


## 어드벤처 디자인 프로젝트 계획서

팀 명	ANOPOS - 그날 본 포스기의 이름을 우리는 아직 모른다				
작 품 명	ANOPOS				
개발기간	2021년 11월 11일 ~ 2021년 12월 10일				
지도교수	컴퓨터공학과			윤 현 주	
구 분	학년	학 번	성 명	휴대전화	E-mail
책임자(팀장)	2	20200123	김다은	X	
팀원	2	20180287	김준용		

어드벤처 디자인 과목에서 수행하는 프로젝트 계획서를 첨부와 같이 제출합니다.

2021년 11월 11일

팀 장 김다은 

컴퓨터공학심화프로그램

---

# 프로젝트 계획서

## - ANOPOS -

### 1. 개요

**프로젝트의 목표** : RPG 게임 속 용사들에게 필요한 아이템 및 식사를 판매하는 POS기 제작

**프로젝트의 목적** : 게임 속 필요한 아이템을 직접 습득하기 위해서는 불편적으로 많은 시간과 노력이 요구됨. 이를 상점 기능을 하는 POS기를 통해 용사들에게 필요한 아이템 및 식사를 제공함으로써 편의성을 향상시킨다.

### 2. 개발 내용

#### 2.1 사용 시나리오 (요구 분석)

##### [시나리오]

용사가 아이템을 주문한다. (식사 주문의 경우도 동일)

##### [액터]

사용자(POS기 사용자)

##### [액션 흐름]

1. 고객이 필요한 아이템을 주문한다.
  2. 사용자가 메인 메뉴의 “명단 등록 메뉴”를 선택한다.
  3. 시스템이 명단 등록 화면(테이블)로 화면을 전환 한다.
  4. 사용자가 명단 등록 대화창에서 테이크아웃을 선택한다.
  5. 시스템이 주문목록 대화창을 출력한다.
  6. 사용자가 주문 받은 아이템을 추가한다.
  7. 시스템이 해당 아이템의 잔여수량을 파악한다.
  8. 시스템이 해당 아이템을 주문목록에 추가한다.
  9. 고객의 요청에 따라, 사용자가 카드결제 혹은 현금결제를 선택한다.
  10. 현금결제의 경우 시스템이 거스름 돈을 계산하여 출력한다.
  11. 사용자가 고객에게 아이템을 제공한다.
-

### [대체 시나리오]

주문한 수량보다 현재 재고가 부족한 경우

### [액터]

사용자(POS기 사용자)

### [액션 흐름]

1. 재고 보충 여부를 출력한다.

(case : 1 보충을 선택)

2. 사용자가 재고 보충을 선택한다.

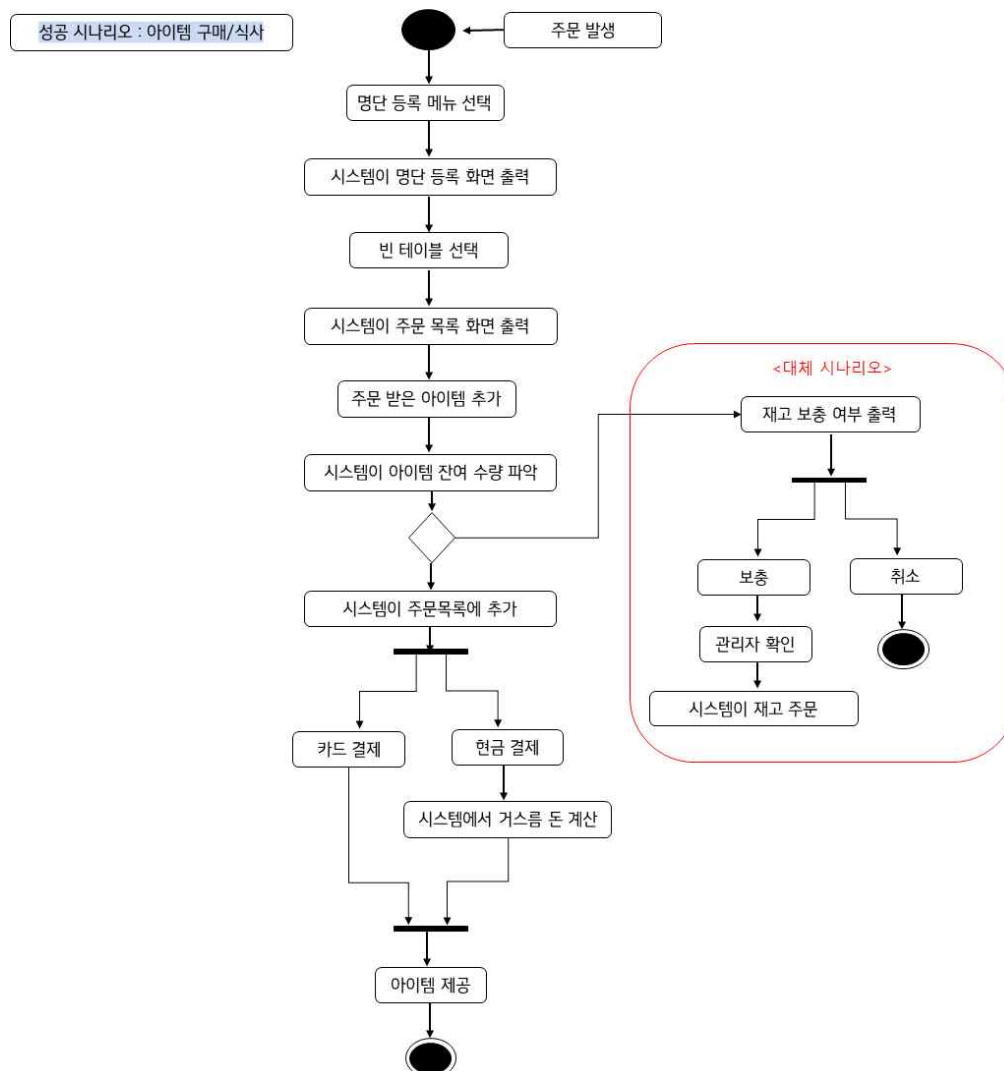
3. 시스템이 재고 주문 권한이 있는지 확인한다.

4. 시스템이 재고를 주문한다.

