



Final Project

PRINCIPLES OF SOFTWARE DESIGN AND DEVELOPMENT

ร้านขายอาหารสัตว์เลี้ยงออนไลน์

Pet Shop online

จัดทำโดย

นาย ธนพนธ์ รวิภาสกร	643020610-1	sec 3
นาย ธนากร ผาเป้า	643020613-5	sec 3
นางสาว สุพัตรา แพงจันทร์	643021342-5	sec 3

เสนอ

รศ.ดร.ปัญญาพล หอระตะ

รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชา

SC313002 PRINCIPLES OF SOFTWARE DESIGN AND DEVELOPMENT

มหาวิทยาลัยขอนแก่น

ภาคการเรียนที่ 1 ปีการศึกษา 2564

สาขาวิชาวิทยาการคอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์

คำนำ

โครงงานฉบับนี้เป็นส่วนหนึ่งของวิชา SC313002 หลักการออกแบบพัฒนาซอฟต์แวร์ (PRINCIPLES OF SOFTWARE DESIGN AND DEVELOPMENT) เพื่อ โดยมีจุดประสงค์เพื่อศึกษาและพัฒนาเว็บไซต์ขายอาหารสัตว์เลี้ยงออนไลน์ (Pet shop online) โดยใช้ Spring Boot ที่มีการนำหลัก SOLID และ Pattern เข้ามาเข้ามาใช้กับระบบเพื่อช่วยให้ระบบมีหลักการออกแบบที่ดี

คณะผู้จัดทำได้เลือกหัวข้อนี้มาทำโครงงาน เนื่องจากเป็นเรื่องที่น่าสนใจ ต้องขอขอบคุณ รศ.ดร.ปัญญาพล หอระตะ ผู้ให้ความรู้และแนวทางการศึกษาเกี่ยวกับการใช้ Spring Boot ,SOLID และ Pattern คณะผู้จัดทำหวังว่าโครงงานฉบับนี้จะให้ความรู้และเป็นประโยชน์ของผู้ที่ต้องการศึกษาทุกท่าน หากผิดพลาดประการใด คณะผู้จัดทำก็ขออภัยมา ณ ที่นี้

คณะผู้จัดทำ

17 ตุลาคม 2566

สารบัญ

เรื่อง	หน้า
สารบัญ	ก
สารบัญภาพ	ข
บทที่ 1	1
1. ที่มาและความสำคัญ	1
2. ทฤษฎีที่เกี่ยวข้อง	1
2.1 Spring Boot	2
2.2 SOLID: The First 5 Principles of Object-Oriented Design	2
2.3 Pattern in Spring boot	2
3. เครื่องมือที่เกี่ยวข้อง	3
3.1 Eclipse Enterprise installation	3
3.2 MySQL server installation	5
บทที่ 2	8
1. แผนภาพ UML	8
2. แผนภาพคงที่ (Class diagram)	8
3. แผนภาพยูสเคส (Use case diagram)	8
4. แผนภาพอีอาร์ (ER-Diagram)	9
บทที่ 3	10
1. ฐานข้อมูล MySQL	10
2. การสร้างเว็บไซต์โดยใช้ Spring Boot	11
3. User Interface	13
อ้างอิง	19

สารบัญภาพ

ภาพที่	หน้า
ภาพที่ 1 เข้าไปดาวน์โหลดEclipse และกดปุ่ม Download ตามภาพ	3
ภาพที่ 2 คลิกเข้าไปที่ไฟล์ที่ได้จากการดาวน์โหลดก่อนหน้านี้เลือกทำการติดตั้ง Eclipse IDE for Enterprise Java and Web Developers	3
ภาพที่ 3 เมื่อทำการติดตั้งเสร็จสิ้นแล้ว เลือกโฟลเดอร์ที่ต้องการติดตั้งหากเลือกค่า default ให้ทำการกด Install ได้เลย	4
ภาพที่ 4 เมื่อเข้ามาในโปรแกรม Eclipse ให้ทำการติดตั้ง Spring Tools ใน Marketplace	4
ภาพที่ 5 คลิกที่ Download MSI installer	5
ภาพที่ 6 คลิกที่ No thanks, just start my download	5
ภาพที่ 7 เลือกไฟล์ที่ Download มา	6
ภาพที่ 8 ทำตามขั้นตอนตามภาพ	6
ภาพที่ 9 เลือก Typical	7
ภาพที่10 เลือก Install	7
ภาพที่ 11 แผนภาพ UML	8
ภาพที่ 12 แผนภาพ Use case diagram	8
ภาพที่ 13 แผนภาพ Class Diagram	9
ภาพที่ 14 แผนภาพ ER Diagram	9
ภาพที่ 15 ตาราง category	10
ภาพที่ 16 ตาราง customer	10
ภาพที่ 17 ตาราง product	10
ภาพที่ 18 คลาส Category	11
ภาพที่ 19 คลาส InventoryService	12
ภาพที่ 20 คลาส Controller	12
ภาพที่ 21 หน้าหลักเมื่อเข้าเว็บไซต์	13
ภาพที่22 หน้าแสดงสินค้าสำหรับลูกค้า	13
ภาพที่ 23 หน้าการทำงานของ Admin	14
ภาพที่ 24 หน้าแสดง Category List	14
ภาพที่ 25 หน้าการอัปเดต Category	15
ภาพที่ 26 หน้า New category	15
ภาพที่ 27 หน้าแสดงรายการสินค้า	16
ภาพที่ 28 หน้าการอัปเดตสินค้า	16
ภาพที่ 29 หน้าการเพิ่มสินค้าใหม่	17

