

โปรเจ็ค

เว็บไซต์สะสมภาพ

รายชื่อสมาชิก

623020465-2 นายชนม์สวัสดิ์ นาคนาม

623020499-5 นางสาวสินีนานู เทียกมูล

623020459-7 นางสาวกนกพร แพงไทย

Requirement

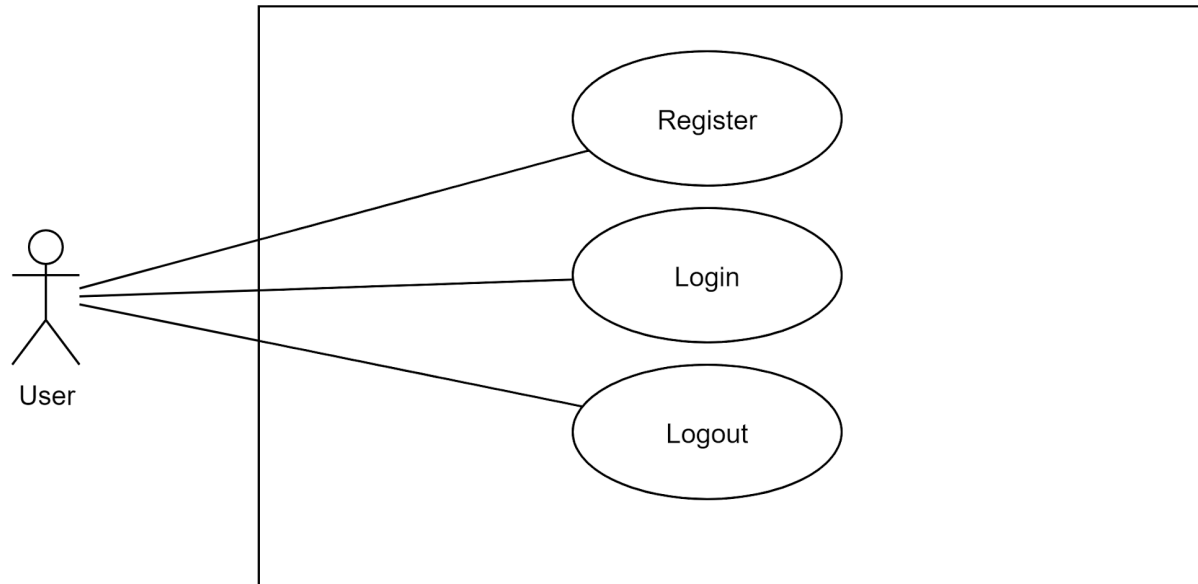
- มีการสมัครสมาชิกใหม่
- มีการลงชื่อเข้าใช้
- มีการออกจากระบบ
- ลูกค้าสามารถเลือกดูภาพที่จะสะสมได้
- ลูกค้าสามารถซื้อภาพมาสะสมได้
- ลูกค้าสามารถดูใบชำระเงินของภาพตอนซื้อได้
- ลูกค้าสามารถดูภาพที่สะสมไว้ได้