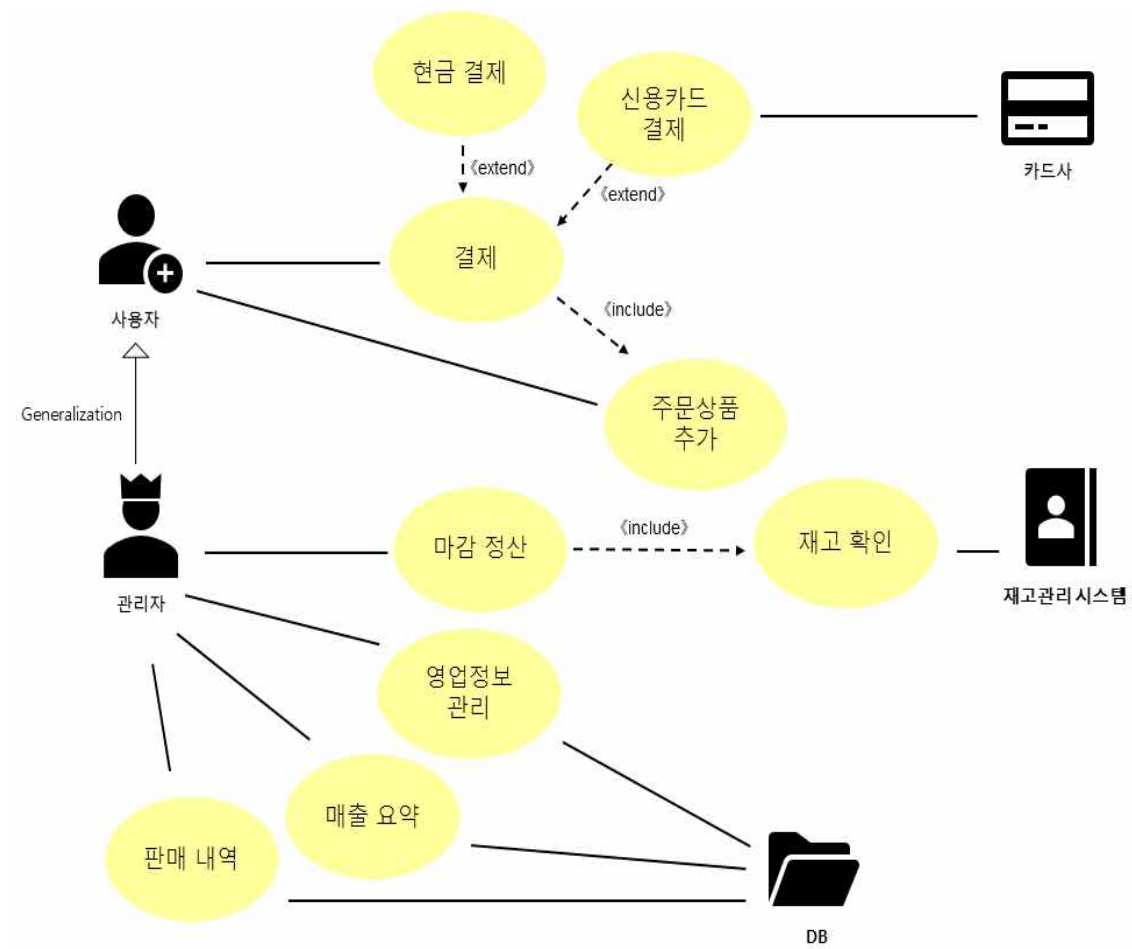
(case : 2 취소를 선택)

2. 취소를 선택할 경우 주문을 취소한다.

## ■ 시나리오를 표현한 상태 차트



## ■ USE\_CASE



## 2.2 설계 구성요소

### [목표 설정]

#### ■ 주문 관리

- 테이블 별, 주문 별(테이크 아웃) 주문이 가능하도록 한다.
- 주문 현황판에 각 메뉴별 단가와 수량 기록이 되도록 한다.
- 고객의 요구에 따른 총액만 기록된 영수증 또는 메뉴명이 포함된 영수증을 따로 출력할 수 있도록 한다.
- 해당 영수증에는 현재 시각이 포함되도록 한다.
- 지불 수단에 따른 결제가 가능하도록 한다. 현금의 경우 고객의 지불 액에 따른 거스름돈을 반환한다.

#### ■ 매출 관리

- 일별/주별/월별/연도별 총 매출액, 각 기간별 메뉴별 매출액, 메뉴별 주문 수 등 시간을 고려한 통계 정보를 출력한다.

#### ■ 재고 관리

- 판매 품목들과 메뉴에 들어가는 재료의 양을 정하고 품목 / 재료별 재고를 관리한다.
- csv 파일에 판매 품목의 정보를 저장하여 사용한다.
- 재고 상황을 항상 확인할 수 있도록 한다. 만약 재고가 없는 경우 재고가 보충되기 전까지 판매할 수 없도록 한다.

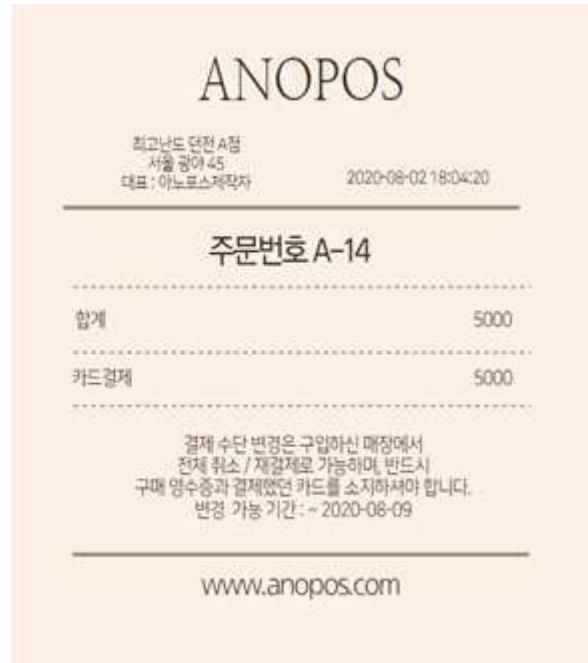
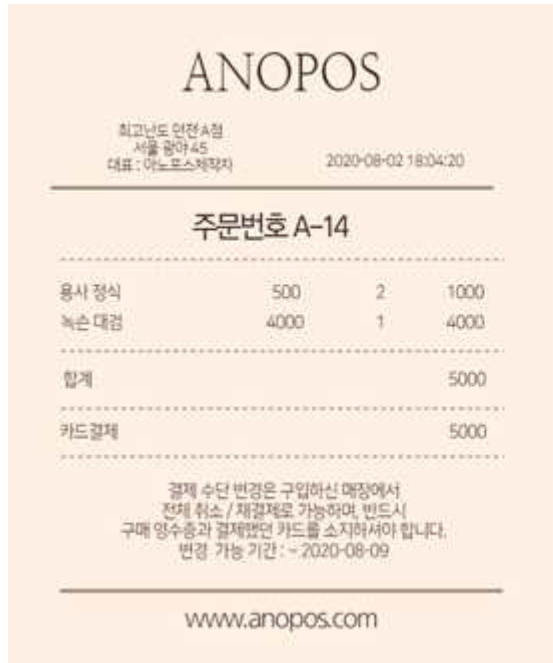
	A	B	C	D	E	F	G	H
1	상품 코드	카테고리	판매가	품명	하위품목1	하위 품목2	하위 품목3	재고
2	1	골렘	2	덩굴	-1	-1	-1	5
3	2	골렘	3000	골렘 왕의	-1	-1	-1	6
4	3	숲	480	생명수	-1	-1	-1	4
5	4	숲	72	마법 버섯	-1	-1	-1	7
6	5	기술	50000	기술의 역	-1	-1	-1	9
7	6	상인	4000	녹슨 대검	-1	-1	-1	6
8	7	상인	20	맹독 젤리	-1	-1	-1	9
9	8	테이블	500	용사 정식	용기	골렘의 땀	요정 가루	3
10	9	부재료	50	용기	-1	-1	-1	5
11	10	부재료	100	골렘의 땀	-1	-1	-1	6
12	11	부재료	30	요정 가루		-1	-1	8
13								

- 판매 품목 csv 예시

## [분 석]

### ■ 주문 관리 요구사항 분석

- 고객의 요구에 따라, 영수증의 출력형태가 달라지므로 영수증 출력기능을 이분화한다.  
또는 영수증 출력 시 메뉴 출력이 체크 되어있지 않다면, 메뉴의 출력은 생략되도록 한다.
- datetime 모듈을 이용하여, 해당 날짜 및 시간을 구해 영수증 내 출력될 수 있도록 한다.