ภาพที่ 30 หน้าแสดง Customer List	17
ภาพที่ 31 หน้าการอัปเดต User	18

บทที่ 1

1. ที่มาและความสำคัญ

ในปัจจุบัน ผู้คนนิยมเลี้ยงสัตว์มากขึ้น ส่งผลให้ความต้องการอาหารสัตว์ก็เพิ่มสูงขึ้นตามไปด้วย เว็บไซต์อาหารสัตว์ (pet shop online) จึงเป็นช่องทางสะดวกสำหรับเจ้าของสัตว์เลี้ยงในการเลือกซื้ออาหารสัตว์ออนไลน์ โดยสามารถเลือกดูและเลือกซื้ออาหารสัตว์ได้หลายยี่ห้อและหลายประเภท เว็บไซต์ขายอาหารสัตว์ (pet shop online) จึงเป็นช่องทางที่สะดวก รวดเร็ว และมีความสำคัญต่อเจ้าของสัตว์เลี้ยงในการเลือกซื้ออาหารที่มีคุณภาพ เหมาะสมกับสัตว์เลี้ยงของตนเอง โดยเว็บไซต์อาหารสัตว์ (pet shop online)

2. ทฤษฎีที่เกี่ยวข้อง

2.1 Spring Boot

Spring Boot คือ Open-Source Framework ที่ใช้สำหรับการออกแบบโครงสร้าง Service หรือการออกแบบซอฟต์แวร์อย่าง Microservice และ Spring Boot ก็ยังมีจุดเด่นในการช่วยให้เราสามารถสร้างแอปพลิเคชันแบบ Stand-alone อีกด้วย โดยตัว Spring Boot ก็จะเข้ามาช่วยทำหน้าที่ในการจัดการกับตัวโค้ดของเราให้เป็นระเบียบและรวบรัดมากขึ้น ทำให้การทำงานของ Spring แอปพลิเคชันก็จะมีประสิทธิภาพมากขึ้นด้วยเช่นกัน

นอกจากนี้ Spring Boot ยังมี Auto Configuration ที่ช่วยลดความยุ่งยากในการกำหนดค่าต่าง ๆ ลง และก็สามารถนำมาใช้งานได้เลยทันที อีกทั้ง Spring Boot เป็น Framework ที่มีการพัฒนาอยู่ตลอดเวลา ทำให้เราสามารถใช้งานได้อย่างมั่นใจได้เลยว่าจะไม่ถูกกลายแพกลางทางอย่างแน่นอน โดยล่าสุดในปี 2022 ที่ผ่านมาก็ได้มีการอัปเดต Spring Boot 3.0 ให้เราได้ใช้งานอย่างเป็นทางการแล้วด้วย

2.2 SOLID: The First 5 Principles of Object-Oriented Design

Single Responsibility Principle (SRP) หลักการ SRP กำหนดว่าคลาสหรือโมดูลควรมีหน้าที่เดียวเท่านั้น สิ่งนี้จะช่วยให้รหัสมีความชัดเจนและเข้าใจง่ายขึ้น นอกจากนี้ยังช่วยให้ปรับเปลี่ยนได้ง่ายขึ้นเมื่อความต้องการเปลี่ยนแปลง

Open-Closed Principle (OCP) หลักการ OCP กำหนดว่าคลาสหรือโมดูลควรเปิดสำหรับการขยาย แต่ปิดสำหรับการปรับเปลี่ยน สิ่งนี้จะช่วยให้ซอฟต์แวร์สามารถปรับเปลี่ยนได้ง่ายขึ้นเมื่อความต้องการเปลี่ยนแปลง

Liskov Substitution Principle (LSP) หลักการ LSP กำหนดว่าคลาสย่อยควรสามารถแทนที่คลาสหลักได้โดยไม่ต้องกระทบต่อพฤติกรรม สิ่งนี้จะช่วยให้มั่นใจได้ว่าซอฟต์แวร์สามารถทำงานได้ตามที่คาดหวังแม้หลังจากคลาสใหม่ถูกเพิ่มเข้ามา

Interface Segregation Principle (ISP)

หลักการ ISP กำหนดว่าอินเทอร์เฟซควรเป็นชุดของอินเทอร์เฟซย่อยที่เล็กกว่าและเกี่ยวข้องกันมากกว่า สิ่งนี้จะช่วยให้ซอฟต์แวร์มีความยืดหยุ่นและความสามารถในการขยาย

Dependency Inversion Principle (DIP)

หลักการ DIP กำหนดว่าคลาสระดับสูงไม่ควรขึ้นอยู่กับคลาสระดับต่ำ คลาสทั้งสองควรขึ้นอยู่กับอินเทอร์เฟซ

หลักการ SOLID เป็นแนวทางที่มีประสิทธิภาพสำหรับการออกแบบและการพัฒนาซอฟต์แวร์เชิงวัตถุที่มีประสิทธิภาพและยืดหยุ่น การปฏิบัติตามหลักการเหล่านี้จะช่วยให้มั่นใจได้ว่าซอฟต์แวร์สามารถปรับเปลี่ยนได้ง่ายขึ้นเมื่อความต้องการเปลี่ยนแปลง และสามารถทำงานได้อย่างถูกต้องแม้หลังจากคลาสใหม่ถูกเพิ่มเข้ามา

2.3 Pattern in Spring boot

รูปแบบการออกแบบเป็นองค์ประกอบสำคัญใน Spring Boot โดยการเข้าใจและการใช้รูปแบบการออกแบบ นักพัฒนาสามารถสร้างแอปพลิเคชัน Spring Boot ที่มีประสิทธิภาพมากขึ้น ทดสอบได้มากขึ้น และบำรุงรักษาได้ง่ายขึ้น

