


전자적 자판기 시점

자판기

상품선택


재고: 5



₩1,500

에비앙(탄산)


재고: 10



₩1,500

햇식스


재고: 10



₩1,800

바리스타


재고: 9



₩1,200

레스비


재고: 10



₩1,600

포카리


재고: 10



₩1,700

데자와


재고: 9



₩2,000

카페라테


재고: 8



₩1,300

몬스터


재고: 10



₩900

삼다수


재고: 10



₩1,300

솔익논


재고: 10



₩1,500

베지밀비

재고: 9



₩1,200

이프루

음료 나오는 곳

추천해주세요!

즐려요

다이어트

탄산

무카페인

아무거나

지폐투입구


1000

5000

10000

카드투입구

카드를 넣으세요



싱글톤

인공지능학과 2023012533 오윤

Contents

설계

구현

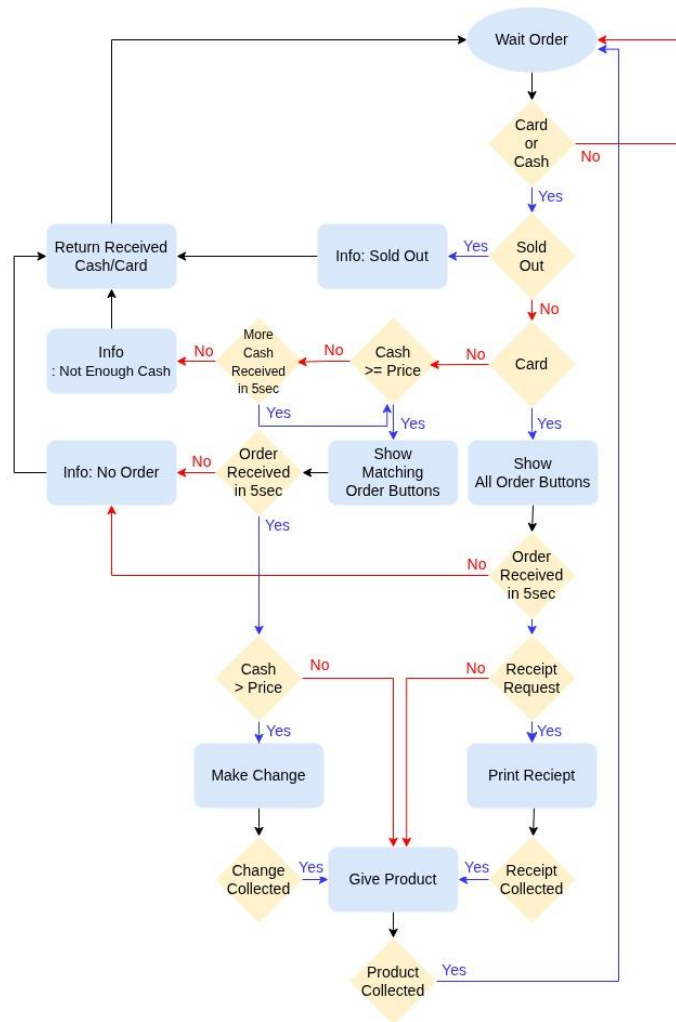
시연

고민



설계(1)

Flow Chart



구매 시나리오

```

sequenceDiagram
    actor User
    participant GUI
    participant Controller
    participant PM as :Payment Manager
    participant DBM as :DB Manager

    User->>GUI: Input Payment
    GUI->>Controller: Payment Data (Type, Amount)
    Controller->>PM: Payment Data (Type, Amount)
    PM->>Controller: Available Products
    Controller->>GUI: Available Products
    alt opt [all product stock = 0]
        Controller->>GUI: Return Card/Cash
    end
    User->>GUI: Place Order
    GUI->>Controller: Order Data (Product Name)
    Controller->>PM: Order Data (Product Name)
    PM->>PM: Process Payment
    PM->>DBM: Insert Order
    DBM->>PM: Insert OK
    PM->>DBM: Update Stock
    DBM->>PM: Update OK
    PM->>Controller: Order OK
    Controller->>GUI: Give Product
    alt alt
        User->>GUI: Receive Product
        alt [cash & change > 0]
            GUI->>Controller: Received Check
            Controller->>PM: Change Req
            PM->>PM: Calc Change
            PM->>Controller: Change
            Controller->>GUI: Give Change
        else [card]
            GUI->>Controller: Received Check
            Controller->>Controller: Check if Receipt needed
        end
    end
    alt alt
        User->>GUI: Receipt Yes/No
        alt [Yes]
            GUI->>Controller: Receipt Req
            Controller->>PM: Make Receipt
            PM->>DBM: Insert Receipt
            DBM->>PM: Insert OK
            PM->>Controller: Receipt
            Controller->>GUI: Give Receipt & Return Card
        else [No]
            GUI->>Controller: Return Card
        end
    end
  