- 영수증의 출력형태 (좌 : 메뉴명 포함 / 우 : 메뉴명 미포함)

### ■ 매출 관리 요구사항 분석

- 시간을 고려한 통계 정보의 데이터 출력을 위해, 각 날짜별 데이터를 정산 시 CSV형태로 출력하여 외부데이터로 저장되도록 한다

기준년월	카테고리명	결제년월	결제년월	결제시간대량	결제금액	고객수
202009	버거	2020	202009	16	17152992738	1698951
202009	버거	2020	202009	12	39824573100	3061393
202009	버거	2020	202009	8	63088329	10955
202009	버거	2020	202009	10	12091243560	1101437
202009	버거	2020	202009	14	21504628204	1982307
202009	버거	2020	202009	17	21740257374	1895619
202009	버거	2020	202009	3	13073394	1245
202009	버거	2020	202009	7	15790621	4038
202009	버거	2020	202009	11	25533035356	2064960
202009	버거	2020	202009	0	1576127417	149693
202009	버거	2020	202009	15	19457049560	1829767
202009	버거	2020	202009	22	4668103422	484290
202009	버거	2020	202009	13	26838932462	2356823
202009	버거	2020	202009	21	9133660774	856609
202009	버거	2020	202009	23	2770240775	293624
202009	버거	2020	202009	9	6937095243	729297
202009	버거	2020	202009	6	24685809	2835
202009	버거	2020	202009	1	280282078	36306
202009	버거	2020	202009	20	13988858603	1320367
202009	버거	2020	202009	2	59535920	5190
202009	버거	2020	202009	19	17300034225	1532993
202009	버거	2020	202009	18	23179589879	1984718
202009	버거	2020	202009	4	11124986	2781
202009	치킨	2020	202009	17	27515145888	1152003
202009	치킨	2020	202009	15	16565825436	310701
202009	치킨	2020	202009	20	39749028788	1504890
202009	치킨	2020	202009	23	12442825364	443476
202009	치킨	2020	202009	19	46871404600	1947991
202009	치킨	2020	202009	6	472829167	1576

- 패스트푸드점의 판매 목록 데이터 저장 형태

## ■ 재고 관리 요구사항 분석

- 메뉴 주문 이후 남은 재고량이 설정한 재고량 이하이거나, 메뉴 주문에 필요한 재료가 부족한 경우, 이를 사용자에게 고지하여 필요한 재료의 추가 주문이 가능하도록 한다.

## [구현/제작]

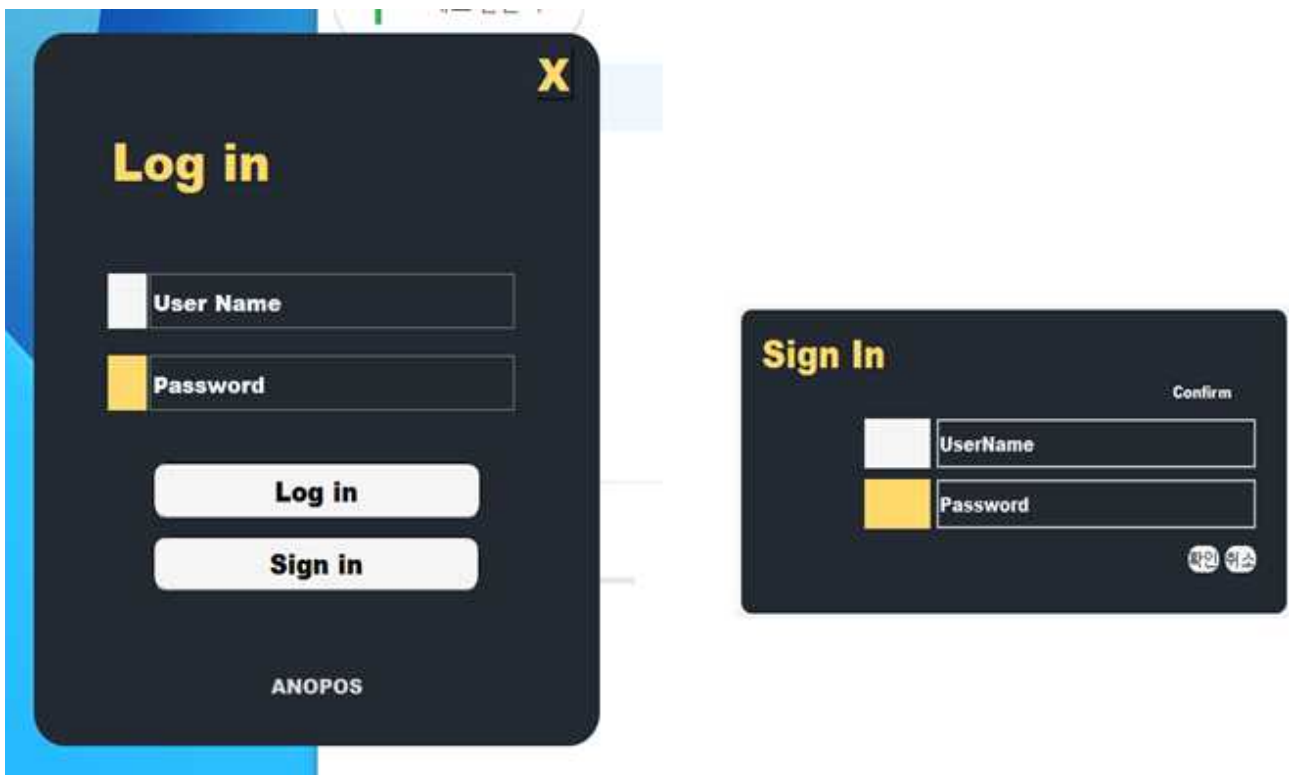
개발 도구 : PyQt 5.0.3, Spyder 4.2.5

## [시험/평가]

### ■ 평가항목

1. 각 재고가 부족 할 때 이를 사용자에게 정상적으로 고지하는가?
2. 사용자로 등록되지 않은 사용자가 프로그램 사용 시, 정상적으로 차단하는가?
3. 관리자가 아닌 사용자가 재고와 같은 기능을 사용 시, 정상적으로 차단되는가?
4. 데이터를 효과적으로 시각화 하였는가?
5. 총 판매내역이 오류 없이 잘 기록 되었는가?

