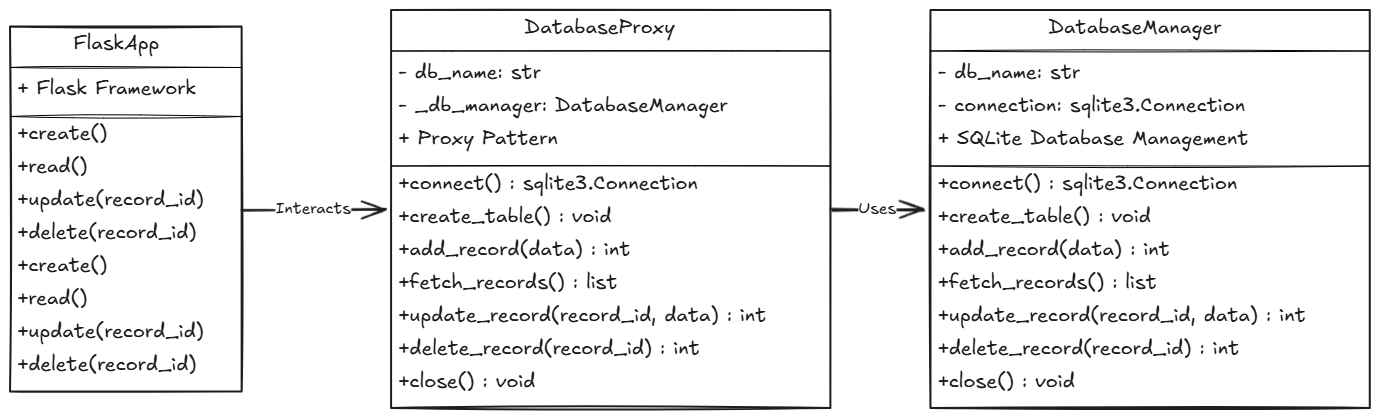
**การประเมินผลลัพธ์ ที่ได้จาก Generative AI ทั้งหมด**

1. **ผลจากการใช้ Generative AI ในการเขียน Code Python มีดังนี้**
   1. **ChatGPT**
      1. ผลในรอบที่ 1



**ภาพที่ 1** แผนภาพการทำงานของระบบ ที่ได้จาก ChatGPT ในรอบที่ 1

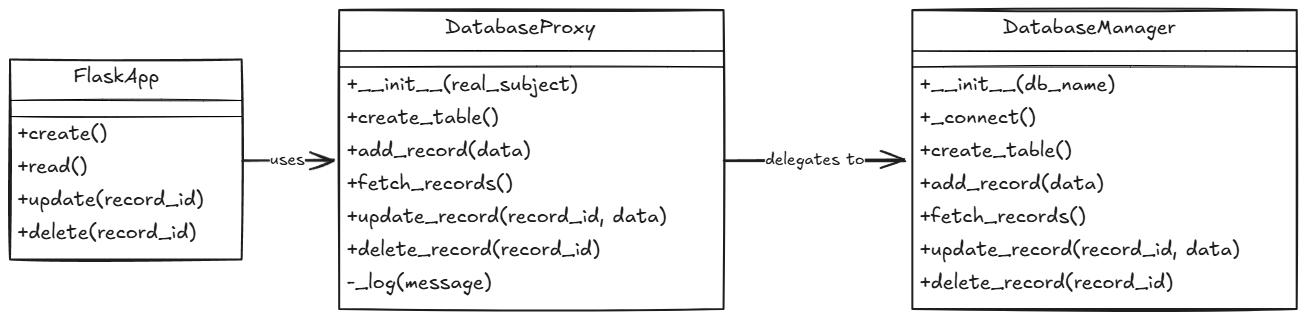
* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

จาก Diagram จะมองว่า ไม่ตรงกับ Pattens ที่กำหนด แต่เนื่องจาก Python เองไม่มี Class interface แต่โดยรวม มองว่า ได้ตาม Pattens ที่กำหนด

* + 1. ผลในรอบที่ 2



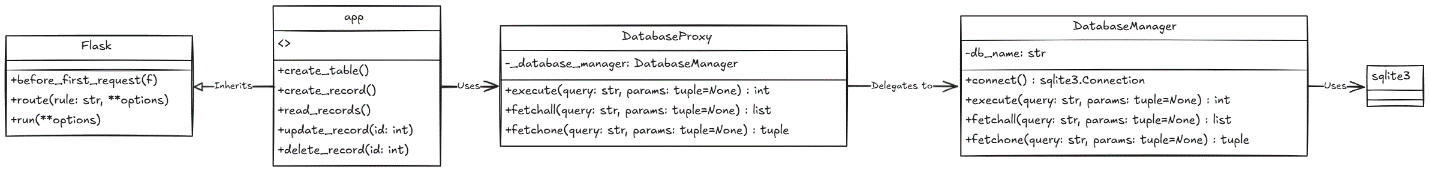
**ภาพที่ 2** แผนภาพการทำงานของระบบ ที่ได้จาก ChatGPT ในรอบที่ 2

* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

จาก Diagram จะมองว่า ไม่ตรงกับ Pattens ที่กำหนด แต่เนื่องจาก Python เองไม่มี Class interface แต่โดยรวม มองว่า ได้ตาม Pattens ที่กำหนด

* + 1. ผลในรอบที่ 3

**ภาพที่ 3** แผนภาพการทำงานของระบบ ที่ได้จาก ChatGPT ในรอบที่ 3

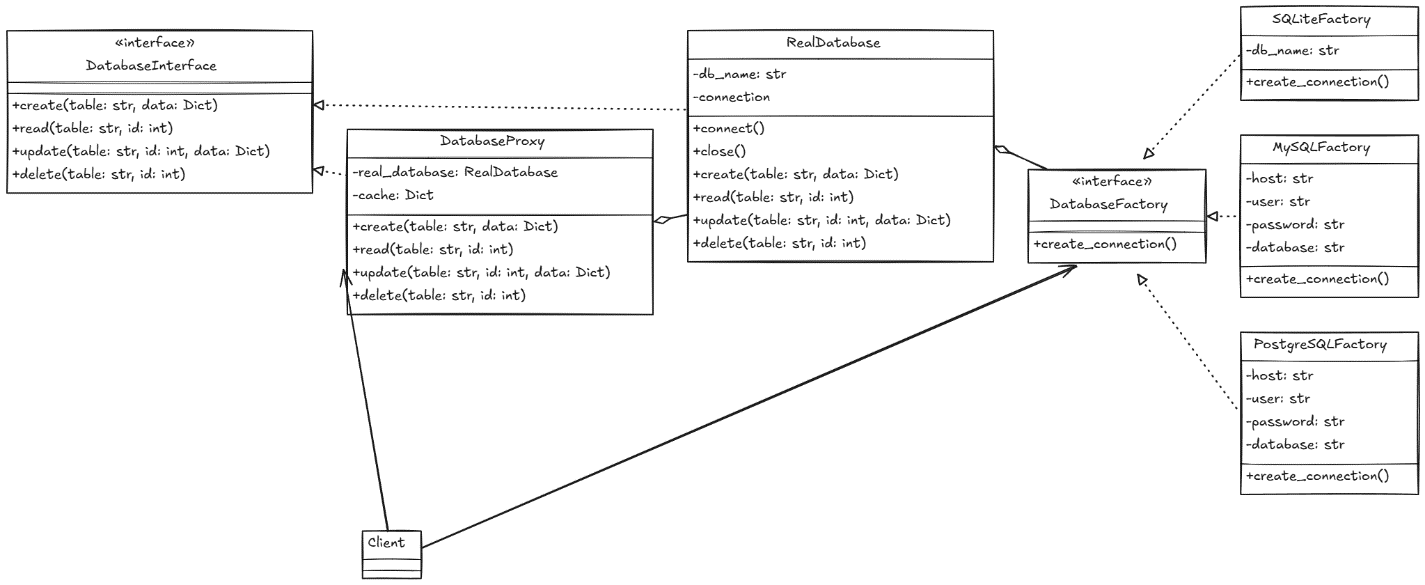
* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

