



## **Juzcare**

(Media & Publishing Project)

**เสนอ**

อาจารย์ปริญญา เอกปริญญา

**จัดทำโดย**

1. นายจิรายุส	เสนาโนฤทธิ์	รหัสนักศึกษา	63010151
2. นายธีรพัฒน์	เกตุสิงห์น้อย	รหัสนักศึกษา	63010467
3. นายนาวีฮาน	เต๊ะหมาน	รหัสนักศึกษา	63010523
4. นายปิยชัย	แก้วชุ่ม	รหัสนักศึกษา	63010604
5. นายพรหมพิริยะ	เจริญพานทองดี	รหัสนักศึกษา	63010648
6. นางสาวพิชชาภา	เวียงทอง	รหัสนักศึกษา	63010679
7. นางสาวภัทราณิษฐ์	เทศเจริญ	รหัสนักศึกษา	63010727

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา Software Architecture and Design

รหัสวิชา 01076024

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## ที่มาและความสำคัญ

เนื่องจากว่าสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังถือเป็นสถานศึกษาขนาดใหญ่ที่มีจำนวนนักศึกษาค่อนข้างมาก ทำให้อาจจะมีการตรวจสอบปัญหาในจุดต่าง ๆ อย่างไม่ครอบคลุม และยังไม่มีช่องทางกลางในการรายงานปัญหาระหว่างนักศึกษากับเจ้าหน้าที่ที่เกี่ยวข้อง โดยมีแนวคิดมาจาก Traffy Fondue ซึ่งจะจัดทำเป็น Web Application ที่สามารถแจ้งปัญหาที่เกิดขึ้นภายในสถาบันได้ เพื่อแสดงให้เห็นถึงปัญหาที่จำเป็นต่อการแก้ไขแก่ทางสถาบัน โดยบุคลากรภายในสถาบันสามารถแจ้งปัญหา แนวนรูปปัญหาต่างๆ และโหวตปัญหาที่ต้องการได้รับการแก้ไข เพื่อให้แอดมินคัดกรองและส่งเรื่องไปยังหน่วยงานที่เกี่ยวข้องให้แก้ไขปัญหา

## วัตถุประสงค์

1. เพื่อรายงานปัญหาที่เกิดขึ้นภายในสถาบัน
2. เพื่อลดปัญหาด้านการสำรวจของสถาบัน
3. เพื่อช่วยแก้ปัญหากที่เกิดขึ้นภายในสถาบัน

## กลุ่มผู้ใช้งาน

บุคคลทั่วไปและบุคลากรภายในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## ข้อจำกัด

1. ไม่สามารถประสานงานกับหน่วยงานที่เกี่ยวข้องได้โดยตรง
2. ไม่สามารถระบุช่วงระยะเวลาที่จะใช้ในการดำเนินการแก้ไขปัญหามาจนเสร็จสิ้นได้อย่างชัดเจน
3. ไม่สามารถทำ fully responsive

## ฟังก์ชันการใช้งาน

### ผู้ใช้งาน

- สร้างกระทู้เพื่อแจ้งปัญหาต่าง ๆ ภายในสถาบัน
- แนวนรูปและวิดีโอของปัญหา
- รัวความพอใจหลังแก้ไขปัญหา
- โหวตเรื่องร้องเรียน
- คอมเมนต์ใต้กระทู้
- Notification
- สร้างกระทู้เพื่อรายงานปัญหา
- อัปเดตกระทู้
- ดูประวัติการตั้งกระทู้
- แก้ไขโปรไฟล์ได้