## [결과 도출]



- 좌 : 첫 로그인 화면 / 우 : 회원가입 화면

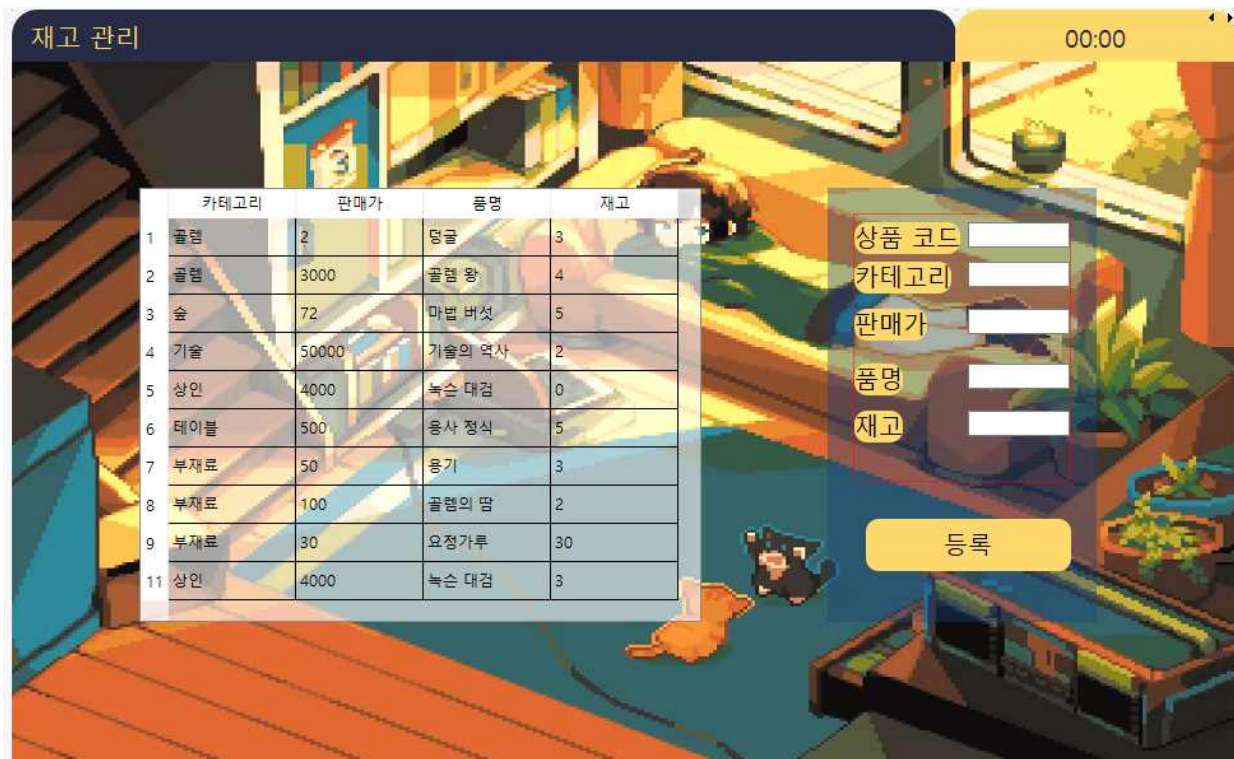


- 메인 화면



- 영업등록(테이블) 화면





- 재고 관리 화면

상품명	수량	금액
1 맹독 젤리	2	40
2 기술의 역사	1	50000

총결제금액		50040
받은 금액		60000
거스름돈		9960

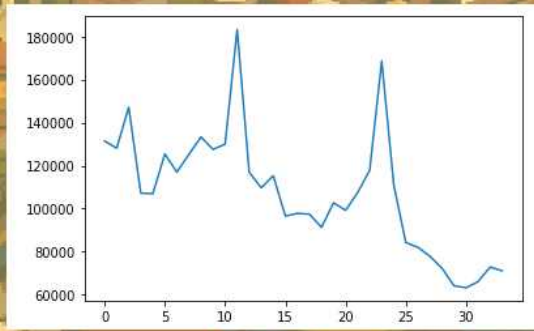
카드 결제	현금 결제	결제
B타입 영수증	A타입 영수증	

맹독 젤리	기술의 역사	

- 영업등록 화면의 테이블 클릭 시, 출력되는 주문 화면

08 월 07 일 기간별 매출액



- 영업등록(테이블) 화면



- 시간별 매출량, 시간별 방문객 수 데이터를 시각화

기간	카테고리명	일	매출액	방문객 수
202009	배거	202009	16	17152992738
202009	배거	202009	12	39824573100
202009	배거	202009	8	63088329
202009	배거	202009	10	12091243560
202009	배거	202009	14	21504626204
202009	배거	202009	17	21740257374
202009	배거	202009	3	13073394
202009	배거	202009	7	15790621
202009	배거	202009	11	25533035356
202009	배거	202009	0	1576127417
202009	배거	202009	15	19457049560
202009	배거	202009	22	4668103422
202009	배거	202009	13	26838932462
202009	배거	202009	21	9133660774
202009	배거	202009	23	2770240775
202009	배거	202009	9	6937095243
202009	배거	202009	6	24685809
202009	배거	202009	1	280282078
202009	배거	202009	20	13988858603
202009	배거	202009	2	59535920
202009	배거	202009	19	17300034225
202009	배거	202009	18	23179589879
202009	배거	202009	4	11124986
202009	치킨	202009	17	27515145888
202009	치킨	202009	15	16565825436
202009	치킨	202009	20	39749028788
202009	치킨	202009	23	12442825364
202009	치킨	202009	19	46871404600
202009	치킨	202009	6	472829167

- 년/월/주/일에 따른 메뉴별 매출액 정보가 저장된 CSV 파일

---

## 2.3 설계 제한 요소

### [성능]

- 메인 메뉴에서 각 기능선택에 의한 화면 전환까지 1초 이내로 전환 되도록 한다.
- 정산 버튼을 누르면 anopos.csv 에 판매 목록이 추가되고, 해당 날짜와 시간 담당자의 이름이 담긴 제목으로 반영된 정산 csv 파일이 생성된다. ex.) 20211110\_1043\_김다은.csv

### [규격/표준]

- POS기 문제에 정해진 규격이나 표준이 없다고 생각된다.

### [신뢰성]

- 기능 및 성능에서 오류 없이 동작하도록 신뢰도 100%를 목표로 설계한다.

### [안정성/내구성]

#### ■ 안전성

- 포스기 프로그램 내 데이터 파일의 입출력에 있어, PC 내 다른 파일 혹은 프로그램에 피해나 위협이 되지 않도록 한다.
- 포스기 프로그램 내 판매 내역과 같이 중요한 기능은 root 권한을 가진 사용자만이 접근할 수 있도록 한다.
- 관리자의 부주의 혹은 잘못된 조작으로 인하여, 재고 주문(음수 값 혹은 너무 큰 값)에 오류가 발생할 시, 이를 예외처리한다.

#### ■ 내구성

- 포스기의 장기적인 실행에도 타 응용프로그램과의 시스템 충돌이 발생하지 않도록 한다. 동적 메모리를 사용할 경우, 사용 후 즉시 해제를 통해 프로그램 내 메모리 누수가 발생하지 않게 설계한다.
-

---

## 2.4 세부 기능 명세

### ■ 주문관리 기능

- 영업등록 기능 : 사용자가 고객으로부터 주문을 받아, 해당 주문에 맞는 메뉴와 수량을 입력하고, 카드 또는 현금을 사용한 결제를 진행한다. 또한, 결제 이후 고객이 요청하는 형태의 영수증 출력을 제공한다.