Singleton pattern: รูปแบบ singleton ช่วยให้แน่ใจว่ามีอินสแตนซ์ของคลาสเพียงอินสแตนซ์เดียวในแอปพลิเคชัน สิ่งนี้มีประโยชน์สำหรับคลาสที่ต้องการรักษาสถานะแบบรวม เช่น สессเชื่อมต่อกับฐานข้อมูลหรือแคช

Factory method pattern: รูปแบบ factory method ช่วยให้สามารถสร้างวัตถุได้โดยไม่ต้องเปิดเผยตรรกะการสร้างเบื้องหลังต่อไคลเอนต์ สิ่งนี้มีประโยชน์ในการสร้างวัตถุที่ซับซ้อนหรือวัตถุที่ต้องสร้างในลักษณะเฉพาะ

Proxy pattern: รูปแบบ proxy ช่วยให้สามารถดักจับและควบคุมการเรียกไปยังวัตถุได้ สิ่งนี้มีประโยชน์ในการ Implement คุณสมบัติ เช่น การแคช การตรวจสอบสิทธิ์ และการบันทึก

Template method pattern: รูปแบบเทมเพลตเมธอดช่วยให้สามารถกำหนดโครงร่างของอัลกอริทึมได้ในขณะที่อนุญาตให้คลาสย่อยกำหนดขั้นตอนเฉพาะ สิ่งนี้มีประโยชน์ในการ Implement อัลกอริทึมที่แตกต่างกันขึ้นอยู่กับบริบทเฉพาะ

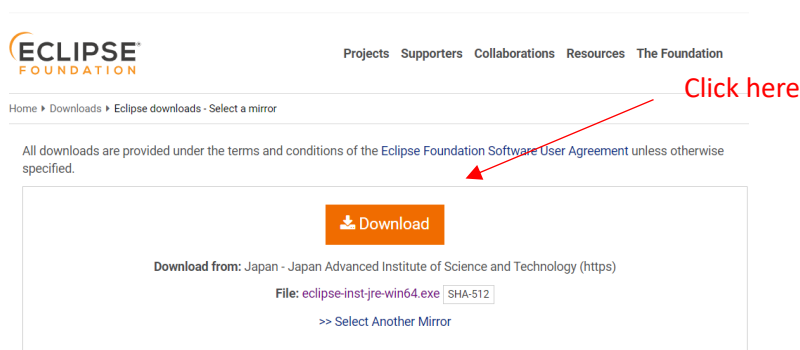
Dependency injection (DI): DI เป็นรูปแบบการออกแบบที่ช่วยให้วัตถุสามารถรับ Dependencies ของตนได้โดยไม่ต้องสร้างด้วยตัวเอง สิ่งนี้ทำให้รหัสสามารถทดสอบได้มากขึ้นและสามารถนำกลับมาใช้ใหม่ได้

3. เครื่องมือที่เกี่ยวข้อง

3.1 Eclipse Enterprise installation

Spring Boot เป็นโครงการที่ช่วยนักพัฒนาสร้างแอปพลิเคชัน Java ได้รวดเร็วและง่ายด้วยการกำหนดค่าและโครงสร้างที่มีคำสั่งต่าง ๆ พร้อมแล้ว สามารถใช้กับแอปพลิเคชันเว็บและควบคุมที่ใช้ Spring Framework และถูกพัฒนาโดยPivotalSoftware

ขั้นตอนที่ 1 Download Eclipse ที่ <https://www.eclipse.org/>



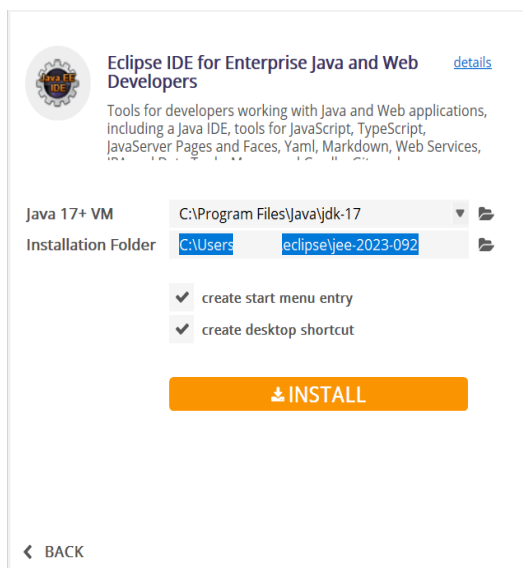
ภาพที่ 1 เข้าไปดาวน์โหลดEclipse และกดปุ่ม Download ตามภาพ

ขั้นตอนที่ 2 Double click ที่ไฟล์ eclipse-inst-win64.exe แล้วเลือก Eclipse IDE for Enterprise Java and Web Developers



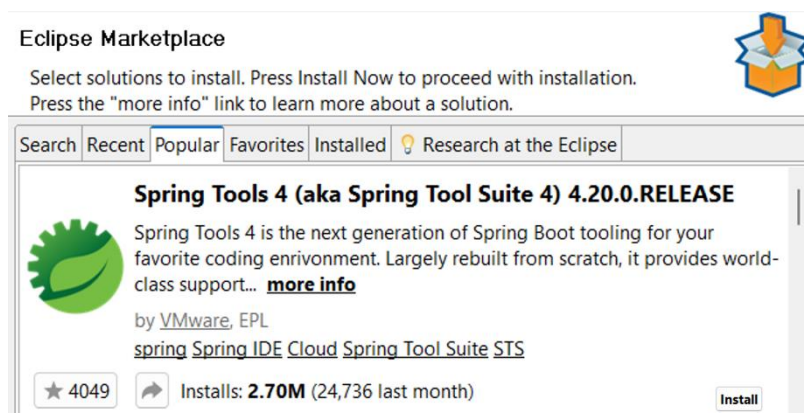
ภาพที่ 2 คลิกเข้าไปที่ไฟล์ที่ได้จากการดาวน์โหลดก่อนหน้านี้เลือกทำการติดตั้ง Eclipse IDE for Enterprise Java and Web Developers

