Bucket ที่คุณสร้าง)
- **RDS Configuration:**
  - spring.datasource.url = jdbc:mysql://[Your-RDS-Endpoint]:3306/[your\_db\_name]?createDatabaseIfNotExist=true
  - spring.datasource.username = (Master username จากขั้นตอนที่ 3.1)
  - spring.datasource.password = (Master password จากขั้นตอนที่ 3.1)
- **Elastic Beanstalk Port Configuration:** (สำคัญมาก)
  - server.port = 5000
  - หมายเหตุ: *Elastic Beanstalk จะ forward traffic จากพอร์ต 80 (HTTP) ไปยังพอร์ต 5000 ของแอปพลิเคชันโดยอัตโนมัติ การตั้งค่านี้จำเป็นเพื่อหลีกเลี่ยงข้อผิดพลาด 502 Bad Gateway*

### 3. อัปเดตโค้ด:

- สร้างคลาส Config (เช่น AwsConfig) เพื่อสร้าง Bean ประเภท AmazonS3
- แก้ไข Service ที่จัดการการอัปโหลดไฟล์ (เช่น FileService) ให้ใช้ amazonS3.putObject() และบันทึก URL แบบเต็มของ S3 (เช่น https://[bucket].s3.amazonaws.com/[filename]) ลงในฐานข้อมูล

### 4. Build Project:

- สั่ง Build โปรเจกต์เป็นไฟล์ .jar โดยใช้คำสั่ง Maven: mvn clean install
- ไฟล์ .jar จะอยู่ที่ไดเรกทอรี target

---

## ขั้นตอนที่ 5: การ Deploy ด้วย Elastic Beanstalk (EB)

Elastic Beanstalk จะทำหน้าที่สร้างและจัดการสภาพแวดล้อม (EC2 instance, Security Groups, Load Balancer) ให้โดยอัตโนมัติ

### 1. Create Application:

- ไปที่คอนโซล Elastic Beanstalk และ "Create a new application"
- ตั้งชื่อแอปพลิเคชัน (เช่น shopping-cart-app)

### 2. Create Environment:

- สร้าง Environment ใหม่ เลือก **"Web server environment"**
- ตั้งค่า Platform เป็น **"Java"**
- เลือก **"Single instance (Free tier)"** สำหรับการทดสอบ

### 3. กำหนดค่า Service Access:

- คลิก **"Configure more options"**
- ในส่วน "Security" หรือ "Service access" ให้เลือก **EC2 instance profile** เป็น Role ที่คุณสร้างไว้ในขั้นตอนที่ 1.2 (เช่น eb-ec2-role)
- สร้าง Environment (สามารถเริ่มด้วย "Sample application" ก่อนได้)

### 4. Upload and Deploy:

- เมื่อ Environment พร้อมใช้งาน (สถานะ "Ok" สีเขียว) ให้คลิกปุ่ม **"Upload and Deploy"**
- เลือกไฟล์ .jar ที่คุณ Build ไว้ในขั้นตอนที่ 4.4
- คลิก **"Deploy"** และรอจนกว่ากระบวนการอัปเดตจะเสร็จสิ้น

---

## ขั้นตอนที่ 6: การทดสอบและสรุป

### 1. การทดสอบ:

- เข้าถึงแอปพลิเคชันผ่าน URL ที่ Elastic Beanstalk สร้างให้
- ทดสอบการทำงานหลัก เช่น การลงทะเบียน, การอัปโหลดรูปภาพ, และการดึงข้อมูล

- ตรวจสอบว่าไฟล์ถูกอัปโหลดไปยัง S3 Bucket และข้อมูลถูกบันทึกลงในฐานข้อมูล RDS อย่างถูกต้อง