```

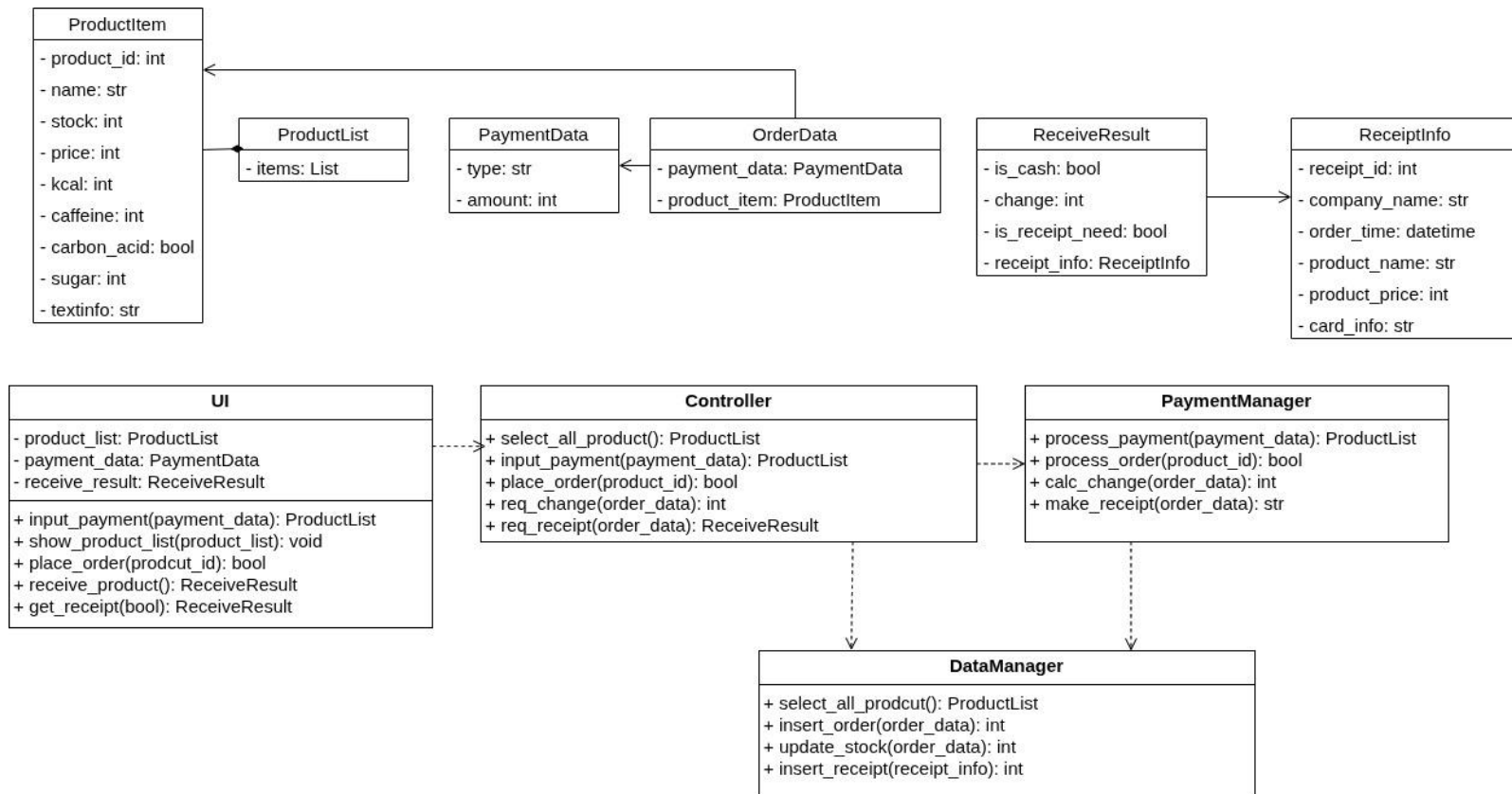
The diagram illustrates the following process flow:

- Initial State:** The GUI is ready to receive input from the user.
- Payment Process:**
 - User provides **Input Payment** to the GUI.
 - GUI sends **Payment Data (Type, Amount)** to the Controller.
 - Controller sends **Payment Data (Type, Amount)** to the Payment Manager.
 - Payment Manager returns **Available Products** to the Controller.
 - Controller returns **Available Products** to the GUI.
 - Optional:** If **all product stock = 0**, the Controller sends **Return Card/Cash** to the GUI.
- Order Process:**
 - User provides **Place Order** to the GUI.
 - GUI sends **Order Data (Product Name)** to the Controller.
 - Controller sends **Order Data (Product Name)** to the Payment Manager.
 - Payment Manager performs **Process Payment** internally.
 - Payment Manager sends **Insert Order** to the DB Manager.
 - DB Manager returns **Insert OK** to the Payment Manager.
 - Payment Manager sends **Update Stock** to the DB Manager.
 - DB Manager returns **Update OK** to the Payment Manager.
 - Payment Manager sends **Order OK** to the Controller.
 - Controller sends **Give Product** to the GUI.
- Receipt Process (Alternative Flows):**
 - User provides **Receive Product** to the GUI.
 - Alternative 1 (Cash):** If **[cash & change > 0]**, the GUI sends **Received Check** to the Controller. The Controller sends **Change Req** to the Payment Manager, who performs **Calc Change** and returns **Change** to the Controller. The Controller then sends **Give Change** to the GUI.
 - Alternative 2 (Card):** If **[card]**, the GUI sends **Received Check** to the Controller, who then performs **Check if Receipt needed** internally.
- Receipt Confirmation (Alternative Flows):**
 - User provides **Receipt Yes/No** to the GUI.
 - Yes Path:** If **[Yes]**, the GUI sends **Receipt Req** to the Controller. The Controller sends **Make Receipt** to the Payment Manager, who sends **Insert Receipt** to the DB Manager. The DB Manager returns **Insert OK**, and the Payment Manager returns **Receipt** to the Controller. The Controller then sends **Give Receipt & Return Card** to the GUI.
 - No Path:** If **[No]**, the GUI sends **Return Card** to the Controller.

설계(3)

구성 요소

(클래스 다이어그램)



설계(4)

ERD

상품 Product

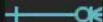
🔑	상품ID	id	int	NOT NULL
	상품명	name	varchar(20)	NOT NULL
	재고	stock	int	NOT NULL
	가격	price	int	NOT NULL
	회사명	company_name	varchar(20)	NOT NULL
	칼로리	kcal	int	NOT NULL
	카페인	caffeine	int	NULL
	탄산	carbon_acid	boolean	NOT NULL
	당	sugar	int	NOT NULL
	텍스트	text_info	varchar(256)	NULL
	이미지경로	image_path	varchar(256)	NOT NULL
	수정시점	edited_at	timestamp	NULL

주문 Order

🔑	주문ID	id	int	NOT NULL
🔑	상품ID	product_id	int	NOT NULL
	주문시점	order_at	timestamp	NOT NULL
	결제타입	payment_type	char(4)	NOT NULL
	현금	cash_amount	int	NULL
	카드	card_info	varchar(50)	NULL
	거스름돈	cash_change	int	NULL