ขั้นตอนที่ 3 โปรแกรมจะสอบถามไฟล์เดอร์ที่ต้องการให้ติดตั้ง หากต้องการติดตั้งที่ Default folder ให้กด Install ได้เลยโดยที่ไม่ต้องแก้ไขค่า



ภาพที่ 3 เมื่อทำการติดตั้งเสร็จสิ้นแล้ว เลือกไฟล์เดอร์ที่ต้องการทำการติดตั้งหากเลือกค่า default ให้ทำการกด Install ได้เลย

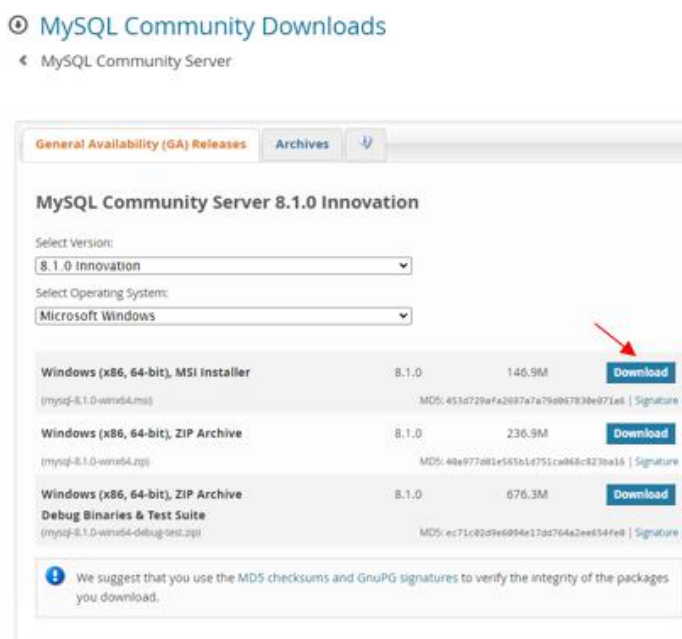
ขั้นตอนที่ 4 หลังจากติดตั้งโปรแกรมเสร็จให้ ทำการ ติดตั้ง Spring tools ใน Marketplace เป็นอันเสร็จสิ้น



ภาพที่ 4 เมื่อเข้ามาในโปรแกรม Eclipse ให้ทำการติดตั้ง Spring Tools ใน Marketplace

3.2 MySQL server installation

ขั้นตอนที่ 1 Download MySQL server from: <https://dev.mysql.com/downloads/mysql/>



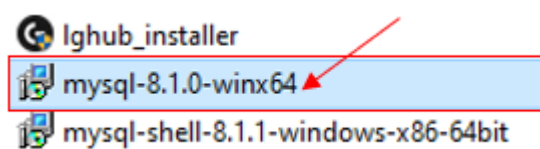
ภาพที่ 5 คลิกที่ Download MSI installer

ขั้นตอนที่ 2



ภาพที่ 6 คลิกที่ No thanks, just start my download.

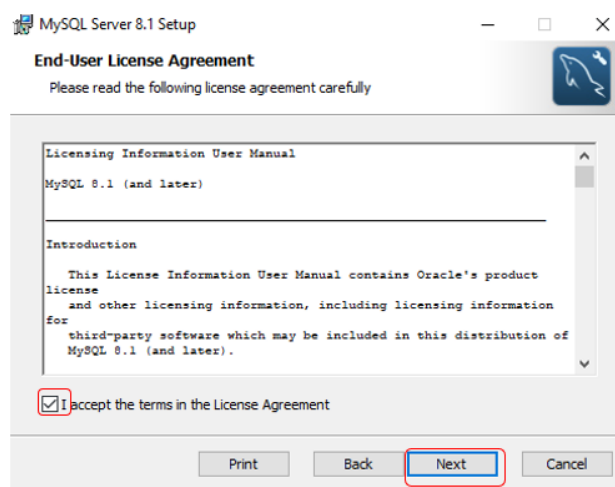
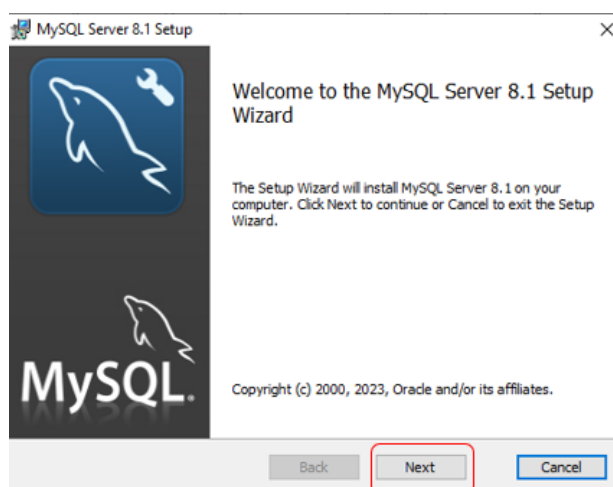
ขั้นตอนที่ 3



