

รายงาน

Final Project

จัดทำโดย

กลุ่ม 6 เจ้าหญิงกนิษฐา กับคนแคะทั้ง 9

สมาชิก

1. 61010014 กนิษฐา ศศิปรียจันทร์
2. 63010035 กฤษฎาง แสงศิริผล
3. 63010212 ชวณ ใบดำรงศักดิ์
4. 63010235 ชินาธิป มีสวนนิล
5. 63010255 ฐนกร วิหกกุล
6. 63010283 ณภัทร มูลพินิจ
7. 63010320 ณัฐนันท์ เรืองพานิช
8. 63010326 ณัฐพงษ์ นาคสามัคคี
9. 63010956 สฤษฎ์พัฒน์ นิยมทอง
10. 63011013 สุเมธ สอนสำราญ

เสนอ

ดร.ปริญญา เอกปริญญา

รายงานนี้เป็นส่วนหนึ่งของวิชา SOFTWARE ARCHITECTURE AND DESIGN

ภาคเรียนที่ 1 ปีการศึกษา 2565

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## Proposal (Functionality)

### ที่มาและความสำคัญ

ในโลกปัจจุบันเทคโนโลยีมีความสำคัญในชีวิตประจำวันเป็นอย่างมาก และการเริ่มต้นการใช้ชีวิตในรั้วมหาวิทยาลัยนับเป็นก้าวสำคัญของคนหลายคน ทำให้คณะผู้จัดทำได้มองเห็นถึงความสำคัญ และจากที่ทางคณะผู้จัดทำได้ประสบ พบเจอปัญหาต่างๆในการใช้ชีวิตในรั้วมหาวิทยาลัย โดยเฉพาะอย่างยิ่งการใช้ชีวิตในรั้วของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อย่างเช่นปัญหาในการเลือกวิชาศึกษาทั่วไปในการลงทะเบียนเรียน ปัญหาในการเลือกหอพัก และปัญหาอื่นๆอีกมากมาย ทำให้คณะผู้จัดทำได้วางแผนและลงมือหาวิธีในการแก้ปัญหาต่างๆเหล่านี้ ด้วยการนำเทคโนโลยีมาเป็นเครื่องมือในการแก้ปัญหา และในปัจจุบันที่อินเทอร์เน็ตเป็นสิ่งที่ทุกคนสามารถเข้าถึงได้ตลอดเวลาทำให้คณะผู้จัดทำได้เลือกการสร้างเว็บแอปพลิเคชันที่สามารถแก้ปัญหาได้และเข้าถึงง่ายต่อผู้ใช้งานเป็นแนวทางในการแก้ปัญหา