영수증 Receipt

🔑	영수증ID	id	int	NOT NULL
🔑	주문ID	order_id	int	NOT NULL



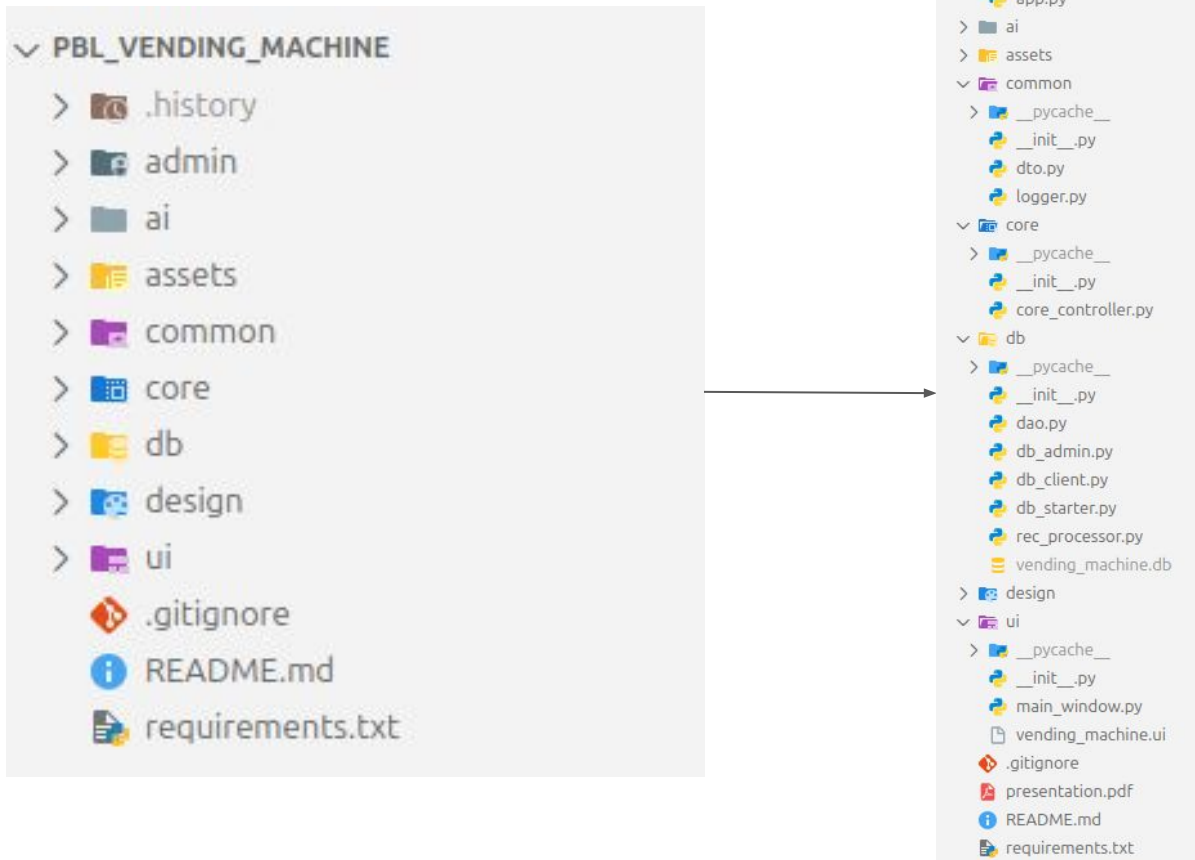
구현(1)

개발 환경

운영체제	Ubuntu 24.04 LTS
버전관리 시스템	Git, Github
개발도구	Visual Studio Code, DBeaver
개발 언어 및 프레임워크	<ol style="list-style-type: none">Python3.12.3<ul style="list-style-type: none">PyQt5==5.15.11Flask==3.1.1sqlalchemy==2.0.41opencv-python==4.11.0.86mediapipe==0.10.21scikit-learn==1.6.1joblib==1.5.1JavaScriptSQL - SQLite, SQL AlchemyQt5

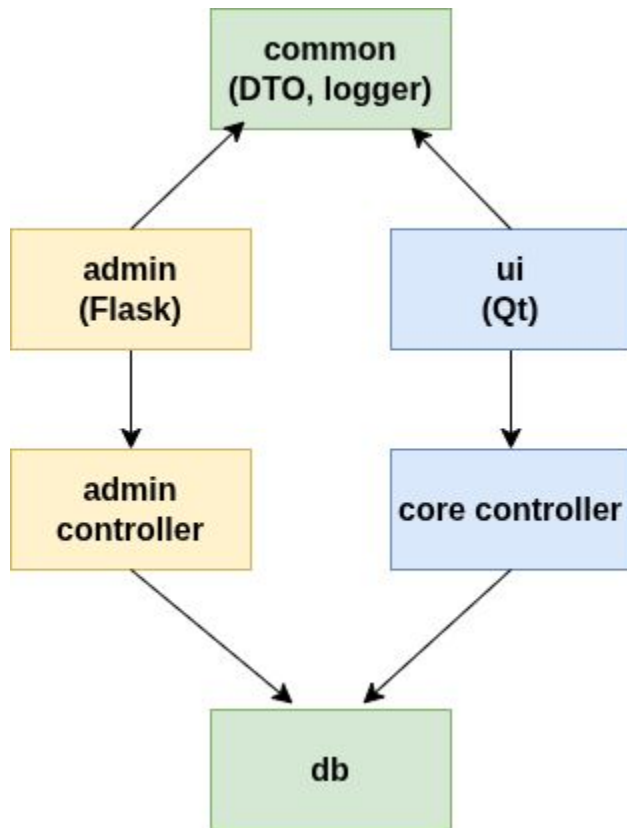
구현(2)