จาก Diagram จะมองว่า ไม่ตรงกับ Pattens ที่กำหนด แต่เนื่องจาก Python เองไม่มี Class interface แต่โดยรวม มองว่า ได้ตาม Pattens ที่กำหนด

* 1. **Claude**
     1. ผลในรอบที่ 1



**ภาพที่ 4** แผนภาพการทำงานของระบบ ที่ได้จาก Claude ในรอบที่ 1

* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

จาก Diagram จะเห็นว่า ตัวของ Claude ทำการสร้าง ได้ถูกหลัก Pattens โดยการทำให้ Python มี Interface นั้นเอง

* 1. **Gemini-Flash**
     1. ผลในรอบที่ 1

A close-up of a document

Description automatically generated

**ภาพที่ 5** แผนภาพการทำงานของระบบ ที่ได้จาก Gemini-Flash ในรอบที่ 1

* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ แต่อาจจะ ไม่ทั้งหมด

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

จาก Diagram จะมองว่า ไม่ตรงกับ Pattens ที่กำหนด แต่เนื่องจาก Python เองไม่มี Class interface แต่โดยรวม มองว่า ได้ตาม Pattens ที่กำหนด

* + 1. ผลในรอบที่ 2

A close-up of a document

Description automatically generated

**ภาพที่ 6** แผนภาพการทำงานของระบบ ที่ได้จาก Gemini-Flash ในรอบที่ 2

* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

จาก Diagram จะมองว่า ไม่ตรงกับ Pattens ที่กำหนด แต่เนื่องจาก Python เองไม่มี Class interface แต่โดยรวม มองว่า ได้ตาม Pattens ที่กำหนด

* + 1. ผลในรอบที่ 3

A diagram of a computer

Description automatically generated

**ภาพที่ 7** แผนภาพการทำงานของระบบ ที่ได้จาก Gemini-Flash ในรอบที่ 3

* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

ไม่ตรงกับ หลักของ Design pattens ที่กำหนด เนื่องจาก ProductAPI ทำการเรียก Product โดยตรง

* 1. **Gemini-Pro**
     1. ผลในรอบที่ 1

A diagram of a computer code

Description automatically generated with medium confidence

**ภาพที่ 8** แผนภาพการทำงานของระบบ ที่ได้จาก Gemini-Pro ในรอบที่ 1

* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

จาก Diagram จะมองว่า ไม่ตรงกับ Pattens ที่กำหนด แต่เนื่องจาก Python เองไม่มี Class interface แต่โดยรวม มองว่า ได้ตาม Pattens ที่กำหนด

* + 1. ผลในรอบที่ 2

A diagram of a mathematical equation

Description automatically generated with medium confidence

**ภาพที่ 9** แผนภาพการทำงานของระบบ ที่ได้จาก Gemini-Pro ในรอบที่ 2

* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

จาก Diagram จะมองว่า ไม่ตรงกับ Pattens ที่กำหนด แต่เนื่องจาก Python เองไม่มี Class interface แต่โดยรวม มองว่า ได้ตาม Pattens ที่กำหนด และ จาก Diagram นี้ เหมือนจะไม่ใช่ แต่มองยังใช้ เนื่องจาก Book API ทำหน้าที่ แทน Proxy

* + 1. ผลในรอบที่ 3

A diagram of a complex structure

Description automatically generated with medium confidence

**ภาพที่ 10** แผนภาพการทำงานของระบบ ที่ได้จาก Gemini-Pro ในรอบที่ 3

* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

จาก Diagram จะมองว่า ไม่ตรงกับ Pattens ที่กำหนด แต่เนื่องจาก Python เองไม่มี Class interface แต่โดยรวม มองว่า ได้ตาม Pattens ที่กำหนด และ จาก Diagram นี้ เหมือนจะไม่ใช่ แต่มองยังใช้ เนื่องจาก Book API ทำหน้าที่ แทน Proxy

* 1. **Copilot**
     1. ผลในรอบที่ 1

A diagram of a data flow

Description automatically generated with medium confidence

**ภาพที่ 11** แผนภาพการทำงานของระบบ ที่ได้จาก Copilot ในรอบที่ 1

* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

จาก Diagram จะมองว่า ไม่ตรงกับ Pattens ที่กำหนด แต่เนื่องจาก Python เองไม่มี Class interface แต่โดยรวม มองว่า ได้ตาม Pattens ที่กำหนด

* + 1. ผลในรอบที่ 2

A diagram of a computer

Description automatically generated

**ภาพที่ 12** แผนภาพการทำงานของระบบ ที่ได้จาก Copilot ในรอบที่ 2

* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

จาก Diagram จะมองว่า ทำได้ตาม Pattens ที่กำหนด

* + 1. ผลในรอบที่ 3

A diagram of a program

Description automatically generated with medium confidence

**ภาพที่ 13** แผนภาพการทำงานของระบบ ที่ได้จาก Copilot ในรอบที่ 3

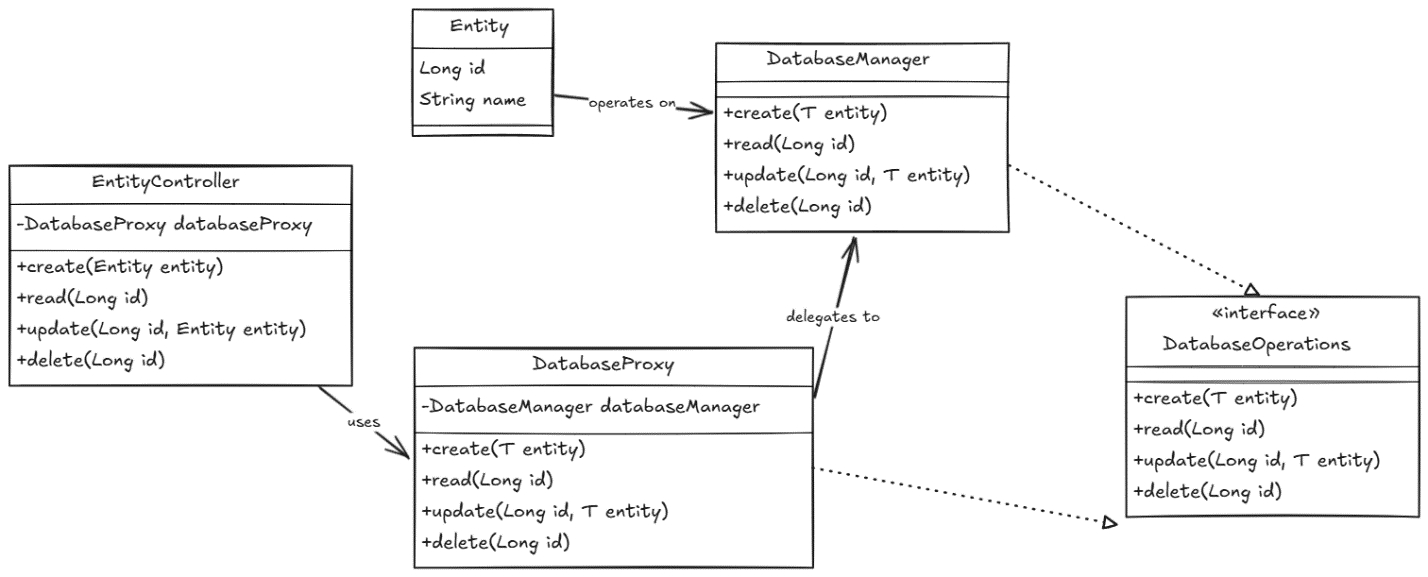
* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่