สารบัญ

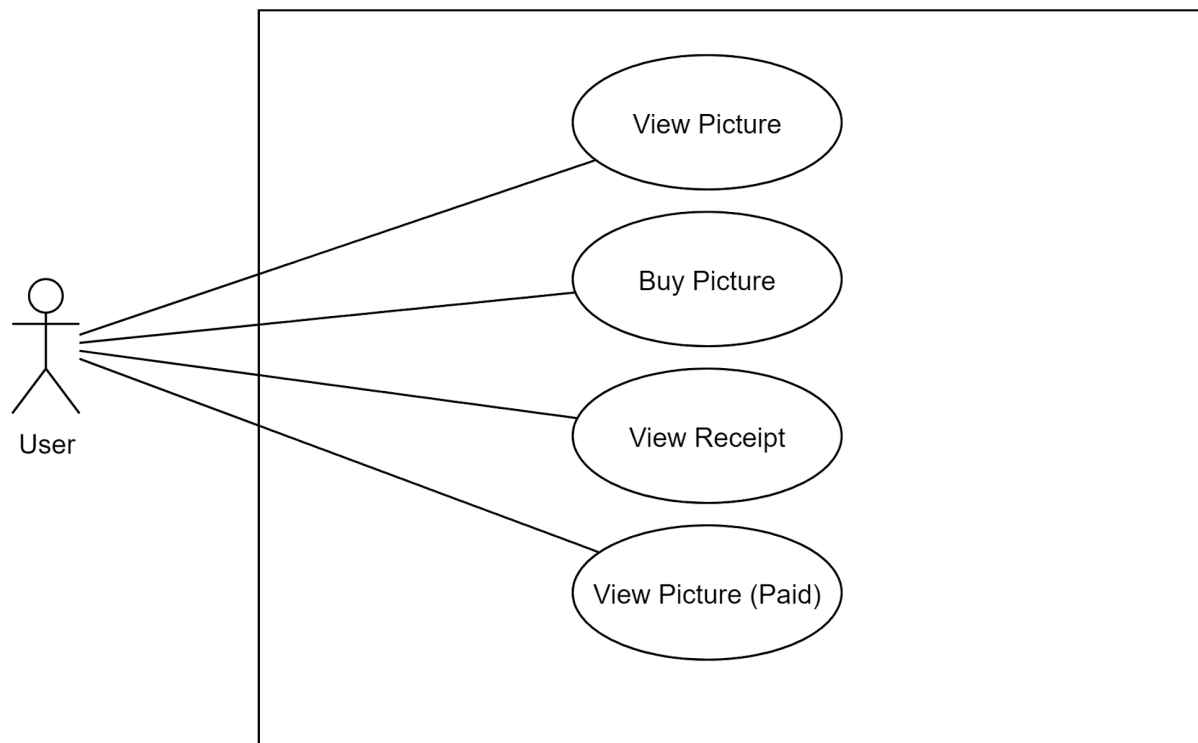
- [Use case Diagram](#)
- [ER Diagram](#)
- [Class Diagrams](#)
- [User Interface](#)
- [Explain Pattern](#)