프로젝트 구조



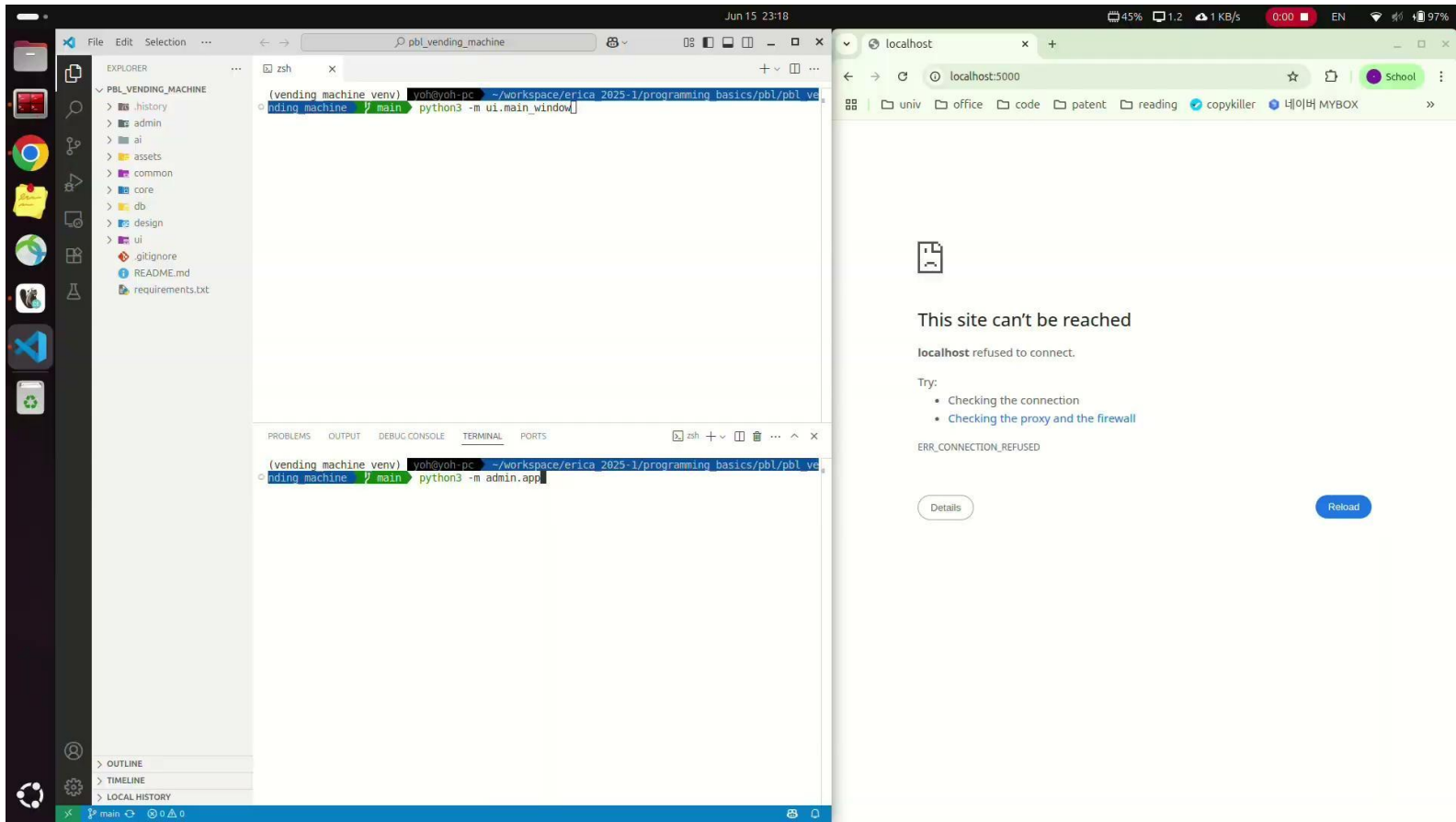
구현(3)

블록 다이어그램



시연(1)

상품 구매: 직접 선택, 현금 결제, 거스름돈



시연(2)

상품 구매: 추천, 카드 결제, 영수증

File Edit Selection ...

pbl_vending_machine

Jun 15 23:20

48% 0.9 201 B/s 0:00 EN 97%

주문 목록

localhost:5000/orders

univ office code patent reading copykiller 네이버 MYBOX

Home

상품현황

주문이력

검색...