ภาพที่ 7 เลือกไฟล์ที่ Download มา

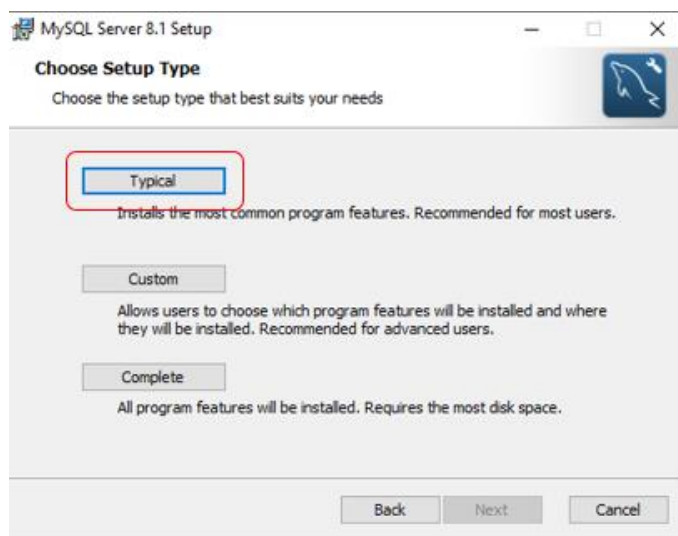
ขั้นตอนที่ 4

ทำตามขั้นตอนดังภาพ

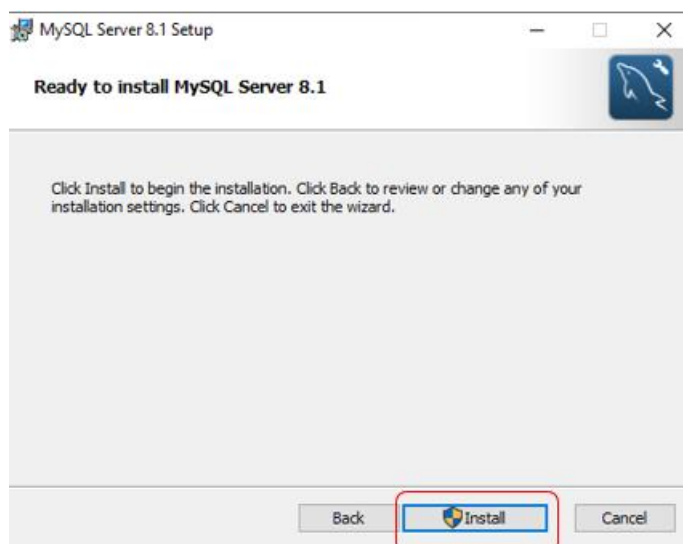


ภาพที่ 8 ทำตามขั้นตอนตามภาพ

เลือกที่ Typical และ install

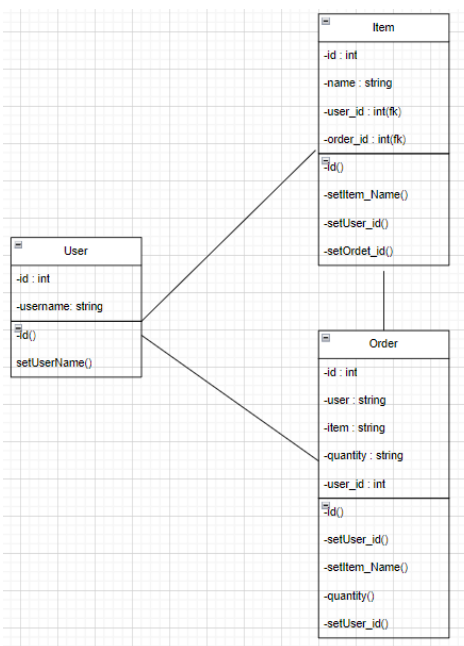


ภาพที่ 9 เลือก Typical

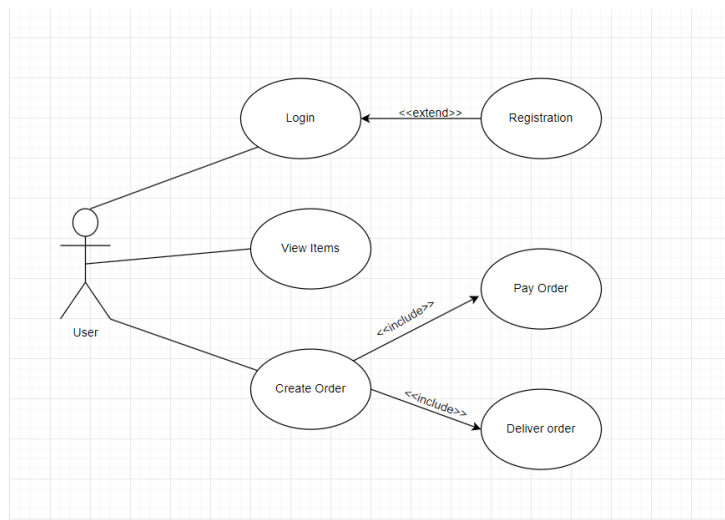


ภาพที่ 10 เลือก Install

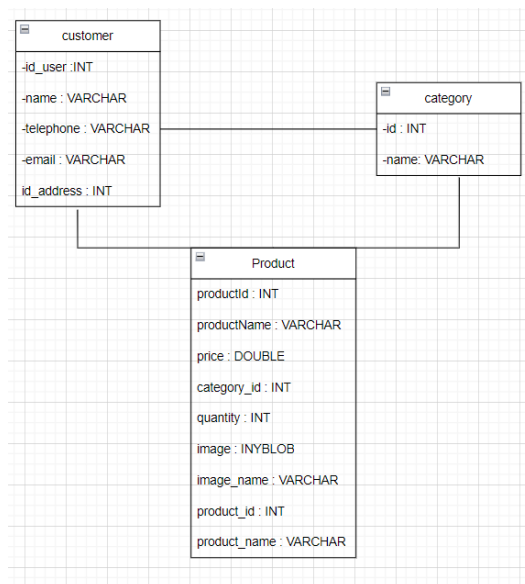
บทที่ 2



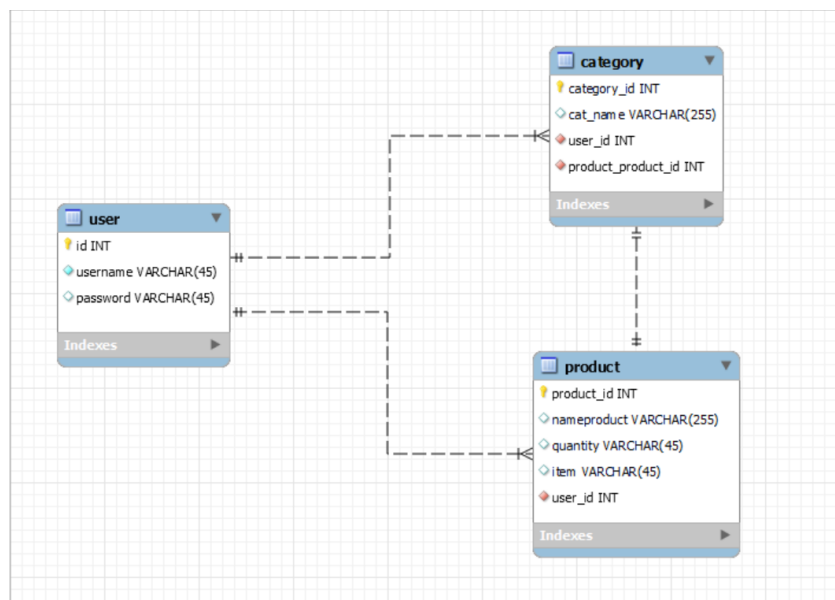
ภาพที่ 11 แผนภาพ UML



ภาพที่ 12 แผนภาพ Use case diagram



ภาพที่ 13 แผนภาพ Class Diagram



ภาพที่ 14 แผนภาพ ER Diagram

บทที่ 3

ฐานข้อมูล MySQL

Table Name: Schema: **lab4**

Charset/Collation: Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

ภาพที่ 15 ตาราง category

Table Name: Schema: **lab4**

Charset/Collation: Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id_user	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
telephone	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
email	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
id_address	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

ภาพที่ 16 ตาราง customer

Table Name: Schema: **lab4**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
productid	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
productname	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
price	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
category_id	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
quantity	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
image	TINYBLOB	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
image_name	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
product_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
product_name	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

ภาพที่ 17 ตาราง product

การสร้างเว็บไซต์โดยใช้ Spring Boot

@Entity 'Category' สามารถใช้ในการจัดเก็บข้อมูลในตาราง 'category' ในฐานข้อมูล โดยมีความสัมพันธ์กับ 'Product' ในรูปแบบ One-to-Many และใช้ Cascade เพื่อจัดการข้อมูล 'Product' ที่เกี่ยวข้อง เมื่อมีการเปลี่ยนแปลงใน 'Category'

```

12
13 @Entity
14 @Table(name="category")
15 public class Category {
16     @Id
17     // @GeneratedValue(strategy=GenerationType.AUTO)
18
19     @Column(name="id")
20     private int id;