Use case

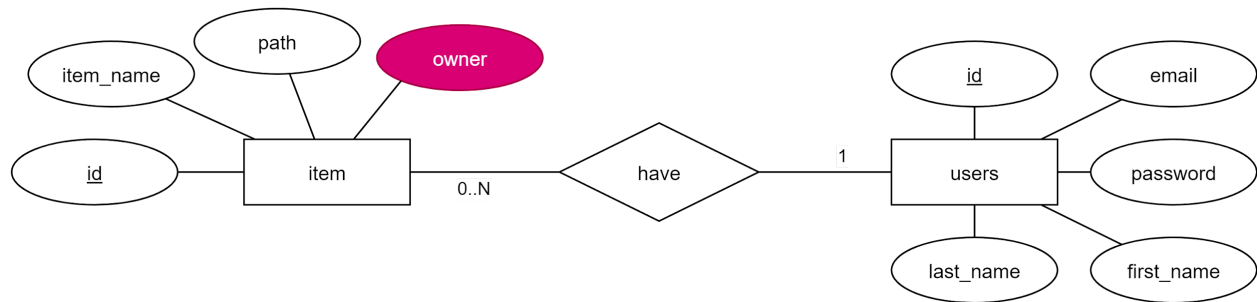
ระบบลงทะเบียนผู้ใช้ใหม่



ระบบซื้อภาพสะสม

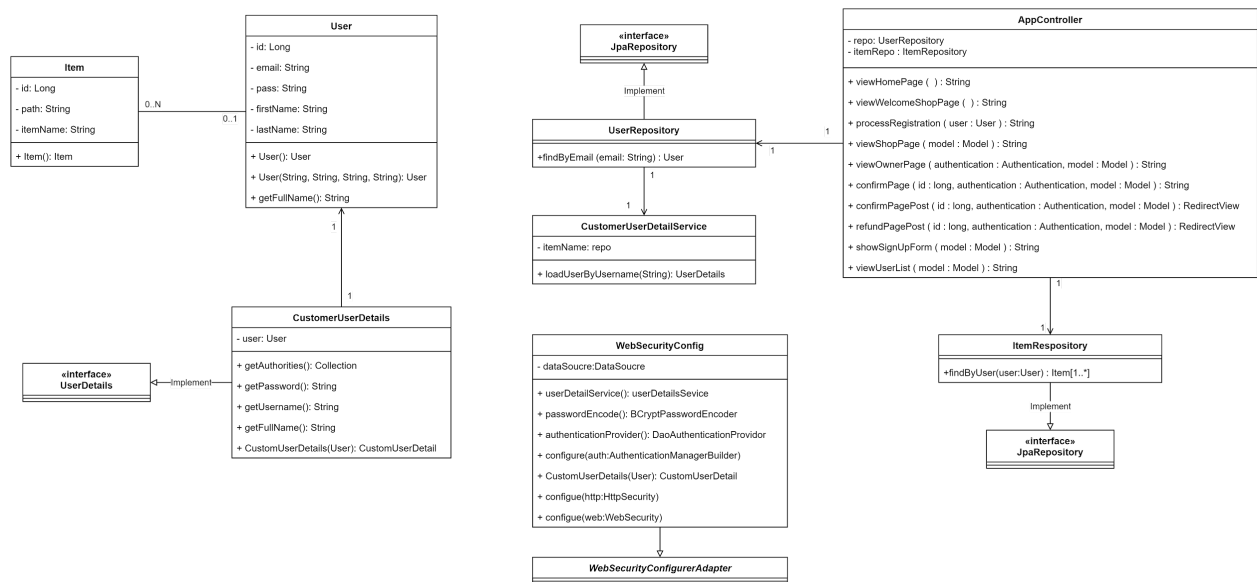


ER Diagram



Owner คือ FK จาก Item ไปที่ User: id

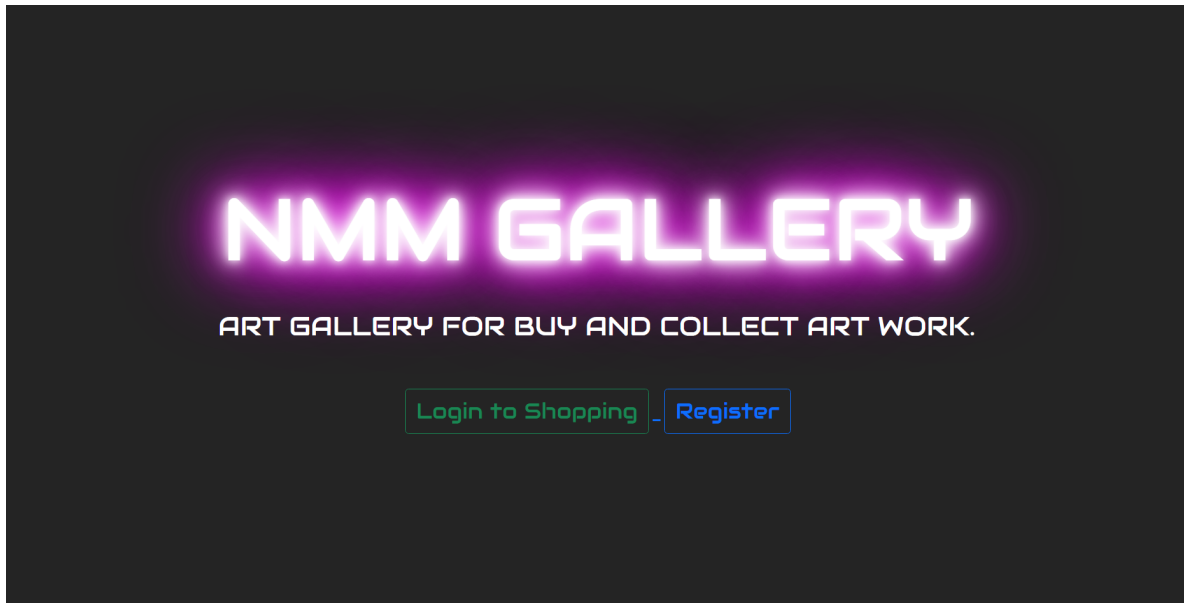
Class Diagram



User Interface

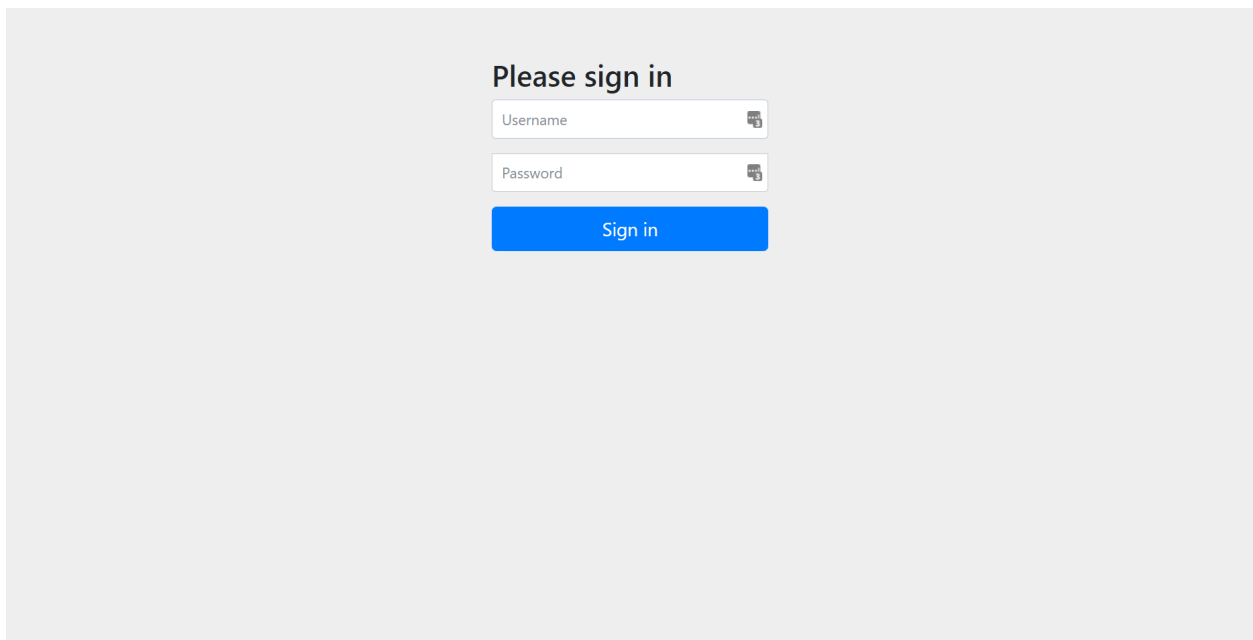