주문 번호	상품 명	업체 명	주문시각	결제 수단	현금결제	거스름 돈	영수 증
1	카페 라테	매일 유업	2025-06-15 23:18:31	현금	5000	3000	N

자판기

상품선택

지폐나 카드 투입 후 주문 가능합니다. *재고 없으면 주문 불가

추천해주세요!

지폐투입구

카드투입구

즐리요

다이어트

탄산

무카페인

아무거나

1000

5000

10000

카드를 넣으세요

제품: 10

제품: 10

제품: 10

제품: 10

제품: 10

제품: 10

제품: 9

제품: 10

제품: 10

제품: 10

제품: 10

제품: 10

제품: 10

제품: 10

제품: 10

제품: 10

W1,500

W1,500

W1,800

W1,200

W1,600

W1,700

W2,000

W1,300

W900

W1,300

W1,500

W1,200

에비앙(탄산)

햇식스

바리스타

레쓰비

포카리

데자와

카페라테

몬스터


삼다수

솔이논

베지밀비

이프로

음료 나오는 곳



2025-06-15 23:18:36,319 [INFO] werkzeug: 127.0.0.1 - - [15/Jun/2025 23:18:36] "GET /orders HTTP/1.1" 200 -

시연(3)

상품 구매: 모든 상품 재고 소진

Jun 15 23:54

46% 1.2 564 B/s 0:00 한 97%

상품 현황









localhost:5000/inventory

univ office code patent reading copykiller 네이버 MYBOX Office365 TTS UCF Crime Dat... HuggingFace-... Simple Icons Perplexity ERDCloud Claude

Home
상품현황
주문이력

검색...

판매중단

상품번호	상품명	재고	가격	업체명	칼로리	카페인	탄산	당	이미지	수정	최종 수정 시점
1	에비앙(탄산)	10	1500	Evian	0	0	없음	0			-
2	햇식스	10	1500	Hot6	100	200	있음	50			-
3	바리스타	10	1800	Maeil	120	60	없음	15			-
4	레쓰비	10	1200	롯데칠성	100	50	없음	18			-

시연(4)









관리자 페이지: 상품 수정

Home

상품현황

주문이력

판매중단

상품번호	상품명	재고	가격	업체명	칼로리	카페인	탄산	당	이미지	수정	최종 수정 시점
1	에비앙(탄산)	10	1500	Evian	0	0	없음	0			2025-06-16 00:09:03
2	핫식스	0	1500	Hot6	100	200	있음	50			2025-06-16 00:08:55
3	바리스타	10	1800	Maeil	120	60	없음	15			-
4	레쓰비	10	1200	롯데칠성	100	50	없음	18			-

고민(1)

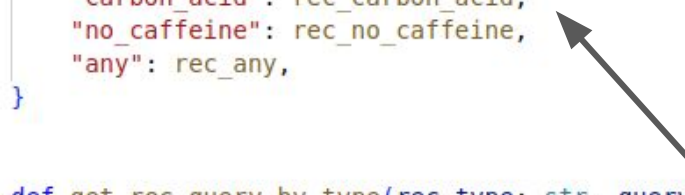
추천 기능: 전략 패턴

db_client.py

```
def get_rec_product(rec_type):  
    with Session() as session:  
        base_query = session.query(Product)  
        rec_query = get_rec_query_by_type(rec_type, base_query)  
        product = rec_query.first()  
        return product
```

rec_processor.py

```
RECOMMEND_STRATEGIES = {  
    "caffeine": rec_caffeine,  
    "kcal": rec_kcal,  
    "carbon_acid": rec_carbon_acid,  
    "no_caffeine": rec_no_caffeine,  
    "any": rec_any,  
}  
  
def get_rec_query_by_type(rec_type: str, query: Query):  
    strategy = RECOMMEND_STRATEGIES.get(rec_type)  
    if not strategy:  
        raise ValueError("알 수 없는 추천 타입입니다.")  
    return strategy(query)
```















고민(2)

주문 후 재고 변경 표시

```
def show_product_list(self, is_starting):
```

상품선택

 ₩1,500 에비앙(탄산)	 ₩1,500 핫식스	 ₩1,800 바리스타	 ₩1,200 레쓰비
 ₩1,600 포카리	 ₩1,700 데자와	 ₩2,000 카페라테	 ₩1,300 몬스터
 ₩900 삼다수	 ₩1,300 솔의눈	 ₩1,500 베지밀비	 ₩1,200 이프로

음료 나오는 곳

추천해주세요!