21     @Column(length=45, nullable=false, unique=true)
22     private String name;
23     public int getId() {
24         return id;
25     }
26     public void setId(int id) {
27         this.id = id;
28     }
29     public String getName() {
30         return name;
31     }
32     public void setName(String name) {
33         this.name = name;
34     }
35
36     @OneToMany(targetEntity=Product.class, mappedBy="category",
37               cascade=CascadeType.ALL, fetch = FetchType.LAZY)
38     private List<Product> products;
39     public List<Product> getProducts() {
40         return products;
41     }
42     public void setProducts(List<Product> products) {
43         this.products = products;
44     }

```

ภาพที่ 18 คลาส Category

คลาส InventoryService นี้ใช้สำหรับการจัดการกับ Category, Product, และ Customer ซึ่งเป็นส่วนหนึ่งของระบบ Inventory และใช้ Repository ที่ถูก Inject เข้ามาเพื่อดึงข้อมูลจากฐานข้อมูล และส่งคืนข้อมูลในรูปแบบของ List ในแต่ละประเภทของข้อมูล

@Service: ประกาศคลาส InventoryService เป็น Spring Service ซึ่งคลาสนี้จะถูกสร้างเป็น Bean โดยอัตโนมัติและสามารถเรียกใช้ผ่าน Dependency Injection ได้


```

1 package com.example.demo;
2
3 import java.util.List;
4
5 @Service
6 public class InventoryService {
7     @Autowired
8     private CategoryRepository catRepo;
9     @Autowired
10    private ProductRepository productRepo;
11    @Autowired
12    private CustomerRepository customRepo;
13    @Autowired
14    public InventoryService() {}
15    // @Autowired // Constructor dependency injection
16    public InventoryService(CategoryRepository catRepo, ProductRepository productRepo, CustomerRepository customRepo) {
17        this.catRepo = catRepo;
18        this.productRepo = productRepo;
19        this.customRepo = customRepo;
20    }
21
22    public List<Category> getAllCategory(){
23        List<Category> cats = (List<Category>) this.catRepo.findAll();
24        return cats;
25    }
26
27    public List<Product> getAllProduct(){
28        List<Product> Product = (List<Product>) this.productRepo.findAll();
29        return Product;
30    }
31
32    public List<customer> getAllCustomer(){
33        List<customer> customer = (List<customer>) this.customRepo.findAll();
34        return customer;
35    }
36 }
37
38