Home

- เป็นหน้าแรกที่จะต้องเลือกว่าเป็นสมาชิกใหม่หรือเข้าสู่ระบบ



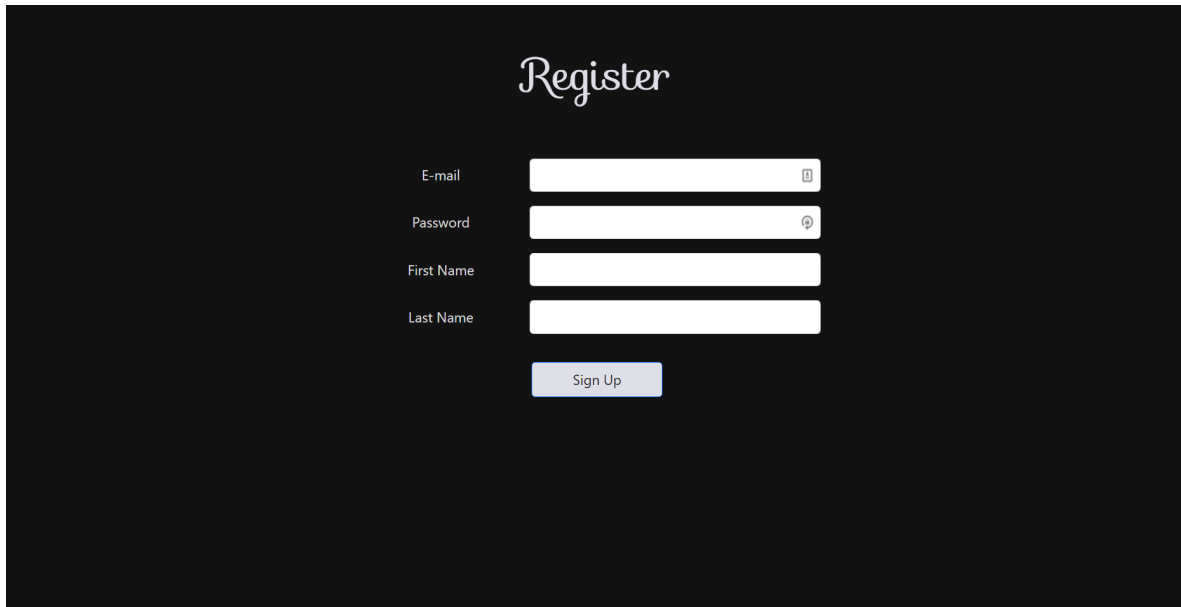
Login

- ลงชื่อเข้าใช้



Register

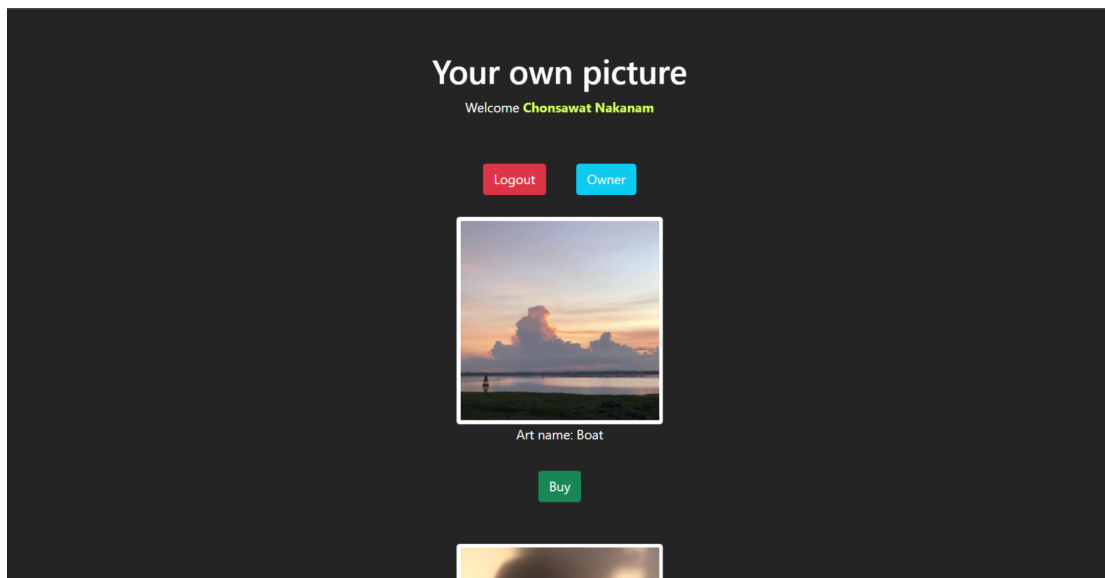
- สมัครสมาชิกใหม่



A screenshot of a registration form titled "Register" in a white serif font on a dark background. The form contains four input fields: "E-mail", "Password", "First Name", and "Last Name". Each field has a small icon on its right side (an '@' for email, a key for password, and a person for names). Below the fields is a light blue "Sign Up" button.