จาก Diagram จะมองว่า ไม่ตรงกับ Pattens ที่กำหนด แต่เนื่องจาก Python เองไม่มี Class interface แต่โดยรวม มองว่า ได้ตาม Pattens ที่กำหนด

1. **ผลจากการใช้ Generative AI ในการเขียน Code Java มีดังนี้**
   1. **ChatGPT**
      1. ผลในรอบที่ 1



* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design patterns ที่ระบุไว้หรือไม่

จาก Diagram จะเห็นได้ว่า สามารถ ออกแบบ ตามหลักของ Design patterns ที่กำหนดได้อย่างดีเยี่ยม

* + 1. ผลในรอบที่ 2

A diagram of a computer code

Description automatically generated with medium confidence

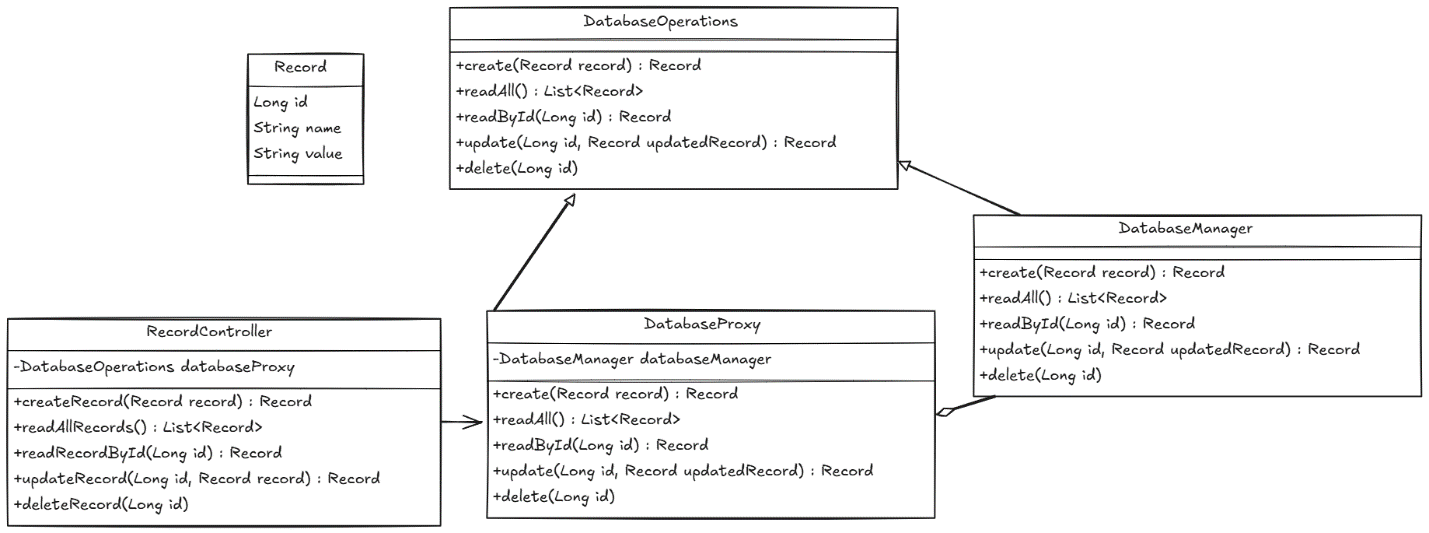
* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design patterns ที่ระบุไว้หรือไม่

จาก Diagram จะเห็นได้ว่า สามารถ ออกแบบ ตามหลักของ Design patterns ที่กำหนดได้อย่างดีเยี่ยม

* + 1. ผลในรอบที่ 3



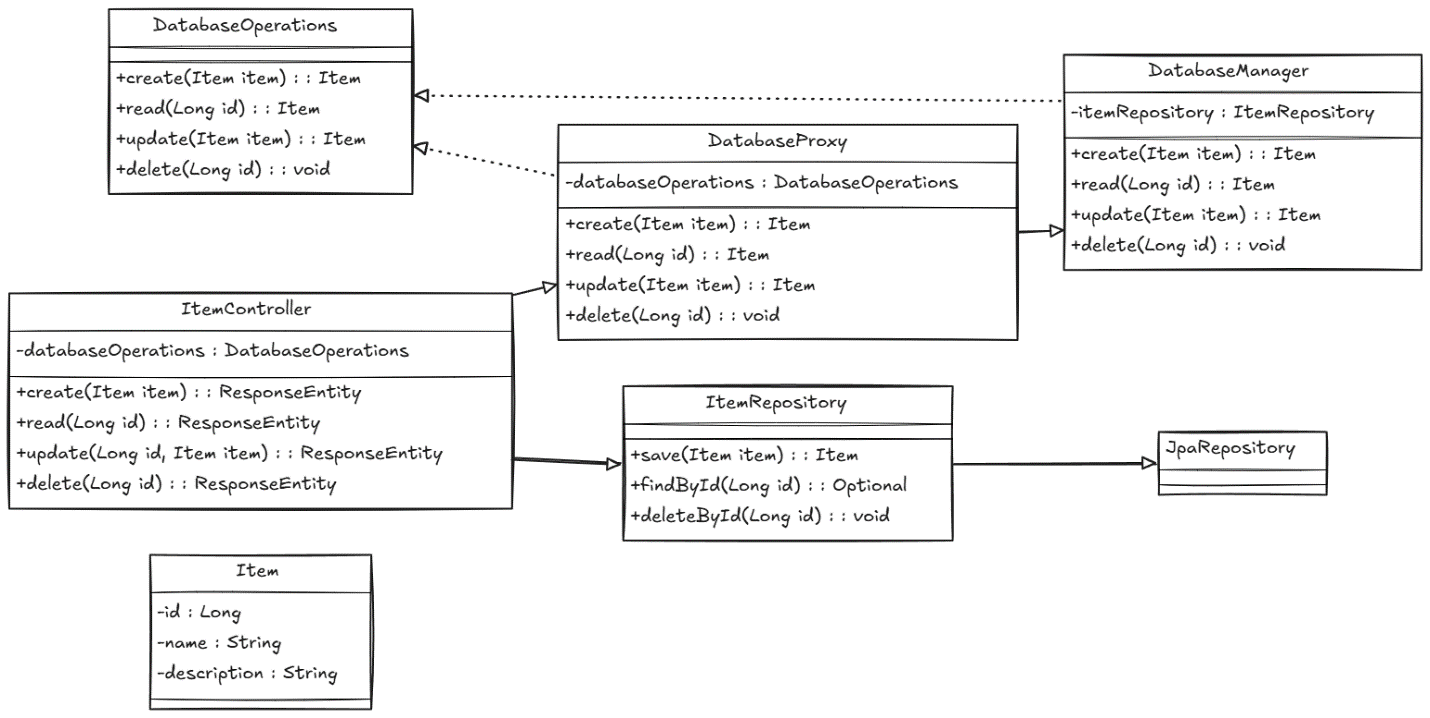
* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design patterns ที่ระบุไว้หรือไม่

จาก Diagram จะเห็นได้ว่า สามารถ ออกแบบ ตามหลักของ Design patterns ที่กำหนดได้อย่างดีเยี่ยม

