

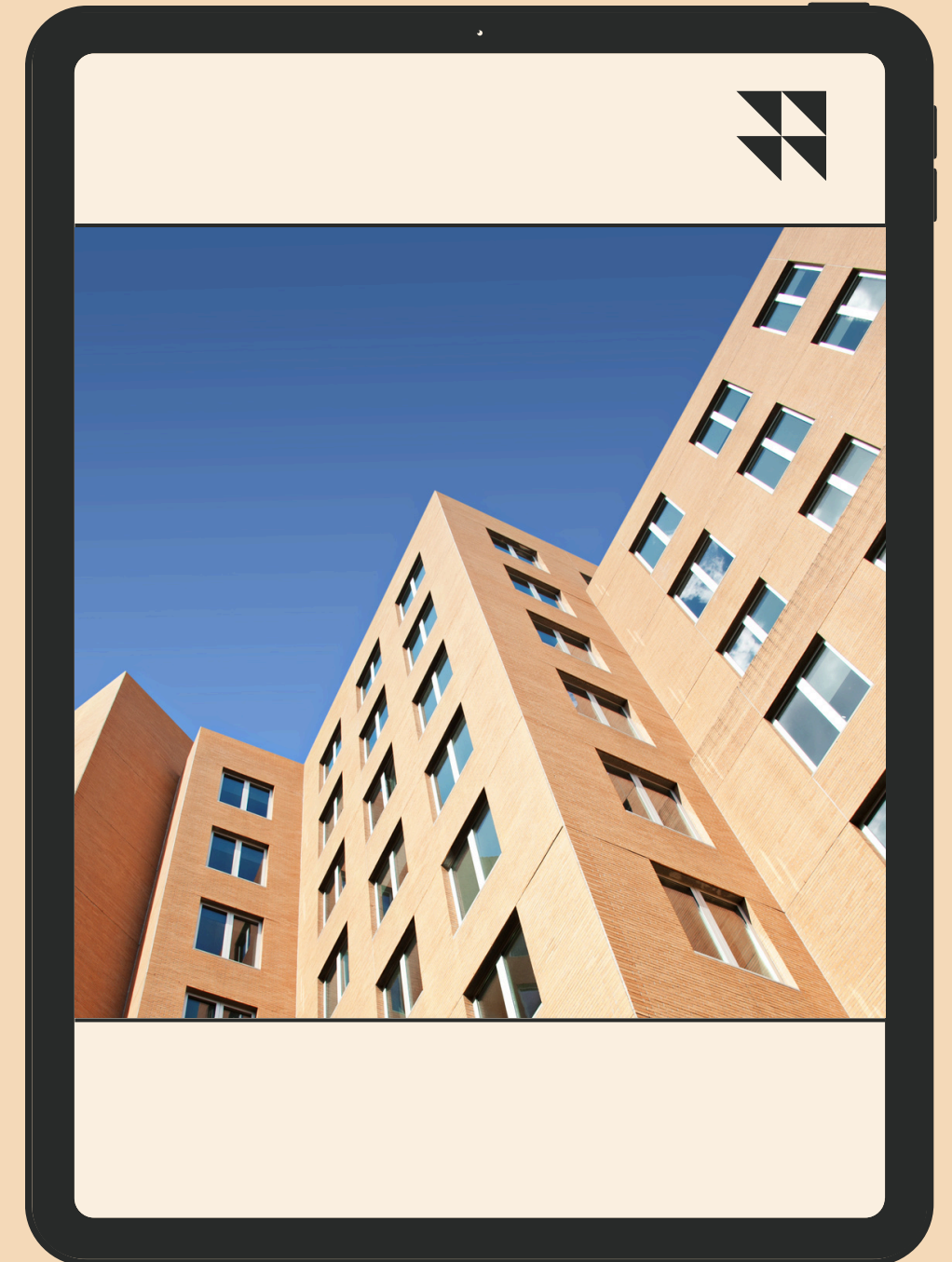
ระบบจัดการ ข้อมูลหอพัก



ที่มาและความสำคัญ

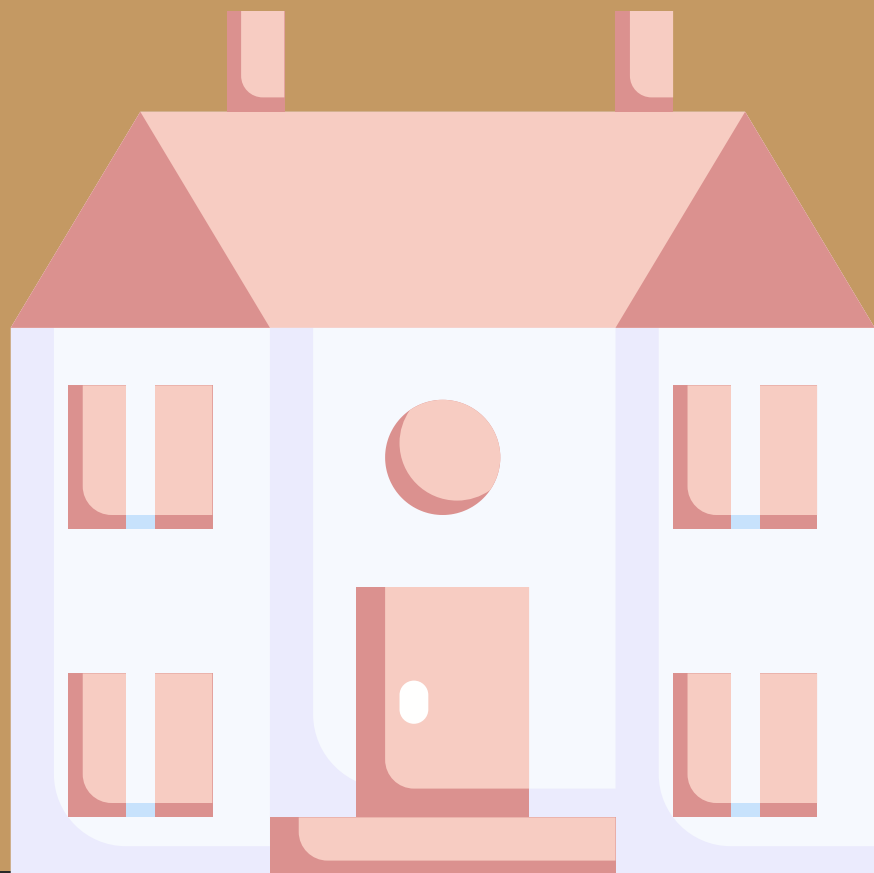
ในปัจจุบันมีธุรกิจหอพักเกิดขึ้นจำนวนมาก เพื่อเพิ่มความสะดวกสบายและประสิทธิภาพในการจัดการหอพักของผู้ประกอบการหอพัก โดยเว็บไซต์นี้จะช่วยจัดการและตรวจสอบข้อมูลของผู้เข้าพักได้อย่างสะดวกและรวดเร็ว

นอกจากนี้ผู้ประกอบการหอพักสามารถตรวจสอบข้อมูลเชิงสถิติและข้อมูลต่างๆ เพื่อปรับปรุงการบริหารจัดการหอพักให้ดียิ่งขึ้นและช่วยเพิ่มพูนความสำคัญในเรื่องความปลอดภัยของผู้เข้าพักได้อีกด้วย



ความต้องการ ของระบบ

- 1 จัดทำเว็บไซต์สำหรับจัดเก็บข้อมูลห้องพักสำหรับผู้ประกอบการห้องพักและผู้เข้าพัก
- 2 ข้อมูลของผู้เข้าพัก เช่น รหัสบัตรประชาชน, ชื่อ, อายุ, อีเมลล์, ที่อยู่, เบอร์โทรศัพท์, รายละเอียดของห้องพักที่จอง, รายละเอียดการชำระเงิน, และสถานะการเข้าพัก
- 3 ระบบการจัดการห้องพัก ที่ผู้ประกอบการสามารถดูข้อมูลใบเสร็จ เพิ่มบิล เพิ่มชื่อลูกค้า จัดการบิล จัดการสัญญา และผู้เข้าพักสามารถตรวจสอบใบเสร็จประจำเดือนได้



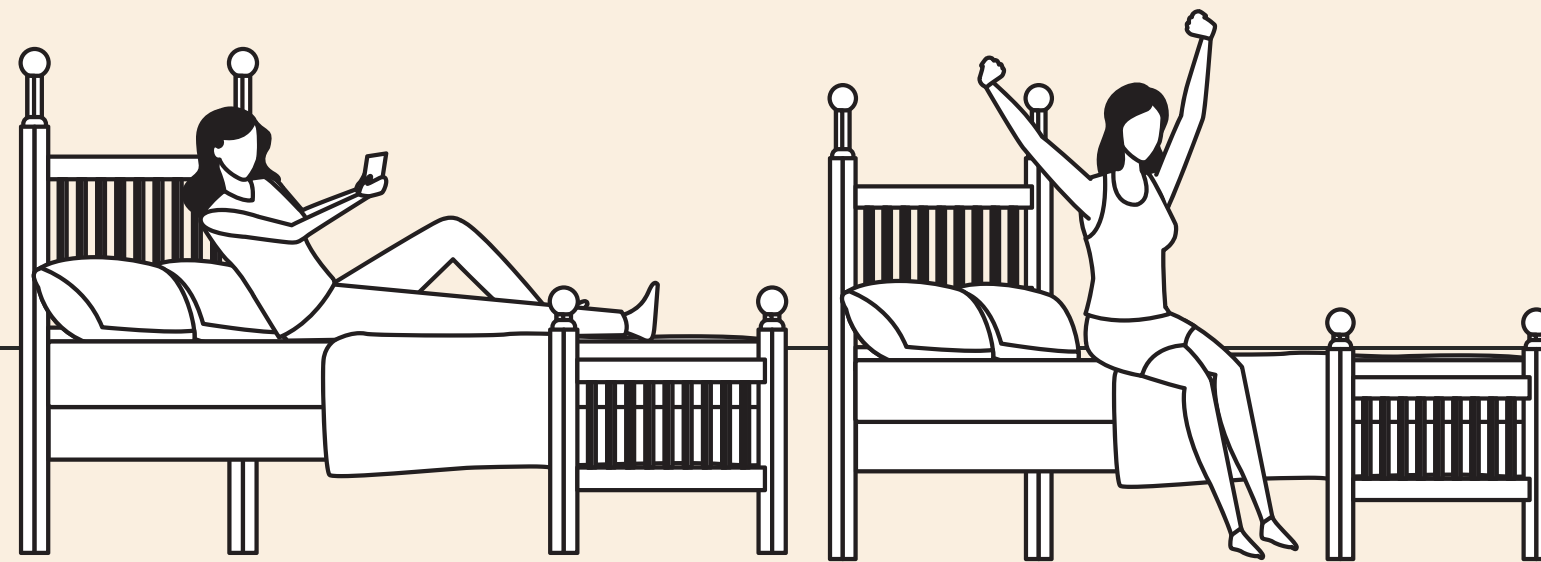
วัตถุประสงค์



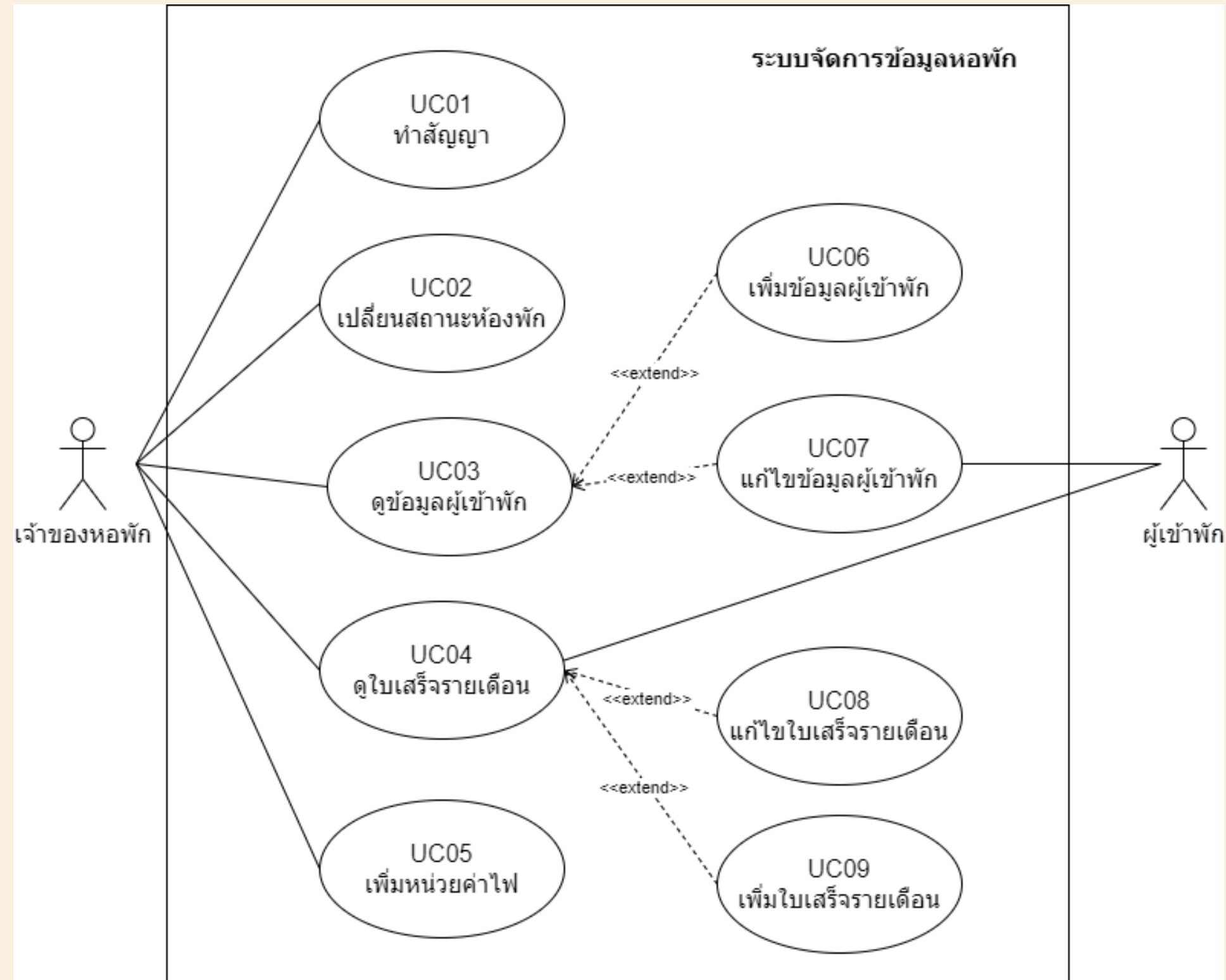
เพื่อพัฒนาเว็บไซต์ที่ใช้ในการจัดการข้อมูลสำหรับผู้ประกอบการหอพัก

ประโยชน์ที่คาดว่าจะได้รับ

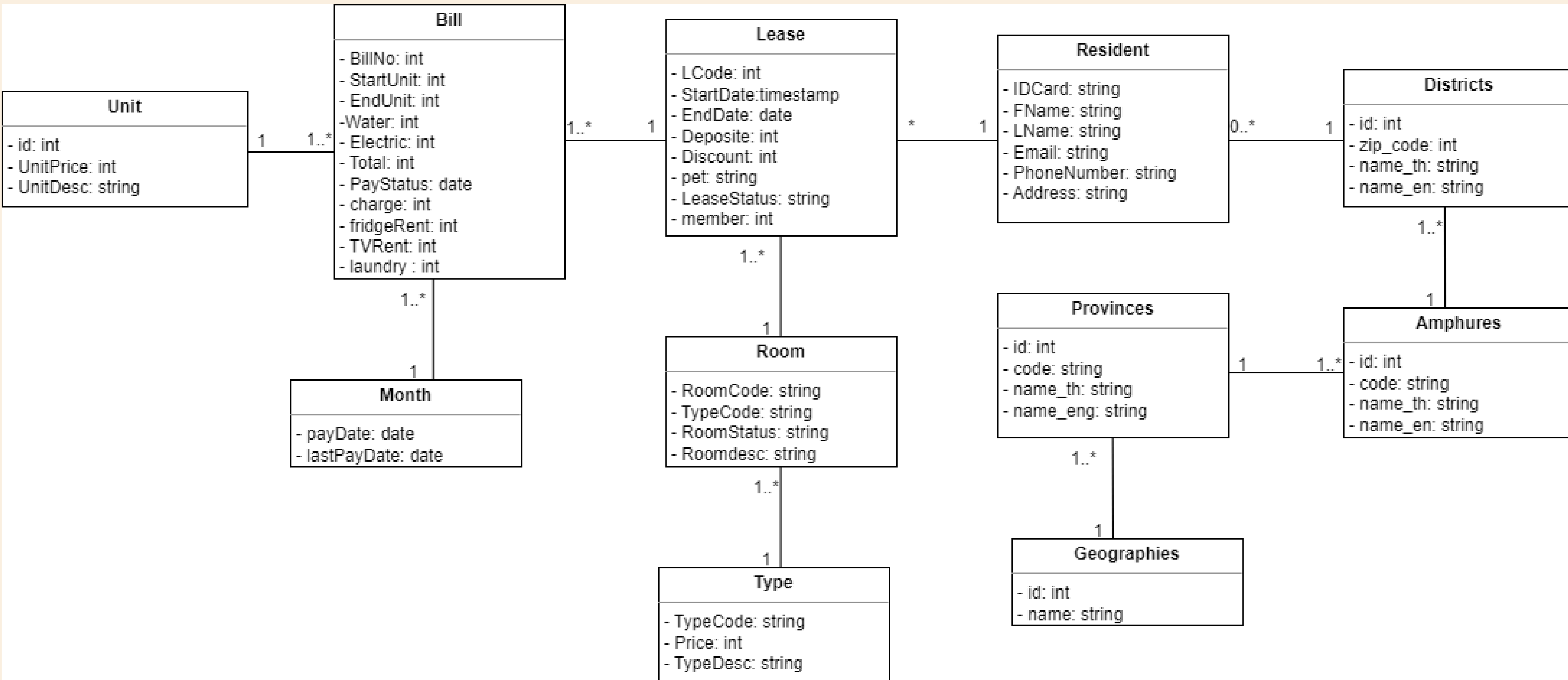
- 3.1 ผู้ประกอบการและผู้เข้าพักจะได้รับความสะดวกสบายในการตรวจสอบเรื่องต่างๆในห้องพัก
- 3.2 ช่วยประหยัดเวลาและค่าใช้จ่ายของผู้ประกอบการหอพัก โดยเฉพาะเมื่อต้องการตรวจสอบข้อมูลหลายๆ อย่างพร้อมกัน
- 3.3 ช่วยลดความผิดพลาดในการจัดการหอพัก โดยเฉพาะอย่างยิ่งเมื่อมีการจัดการหลายๆ อย่างพร้อมกัน
- 3.4 ช่วยเพิ่มความโปร่งใสในการจัดการหอพัก โดยทำให้ผู้ประกอบการหอพักสามารถตรวจสอบข้อมูลเกี่ยวกับหอพักย้อนหลังได้อย่างชัดเจนและถูกต้อง



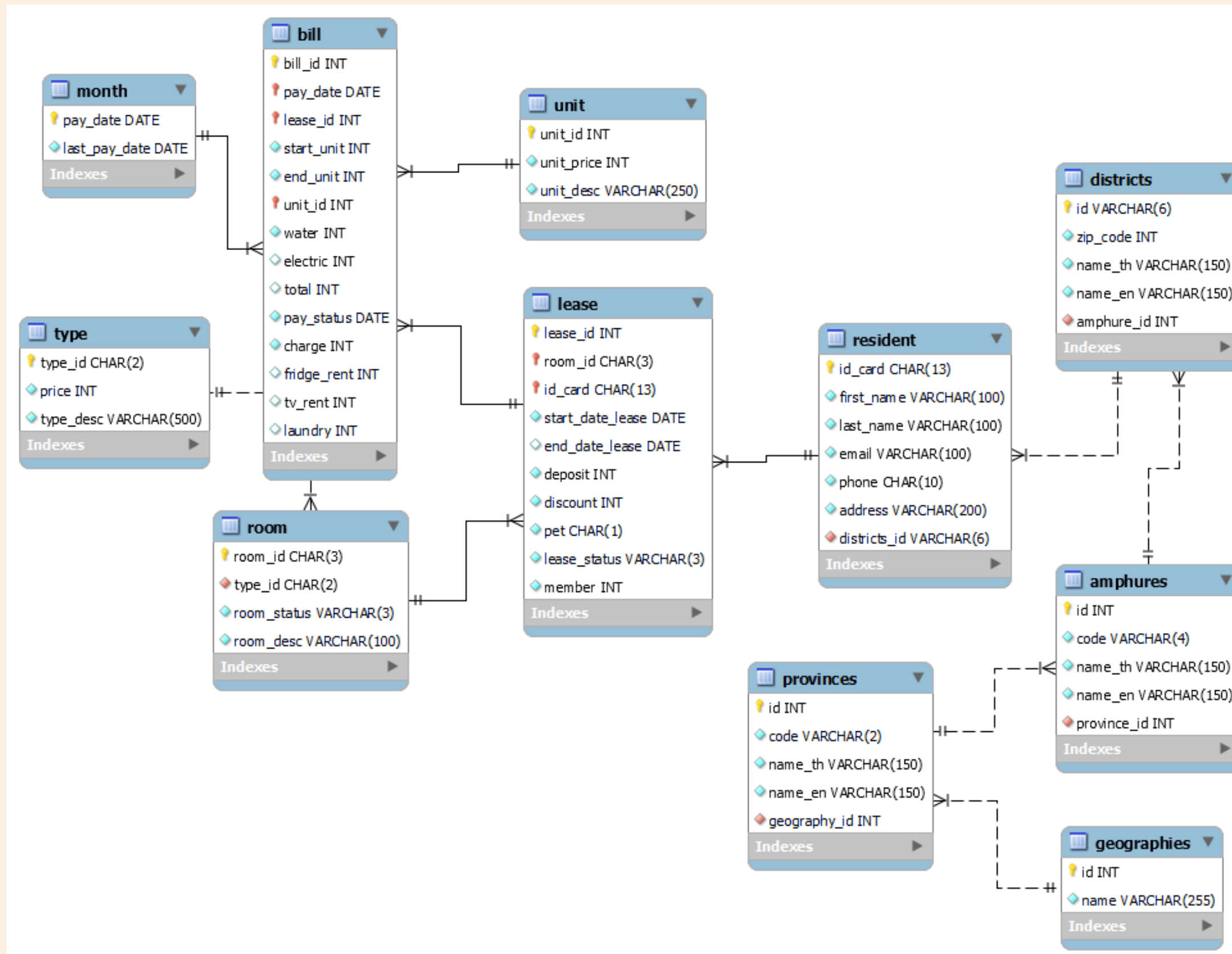
แผนภาพยูสเคส (Use case diagram)



แผนภาพคลาสโดเมน (Domain class diagram)



แผนภาพอีอาร์ (ER Diagram)



การนำหลัก SOLID / Pattern

มาประยุกต์กับระบบ



Single Responsibility Principle (SRP)

- คลาส ResidentService มีหน้าที่เดียวคือจัดการกับข้อมูลของผู้เช่าคลาส
- ResidentRepository มีหน้าที่เดียวคือจัดเก็บข้อมูลของผู้เช่า
- คลาส ResidentController มีหน้าที่เดียวคือควบคุมการทำงานที่เกี่ยวข้องกับผู้เช่า เช่น การเพิ่ม แก้ไข และลบผู้เช่า

Open/Closed Principle (OCP)

คลาส ResidentService ใช้ interface ResidentRepository เพื่อเข้าถึงข้อมูลของผู้เช่า คลาสนี้ออกแบบให้สามารถรองรับการเปลี่ยนแปลงการจัดเก็บข้อมูลของผู้เช่าได้โดยไม่ต้องแก้ไขโค้ดของคลาส ResidentService

Interface Segregation Principle (ISP)

อินเตอร์เฟซ ResidentRepository มีเพียงหน้าที่ที่จำเป็นสำหรับคลาส ResidentService เท่านั้น

Dependency Inversion Principle (DIP)

- คลาส ResidentService ขึ้นอยู่กับ interface ResidentRepository ไม่ใช่คลาส ResidentRepository โดยตรง
- คลาส ResidentController ขึ้นอยู่กับ interface ResidentService ไม่ใช่คลาส ResidentService โดยตรง

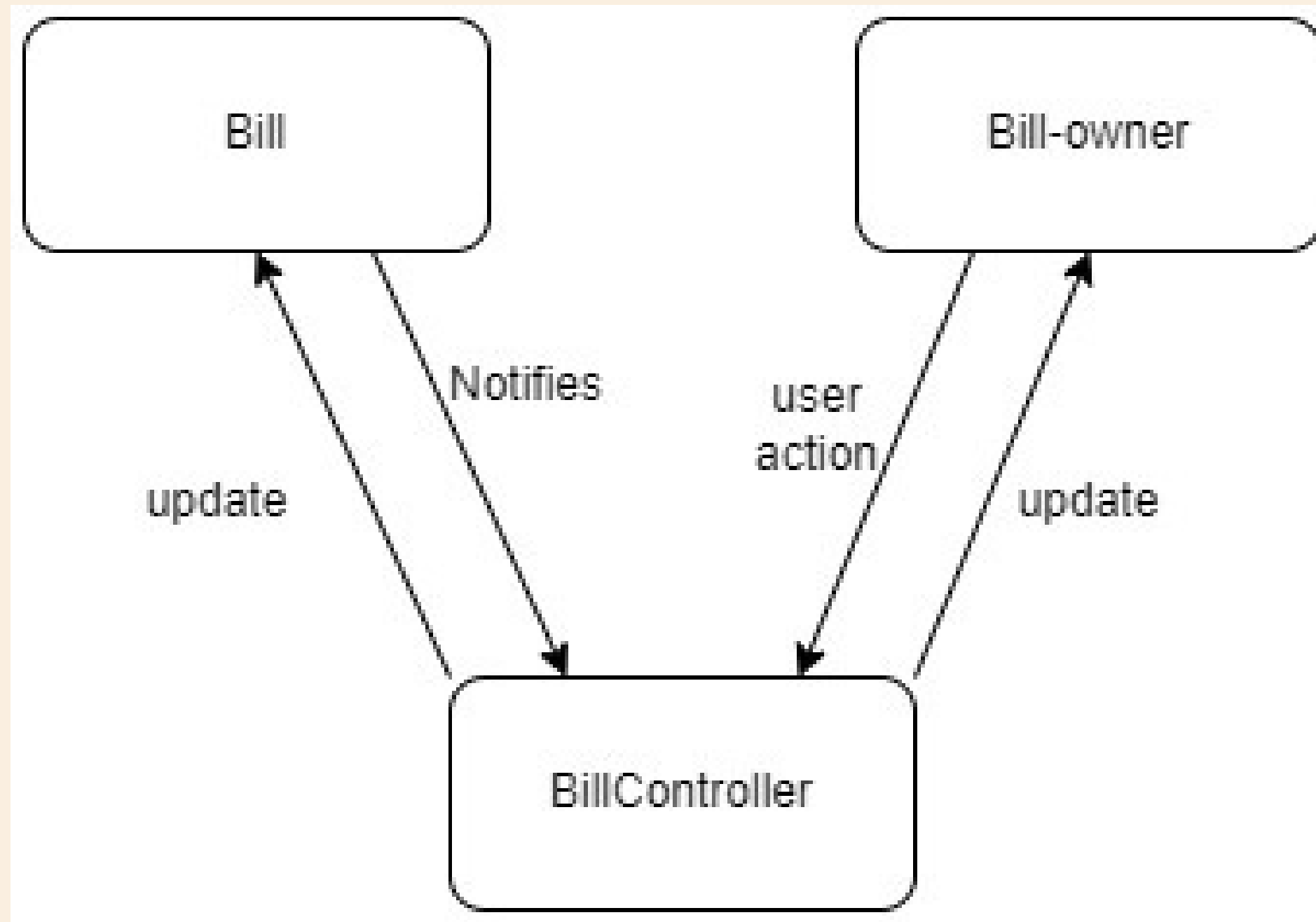
Repository Pattern

คลาส ResidentRepository ใช้ Repository Pattern เพื่อแยกชั้นการจัดเก็บข้อมูล
ออกจากชั้นการให้บริการข้อมูล

Service Pattern

คลาส ResidentService ใช้ Service Pattern เพื่อแยกชั้นการให้บริการข้อมูล
ออกจากชั้นการควบคุมการทำงาน

Model-View-Controller(MVC)



THANKYOU