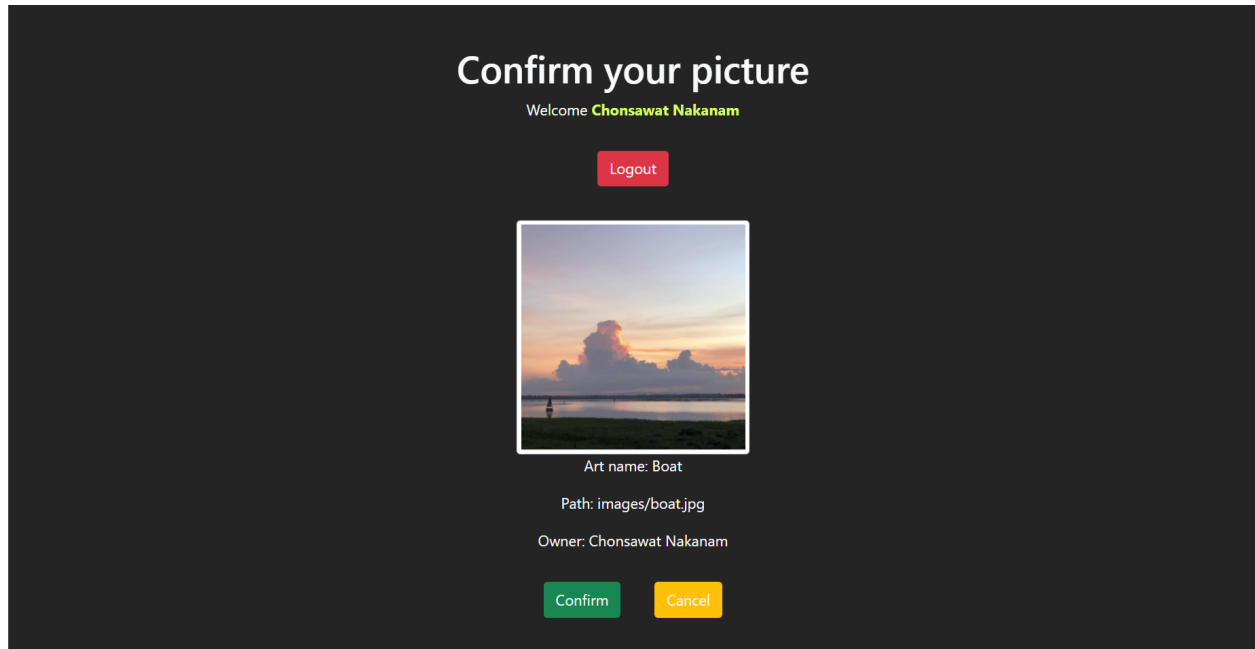
Shop

- จะเข้าได้หลังจากล็อกอินแล้ว สามารถซื้อภาพเพื่อสะสมได้



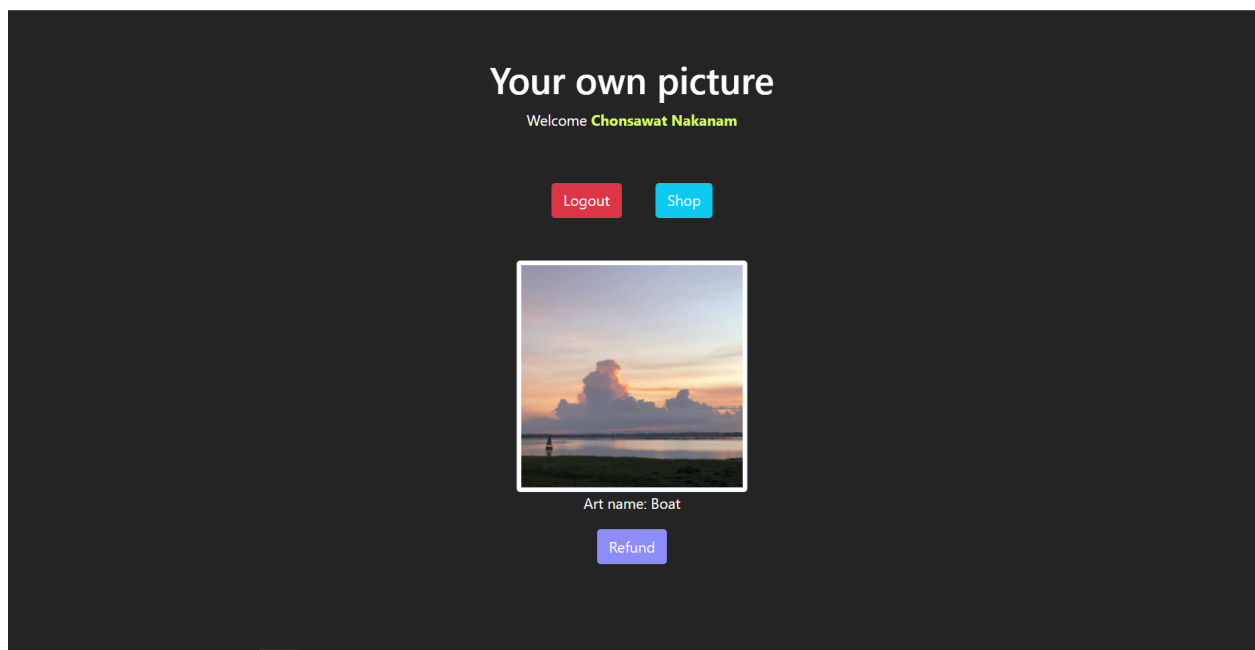
Confirm

- ยืนยันการซื้อภาพจาก Shop และส่งไปยังหน้าภาพที่ผู้ใช้สะสม



Owned

- ดูภาพที่เคยซื้อจาก Shop ได้ (ภาพที่ซื้อแล้วจะไม่แสดงผลใน Shop อีกต่อไป)



Explain Pattern

SOLID คือ

S: Single-Responsibility principle

เขียน Class ให้มีหน้าที่เดียว

O: Open/Closed principle

ต้องเปิดต่อการขยาย แต่ปิดต่อการแก้ไข

L: Liskov Substitution principle

คลาสย่อยจะต้องสามารถแทนที่สำหรับคลาสหลักของตัวเอง

I: Interface Segregation principle

ไม่ควรบังคับให้ implement interface โดยที่ไม่ได้ใช้ implement นั้น

D: Dependency Inversion principle

โมดูลระดับสูง (Abstraction) จะไม่ขึ้นกับโมดูลระดับล่าง (Concretion)

Spring Annotation ที่ผู้จัดทำได้ใช้ในโปรเจกต์นี้คือ

@Autowired

ทำหน้าที่ให้เราสามารถ "Inject" object ขึ้นมาใช้งานได้ โดย object ดังกล่าวจะต้องถูกประกาศไว้ก่อนหน้า และเป็น object ที่มีความเป็น "Spring Bean"