### วัตถุประสงค์

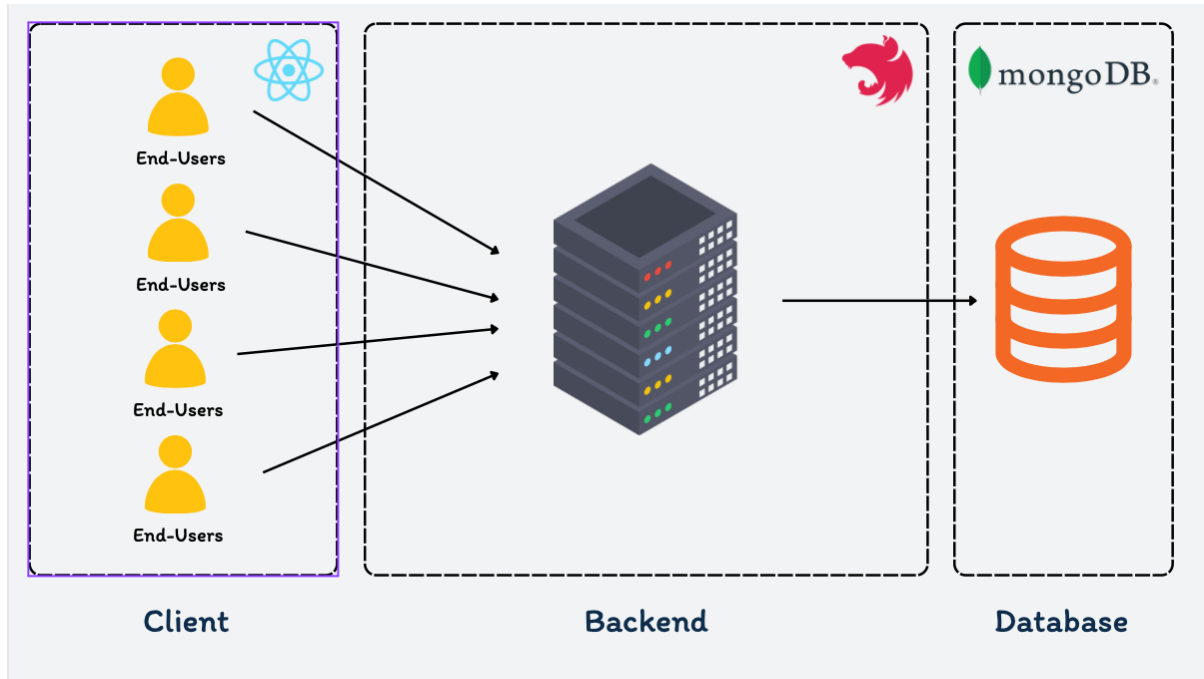
1. เพื่อแก้ปัญหาสำหรับผู้ที่ยังตัดสินใจลงวิชาเรียนไม่ได้ หรืออยากเห็นความคิดเห็นของผู้อื่นเพื่อเป็นการประกอบการตัดสินใจในการลงวิชาเรียน
2. เพื่ออำนวยความสะดวกในการหาหอพักที่เหมาะสม และถูกใจให้นักศึกษาใหม่หรือนักศึกษาที่จะตัดสินใจเลือกหอพัก
3. เพื่อให้นักศึกษาเข้าใจบทเรียนในวิชาต่างๆมากขึ้นผ่านแพลตฟอร์มสำหรับการรวบรวมแหล่งเนื้อหาข้อมูลบทความ สรุปเนื้อหาวิชาเรียนของนักศึกษา
4. เพื่อแก้ปัญหานักศึกษาหาหอพักไม่ได้, ได้หอที่ไม่ถูกใจ หรือบางคนเพิ่งมาเจอหอที่ถูกใจหลังทำสัญญาเช่ากับอีกที่ไปแล้ว เป็นต้น

## แนวทางในการแก้ปัญหา

1. Platform เป็น Line Official ที่เข้าถึงง่ายและสะดวกต่อ ผู้ใช้ที่เป็นนักศึกษา
2. Feature ที่ทำให้นักศึกษาหาหอพักที่ตรงกับความต้องการได้สะดวกมากยิ่งขึ้น เพียงแค่กรอกความต้องการลงไประบบก็จะสรรหาหอพักที่ตรงกับความต้องการพร้อม reviews จากผู้ใช้จริงมาให้ทันที
3. Feature ที่สามารถคัดกรองวิชาเลือกที่สามารถลงได้โดยเวลาเรียนและเวลาสอบไม่ชนกับวิชาหลัก
4. Feature ที่สามารถแบ่งปันเนื้อหาหรือบทความสรุปเกี่ยวกับเนื้อหาวิชาเรียน
5. Feature สำหรับการรีวิววิชาเรียน