### ■ 매출 관리 기능

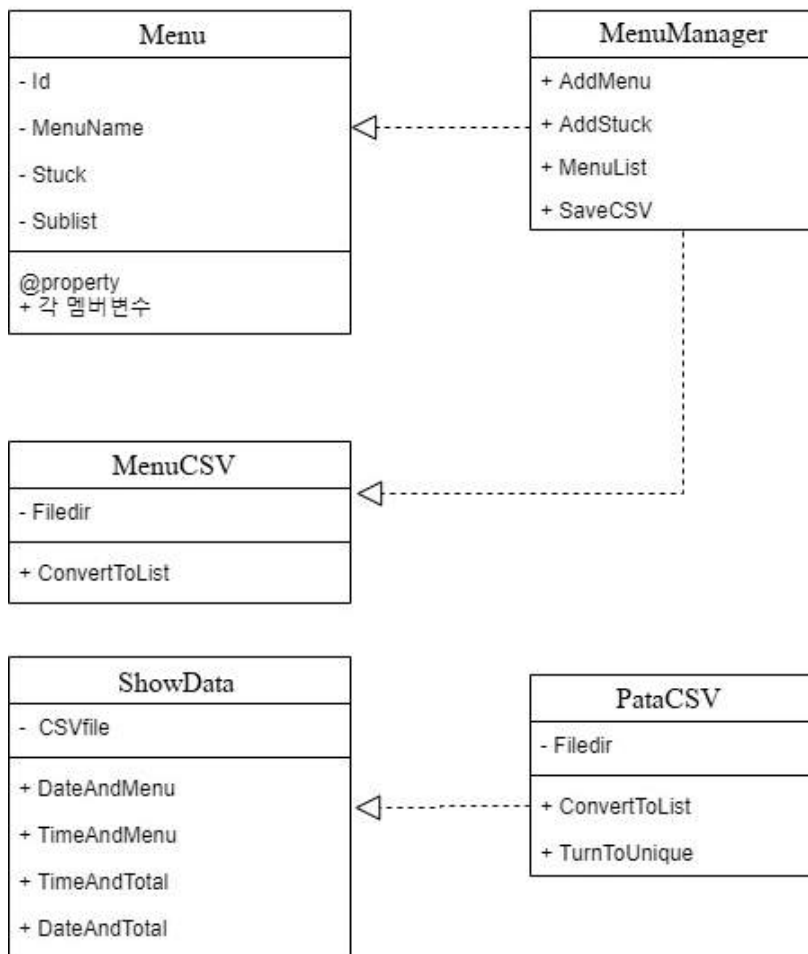
- 매출 요약 : 현재까지의 일별/주별/월별/연도별 총매출액, 기간별 메뉴별 매출액, 메뉴별 주문 수 등 시간을 고려한 통계 정보를 CSV 파일로부터 입력받아 이를 그래프 형태로 제공한다.
- 판매 내역 : 마감정산 이후 업데이트 되며, 당일 판매 내역을 시간/판매 메뉴 형태로 제공한다.
- 마감정산 : 최종 확인 후 매출 내역이 저장된 CSV 파일에 현재 데이터를 추가하여 갱신한다

### ■ 재고 관리 기능

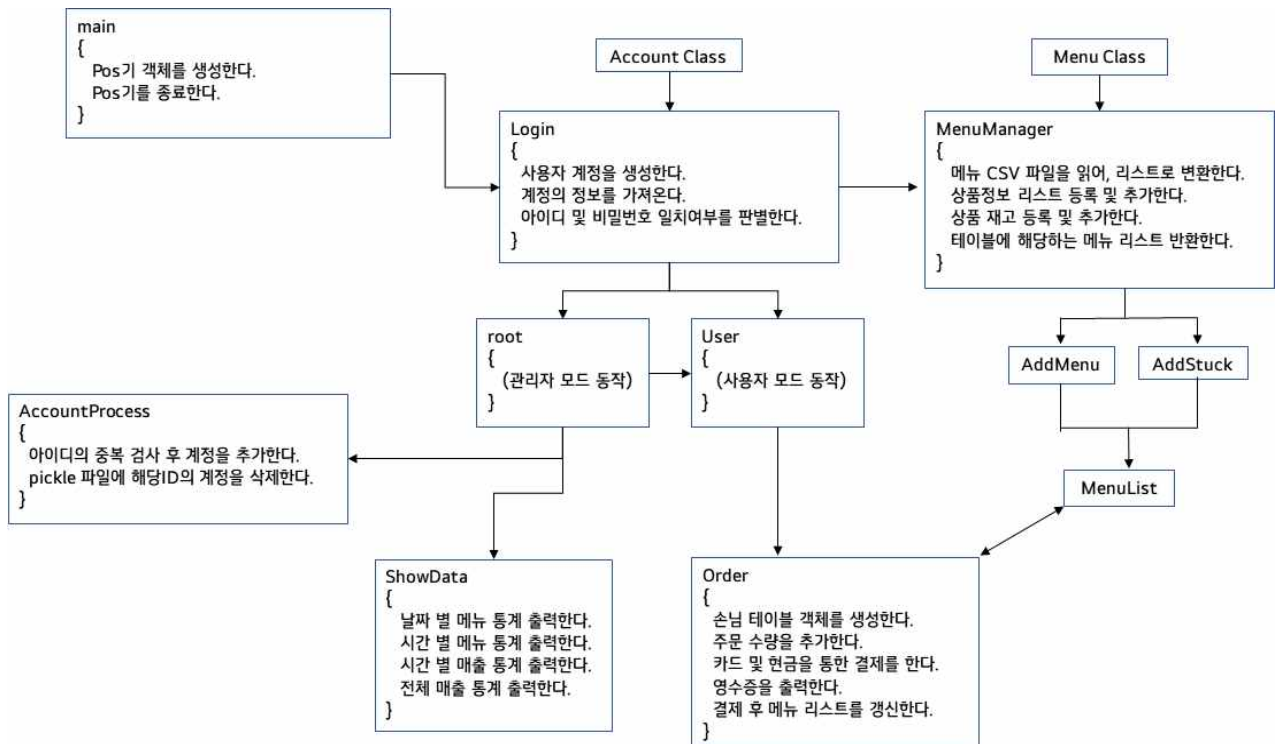
- 영업정보관리 기능 : 현재 남아있는 재료(아이템)의 수량 확인 및 재고 주문이 가능하다.

## 3. 설계

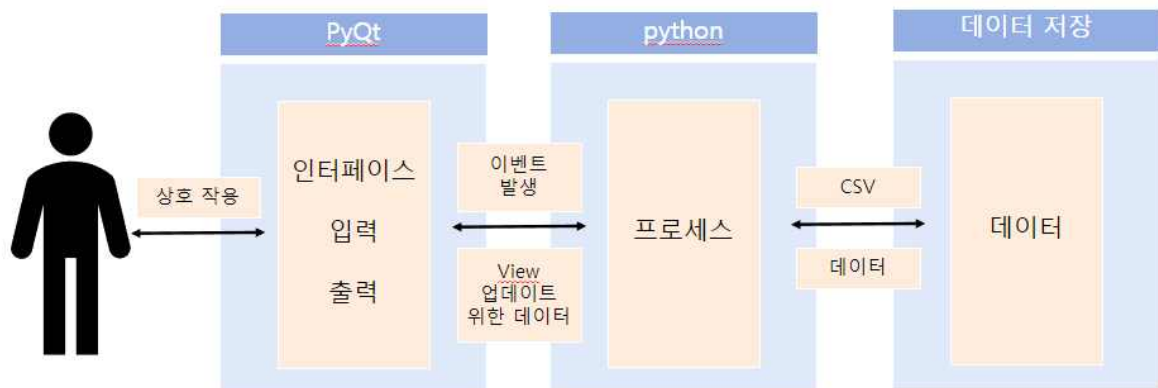
### 3.1 데이터 / 클래스 설계



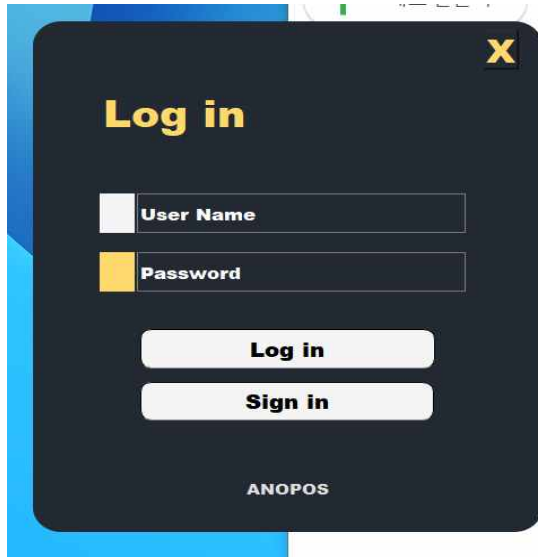
### 3.2 프로그램 구조 설계



### 3.3 인터페이스 설계







## 로그인

- 로그인 위젯과 메인 메뉴가 프로그램 실행시 한번에 띄워진다.
- 로그인 위젯에서 login success 하기 전까지 메인 메뉴가 보여지지 않은 setEnabled(false) 혹은 화면 출력을 막는다.
- 로그인을 클릭하여 success 한다면 로그인창은 사라지고 메인 메뉴 visible 및 enable
- User Name과 PassWord가 저장되어있는 pickle 파일과 일치해야 success 된다.
- id 와 비번이 없을 경우 sign in을 통해 회원 가입을 진행한다.
- 초기설정 (User Name : root, Password : root) 루트 권한으로 접근할시에는 메인 메뉴에 모든 버튼이 enable 된다.



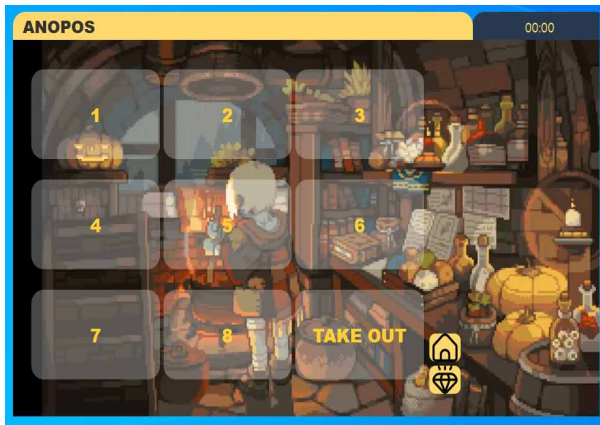
## 회원가입

- 탐색을 통해 userName 이 중복되는지 검색
- 모든 입력이 완료될 시에 확인을 누르면 UserInfo.pkl 에 새로운 회원 정보가 입력된다.



## 메인 화면

- 로그인 성공 시 뜨게 되는 화면
- 로그인한 유저의 권한에 따라 enable 되는 버튼이 달라진다.
- 영업 등록 : 각 테이블 및 TAKE OUT 주문을 처리할 수 있는 화면으로 이동한다.
- 영업 정보 관리 : 전체 상품 목록에 대한 재고 현황을 살피고 재고 추가 및 새 상품 등록이 가능하다.
- 매출 요약 : 시간이 지남에 따른 매출의 현황을 시각화 하여 확인할 수 있다.
- 총 매출금액에 대한 정보는 오른쪽에 항상 뜨게 된다. 영업 등록에서 저장을 눌렀을 때를 기준으로 업데이트 된다.



## 영업 등록

- 메인 화면에서 영업 등록을 누르면 전환되는 화면
- 정산 버튼을 누르면 csv 파일에 현재까지의 판매정보가 기록된다.
- 홈 버튼을 누르면 메인 메뉴 화면으로 전환되게 된다.
- 숫자가 입력된 테이블에서는 조리된 음식만을 판매할 수 있도록 하고, TAKE OUT이 표기된 버튼에서는 소모품 등을 판매할 수 있도록 한다.
- 화면의 상단에는 현재 시간이 출력되게 된다.
- 테이블에 주문이 접수되면 테이블의 숫자와 현재 주문 품목의 목록이 버튼에 표시된다.

상품명	수량	금액
1 명품 쥬리	2	40
2 기술의 역사	1	50000

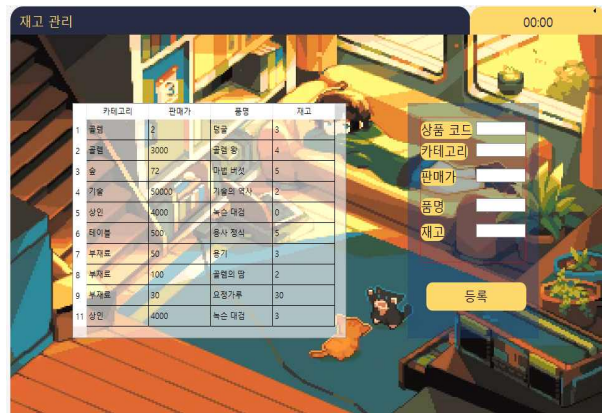
총결제금액	50040
받은 금액	60000
거스름돈	9960

카드 결제	현금 결제	결제
BE입 영수증	AE입 영수증	

## 판매 등록

- 영업 등록 화면에 있는 버튼을 클릭할 시 출력되는 다이얼로그.
- 총 결제 금액과 받은 금액 거스름돈이 항상 업데이트 되게 된다.
- 영수증은 A 타입과 B 타입으로 나누어져 출력이 가능하다.
- 상품 등록 시 현재 존재하는 재고보다 등록하려는 재고가 더 많을 시 재고가 부족하다는 메시지 출력 및 해당 품목의 재고를 추가할 수 있는 다이얼로그를 생성한다.
- 결제를 누르기 전까지 각 버튼마다 다이얼로그는 현재 정보를 계속 유지하고 있게 된다.
- 상품 품목에 등록된 상품만큼 버튼에 상품이 표시되게 된다.
- 버튼을 클릭하면 수량이 하나 증가하게 된다.
- 카드 결제와 현금 결제를 선택할 수 있다.



## 영업 정보 관리

- 메인에서 영업 정보 관리 버튼을 누르면 전환되는 화면이다.
- 제품 등록 csv 파일을 읽어 테이블 위젯에 업데이트 해준다.
- 테이블 리스트에 품목을 클릭하면 상품 코드, 카테고리, 판매가, 품명이 입력되고 재고에 값을 바꾸면 상품 목록의 재고가 업데이트 된다.
- 새로운 상품 코드가 입력되게 되면 새로운 상품이라고 판단하고 csv 파일 하단에 추가하여 준다.
- 홈 버튼을 누르면 메인으로 돌아갈 수 있다.



## 매출 요약

- 메인에서 매출 요약 버튼을 클릭하면 전환되는 화면이다.
- 달력에 기간을 클릭하면 해당 기간의 매출액을 시각화하여 확인할 수 있다.
- 판매 품목당 매출량 확인을 위해 콤보 박스에 품목을 선택하면 해당 일의 시간별 선택 품목당 매출량을 확인할 수 있다.
- 홈 버튼을 누르면 메인으로 돌아갈 수 있다.



---

### 3.4 알고리즘 설계

#### [로그인 - view]

```
class Login(QWidget, form_class):
    def __init__(self):
        super().__init__()
        // UI 셋업
        // 계정 관리 클래스 객체 생성
        // 계정의 정보를 가져옴
        // 로그인 버튼과 연결
        // 종료 이벤트 cacle 버튼과 연결

    def 로그인 슬롯
        // 텍스트 라인에 입력된 텍스트 정보 받기
        // account 에 존재하는 아이디인지 true false 반환하는 함수 실행
        // true 라면
            // 비밀번호 텍스트 정보를 a 와 비교하여 만약 일치한다면 suceess
            // 실패한다면 fail messagebox 출력

    def 종료 슬롯
        // 다이얼로그 닫음
```

#### [Account class]

```
class Account:
    def __init__(self, _id="", _password="", _sudo=False):
        // 디폴트 _sudo false
        // 초기화

    # Property Setter 생성
```

---

### [AccountProcess class]

```
# 사용자 정보를 담고있는 pickle 파일 처리 클래스 импорт
class 계정 관리자:
    계정을 딕셔너리로 관리
    pickle 파일을 딕셔너리 형태로 변환

    def __del__(self):
        file_manager = AccountFileManager('./account.pkl')
        file_manager.save(self.__account_list)

    @property
    def account_list(self):
        return self.__account_list

    def 로그인(아이디, 비밀번호):
        if( 아이디를 키값으로 조회 했을 때 아이디가 존재하지 않음 ):
            false를 리턴
        else:
            if 비밀번호가 일치함 :
                true를 리턴함
            else:
                false를 리턴함

    # 계정 추가
    아이디가 중복되는지 검사 후
    중복되지 않는다면 파일에 사용자 추가
    # 계정 삭제
    pickle 파일에 해당 아이디에 해당하는 사용자를 삭제
```

---

## [상품 품목 등록]

### ■ 알고리즘 분석

사용자는 프로그램 실행 전 상품 품목이 위의 형태와 같이 입력된 csv 파일을 지정된 경로에 생성해 두어야 한다. 상품 코드는 A 번 카테고리는 B 번 판매가는 C 번 품명은 D 번에 위치하도록 하며 재고는 항상 맨 마지막 열에 입력되어야 한다. 하위 품목은 E 번에서부터 시작하여 입력된다. 하위 품목이 존재하지 않을 경우 -1을 채워준다.

	A	B	C	D	E	F	G	H
1	상품 코드	카테고리	판매가	품명	하위품목1	하위 품목2	하위 품목3	재고
2	1	골렘	2	덩굴	-1	-1	-1	5
3	2	골렘	3000	골렘 왕의	-1	-1	-1	6
4	3	숲	480	생명수	-1	-1	-1	4
5	4	숲	72	마법 버섯	-1	-1	-1	7
6	5	기술	50000	기술의 역사	-1	-1	-1	9
7	6	상인	4000	녹슨 대검	-1	-1	-1	6
8	7	상인	20	맹독 젤리	-1	-1	-1	9
9	8	테이블	500	용사 정식	용기	골렘의 땀	요정 가루	3
10	9	부재료	50	용기	-1	-1	-1	5
11	10	부재료	100	골렘의 땀	-1	-1	-1	6
12	11	부재료	30	요정 가루		-1	-1	8
13								

- 상품등록 CSV 파일

### ■ sudo\_code

```
# 메뉴 클래스
class Menu:
    def __init__(self, id , category, menuName, stuck, sublist(배열로 주어짐)):
        #생성
        # 각 멤버변수의 property 필요
```

```
csv 파일을 읽고 리스트 형태로 바꾸어 주는 MenuCSV 클래스 import
class MenuManager:
    def __init__(self):
        #메뉴를 읽어 리스트로 변환해주는 과정
        # 메뉴 리스트에 대한 property 필요
    def addmenu(상품 정보 리스트):
        코드가 존재하지 않고 항목이 존재하지 않는다면 실행됨
    def addstuck(상품 코드, 재고 수):
        코드에 해당하는 재고의 수를 재고 수 만큼 증가시킨 후 변경
    def MenuList(isTable):
        if 테이블이 맞다면:
            테이블에 해당하는 메뉴들의 리스트 return
        else:
            잡화 및 소모품에 해당하는 메뉴들의 리스트 return
        # view에서 dialog 메뉴에 띄워줄 항목들을 달리 하기 위해
    def SaveCSV():
        csv파일에 저장
```

## [상품 구매]

```
Menu Class, table Class import
class OrderDialog(QWidget, form_class):
    def __init__(self, table):
        super().__init__()
        // UI 셋업
        // 손님 테이블 객체 생성
        // Menu에 등록된 상품 품목만큼 주문 버튼에 표시
        // 각 상품 별 추가 이벤트 연결
        // 카드 및 현금 결제 이벤트 연결
        // 타입 별 영수증 출력 이벤트 연결
        // 결제 이벤트 연결

    def 카드 결제 슬롯 :
        // 총 카드 매출에 결제 금액을 더함
        // 해당 품목(재고)의 수 감소

    def 현금 결제 슬롯 :
        // 총 현금 매출에 결제금액을 더함
        // 해당 품목(재고)의 수 감소

    def A 타입 별 영수증 출력 :
        //총결제 금액의 라인 출력
    def A 타입 별 영수증 출력 :
        //총결제 금액의 라인 출력

    def 메뉴 별 버튼 클릭(수량 추가) :
        if 등록하고자 하는 수량 > 잔여 재고량 :
            //재고가 부족하다는 메시지 출력
            if 현재 사용자가 관리자인 경우 :
                //재고를 추가할 수 있는 다이얼로그 출력
            else :
                //해당 상품의 수량을 하나 증가
    def 상품 품목 changed 슬롯 :
        if 해당 상품의 수량이 0 이 아닌 경우
            //품목의 수량 * 가격을 총 결제 금액에 더함
            //총 결제금액 text line을 업데이트
    def 받은 금액 changed 슬롯:
        if 총 결제금액 < 받은 금액 :
            //결제금액에서 받은 금액을 뺀 값을 거스름돈에 출력
        else :
            //금액 부족을 거스름돈에 출력
            현금 결제 버튼 disable
```

## [통계 기능]

판매 정산을 통해 쌓인 csv 파일을 이용해 matplotlib pandas 의 모듈을 사용하여 데이터를 시각화 의미있는 정보들을 제공한다. 별도의 클래스를 만들어 기간별 시간별 품목별 등의 함수를 제공한다. 파라미터로 csv 파일을 받는다.

## 4. 추진 전략 및 일정

### [정보 수집 방법]

- 공공 데이터 및 제공되는 판매 데이터 참고
- 파이썬 쿡북 및 책
- PyQt 공식 문서
- Matplotlib 공식 문서
- python 공식 문서
- 디자인 패턴 책 (mvc 패턴)

### [개발 절차]

개발 검토	설계 검증 1차	설계 검증 2차
<ul style="list-style-type: none"><li>- 클래스 구조 및 설계 확인</li><li>- 오버 엔지니어링, 알고리즘 미흡 등 문제 검토</li><li>- 사용자 유즈 케이스 확인</li><li>- csv 파일 검증</li></ul>	<ul style="list-style-type: none"><li>- 판매 기능 설계 검증</li><li>- 판매 기능 내부 알고리즘 검증</li><li>- 메인 기능 설계 검증</li></ul>	<ul style="list-style-type: none"><li>- csv 파일 시각화 검증</li><li>- 매출 통계 설계 검증</li><li>- 판매 기능 설계 검증</li></ul>

개발 검증 및 인증	배포
<ul style="list-style-type: none"><li>- mvc 패턴 검증</li><li>- 데이터 검증</li><li>- 판매 목록 검증</li></ul>	<ul style="list-style-type: none"><li>- 코드 리팩토링</li><li>- 사용 방법 및 공식 문서 작성</li><li>- 테스트 및 검수</li><li>- 깃헙 소스코드 및 개발 후기 작성</li><li>- 릴리즈 모드로 빌드하여 배포</li></ul>

---

## [개발 방법론]

### ■ 객체 지향 개발 방법론

- 객체 지향 개발 방법론이란 프로그램을 객체와 객체 간의 인터페이스 형태로 구성하기 위하여 객체, 클래스 및 이들 간의 관계를 설계하여 모델로 변환하는 개발 방법론이다.

### ■ 개념화 단계

- 시스템에 대한 비즈니스 사례를 만들고 프로젝트 문제 해결 범위 정의
- 모든 유스케이스를 명시하고 설명 포함
- 단계별 계획 포함
- 마지막으로 프로젝트의 목적 검사 및 개발 진행 여부 결정

### ■ 상세화 단계

- 문제 영역을 분석 프로젝트 계획 개발을 통해 위험 요소 제거
- 유스케이스 기술 및 제약사항 고려
- 선정된 아키텍처 실현 및 시스템 구현
- 상세한 시스템의 목적과 범위 및 아키텍처 선정과 위험 검사.

### ■ 구축 단계

- 사용자들에게 제공할 수 있도록 반복 및 점증적으로 개발
- 유스케이스 보충 기술, 설계 보충 구현 마무리 및 테스트
- 마지막으로 소프트웨어 환경과 사용자들이 운영될 준비가 되었는지 결정

### ■ 전이 단계

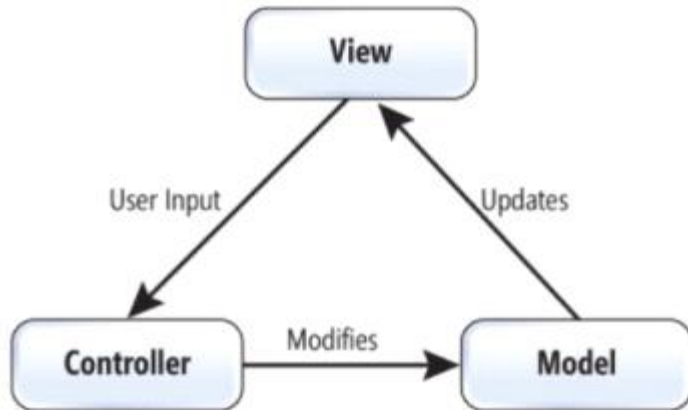
- 소프트웨어를 사용자들에게 전달
  - 사용자의 손에서 시스템에 적합하도록 추가적인 개발 및 발견되지 않은 문제점 수정 및 미루어 놓은 사항들 마무리
-

---

## [아키텍처]

### ■ MVC 패턴

model view controller 의 약자로 하나의 애플리케이션 프로젝트를 구성할 때 그 구성요소를 세가지의 역할로 구분한 패턴이다.



- MVC 패턴

## [모델]

애플리케이션의 정보 데이터를 나타낸다. 데이터 베이스, 처음 정의 하는 상수, 초기화값, 변수등을 의미. 또한 이러한 데이터 정보들의 가공을 책임지는 컴포넌트

규칙

1. 모델은 사용자가 편집하길 원하는 모든 데이터를 가지고 있어야한다.
2. 모델은 뷰나 컨트롤러에 대해서 어떠한 정보도 알아선 안된다.
3. 변경이 일어나면 변경 통지에 대한 처리 방법을 구현해야 한다.

---

## [뷰]

input 텍스트, 체크 박스 항목과 같은 사용자 인터페이스 요소를 나타냄. 데이터 및 객체의 입력 그리고 출력을 담당. 데이터를 기반으로 사용자들이 볼 수 있는 화면

규칙

1. 모델이 가지고 있는 정보를 따로 저장해서는 안된다.
2. 뷰는 모델이나 컨트롤러와 같은 다른 구성요소들의 존재를 알아선 안된다.
3. 변경이 일어나면 변경 통지에 대한 처리 방법을 구현해야 한다.

---

## [컨트롤러]

데이터와 UI의 요소들을 잇는 역할이다. 즉 사용자가 데이터를 클릭하고 수정하는 것에 대한 이벤트를 처리하는 부분이다.

규칙

1. 모델이나 뷰에 대해서 알고 있어야한다.
  2. 모델이나 뷰의 변경을 모니터링 해야 한다.
-

### [역할 분담]

김준용	김다은
로그인 및 회원가입 구축	판매 구현
메인 화면 구축	GUI 구성 및 디자인
예외 처리 및 사용자 유즈 케이스 고려 설계	CSV 파일 생성 및 판매 통계 구현
영업 정보 구현	전체 코드 병합

### [일정 제시]

(수행 기간 : 2021년 11월 13일 ~ 2021년 12월 10일)						
수행 내용 \ 구분	1 주	2 주	3 주	4 주	주	
POS 회원 관리 구현						
POS 테이블 버튼 다이얼로그 로직 구현 및 테이블 뷰 변경과 정산 기능						
판매량 통계 구현						
재고 관리 및 상품 추가 구현						
영수증 출력과 판매 구현						