@GetMapping @PostMapping และ @RequestMapping

ทำหน้าที่รับ Path ของเว็บมาเข้า Method ที่ใช้ @GetMapping หรือ @PostMapping ถ้าเป็น Get สามารถใช้ @PathVariable เพื่อรับค่าจาก Path มาได้

@Controller

ทำหน้าที่เป็นชั้นของหน้าบ้านเช่น Spring MVC ที่เล่นกับหน้าจอ หรือรับส่งค่าผ่าน Web Service

@Entity

ทำหน้าที่ประกาศว่า Class นั้นจะถูกนำไปสร้างเป็น Table ในฐานข้อมูล

@Table

ทำหน้าที่ในการตั้งชื่อ Table ที่มีการใช้ **@Entity** ได้

@Id

ทำหน้าที่นำ Field ที่ถูกประกาศไปทำเป็น Primary Key ของ Attribute นั้น

@GeneratedValue

ทำหน้าที่กำหนดค่าเบื้องต้นให้กับ Attribute ที่ใช้ **@GeneratedValue** บน Field นั้นๆ

@ManyToOne

ทำหน้าที่จับคู่ความสัมพันธ์ 1 - M ระหว่างผ่าน Class Field นั้นๆ

@JoinColumn

ทำหน้าที่ตั้งชื่อคอลัมน์ในฐานข้อมูลจาก Field ที่ใช้ Annotation เกี่ยวกับความสัมพันธ์เช่น **@ManyToOne** เป็นต้น

ทุก Annotation ที่ใช้ มีการใช้หลักการ SOLID ในการออกแบบ