* 1. **Gemini-Flash**
     1. ผลในรอบที่ 1



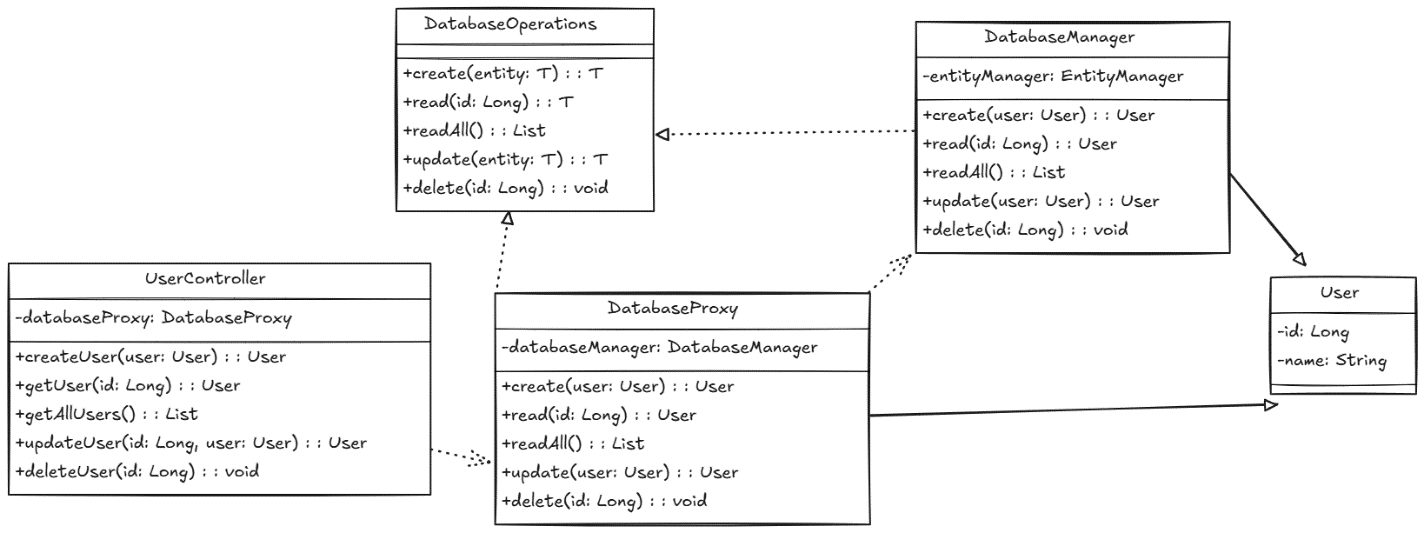
* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design patterns ที่ระบุไว้หรือไม่

จาก Diagram จะเห็นได้ว่า สามารถ ออกแบบ ตามหลักของ Design patterns ที่กำหนดได้อย่างดีเยี่ยม

* + 1. ผลในรอบที่ 2



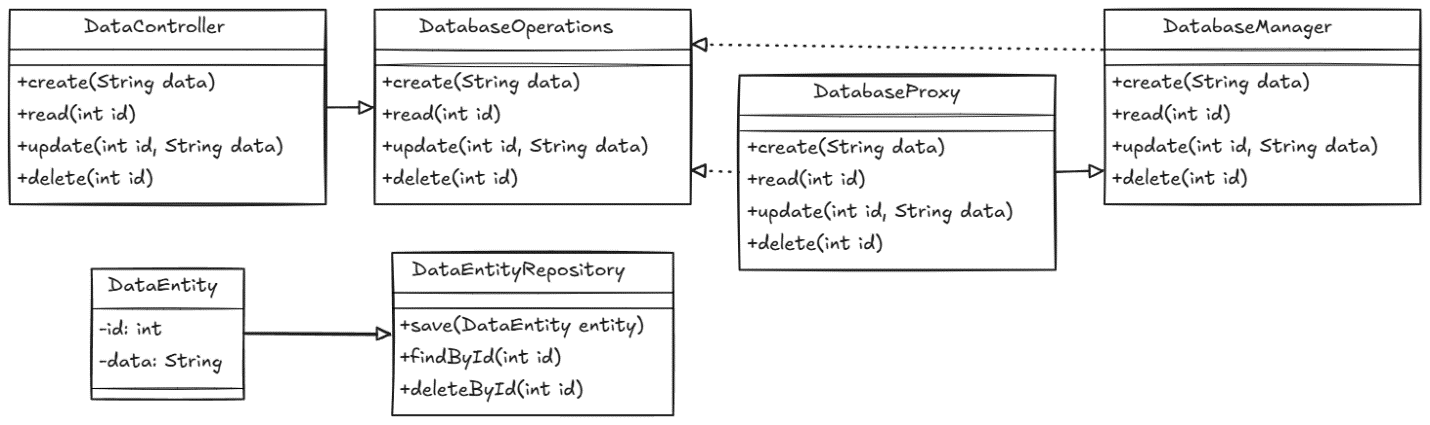
* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design patterns ที่ระบุไว้หรือไม่

จาก Diagram จะเห็นได้ว่า สามารถ ออกแบบ ตามหลักของ Design patterns ที่กำหนดได้อย่างดีเยี่ยม

* + 1. ผลในรอบที่ 3



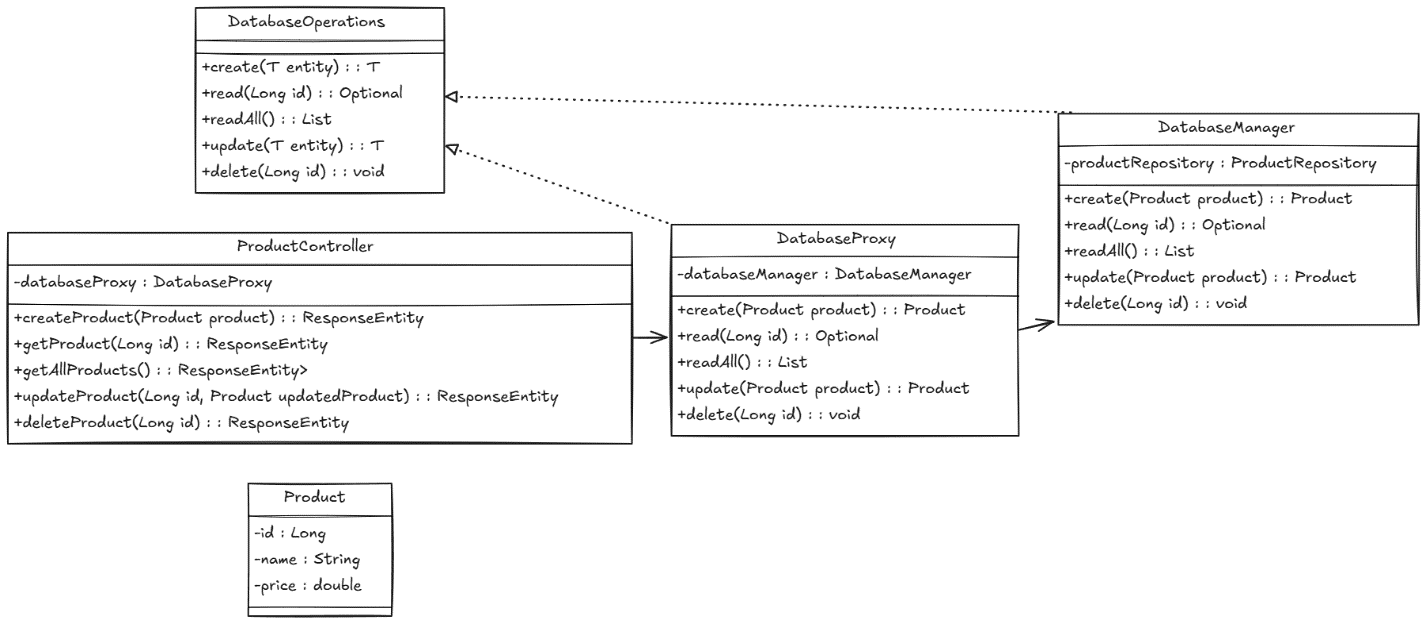
* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