```

ภาพที่ 19 คลาส InventoryService

@GetMapping เป็นการดึงค่า จาก Class category มาใช้โดย GET ไปยังปลายทาง "categoryList" จะดึงรายการที่ใช้ invSer.getAllCategory และจัดเก็บไว้ในตัวแปรจากนั้นรายการหมวดหมู่จะถูกเพิ่มลงในโมเดล

จะทำให้เราสามารถเพิ่มรายการ Category ได้

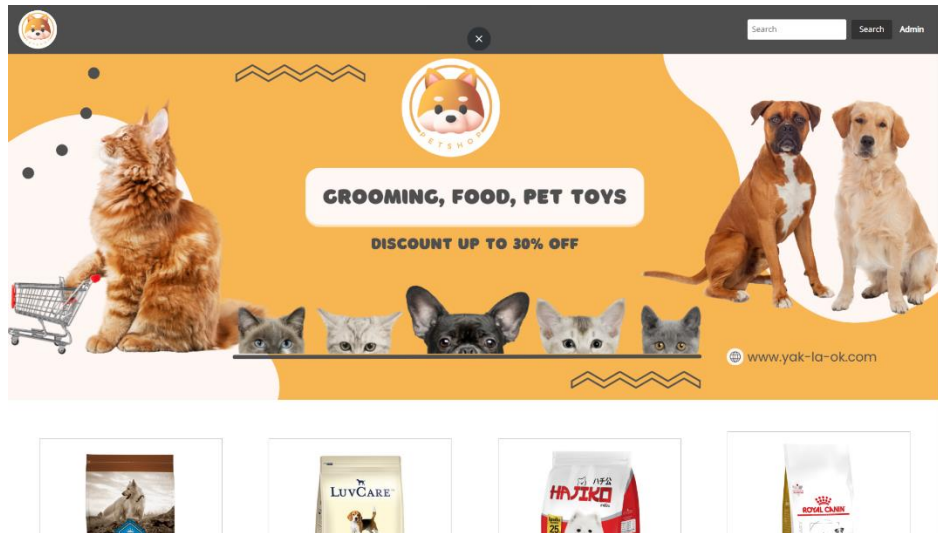
```

42
43 @GetMapping("/categoryList")
44 public String getListOfCategoryList (Model model) {
45     List<Category> categorys= invSer.getAllCategory();
46     model.addAttribute("categorys",categorys);
47     return "categoryList";
48 }
49
50
51 @GetMapping("/list_product")
52 public String getproduct(Model model) {
53     List<Product> Product = invSer.getAllProduct();
54     model.addAttribute("products",Product);
55     return "productList";
56 }
57
58
59 @GetMapping("/customer")
60 public String getcustomer(Model model) {
61     List<customer> customer= invSer.getAllCustomer();
62     model.addAttribute("customer",customer);
63     return "customerList";
64 }
65
66 @GetMapping("/new_category")
67 public String showForm(Model model) {
68     Category category = new Category();
69     model.addAttribute("category", category);
70     return "new_category";
71 }

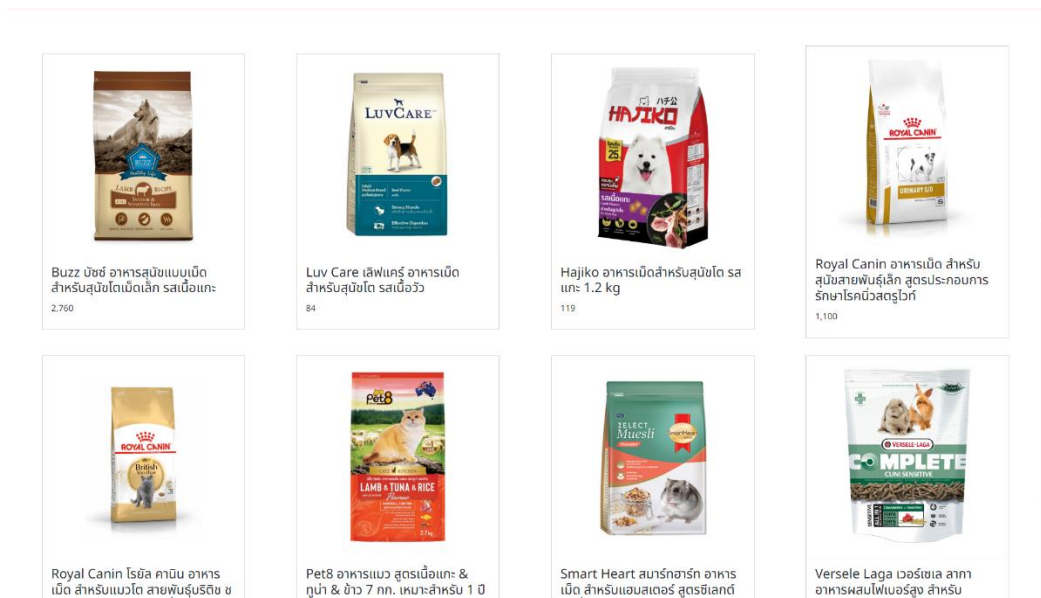
```