## แอดมิน

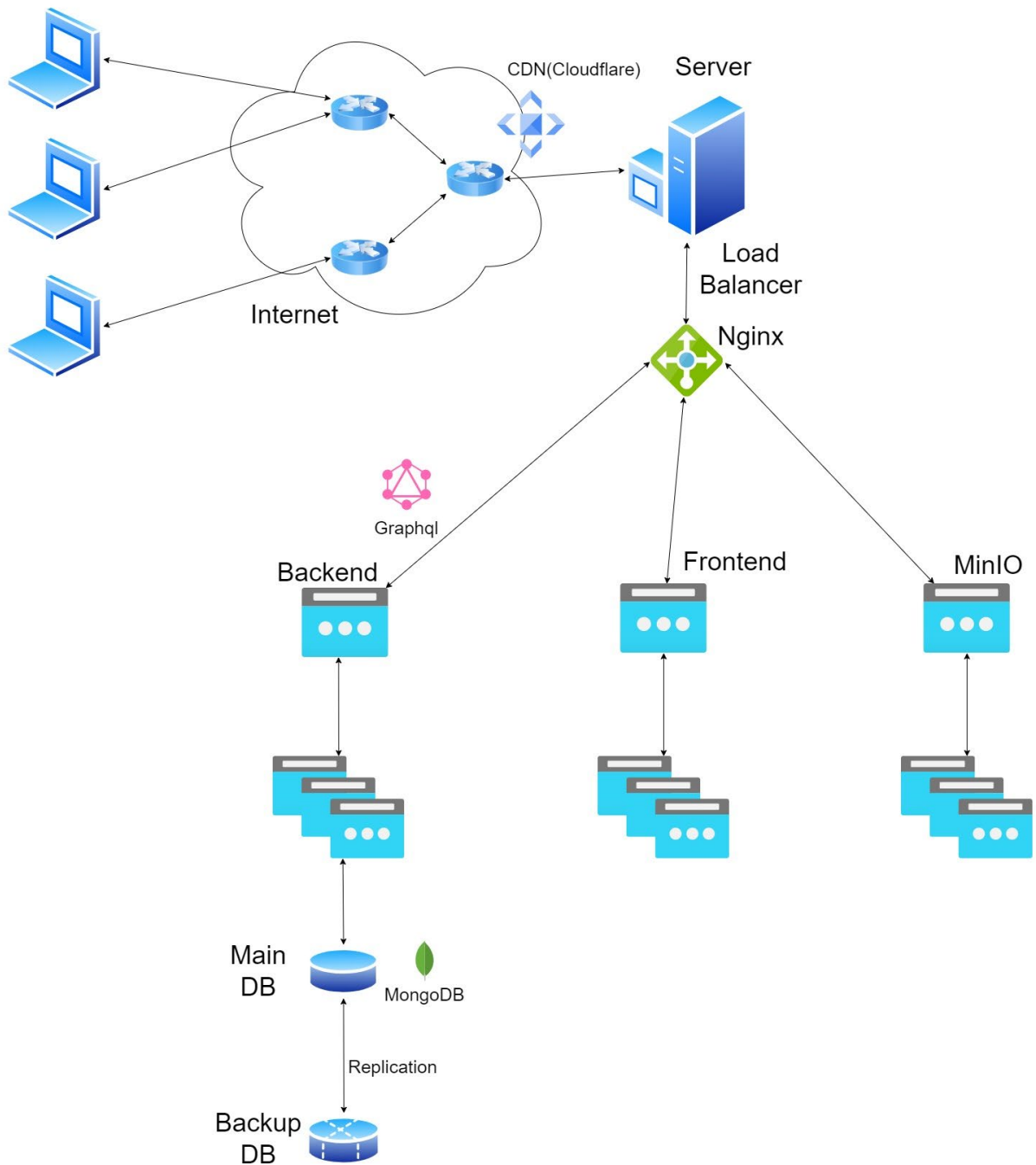
- ดูปัญหาในแต่ละพื้นที่
- คัดกรองปัญหา เพื่อให้หน่วยงานที่เกี่ยวข้องแก้ไขปัญหา
- เพิ่มแท็กที่คัดกรองกระทู้
- ดูรีวิวของผู้ที่แจ้งปัญหา

## ฟังก์ชันทั่วไป

- ดูกระทู้รายงานปัญหาปัจจุบันของผู้ใช้งานคนอื่น
- ดูเทรนด์กระทู้
- คัดกรองการมองเห็นกระทู้ผ่านแท็ก
- ค้นหากระทู้

## Software architecture

Clients



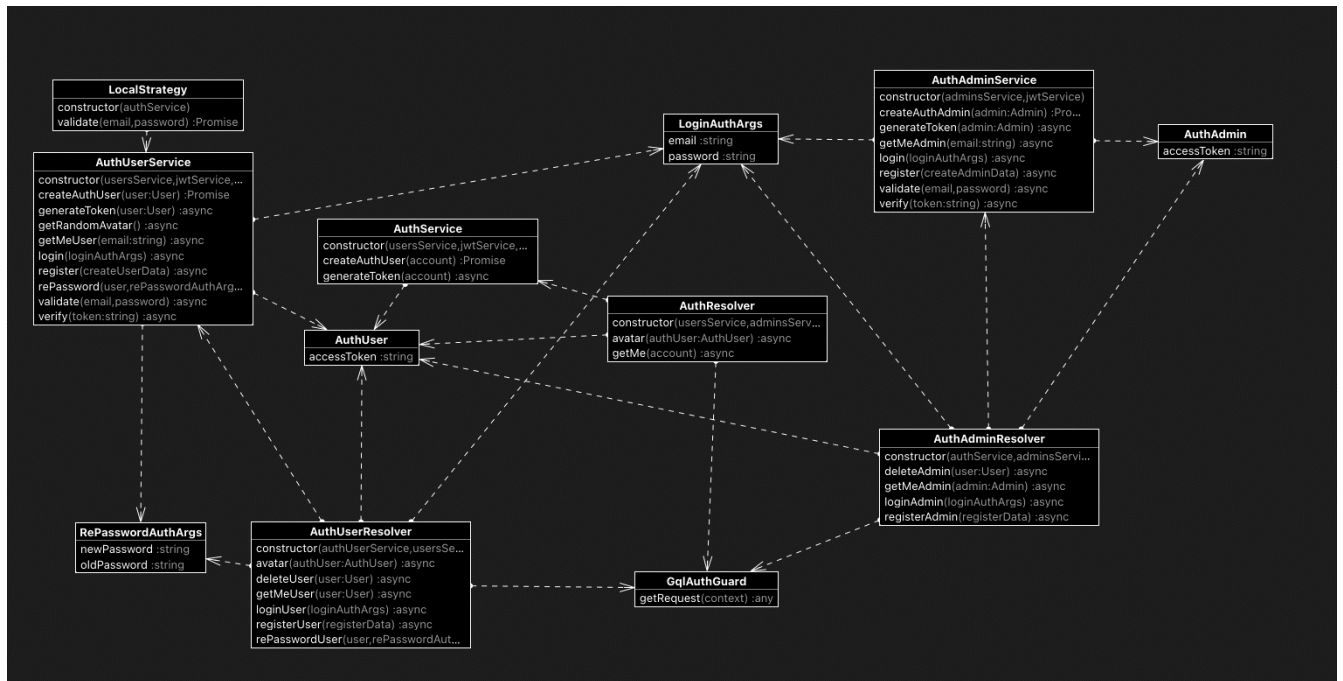
Juzcare เป็น Web Application ที่มีสถาปัตยกรรมแบบ Client-Server หรือคือ ผู้ใช้บริการและผู้ให้บริการ ซึ่งในที่นี้ Juzcare จะแบ่งการทำงานออกเป็นสองส่วน ได้แก่ ในส่วน front end ที่ทำหน้าที่เป็น Client และส่วน Backend ที่ทำหน้าที่เป็น server โดยการทำงานจะเป็นไปในรูปแบบของการที่ผู้ให้บริการต้องการใช้ข้อมูลก็จะส่งคำขอไปยังผู้ให้บริการ และผู้ให้บริการก็จะส่งข้อมูลกลับมายังผู้ให้บริการตามที่ผู้ให้บริการร้องขอ จากในรูปเราจะมาเริ่มจากส่วน ของ Backend เราจะใช้ MongoDB เป็น NoSQL Database ที่มี High availability เนื่องจากตัว MongoDB สามารถที่จะสร้างเป็น Cluster ในการ Replication Data เพื่อนำมาเป็น Backup DB และใช้ตัว GraphQL ซึ่งเป็น Query Language โดยใช้งานผ่าน API ของระบบและจะประมวลผลคำสั่งที่ฝั่ง server หรือที่เรียกว่า server-side runtime โดยใช้โครงสร้างข้อมูลที่เรากำหนดไว้ ช่วยให้ client มีความสามารถที่จะ request สิ่งที่ต้องการ และได้รับ response ตามที่ต้องการกลับมาได้ ในส่วนของ Front end จะใช้ Apollo-Client มีประสิทธิภาพมากในเรื่อง Caching Feature ซึ่งทำให้ Components Fetch จาก Cache ก่อน อย่างนั้นถึงจะส่ง Request

ต่อมาจะเป็นส่วนของตัว Server ในที่นี้เราใช้ Server 1 ตัว และเพิ่มการใช้ Nginx เป็น HTTP Load Balancing ทั้งใน 3 ส่วนก็คือ ส่วน Backend, Frontend และ MinIO เพื่อให้ทั้ง 3 ส่วนมี High availability โดยเมื่อเกิดการ Failover ระบบจะเปลี่ยนเส้นทางไปยังอีกระบบหนึ่งที่สแตนด์บายอยู่โดยอัตโนมัติ เมื่อตัวหลักหยุดทำงาน Floating IP จะย้ายไปยังตัวที่สอง โดยอัตโนมัติเพื่อให้บริการต่อ และในส่วนที่ Client ได้ทำการเชื่อมต่อกับ Internet นั้นได้มีการใช้ระบบการกระจายเนื้อหาข้อมูลโดยที่เราได้ใช้ Cloudflare CDN โดยระบบ CDN นั้นจะทำการส่งข้อมูลจาก Server ที่ใกล้ที่สุดผ่านการเชื่อมต่ออินเทอร์เน็ตทำให้สามารถส่งข้อมูลไปถึงผู้รับปลายทางได้อย่างรวดเร็ว อีกทั้งยังเพิ่มประสิทธิภาพในการเข้าถึงข้อมูล ซึ่งตัวของ CDN ยังช่วยในอีกหลายด้าน เช่น ในด้าน Security ช่วยลดการโจมตี DDos Attack ในด้าน Performance มีการแคชเว็บที่มีเนื้อหาคงที่ (Static content) และเนื้อหาที่มีการเปลี่ยนแปลงบ่อย (Dynamic Content) ทำให้ง่ายต่อการจัดการกับเนื้อหาออกเหล่านี้อย่างมีประสิทธิภาพ

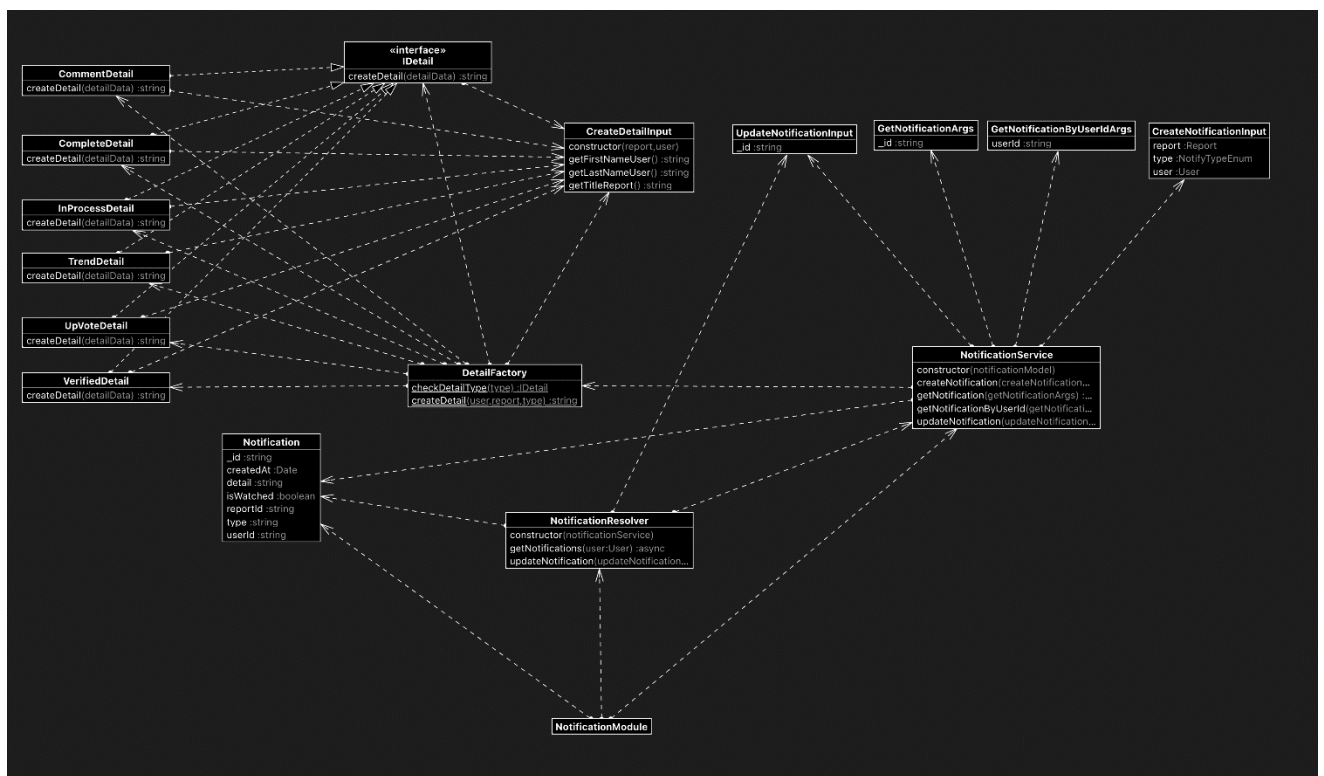
## Software design

Juzcare ประกอบไปด้วย 3 bounded contexts

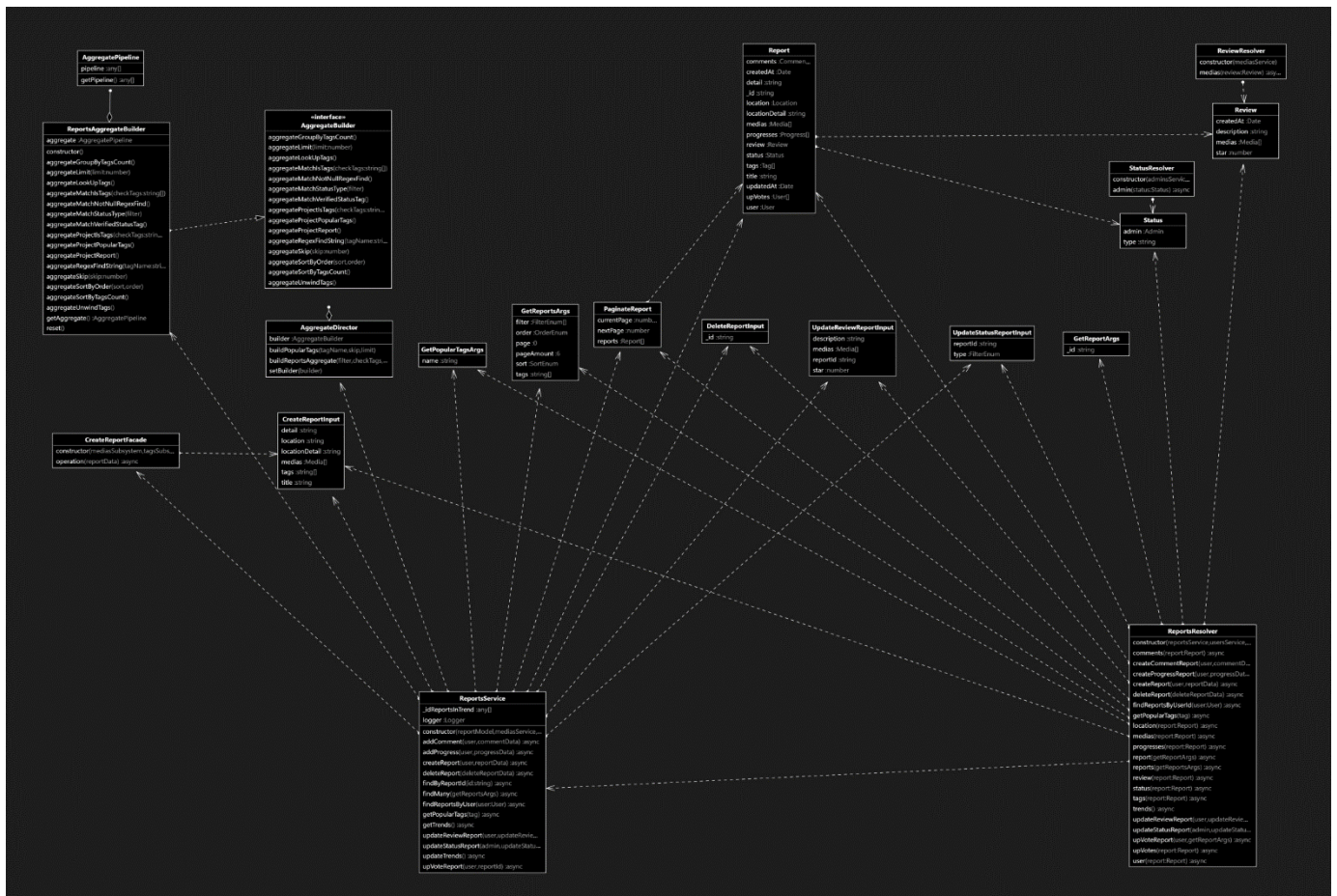
### 1. Authentication



### 2. Notification



### 3. Reporting



## Design pattern

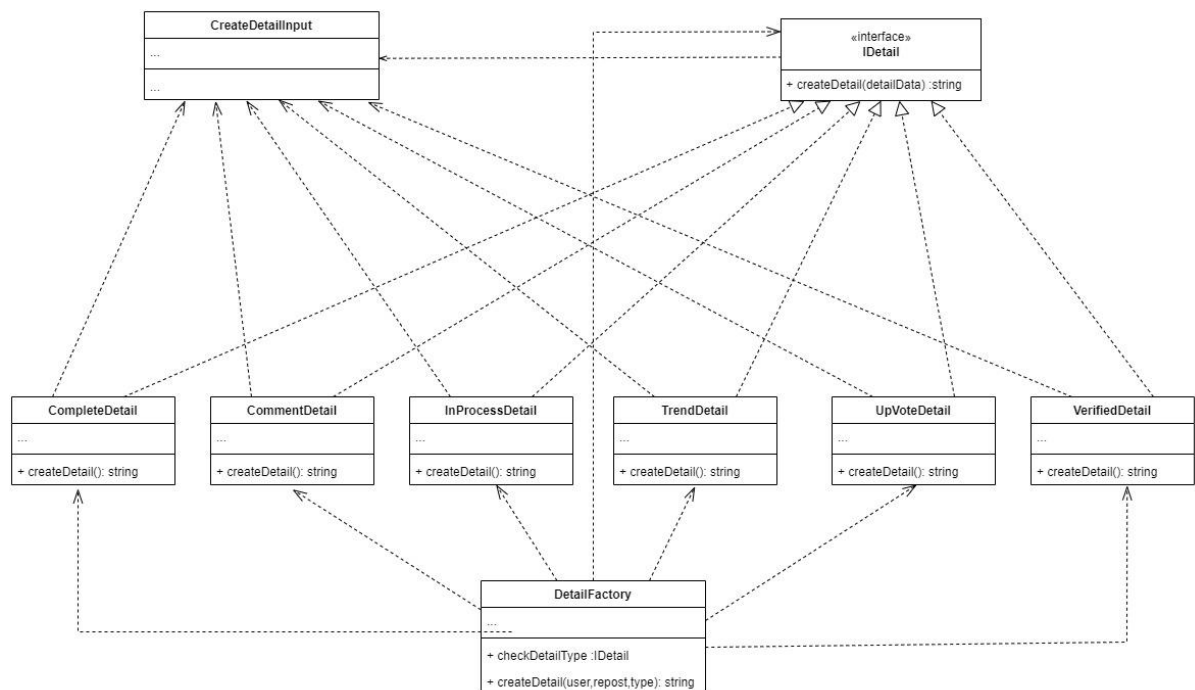
### 1. Factory method (อยู่ในส่วนของ notification)

- ทำไมถึงใช้

เพราะว่าเว็บของเรามี detail ของ notification ที่หลายแบบ เลยทำการสร้าง concrete class เพื่อรองรับรูปแบบการแจ้งเตือนเหล่านั้น

- ใช้อย่างไร

ใช้ในเรื่อง notification ที่มีหลายแบบ เช่น การแจ้งเตือนสำหรับ Trend, Comment, InProcess, Complete, UpVote, Verified



- ใช้แล้วมีผลดีอย่างไร

ทำให้โค้ดที่เรียกใช้งานมีความยืดหยุ่น กระบวนการการสร้าง object กับ object แยกออกจากกัน



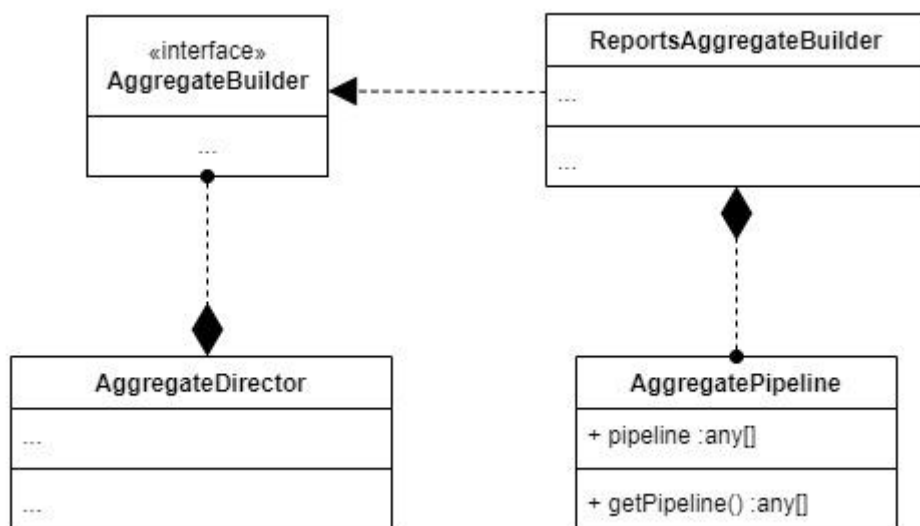
## 2. Builder (อยู่ในส่วนของ report)

- ทำไมถึงใช้

เพราะช่วยแก้ปัญหาเวลาที่มีการสร้าง object หลาย ๆ อย่าง ที่มีขั้นตอนคล้าย ๆ กัน และช่วยแก้ปัญหาเวลาที่มีการสร้าง class ที่ซับซ้อน

- ใช้อย่างไร

เช่น การสร้างรูปแบบการ Query ถ้ารูปแบบการ Query ใหม่ มีบางส่วนที่เหมือนกับรูปแบบเดิมก็จะนำส่วนของการ Query แบบเก่า มาต่อด้วยรูปแบบการ Query ใหม่ที่ต่างไปจากเดิม โดยที่ไม่ต้องเขียนรูปแบบการ Query เก่าขึ้นมาใหม่อีกครั้ง



- ใช้แล้วมีผลดียังไง

ช่วยให้ประหยัดเวลาเพราะไม่ต้องเขียนโปรแกรม หรือรูปแบบการ Query เดิมซ้ำอีกครั้ง ช่วยทำให้สะดวกในการสร้าง class จาก builder ที่เรากำหนด โดยกำหนดคุณสมบัติจากการเรียกใช้ method ต่างๆ

### 3. Facade

- ทำไมถึงใช้

แก้ไขปัญหาในเรื่องการที่ Client จะต้องการเรียกใช้ Subsystem หลายๆตัว

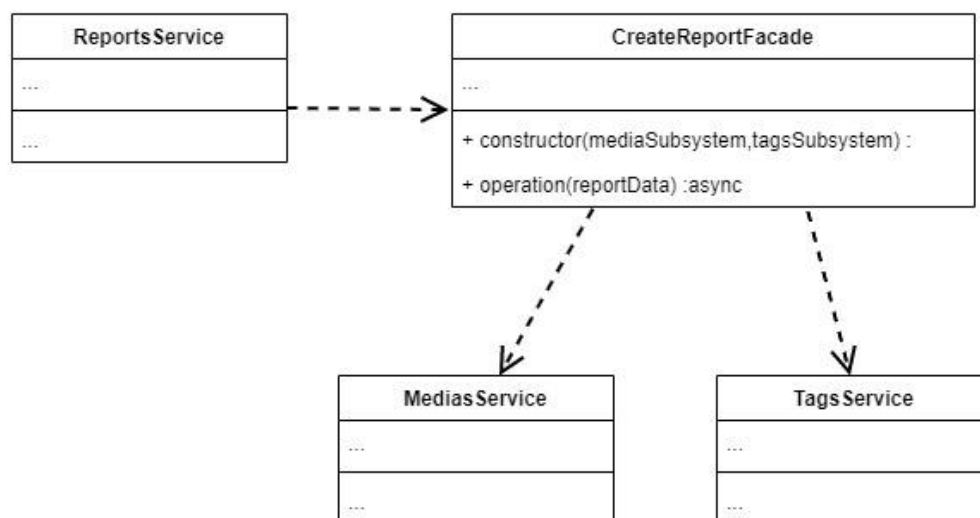
จึงแก้ไขปัญหาโดยการสร้าง Facade Class ที่ใช้ Method มาเรียกใช้ Subsystem นั้นๆ

และให้ Client เรียกใช้ Facade Class นั้นแทนการเรียก Subsystem หลายๆตัว

- ใช้อย่างไร

เช่น Client สามารถดึงข้อมูลการสร้างกระทู้มาได้เลยว่าใครเป็นคนสร้าง หัวข้ออะไร

รูปคืออะไร ใช้แท็กอะไร เวลาที่สร้างเมื่อไร โดยไม่ต้องลงไปยุ่งกับ Subsystem



- ใช้แล้วมีผลดียังไง

- ช่วยให้สามารถประยุกต์ใช้งานกับไลบรารีอื่น ๆ ได้ง่ายขึ้น  
แล้วสามารถเพิ่มหรือแก้ไขฟังก์ชันการทำงานได้ง่ายขึ้น
- มี unified interface เพื่อเชื่อมต่อกับ subsystem ใน Facade จะกำหนด interface ในระดับสูง ที่ทำให้ subsystem ง่ายต่อการใช้งาน

## Quality Attribute

### 1. Availability : Load Balancer, MongoDB Cluster

- เพิ่มประสิทธิภาพให้ระบบ application หรือ service โดยสามารถ scale-out ด้วยการทำ load balancing ในกรณีที่ต้องการรองรับการใช้งานของ user ที่มีจำนวนมากขึ้น เราสามารถที่จะเพิ่มจำนวนของ application หรือ service โดยการเพิ่มจำนวนของ server และกำหนด Algorithms สำหรับจัดการ traffic การใช้งานของ user ให้สอดคล้องกับ resource ของ server นั้นๆ ได้
- ทำให้ application หรือ service มี High Availability (HA) มากขึ้น การใช้ load balancer ในการจัดการ traffic การใช้งานของ user ไปยัง application ที่อยู่หลายๆ server นั้น เมื่อมี application บน server ใดๆ มีปัญหาไม่สามารถให้บริการหรือสามารถใช้งานได้ load balancer จะมีการ health check เพื่อตรวจสอบและหยุดส่ง traffic ของ user ไปยัง application ที่อยู่บน server นั้นทันที และส่งต่อไปยัง application บน server ที่พร้อมให้บริการ ซึ่งทำให้ application สามารถให้บริการและใช้งานได้อย่างต่อเนื่อง
- มีการใช้ MongoDB Cluster (Replication) คือการกระจายข้อมูลท่ามกลาง MongoDB servers หลายๆ node โดย MongoDB สามารถกระจายข้อมูลไปยัง 1 หรือมากกว่านั้นและข้อมูลจะ sync กันตลอดเวลาเมื่อมีข้อมูลเปลี่ยนแปลง ยกตัวอย่างเช่น เราสามารถรัน backup บน secondary node เพื่อลดโหลดบน primary node หรือการทำ index ที่เราสามารถเลือกที่จะ build บน secondary node ก่อน จากนั้นสลับไปเป็น primary ก็ได้แล้ว build ใหม่สลับกันไป

## Availability Scenario

Portion of Scenario	Possible values
Source	ผู้ใช้งาน
Stimulus	ต้องการกดไลก์
Artifact	ไลก์ของกระทู้
Environment	Normal mode
Response	ไลก์ถูกอัปเดตลงฐานข้อมูลทันที
Response Measure	กระทู้มีการจัดเรียงใหม่ในเทรนด์กระทู้ทันที

Portion of Scenario	Possible values
Source	ผู้ใช้งาน
Stimulus	การสร้างบัญชีผู้ใช้
Artifact	บัญชีผู้ใช้ภายในระบบ
Environment	Normal mode
Response	สามารถล็อกอินได้หลังจากสมัคร
Response Measure	ระยะเวลาในการอัปเดตข้อมูลลงฐานข้อมูล

Portion of Scenario	Possible values
Source	ผู้พัฒนา
Stimulus	เพิ่ม / ลด / แก้ไขฟังก์ชันการทำงาน
Artifact	ระบบ
Environment	Dev mode
Response	แก้ไขได้โดยไม่กระทบกับข้อมูลบัญชีผู้ใช้งาน
Response Measure	ผู้ใช้งานสามารถกลับมาใช้งานได้ตามปกติ

## 2. Performance : Load Balancer , GraphQL

- ช่วยเพิ่มความเร็วของ application หรือ service  
เนื่องจาก load balancer ทำการกระจาย traffic ของ user ไปยัง application ที่ทำงานอยู่บน server หลายๆ เครื่อง ทำให้การใช้งานของ user ไม่ต้องรอคิวหรือรอการ process ข้อมูลที่ต้องใช้เวลานานเหมือนกับ application ที่ทำงานอยู่บน server เครื่องเดียวที่มี resource อยู่อย่างจำกัดทั้ง CPU, Memory และ Harddisk ทำให้การใช้งานของ user เร็วขึ้น เช่น สามารถโหลดข้อมูลรูปภาพจากเว็บได้เร็วขึ้น การเข้าถึงข้อมูลของ application ได้เร็วขึ้น เป็นต้น
- ใช้ GraphQL ในเรื่องของ caching เพื่อให้การเรียกใช้ข้อมูลในครั้งต่อไปรวดเร็วมากยิ่งขึ้น และเพื่อให้เราสามารถดึงข้อมูลได้ตรงกับความต้องการ ลดความซับซ้อนในการเขียนโค้ดเพื่อดึงข้อมูล สามารถจัดการและดูแลรักษาโค้ดในฝั่ง server-side ได้ง่ายมากยิ่งขึ้น และที่สำคัญ GraphQL ไม่ได้ผูกติดกับ database และที่จัดเก็บข้อมูลใดๆทั้งนั้น

## Performance Scenario

Portion of Scenario	Possible values
Source	ผู้ใช้งาน
Stimulus	ต้องการตั้งกระตุ้โดยแนบรูปหรือวิดีโอที่มีขนาดความจุมากหรือมีจำนวนที่เยอะ
Artifact	ระบบ
Environment	Normal mode
Response	ระบบทำการสร้างกระตุ้
Response Measure	ระยะเวลาในการอัปโหลดกระตุ้

Portion of Scenario	Possible values
Source	Developer
Stimulus	เขียนโค้ดเพื่อดึงข้อมูล
Artifact	ข้อมูล
Environment	runtime, compile time
Response	ได้รับข้อมูลที่ถูกต้องตามต้องการ
Response Measure	ระยะเวลาในการเรียกใช้ข้อมูล

Portion of Scenario	Possible values
Source	ผู้มาเยือน
Stimulus	ต้องการเห็นกระตู่จำนวนมาก
Artifact	ระบบ
Environment	Normal mode
Response	ระบบทำการโหลดกระตู่
Response Measure	ระยะเวลาที่กระตู่แสดงให้เห็น



### 3. Modifiability : NestJS

- NestJS คือ Node Framework ใช้สำหรับเขียน Backend โดยมีข้อดีคือ การที่โครงสร้างของมันเหมาะสมกับการ Scale และยังมี Dependency Injection เป็นเทคนิคการเขียนโค้ดที่จะช่วยให้โค้ดมีความยืดหยุ่นมากขึ้น ไม่ผูกกับคลาสด้วยกันจนเกินไป ซึ่งเป็นหัวใจสำคัญสำหรับการเขียนโค้ดที่มีโครงสร้างขนาดใหญ่

#### Modifiability Scenario

Portion of Scenario	Possible values
Source	ผู้พัฒนา
Stimulus	ต้องการเปลี่ยน UI
Artifact	UI
Environment	runtime, compile time
Response	เปลี่ยนโดยไม่มีผลกระทบกับฟังก์ชันอื่นๆ, ตรวจสอบการเปลี่ยน, การนำไปใช้งาน
Response Measure	ระยะเวลาในการเปลี่ยน

Portion of Scenario	Possible values
Source	ผู้พัฒนา
Stimulus	ต้องการสร้างหน้าเพิ่ม
Artifact	UI
Environment	runtime, compile time
Response	เรียกใช้งานองค์ประกอบบางอย่างจากคลาสอื่นได้โดยไม่ส่งผลกระทบกับฟังก์ชันอื่น
Response Measure	ระยะเวลาในการสร้างหน้าใหม่

Portion of Scenario	Possible values
Source	ผู้พัฒนา
Stimulus	ต้องการเชื่อมต่อระบบเข้ากับปลั๊กอินอื่น ๆ
Artifact	ระบบ
Environment	runtime, compile time
Response	ทำได้ง่ายโดยไม่ส่งผลกระทบกับฟังก์ชันอื่น
Response Measure	ระยะเวลาในการเชื่อมต่อระบบเข้ากับปลั๊กอินอื่น

#### 4. Security : Authentication

- มีการจำกัดการเข้าถึงข้อมูลต่างๆ เช่น ฝั่งผู้ใช้งาน ฝั่งแอดมิน
- การระบุตัวตนและยืนยันตัวตน ใช้ในการตรวจสอบสิทธิ์เข้าใช้บริการ
- มีการใช้ Password encryption

#### Security Scenario

Portion of Scenario	Possible values
Source	ผู้ใช้งาน
Stimulus	แก้ไขข้อมูลส่วนตัว
Artifact	ข้อมูลภายในระบบ
Environment	Normal Mode
Response	ตรวจสอบการยืนยันข้อมูลของผู้ใช้
Response Measure	สามารถแก้ไขข้อมูลได้

Portion of Scenario	Possible values
Source	ผู้ใช้งาน
Stimulus	พยายามเข้าใช้งานบัญชีของผู้ใช้งานคนอื่น
Artifact	บัญชีภายในระบบ
Environment	Normal mode
Response	ไม่สามารถเข้าสู่ระบบของบัญชีอื่นได้
Response Measure	ตรวจสอบข้อมูลที่ใช้ล็อกอิน

Portion of Scenario	Possible values
Source	ผู้ใช้งาน
Stimulus	พยายามสแปมกระทู้
Artifact	กระทู้ภายในระบบ
Environment	Normal mode
Response	บัญชีถูกลบออกจากระบบโดยแอดมิน
Response Measure	จำนวนการกดรีพอร์ตจากผู้ใช้งานบัญชีอื่น

## 5. Usability : UxUi

- มีการทำ Ux/Ui Design เพื่อให้เว็บไซต์มีการใช้งานที่ง่ายขึ้นและมีความสวยงาม

### Usability Scenario

Portion of Scenario	Possible values
Source	ผู้ใช้งาน
Stimulus	เข้ามาใช้งานระบบ
Artifact	รูปแบบการใช้งานระบบ
Environment	Normal mode
Response	ใช้งานได้โดยไม่ต้องมีคู่มือ
Response Measure	อัตราการเพิ่มขึ้นของจำนวนบัญชีผู้ใช้ภายในระบบ

Portion of Scenario	Possible values
Source	ผู้ใช้งานทั่วไปที่ไม่มีบัญชีอยู่ในระบบ
Stimulus	เรียกดูความคิดเห็นภายในกระทู้
Artifact	ความคิดเห็นของกระทู้
Environment	Normal mode
Response	ข้อมูลที่ผู้ใช้งานทั่วไปเรียกดู
Response Measure	สามารถดูความคิดเห็นในแต่ละกระทู้ได้โดยไม่ต้องล็อกอิน

Portion of Scenario	Possible values
Source	ผู้ใช้งาน
Stimulus	ต้องการคัดกรองการมองเห็นกระทู้
Artifact	กระทู้ภายในระบบ
Environment	Normal mode
Response	แสดงผลกระทู้ตามที่คัดกรอง
Response Measure	จำนวนกระทู้ที่แสดง