ไม่สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design patterns ที่ระบุไว้หรือไม่

จาก Diagram จะเห็นได้ว่า ไม่สามารถ ออกแบบ ตามหลักของ Design patterns ที่กำหนดได้ โดยมีการเชื่อมต่อที่ผิดหลักของ patterns ที่กำหนด

* 1. **Gemini-Pro**
     1. ผลในรอบที่ 1



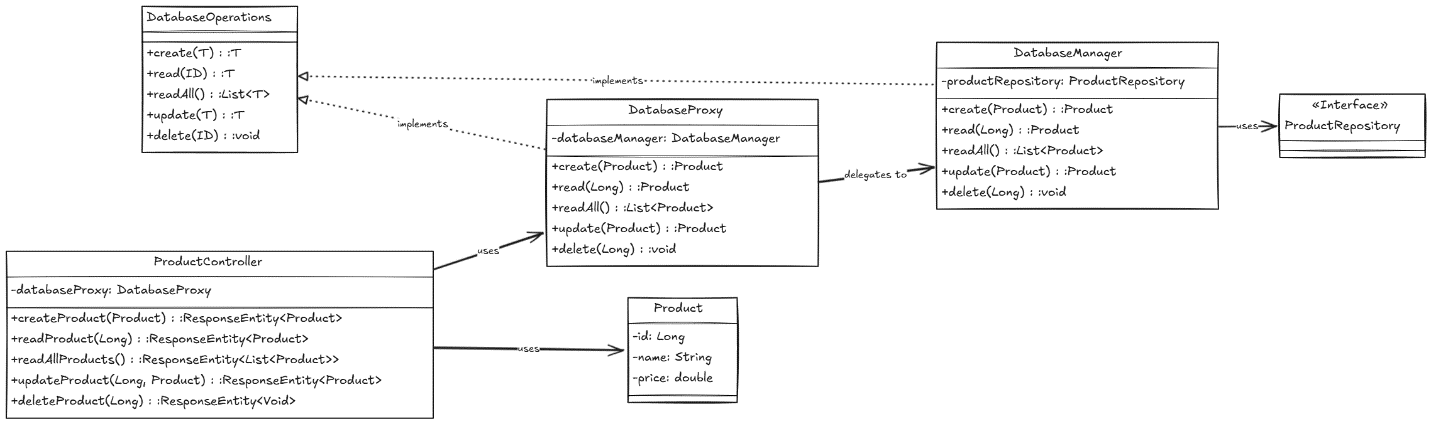
* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design patterns ที่ระบุไว้หรือไม่

จาก Diagram จะเห็นได้ว่า สามารถ ออกแบบ ตามหลักของ Design patterns ที่กำหนดได้อย่างดีเยี่ยม

* + 1. ผลในรอบที่ 2



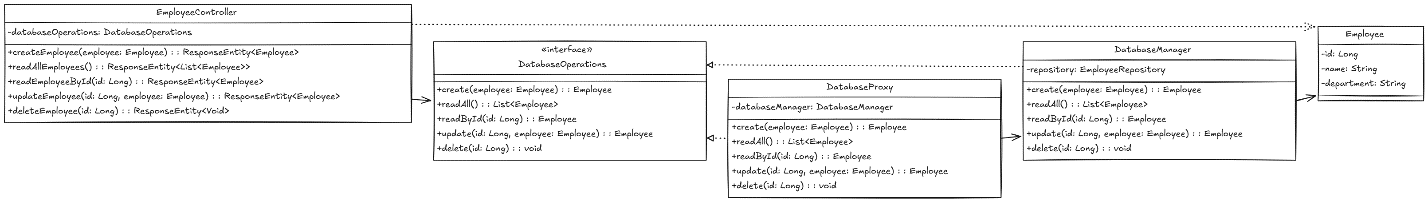
* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design patterns ที่ระบุไว้หรือไม่

จาก Diagram จะเห็นได้ว่า สามารถ ออกแบบ ตามหลักของ Design patterns ที่กำหนดได้อย่างดีเยี่ยม

* + 1. ผลในรอบที่ 3



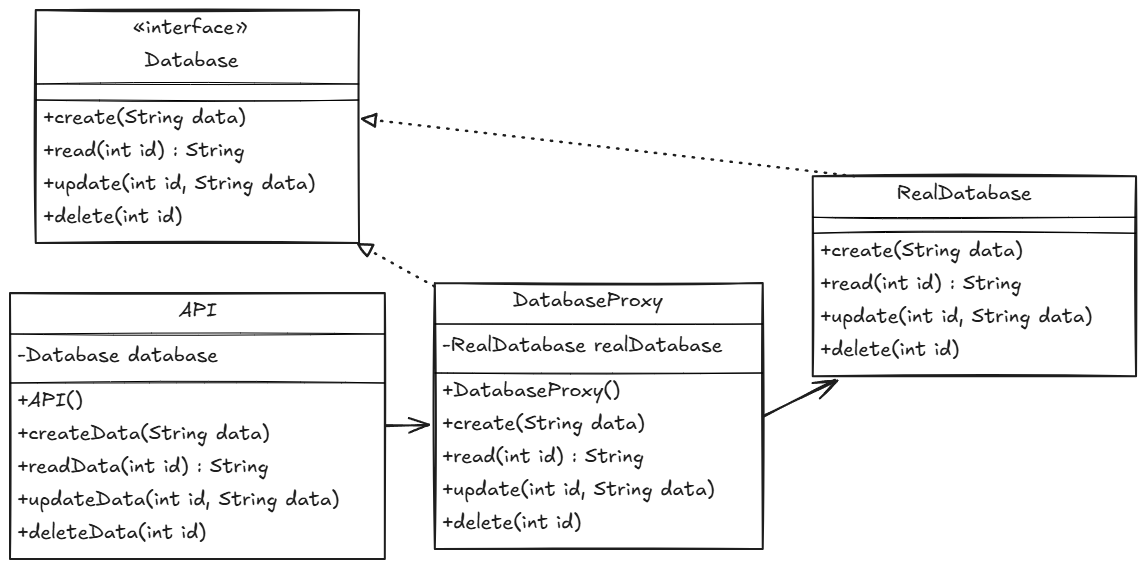
* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่

ไม่สามารถสร้างได้ตามที่ต้องการ

* + - 1. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design patterns ที่ระบุไว้หรือไม่

จาก Diagram จะเห็นได้ว่า ไม่สามารถ ออกแบบ ตามหลักของ Design patterns ที่กำหนดได้ โดยมีการเชื่อมต่อที่ผิดหลักของ patterns ที่กำหนด