ภาพที่ 20 คลาส Controller

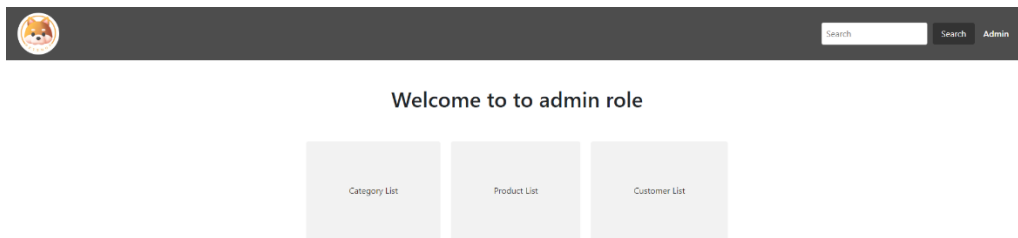
User Interface



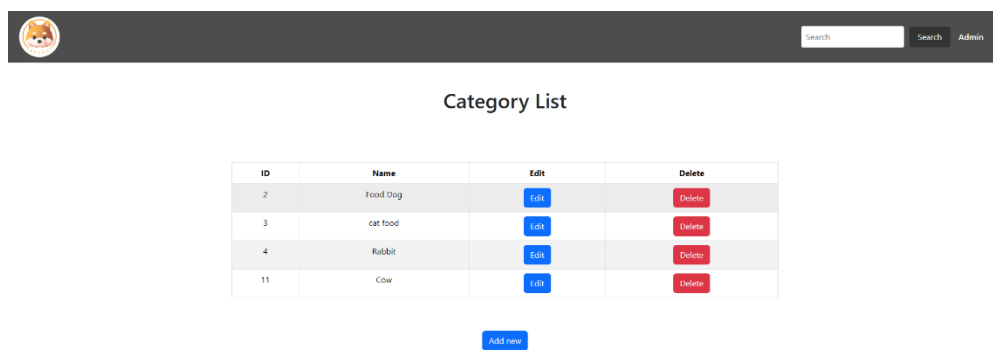
ภาพที่ 21 หน้าหลักเมื่อเข้าเว็บไซต์



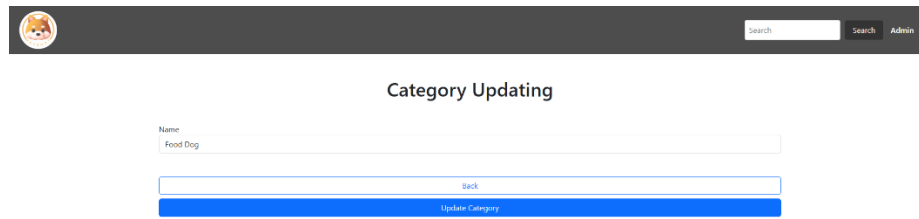
ภาพที่ 22 หน้าแสดงสินค้าสำหรับลูกค้า



ภาพที่ 23 หน้าการทำงานของ Admin



ภาพที่ 24 หน้าแสดง Category List



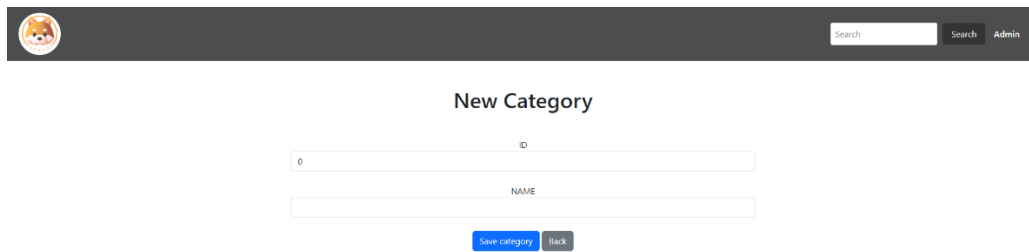
Category Updating

Name
Food Dog

Back

Update Category

ภาพที่ 25 หน้าการอัปเดต Category



New Category

ID
0

NAME

Save category Back

ภาพที่ 26 หน้า New category



Product List

Id	Name	Quantity	Price	Category ID	Category Name	Edit	Delete
2	Versle Lage เวอร์เซล ลาเก อาหารแมวโฮลเกรนสูง สำหรับกระต่ายบ้าน	10	250.0	2	Food Dog	Edit	Delete
13	Smart Heart สมารท์หัวใจ อาหารแมว สำหรับแมวและแมว ลูกเขี้ยว แมว	25	195.0	3	cat food	Edit	Delete

[Add a product](#)

ภาพที่ 27 หน้าแสดงรายการสินค้า

Update Product

id:

Name:

quantity:

price:

category:

ภาพที่ 28 หน้าการอัปเดตสินค้า



New Product

Name:	<input type="text"/>
Quantity:	<input type="text"/>
Price:	<input type="text" value="0.0"/>
Category:	Food Deg ▼

[Save Product](#)
[Back](#)

ภาพที่ 29 หน้าการเพิ่มสินค้าใหม่




Customer List

ID	Name	telephone	email	Adress_Id	Edit	Delete
1	khetnaja	0643149634	tharakon@kkunmail.com	com.example.demo.address@2286aab5	Edit	Delete
2	ssssssssss	76707667	gheighiublebf	com.example.demo.address@2286aab5	Edit	Delete
3	thabakg	25515165	0643149823	com.example.demo.address@2286aab5	Edit	Delete

[Back](#)
[Add new](#)

ภาพที่ 30 หน้าแสดง Customer List



User Updating

Name
khetnaja

Email
thanakon@kkunmail.com

Telephone
0643149834

ภาพที่ 31 หน้าการอัปเดต User

อ้างอิง

xuchen, Liu, and Li chaoyu. Design and Implementation of a Spring Boot-Based Data Collection System. *2020 12th International*

Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Aug. 2020,

<https://doi.org/10.1109/ihmsc49165.2020.00059>. Accessed 30 Sept. 2022.

Baeldung. "A Simple E-Commerce Implementation with Spring | Baeldung." *Baeldung*, 6 Aug. 2018, www.baeldung.com/spring-angular-e-commerce.

Spring. "Spring Projects." *Spring.io*, 2019, spring.io/projects/spring-boot.