즐려요

다이어트

탄산

무카페인

아무거나

지메투입구

1000

5000


10000

현금 ₩5,000













꺼내기

카드투입구

결제중...



상품선택

 ₩1,500 에비앙(탄산)	 ₩1,500 핫식스	 ₩1,800 바리스타	 ₩1,200 레쓰비
 ₩1,600 포카리	 ₩1,700 데자와	 ₩2,000 카페라테	 ₩1,300 몬스터
 ₩900 삼다수	 ₩1,300 솔의눈	 ₩1,500 베지밀비	 ₩1,200 이프로

음료 나오는 곳

추천해주세요!

즐려요

다이어트

탄산

무카페인

아무거나

지메투입구

1000

5000


10000

현금 ₩3,000

꺼내기

카드투입구

결제중...



받았습니다!

고민(3)

로깅: 싱글톤 패턴

admin.admin_controller.py

```
from common.logger import LogManager
```

```
logger = LogManager().get_logger()
```

db.db_admin.py

```
from common.logger import LogManager
```

```
logger = LogManager().get_logger()
```

2025-06-16 11:50:30,270	[DEBUG]	[vending_machine]	admin_controller	- get_all_orders
2025-06-16 11:50:30,270	[INFO]	[vending_machine]	db_admin	- get_all_orders

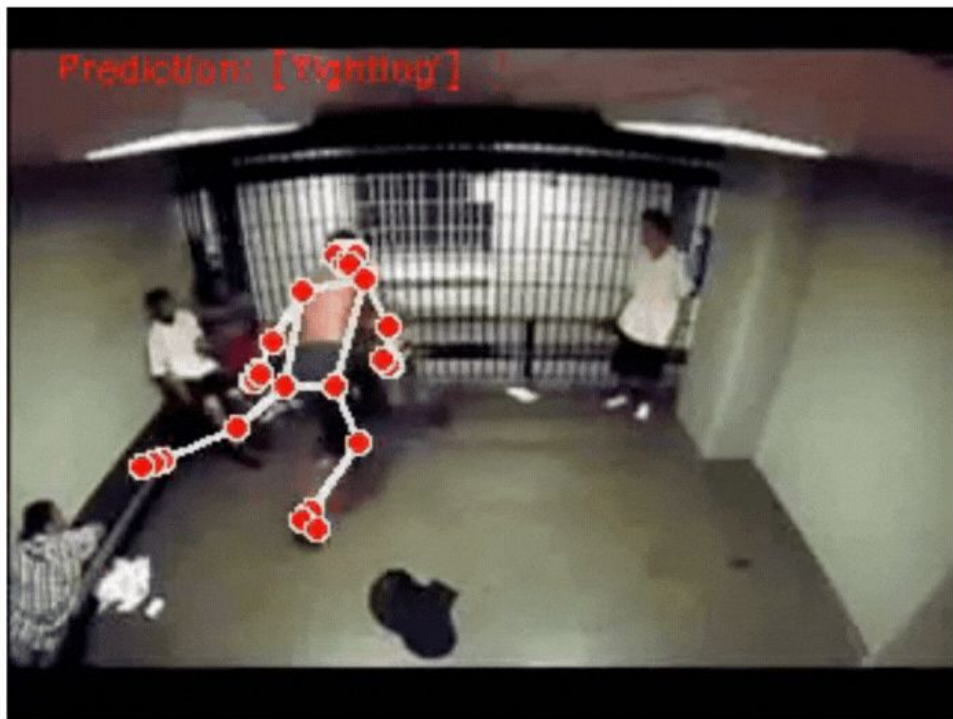
로그 생성 시점

로그레벨

로거 이름

파일 이름

로그 메시지



Q&A





감사합니다
Thank you