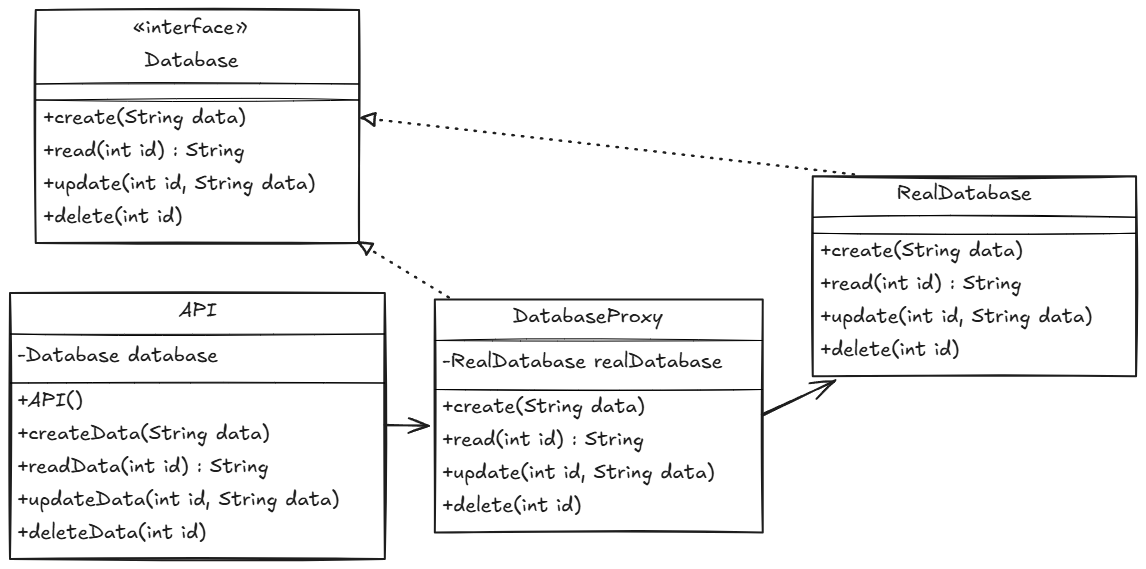
* 1. **Copilot**
     1. ผลในรอบที่ 1



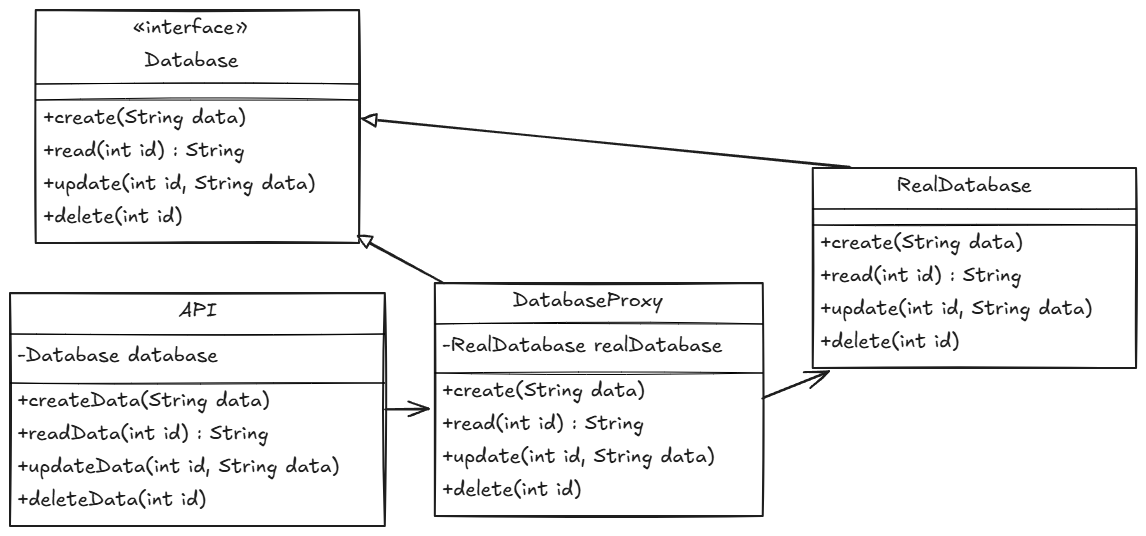
* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่  
         สามารถสร้างได้ตามที่ต้องการ
      2. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design patterns ที่ระบุไว้หรือไม่

จาก Diagram จะเห็นได้ว่า สามารถ ออกแบบ ตามหลักของ Design patterns ที่กำหนดได้อย่างดีเยี่ยม

* + 1. ผลในรอบที่ 2



* + 1. ผลในรอบที่ 3



* + - 1. โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่  
         สามารถสร้างได้ตามที่ต้องการ
      2. โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design patterns ที่ระบุไว้หรือไม่

จาก Diagram จะเห็นได้ว่า สามารถ ออกแบบ ตามหลักของ Design patterns ที่กำหนดได้อย่างดีเยี่ยม

1. **ตารางสรุปผลการทดลอง**
   1. **ตารางสรุปผลการทดลองในภาษา Python**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Generative AI | Round | โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่ | โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่ | สรุปผล |
| 1.ChatGPT | 1 | ได้ | ได้ | ถือว่าทำได้ดี ครบตามที่ร้องขอไป |
| 2 | ได้ | ได้ |
| 3 | ได้ | ได้ |
| 2.Claude | 1 | ได้ | ได้ | ถือว่าดีมาก จากทั้งหมด AI ตัวนี้ดู ตอบได้ดีที่สุด โดยสามารถ แก้ปัญหาเรื่อง Interface ได้  แต่ไม่ฟรี |
| 3.Gemini-Flash | 1 | เกือบได้ | ได้ | เนิ่องจากเป็นตัวเล็กสุด อาจจะทำให้ตอบให้ไม่ครบ หรือลืมในบางส่วนไป |
| 2 | ได้ | ไม่ได้ | เห็นว่า รอบนี้ จะไม่มีการใช้งานตัว Design pattens ที่กำหนดเลย |
| 3 | ได้ | ไม่ได้ |
| 4.Gemini-Pro | 1 | ได้ | ได้ | ถือว่าทำได้ดี ครบตามที่ร้องขอไป  อาจจะตั้งชื่อไม่ตรง กับ คุณสมบัติ |
| 2 | ได้ | ได้ | ถือว่าทำได้ดี ครบตามที่ร้องขอไป แต่อาจจะตั้งชื่อไม่ตรง กับ คุณสมบัติ |
| 3 | ได้ | ได้ | ถือว่าทำได้ดี ครบตามที่ร้องขอไป  อาจจะตั้งชื่อไม่ตรง กับ คุณสมบัติ |
| 5.Copilot | 1 | ได้ | ได้ | ถือว่าทำได้ดี ครบตามที่ร้องขอไป |
| 2 | ได้ | ได้ | ถือว่าทำได้ดี ครบตามที่ร้องขอไป |
| 3 | ได้ | ได้ | ถือว่าทำได้ดี ครบตามที่ร้องขอไป |

* 1. ตารางสรุปผลการทดลองในภาษา Java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Generative AI | Round | โค้ดที่สร้างขึ้นถูกต้องตามข้อกําหนดที่ระบุไว้หรือไม่ | โค้ดที่ถูกสร้างขึ้นใช้ภาษาโปรแกรมและ Design pattern ที่ระบุไว้หรือไม่ | สรุปผล |
| 1.ChatGPT | 1 | ได้ | ได้ | ถือว่าทำได้ดี ครบตามที่ร้องขอไป |
| 2 | ได้ | ได้ |
| 3 | ได้ | ได้ |
| 2.Gemini-Flash | 1 | ได้ | ได้ | ถือว่าทำได้ดี ครบตามที่ร้องขอไป |
| 2 | ได้ | ได้ |
| 3 | ได้ | ไม่ได้ | เกิดการเลือกใช้ Class ผิดทำให้ไม่ถูกตามหลักของ Design patterns |
| 3.Gemini-Pro | 1 | ได้ | ได้ | ถือว่าทำได้ดี ครบตามที่ร้องขอไป |
| 2 | ได้ | ได้ |
| 3 | ได้ | ไม่ได้ | เกิดการเลือกใช้ Class ผิดทำให้ไม่ถูกตามหลักของ Design patterns |
| 4.Copilot | 1 | ได้ | ได้ | ถือว่าทำได้ดี ครบตามที่ร้องขอไป ทั้ง 3 ผลลัพธ์ ให้คำตอบเหมือนกันทั้งหมด |
| 2 | ได้ | ได้ |
| 3 | ได้ | ได้ |

เนื่องจาก Java เป็นภาษา ที่ออกแบบมาสำหรับ OOP (Object-oriented programming) อยู่แล้วทำให้ Generative AI สามารถสร้างได้ตาม Design patterns ที่ต้องการ ผิดกับ Python ที่ไม่ได้ออกแบบให้เป็น OOP ตั้งแต่ แรกทำให้บางอย่างอาจจะไม่สามารถทำได้เหมือนกัน Java แต่ก็ยังถือว่า อยู่ในหลักของ Design patterns นั้นอยู่