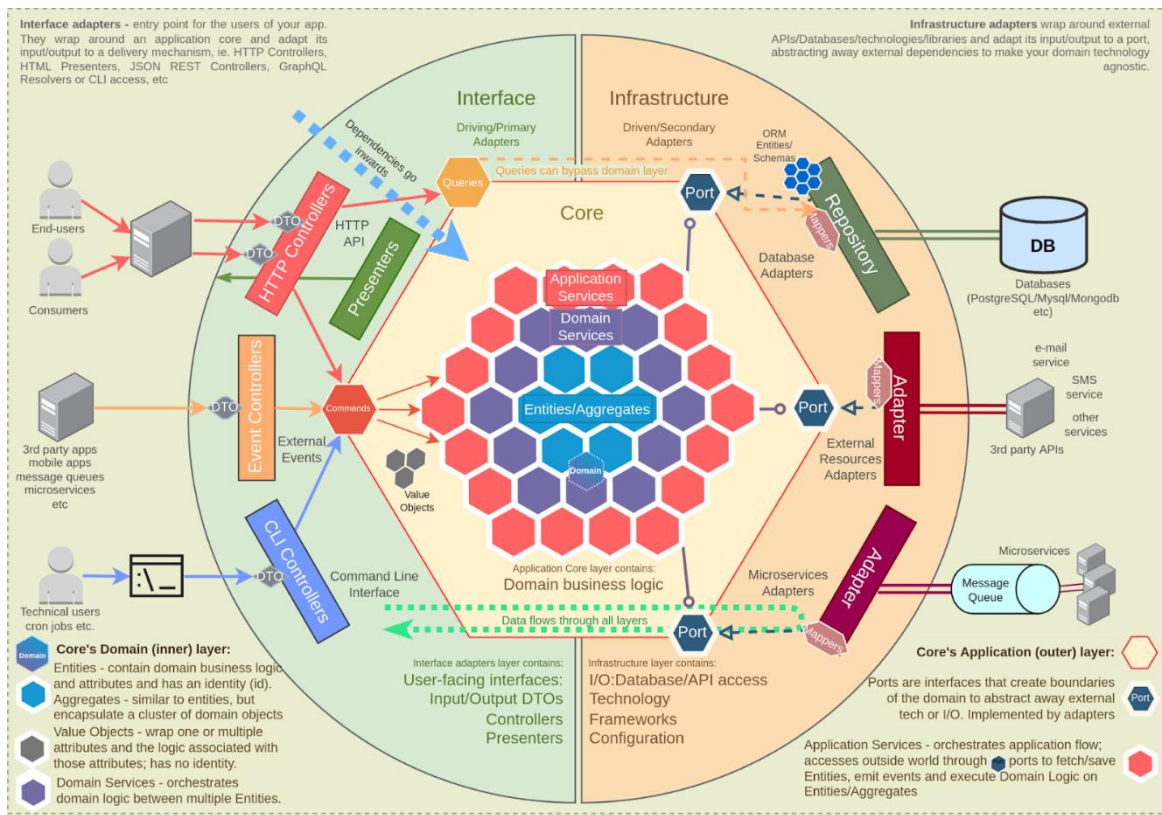
## Software Architecture

### N-tier



จากรูป ทาง client ของทางโปรเจกเราจะแบ่งเป็น 4 เว็บไซต์ ที่แยกออกเป็นแต่ละ feature และจะทำการส่ง request ผ่านไปยังตัว backend server และ backend server จะทำการเข้าถึงและแก้ไขข้อมูลผ่านตัวระบบ database ของทางโปรเจกเรา

## Software Architectural Style



จากรูป ทางคณะผู้จัดทำได้นำ Hexagonal architecture กับ Domain-Driven Design (DDD) มาประยุกต์ใช้ในการออกแบบและพัฒนาซอฟต์แวร์ ซึ่งภายในตัว Hexagonal architecture ก็จะมีการแบ่งส่วนออกเป็น user-side, business logic, และ server-side และภายในก็จะประยุกต์ DDD มาด้วยโดยทำการ implement Entities, Repositories และ Factories ต่างๆ

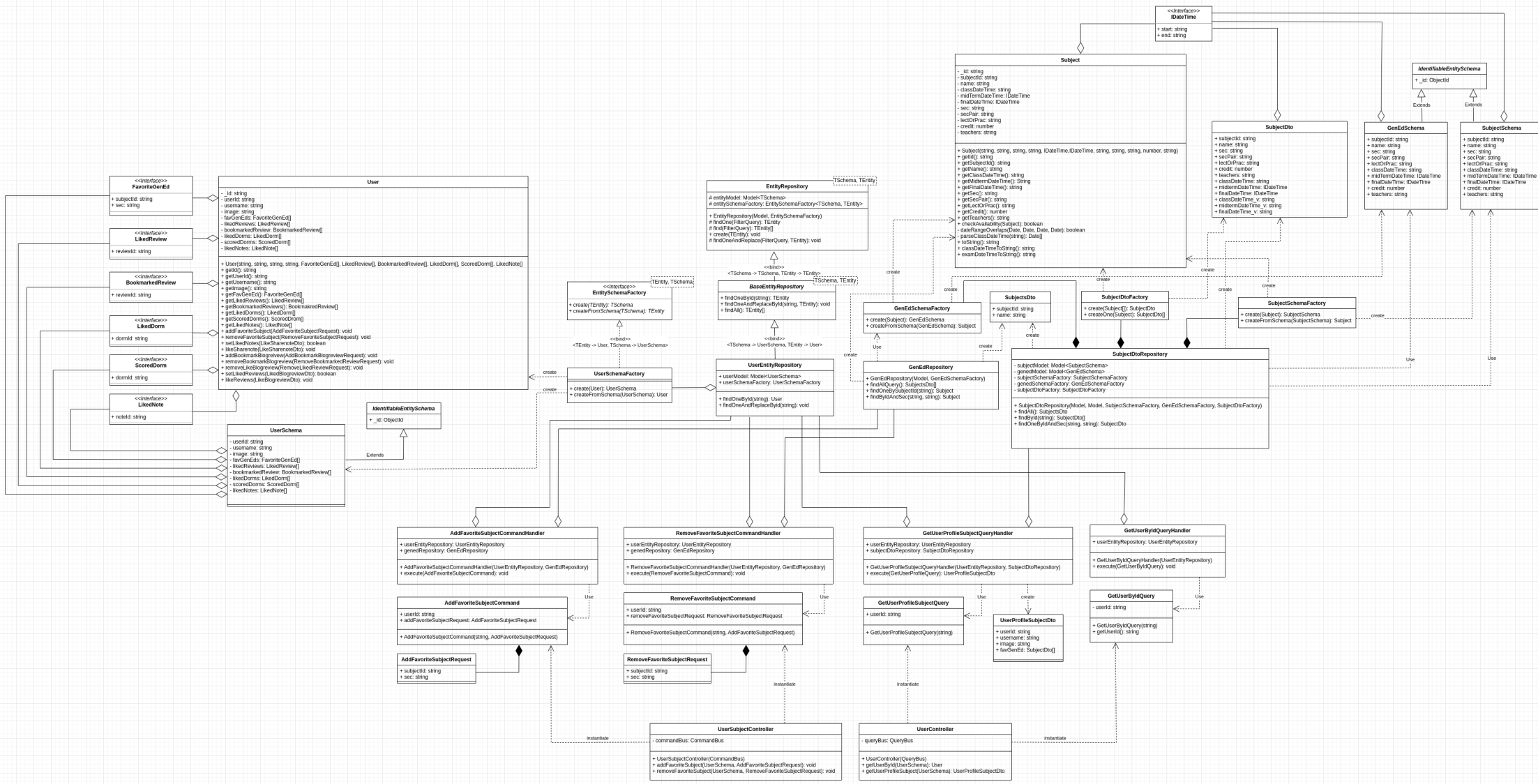
# Software Modeling

# Component diagram

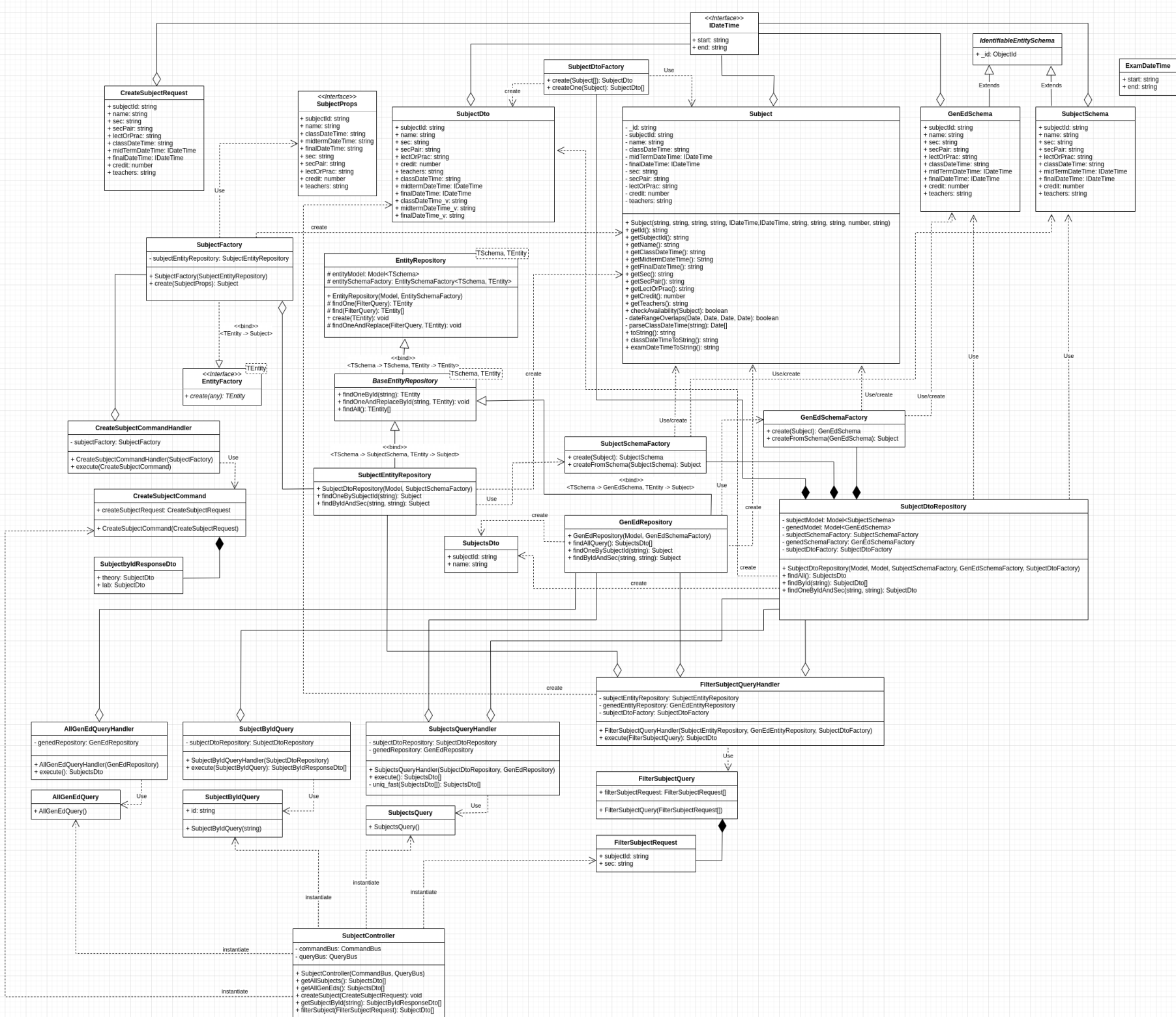




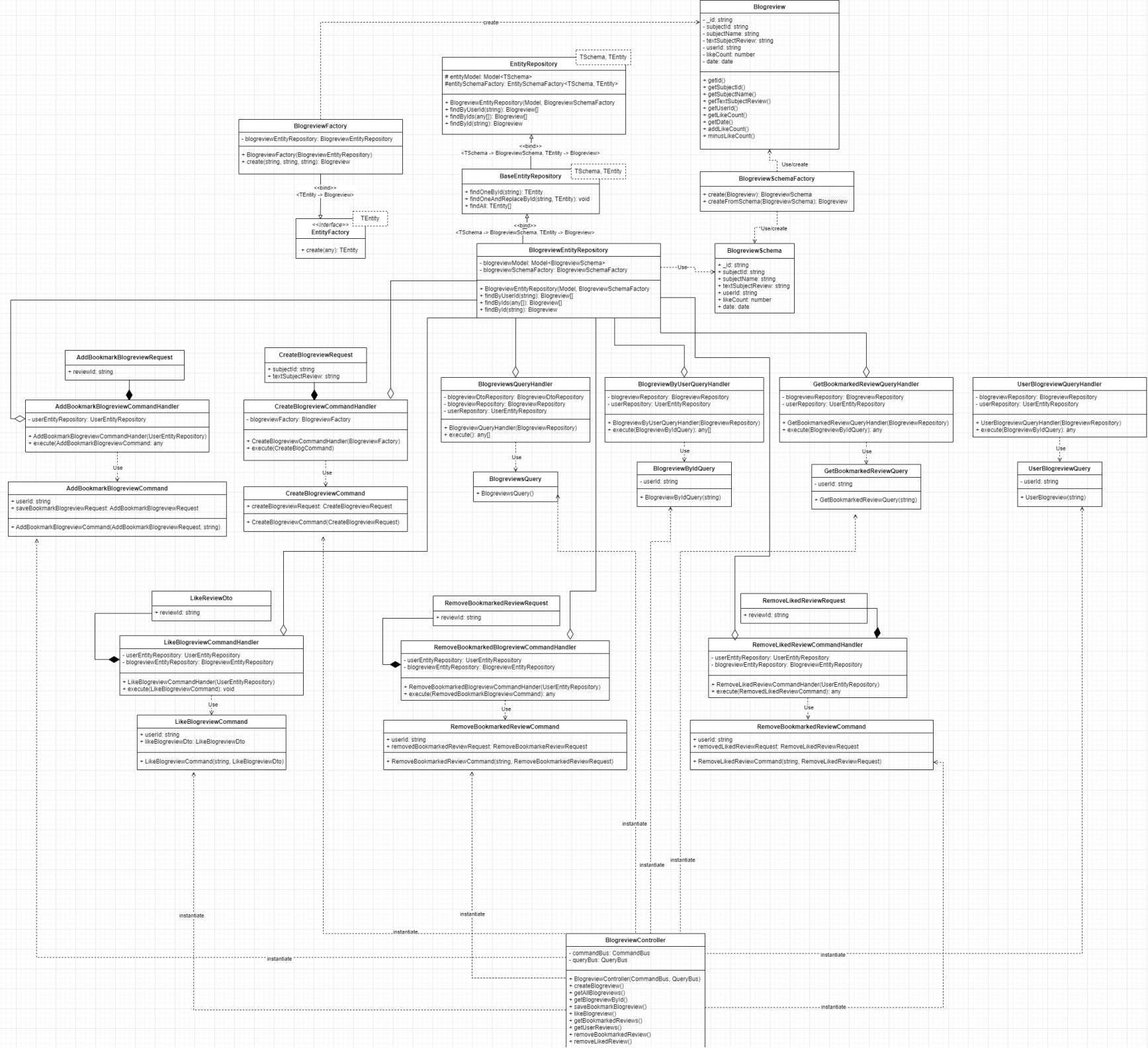
User



Subject



Blog review

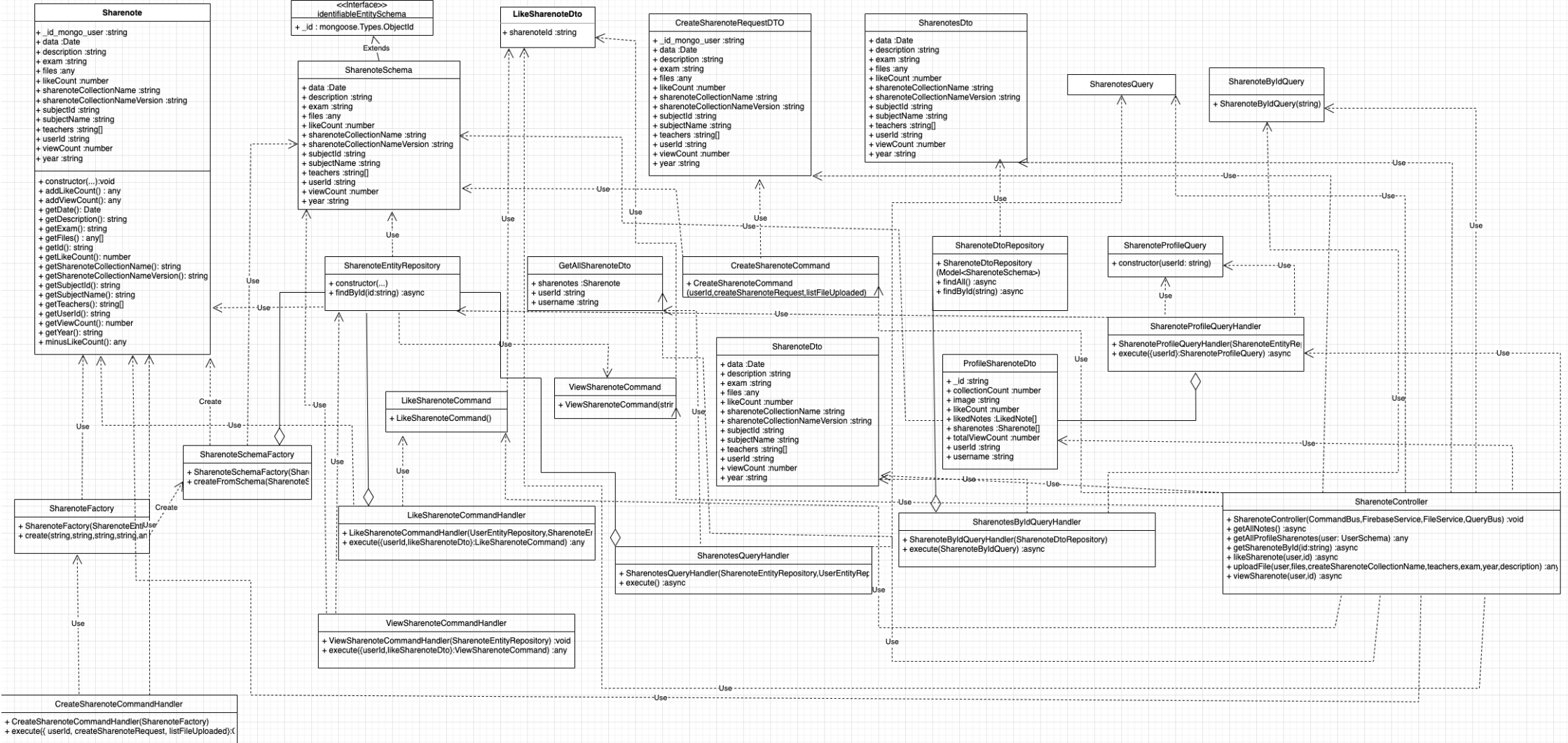


Dorm





Sharing note



## Design Patterns

### Abstract Factory

#### ปัญหาที่แก้ไข

นำมาใช้แก้ไขปัญหาการขยาย Scale ของโค้ดในโปรเจกของเราเมื่อต้องสร้าง entity ใหม่ขึ้นมาจะทำได้ยากเพราะว่าต้องแก้โค้ดทุกครั้ง เนื่องจากการทำงานที่คล้ายๆกัน แต่ได้ตัว product ที่ออกมาไม่เหมือนกัน

#### วิธีนำมาใช้

โดยนำ Abstract Factory มาใช้สร้างตัว EntitySchemaFactory ขึ้นมาเพื่อเป็น Abstract Factory เพื่อให้แต่ละ Schema-Factory ไป implement abstract method ไว้สำหรับสร้างแต่ละ product ยกตัวอย่างเช่นในไฟล์ blogreview-schema.factory.ts, dorm-schema.factory.ts, sharenote-schema.factory, subject-schema.factory.ts, user-schema.factory.ts

### Adapter

#### ปัญหาที่แก้ไข

แก้ไขปัญหาการเกิดโค้ดที่ไม่เกี่ยวข้องกันกับตัว handler ในกรณีที่ต้องมีการจัดการความซับซ้อนของ format ของข้อมูลที่ทาง client ต้องการ

#### วิธีนำมาใช้

สร้างตัว adapter มาเพื่อทำการจัดการโค้ดส่วนที่ต้องเปลี่ยนแปลงไปมาระหว่างตัว Object ที่ได้มาจาก repository กับตัว Object ตัวที่ทาง client ต้องการ

## Command

### ปัญหาที่แก้ไข

เปลี่ยน Action ต่างๆให้กลายเป็น Object Command ทำให้โปรแกรมจัดการ action ที่เข้ามาได้หลายรูปแบบ ยิ่งเรามี Module หรือ Component ที่มีหน้าที่การทำงานที่เยอะมากขึ้น ยิ่งทำให้เกิดความซับซ้อนในการเรียกใช้งานสัก 1 การทำงาน และการแก้ไขหรือการขยายความสามารถของโค้ดก็ทำได้ยาก อาจส่งผลกระทบต่อโค้ดส่วนอื่นภายใน

### วิธีนำมาใช้

นำ Command มาใช้ในแต่ละ Controller ของแต่ละ Module นั้น มี dependencies กับหลาย Class Object ดังนั้น จึงเพิ่ม QueryBus / CommandBus ขึ้นมาทำหน้าที่คล้ายตัว Invoker ที่จะส่ง execute() คำสั่งที่มี dependencies กับ คลาสที่เราต้องการ โดยเราสร้าง Class Object Command ที่มี method ภายในจะคอยเรียกใช้ Class อื่น จากนั้น เมื่อ Controller ต้องการจะ execute( params ) ก็ จะ pass Parameter Class Object Command ลงไป

## Quality Attribute scenarios

### Performance Scenario

Source of stimulus : ผู้ใช้งานหลายๆคน

Stimulus : requests ของแต่ละผู้ใช้งาน

Artifact: ระบบ , เว็บ

Environment : Normal mode (การทำงานปกติ)

Response : ส่ง response ที่ผู้ใช้งานต้องการ

Response Measure : ความเร็วในการได้รับ response ไม่เกิน 2 วินาที

### Security Scenario

Source of stimulus : Unauthor use

Stimulus : ส่ง request ไปโพส Blog review

Artifact: Normal operation

Environment : Database

Response : ส่ง errors กลับไป

Response Measure : Data ใน database ครบถ้วนปกติ

### Integrability Scenario

Source of stimulus : ผู้พัฒนา (Developer)

Stimulus : เพิ่ม component ใหม่

Artifact : component

Environment : Development

Response : ระบบถูก deploy อัตโนมัติ

Response Measure : redeploy ระบบไม่เกิน 3 นาที

### Availability Scenario

Source of stimulus: Internal System (ภายในระบบ)

Stimulus: ระบบไม่สามารถดึงข้อมูลจากฝั่ง backend มาได้

Artifact: ระบบ

Environment: ระหว่างการทำงานของโปรแกรม

Response: แจ้งไปยังผู้ทำงานในฝั่ง backend

Response Measure: ไม่เกิน 2 นาที

### Availability Scenario

Source of stimulus : monitor (bot discord, email noti)

Stimulus : system crash

Artifact : system

Environment : ระหว่างการทำงานของโปรแกรม

Response : แจ้งเตือนใน discord, email

Response Measure : redeploy ระบบเซิร์ฟเวอร์ ไม่เกิน 1 ชั่วโมง

### Modifiability Scenario

Source of stimulus: Frontend developer

Stimulus: แก้ไขฟังก์ชัน dropdown ในหน้า Blog review

Artifact: UI ในหน้า blog review

Environment: ขณะเขียนโปรแกรม

Response: แก้ไขฟังก์ชันได้โดยไม่เกิดผลกระทบกับงานส่วนอื่น

Response Measure: ใช้เวลาแก้ไขไม่เกิน 10 นาที

### Modifiability Scenario

Source of stimulus : Developer

Stimulus : ต้องการเพิ่ม/ลบ/แก้ไข ฟังก์ชัน

Artifact : ระบบ

Environment : Runtime

Response : Deploy changes

Response Measure : Auto deploy took 2-5 minutes

### Usability Scenario

Source of stimulus : User , ผู้ใช้งานระบบ

Stimulus : การใช้งานระบบแต่ละ features

Artifact : ระบบ,เว็บไซต์

Environment : Runtime

Response : ลดความผิดพลาดในการใช้งาน , ให้ User ใช้งานอย่างมีประสิทธิภาพ

Response Measure : ความพึงพอใจของ User  $\geq 10$  คน , จำนวนการผิดพลาด  $< 10$  ครั้ง

### Testability Scenario

Source of stimulus : Developer

Stimulus : ตรวจสอบความถูกต้องของ Request body, จำนวน feature ที่เพิ่มขึ้น

Artifact(s) : ส่วนของโค้ดที่ทำหน้าที่ประมวลผล request

Environment : design time (ออกแบบยังไงให้ test ง่าย), development time, compile time, deployment time

Response : response บ่งบอกถึงข้อผิดพลาด

Response Measure : เปอร์เซ็นต์ของการ test ที่สำเร็จ, เปอร์เซ็นต์ของการ test ที่ผลออกมาแล้วผิดพลาด, เวลาที่ใช้ในการทดสอบ