1. **ตารางสรุปผลการ****ทำงานของตัวโค้ด**
   1. **ตารางสรุปผลการทำงานของตัวโค้ดในภาษา Python**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Generative AI | Round | สามารถทำงานได้หรือไม่ | ปัญหาที่พบ | สามารถแก้ได้ง่ายหรือไม่ |
| 1.ChatGPT | 1 | ได้ |  |  |
| 2 | ได้ |  |  |
| 3 | ไม่ | AttributeError: 'Flask' object has no attribute 'before\_first\_request'. Did you mean: '\_got\_first\_request'? | ใช่ |
| 2.Gemini-Flash | 1 | ไม่ | NameError: name 'Union' is not defined | ไม่ |
| 2 | ไม่ | Exception: Failed to create product: 'sqlite3.Connection' object has no attribute 'lastrowid' | ไม่ |
| 3 | ได้ |  |  |
| 3.Gemini-Pro | 1 | ไม่ | sqlite3.OperationalError: no such table: books | ใช่ |
| 2 | ได้ |  |  |
| 3 | ได้ |  |  |
| 4.Copilot | 1 | ได้ |  |  |
| 2 | ได้ |  |  |
| 3 | ได้ |  |  |

* 1. **ตารางสรุปผลการทำงานของตัวโต้ดในภาษา Java**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Generative AI | Round | สามารถทำงานได้หรือไม่ | ปัญหาที่พบ | สามารถแก้ได้ง่ายหรือไม่ |
| 1.ChatGPT | 1 | ไม่ | เนื่องจาก มีการที่ใช้ code ไม่ครบโครงสร้าง และเป็นการยากมาก ที่จะแก้ใข โดยที่ไม่ทำการสร้างใหม่ ทำให้จะไม่มีผลการรันทั้งหมดในส่วนของ JAVA | ไม่ |
| 2 | ไม่ |
| 3 | ไม่ |
| 2.Gemini-Flash | 1 | ไม่ |
| 2 | ไม่ |
| 3 | ไม่ |
| 3.Gemini-Pro | 1 | ไม่ |
| 2 | ไม่ |
| 3 | ไม่ |
| 4.Copilot | 1 | ไม่ |
| 2 | ไม่ |
| 3 | ไม่ |

1. **ตารางสรุปผลการทำงานของตัวทดสอบ**
   1. **ตารางสรุปผลการทำงานของตัวทดสอบในภาษา Python**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Generative AI | Round | จำนวนการทดสอบ | ผลการทดสอบ | | |
| ผ่าน | ไม่ผ่าน | ไม่ทำงาน |
| 1.ChatGPT | 1 | 12 | 0 | 12 | 0 |
| 2 | 7 | 1 | 6 | 0 |
| 3 | 6 | 6 | 0 | 0 |
| 2.Gemini-Flash | 1 | 11 | 3 | 8 | 0 |
| 2 | 15 | 4 | 15 | 0 |
| 3 | 17 | 11 | 6 | 0 |
| 3.Gemini-Pro | 1 | 10 | 1 | 9 | 0 |
| 2 | 26 | 3 | 23 | 0 |
| 3 | 12 | 12 | 0 | 0 |
| 4.Copilot | 1 | 10 | 6 | 4 | 0 |
| 2 | 9 | 7 | 2 | 0 |
| 3 | 5 | 5 | 5 | 0 |

* 1. **ตารางสรุปผลการทำงานของตัวทดสอบในภาษา Java**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Generative AI | Round | จำนวนการทดสอบ | ผลการทดสอบ | | |
| ผ่าน | ไม่ผ่าน | ไม่ทำงาน |
| 1.ChatGPT | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 2.Gemini-Flash | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 3.Gemini-Pro | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 4.Copilot | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |

จากการทำงานของตัวโค้ด และ การทดสอบ ทำให้เห็นว่า โค้ดที่ได้ในส่วนของภาษา Python จะเห็นได้ว่า สามารถทำงานได้ เกือบหมด แต่ก็ยังมีปัญหาเพียงเล็กน้อยใน เรื่องของการนำเอาโมดูลที่เก่าแล้วมาใช้ หรือนำสิ่งที่มีสิ่งใหม่แล้ว ทำให้ไม่สามารถทำงานได้ แต่ในทางกับกัน ในตัวของภาษา Java จะเห็นว่า ไม่สามารถทำงานได้เลย เนื่องจากในบางครั้งตัวของ Generative AI ก็ได้มีการนำเอาโมดูลที่เก่ามากมาใช้ หรือการที่ไม่ได้กำหนดโครงสร้างของ Spring Boot ทำให้ไม่สามารถแก้ไขได้ง่ายว่ามีการนำเอา Dependency อะไรเข้ามาใช้ในตัวโครงการบ้างทำให้ผู้ใช้เป็นการยากมากที่จะแก้ไขได้

ในการทดสอบตัวโค้ดจะเห็นได้ว่ามีการทดสอบได้ถึงแม้ว่า จะมีบางรายการที่ทำการทดสอบผ่านน้อยกว่าครึ่ง ในกรณีของ Java ไม่สามารถทดสอบได้เนื่องจากตัวระบบไม่สามารถที่ทำงานได้ เลยไม่สามารถที่จะทดสอบได้เลย

1. **ตารางสรุปผลการครอบคุมของของการทดสอบ**
   1. **ตารางสรุปผลการทำงานของตัวทดสอบในภาษา Python**

|  |  |  |  |
| --- | --- | --- | --- |
| Generative AI | Round | Coverage | Line of Code |
| 1.ChatGPT | 1 | 63% | 94 |
| 2 | 97% | 93 |
| 3 | 97% | 91 |
| 2.Gemini-Flash | 1 | 81% | 139 |
| 2 | 50% | 157 |
| 3 | 67% | 122 |
| 3.Gemini-Pro | 1 | 67% | 82 |
| 2 | 77% | 69 |
| 3 | 100% | 75 |
| 4.Copilot | 1 | 97% | 81 |
| 2 | 94% | 77 |
| 3 | 98% | 74 |

* 1. **ตารางสรุปผลการทำงานของตัวทดสอบในภาษา Java**

|  |  |  |  |
| --- | --- | --- | --- |
| Generative AI | Round | Coverage | Line of Code |
| 1.ChatGPT | 1 | 0% |  |
| 2 | 0% |  |
| 3 | 0% |  |
| 2.Gemini-Flash | 1 | 0% |  |
| 2 | 0% |  |
| 3 | 0% |  |
| 3.Gemini-Pro | 1 | 0% |  |
| 2 | 0% |  |
| 3 | 0% |  |
| 4.Copilot | 1 | 0% |  |
| 2 | 0% |  |
| 3 | 0% |  |

จากตารางสรุปผลการทำงานของตัวทดสอบในภาษา Python และ Java พบว่าการทดสอบโค้ดที่ใช้ Generative AI มีประสิทธิภาพในระดับที่น่าพอใจ โดยในภาษา Python มีการวัด Coverage และจำนวนบรรทัดของโค้ดในหลายรอบการทดสอบ ผลลัพธ์แสดงให้เห็นว่า AI รุ่นต่าง ๆ เช่น ChatGPT, Gemini Flash, Gemini Pro และ Copilot มีความสามารถในการครอบคลุมโค้ดได้ในช่วง 50% ถึง 100% โดยเฉพาะ ChatGPT และ Copilot ที่ให้ผลครอบคลุมถึง 97-100% ในหลายรอบการทดสอบ ซึ่งเป็นผลที่ดีสำหรับการทดสอบการครอบคลุมของโค้ด

จากผลสรุปพบว่า แม้ว่าการทดสอบด้วย Generative AI จะไม่สามารถทำให้ Coverage ถึง 100% ในทุกครั้ง แต่ผลที่ได้ในช่วง 75% - 90% นับว่าดีมากสำหรับการทดสอบโปรแกรม แม้บางครั้งผลลัพธ์จะต่ำกว่านั้น แต่ในภาพรวมก็แสดงให้เห็นว่า AI ทำงานได้ดีในระดับที่น่าพอใจ

1. **ตารางสรุปผลการจากการทำ static code analysis tools**

static code analysis tools ที่เลือกใช้คือ SonarCube

SonarQube เป็นเครื่องมือที่ได้รับความนิยมอย่างแพร่หลายในวงการพัฒนาซอฟต์แวร์สำหรับการวิเคราะห์โค้ดแบบสถิต เนื่องจากมีความสามารถในการรองรับภาษาโปรแกรมที่หลากหลาย เช่น Java Python C# และ PHP ทำให้สามารถใช้ได้กับโครงการที่มีการพัฒนาในหลายภาษา นอกจากนี้ SonarQube ยังมีประสิทธิภาพในการตรวจจับปัญหาที่เกี่ยวข้องกับคุณภาพของโค้ด รวมถึงการจัดการตัวแปร การจัดรูปแบบโค้ด และการตรวจสอบช่องโหว่ด้านความปลอดภัย เช่น การโจมตีแบบ SQL Injection หรือ Cross-Site Scripting ซึ่งช่วยเพิ่มความมั่นใจในความปลอดภัยของโค้ดที่พัฒนา

อีกทั้ง SonarQube ยังสามารถรวมเข้ากับกระบวนการ Continuous Integration และ Continuous Deployment (CI/CD) ได้อย่างราบรื่น ทำให้สามารถตรวจสอบคุณภาพโค้ดได้อย่างต่อเนื่องตลอดกระบวนการพัฒนา และด้วยระบบรายงานผลที่ชัดเจนและเข้าใจง่าย นักพัฒนาสามารถดูข้อมูลเกี่ยวกับคุณภาพโค้ดและปรับปรุงได้ทันที การเลือกใช้ SonarQube จึงเป็นการเพิ่มประสิทธิภาพและความปลอดภัยของโครงการพัฒนาในทุกขั้นตอนของการทำงาน

* 1. **ตารางสรุปผลการจากการทำ static code analysis tools ในภาษา Python**
     1. **ในด้านความปลอดภัย**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Generative AI | Round | Line of Code | Low Impact | Medium Impact | High Impact |
| 1.ChatGPT | 1 | 94 | 0 | 0 | 0 |
| 2 | 93 | 0 | 0 | 0 |
| 3 | 91 | 0 | 0 | 0 |
| 2.Gemini-Flash | 1 | 139 | 0 | 0 | 0 |
| 2 | 157 | 0 | 0 | 0 |
| 3 | 122 | 0 | 0 | 0 |
| 3.Gemini-Pro | 1 | 82 | 0 | 0 | 0 |
| 2 | 69 | 0 | 0 | 0 |
| 3 | 75 | 0 | 0 | 0 |
| 4.Copilot | 1 | 81 | 0 | 0 | 0 |
| 2 | 77 | 0 | 0 | 0 |
| 3 | 74 | 0 | 0 | 0 |

* + 1. **ด้านความเชื่อถือได้**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Generative AI | Round | Line of Code | Low Impact | Medium Impact | High Impact |
| 1.ChatGPT | 1 | 94 | 0 | 0 | 0 |
| 2 | 93 | 0 | 0 | 0 |
| 3 | 91 | 0 | 0 | 0 |
| 2.Gemini-Flash | 1 | 139 | 0 | 0 | 0 |
| 2 | 157 | 0 | 0 | 0 |
| 3 | 122 | 0 | 0 | 0 |
| 3.Gemini-Pro | 1 | 82 | 0 | 0 | 0 |
| 2 | 69 | 0 | 0 | 0 |
| 3 | 75 | 0 | 0 | 0 |
| 4.Copilot | 1 | 81 | 0 | 0 | 0 |
| 2 | 77 | 0 | 0 | 0 |
| 3 | 74 | 0 | 0 | 0 |

* + 1. **ด้านการบำรุงรักษา**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Generative AI | Round | Line of Code | Low Impact | Medium Impact | High Impact |
| 1.ChatGPT | 1 | 94 | 0 | 0 | 0 |
| 2 | 93 | 0 | 0 | 0 |
| 3 | 91 | 0 | 0 | 0 |
| 2.Gemini-Flash | 1 | 139 | 0 | 14 | 0 |
| 2 | 157 | 0 | 5 | 0 |
| 3 | 122 | 0 | 1 | 0 |
| 3.Gemini-Pro | 1 | 82 | 0 | 0 | 0 |
| 2 | 69 | 0 | 0 | 0 |
| 3 | 75 | 0 | 0 | 0 |
| 4.Copilot | 1 | 81 | 0 | 0 | 0 |
| 2 | 77 | 0 | 0 | 0 |
| 3 | 74 | 0 | 0 | 0 |

จากการวิเคราะห์โค้ด Python ในส่วนของ Proxy โดยพิจารณาจากสามด้านหลัก ได้แก่ Security Reliability และ Maintainability พบว่า

1. **ด้านความปลอดภัย (Security)** ไม่มีโค้ดใดที่มีปัญหาในระดับ Low Medium หรือ High Impact ซึ่งแสดงให้เห็นว่าไม่มีช่องโหว่ทางความปลอดภัยที่ตรวจพบในโค้ดทั้งหมด ทั้งจาก ChatGPT Gemini Flash Gemini Pro และ Github Copilot
2. **ด้านความเชื่อถือได้ (Reliability)**: เช่นเดียวกันกับด้านความปลอดภัย ไม่มีปัญหาที่ตรวจพบในทุกระดับความรุนแรง แสดงถึงความเสถียรในการทำงานของโค้ดที่พัฒนา
3. **ด้านการบำรุงรักษา (Maintainability)**: พบปัญหาที่ระดับ Medium Impact จำนวน 20 จุด โดยทั้งหมดมาจาก Gemini Flash ซึ่งมี 14 จุดในรุ่นที่ 1 5 จุดในรุ่นที่ 2 และ 1 จุดในรุ่นที่ 3 ส่วนโค้ดจาก ChatGPT Gemini Pro และ Github Copilot ไม่มีปัญหาใด ๆ ในด้านนี้

โดยรวมแล้ว โค้ดจากหลายแหล่งมีคุณภาพดีในแง่ของความปลอดภัยและความเสถียร แต่ในด้านการบำรุงรักษายังมีบางจุดที่สามารถปรับปรุงได้ โดยเฉพาะในโค้ดของ Gemini Flash

* 1. **ตารางสรุปผลการจากการทำ static code analysis tools ในภาษา Java**
     1. **ในด้านความปลอดภัย**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Generative AI | Round | Line of Code | Low Impact | Medium Impact | High Impact |
| 1.ChatGPT | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 2.Gemini-Flash | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 3.Gemini-Pro | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 4.Copilot | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |

* + 1. **ด้านความเชื่อถือได้**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Generative AI | Round | Line of Code | Low Impact | Medium Impact | High Impact |
| 1.ChatGPT | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 2.Gemini-Flash | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 3.Gemini-Pro | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 4.Copilot | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |

* + 1. **ด้านการบำรุงรักษา**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Generative AI | Round | Line of Code | Low Impact | Medium Impact | High Impact |
| 1.ChatGPT | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 2.Gemini-Flash | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 3.Gemini-Pro | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 4.Copilot | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |