

# Software Design Document

소프트웨어 공학 프로젝트: **POS**기 시스템

**2018060646** 이한승

**2019031858** 김재명

**2021030873** 이동오

**2019008813** 문원찬

**2021051321** 가제고 달린

# **Table Of Content**

## **1. Overview**

- 1.1 purpose
- 1.2 scope
- 1.3 features summary

## **2. Data Design**

- 2.1 object diagram
- 2.2 database schema design and table

## **3. Architectural Design**

- 3.1 data flow diagram
- 3.3 mini specification
- 3.4 architectural context diagram

## **4. Interface Design**

- 4.1 Use case
- 4.2 User function categories

## **5. Procedural Design**

- 5.1 Activity diagram

## 1. Overview

POS (Point of sale) 시스템은 매장 운영에 매우 중요한 역할을 합니다. 현금, 카드, 모바일 결제 등 다양한 결제 방식을 편리하고 안전하게 처리하여 고객이 쉽게 구매할 수 있도록 도와주는 시스템입니다. 결제 외에도 POS 시스템은 거래를 효율적으로 관리하고 제품 명세와 결제 방법을 기록합니다. 이 거래 데이터는 실시간 재고 관리에 도움이 되어 과잉 재고를 방지합니다. 따라서 POS 는 사업을 더 효과적으로 운영할 수 있도록 도와주는 시스템이라고 할 수 있습니다.

### 1.1 purpose

이 Software Design Document(SDD) 문서의 주요 목표는 Point of sale(POS) 시스템의 설계 및 아키텍처를 명확하게 다루는 것입니다. 해당 문서는 개발 팀, 이해관계자 및 프로젝트에 참여하는 기타 당사자들에게 POS 시스템의 디자인을 알리는 중요한 역할을 합니다. 이는 시스템 개발, 테스트 및 배포를 위한 기반이 됩니다.

이 문서는 다음을 대상으로 합니다.

개발팀: POS 시스템의 구조와 동작을 쉽게 이해함으로써 효율적인 구현을 할 수 있습니다.

이해관계자: 시스템의 구체적인 설계를 파악해 효과적인 의사 소통을 진행할 수 있습니다.

프로젝트 매니저: 시스템의 변경 사항을 관리하고 추적하는데 도움을 얻습니다. 따라서 시스템 업그레이드, 버그 수정 및 새로운 요구사항을 효과적으로 다룰 수 있게 됩니다.

### 1.2 scope

이 문서의 범위는 다음과 같습니다.

- POS 시스템의 전반적인 아키텍처와 구성 요소를 다룹니다.

- POS 시스템의 각 모듈 및 구성요소의 상세 설계, 동작 및 상호 작용 방식을 다룹니다.
- 사용자 인터페이스, API, 데이터 흐름 및 외부 시스템과의 상호 작용에 관한 설계를 다룹니다.
- 안전한 설계 및 보안 조치에 대한 내용을 포함합니다.
- 성능과 확장성에 관한 설계 결정과 관련된 정보를 제공합니다.
- POS 시스템을 웹 애플리케이션으로써 구현하는 것으로 규모를 제한합니다.

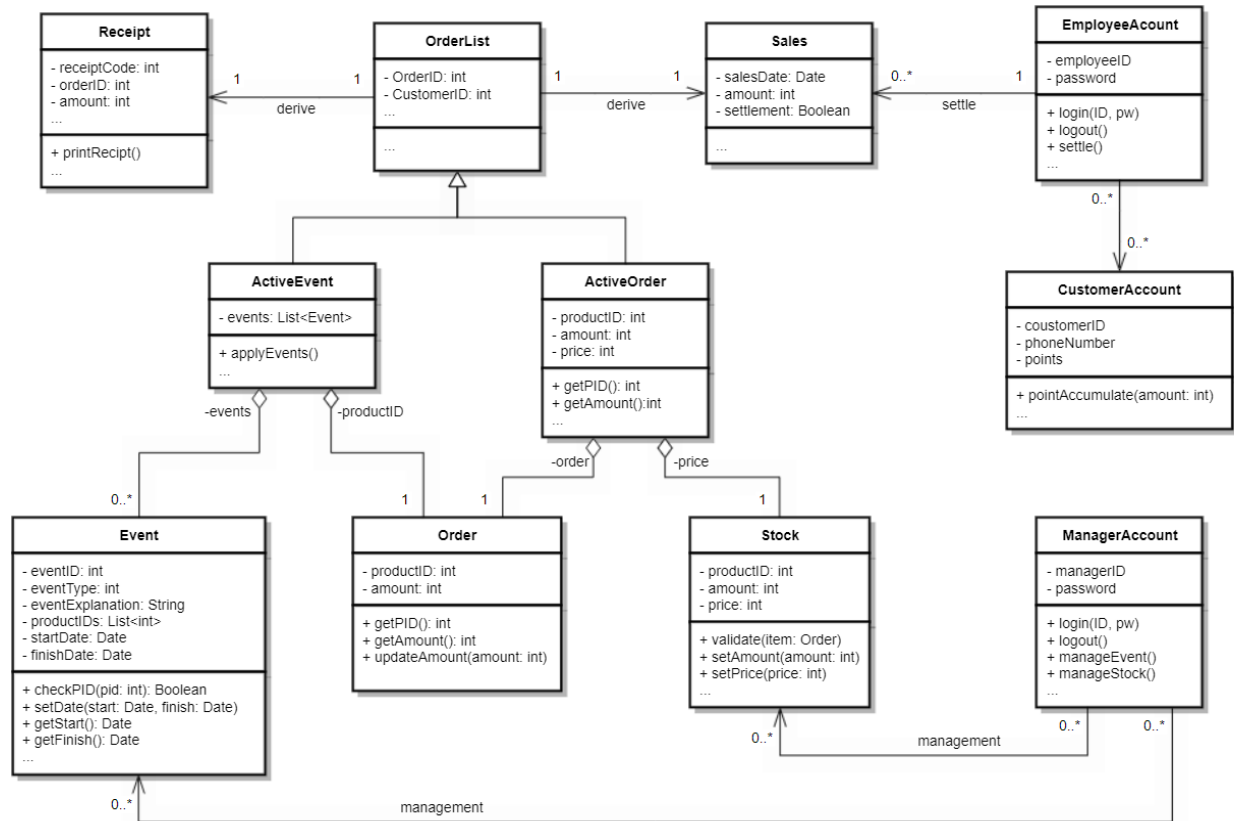
### **1.3 features summary**

POS 시스템은 고객의 상품 구매 및 환불 등의 요청을 하나의 시스템으로 관리하고 처리할 수 있도록 돕습니다. 그리고 관리자가 상품의 재고나 이벤트를 설정할 수 있게하여 상품 처리 과정의 이상 여부나 혜택을 자동적으로 확인할 수 있도록 합니다.

이 프로젝트에서는 **Java**와 **JavaScript**가 애플리케이션 개발에서, **MongoDB**가 데이터베이스 관리에서 사용될 예정입니다. 그리고 **Amazon Web Services**를 통해 프로젝트가 배포 및 운영되며 협업을 위해 **git**을 통해 프로젝트를 관리할 것입니다.

## 2. Data Design

### 2.1 object diagram



## 2.2 database schema design and collection

위 object diagram에서 Event, Stock, CustomerAccount, Employee와 Manager의 Account, 그리고 Sales object가 database형태로 저장됩니다.

### 2.2.1 Event Collection

Event를 저장하는 collection은 event의 공통적인 부분을 저장하는 main collection 하나와, event type별로 각각의 내용을 저장하는 detail collection 여러개로 구분됩니다.

#### A. Main collection

Field	Type	Description
event_ID	Number (Int32)	event들을 서로 구분하고 다른 collection과 join을 하기 위한 primary key로 사용됩니다.
event_type	Number (Int32)	event의 종류를 구분하는 용도로 사용됩니다. (type 0: 할인 event, type 1: n+m 형태의 행사 event) 추후에 event 종류가 추가될 경우, 신규 collection을 생성하여 대응할 수 있습니다.
start_date	Date	event가 시작되는 날짜를 저장합니다.
finish_date	Date	event가 종료되는 날짜를 저장합니다.
explanation	String	event의 설명을 저장합니다. POS기 monitor에 해당 설명을 출력하여 보여줄 수 있고, manager가 event의 내역을 바로 파악하는 용도로도 사용할 수 있습니다.

## B. Detail collection (type = 0)

Field	Type	Description
event_ID	Number (Int32)	event들을 서로 구분하고 다른 collection과 join을 하기 위한 primary key로 사용됩니다.
count	Number (Int32)	할인 event를 적용받기 위해 구매해야하는 물품의 최소 개수를 저장합니다. 조건이 없다면 default로 10이 저장됩니다.
discount_rate	Number (Double)	할인 비율을 저장합니다. 0 초과 1이하의 값을 가집니다.
product_IDs	Array	event를 적용받는 물품들의 ID를 저장합니다.

## C. Detail collection (type = 1)

Field	Type	Description
event_ID	Number (Int32)	event들을 서로 구분하고 다른 collection과 join을 하기 위한 primary key로 사용됩니다.
base	Number (Int32)	행사 event를 적용받기 위해 구매해야하는 물품의 개수를 저장합니다.
bonus	Number (Int32)	행사 event를 적용받아서 추가로 지급되는 물품의 개수를 저장합니다.
product_IDs	Array	event를 적용받는 물품들의 ID를 저장합니다.

### 2.2.2 Stock Collection

Field	Type	Description
product_ID	Number (Int32)	각 물품을 구분하기 위한 <b>primary key</b> 로 사용됩니다.
amount	Number (Int32)	현재 남아있는 물품의 재고 수를 저장합니다.
price	Number (Int32)	<b>event</b> 적용 이전에 물품의 본래 판매 가격을 저장합니다.

### 2.2.3 CustomerAccount Collection

Field	Type	Description
customer_ID	Number (Int32)	고객의 ID를 저장합니다. 각 고객들의 계정을 구분하는 <b>primary key</b> 로도 사용됩니다.
phone	String	<b>point</b> 적립을 위한 휴대폰 번호를 저장합니다.
points	Number (Int32)	고객이 사용할 수 있는 <b>point</b> 의 양을 저장합니다.

### 2.2.4 Employee/ManagerAccount Collection

Field	Type	Description
ID	String	login을 위한 ID를 저장합니다. 직원들의 계정을 구분하는 <b>primary key</b> 로도 사용됩니다.



password	String	계정의 비밀번호를 저장합니다.
role	Number (Int32)	직원의 역할을 저장합니다. (role 0: Employee, role 1: Manager)

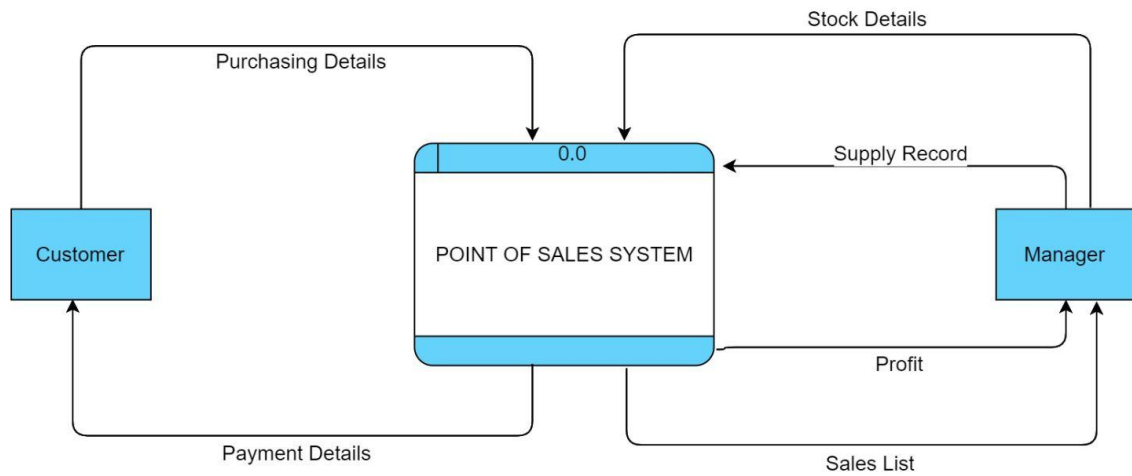
### 2.2.5 Sales Collection

Field	Type	Description
Sales_Date	Date	매출이 발생한 날짜를 저장합니다. 각 매출을 구분하는 <b>primary key</b> 로도 사용됩니다.
amount	Number (Int32)	매출의 총액을 저장합니다.
settlement	Boolean	employee의 정산 여부를 저장합니다.

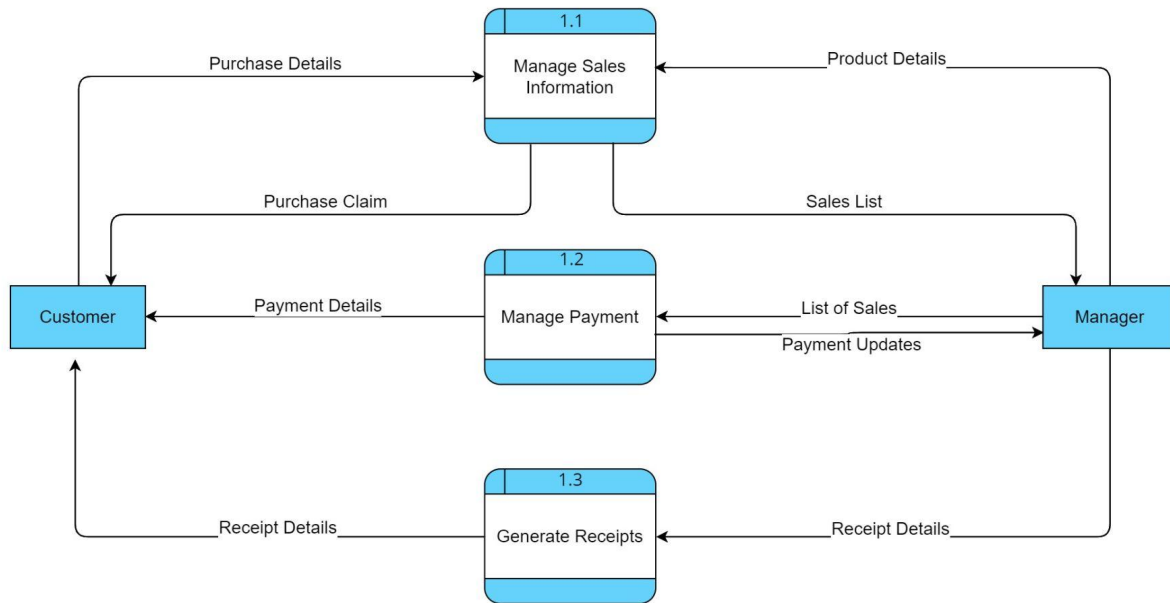
### 3. Architectural Design

#### 3.1 Data flow diagram

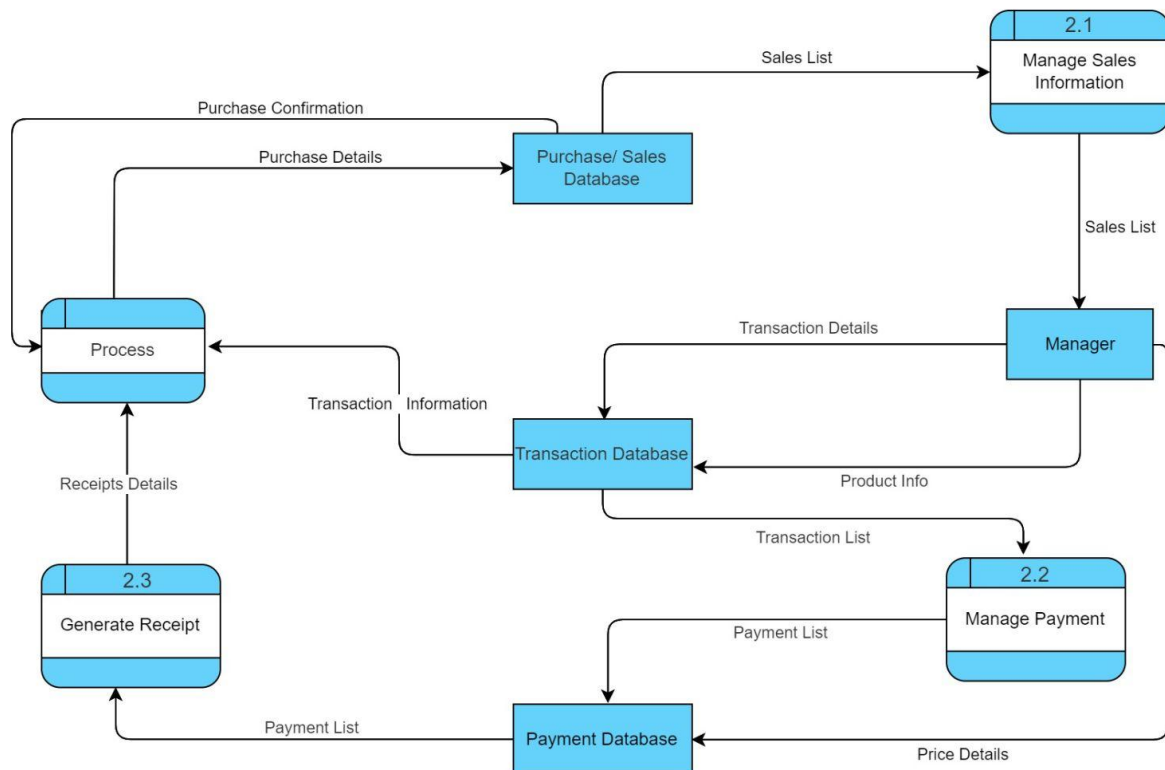
##### Level 0



## Level 1



## Level 2



## 3.2 Mini Specification

### 3.2.1 현금 결제 및 바코드 이미지 처리 시스템

- 기능적 요구 사항

바코드 이미지 입력 및 처리:

사용자는 바코드 이미지를 시스템에 직접 업로드합니다.

시스템은 업로드된 바코드 이미지를 인식하고 해당 제품 정보를 데이터베이스에서 검색합니다.

인식된 제품의 가격 정보는 자동으로 총액에 추가됩니다.

현금 거래 처리:

사용자는 고객으로부터 받은 현금을 시스템에 입력합니다.

시스템은 입력된 현금과 총액을 비교하여 거스름돈을 계산합니다.

거스름돈 계산 결과는 사용자 인터페이스에 표시되어 사용자가 확인할 수 있습니다.

거래 완료 및 기록:

거래가 완료되면, 시스템은 거래 내역을 기록하고 사용자에게 거래 완료를 알립니다.

시스템은 거래 내역에 대한 간단한 요약을 화면에 표시합니다.

- 비기능적 요구 사항

성능:

바코드 이미지 처리와 거래 처리는 2초 이내에 완료되어야 합니다.

시스템은 고객 대기 시간을 최소화하기 위해 최적화되어야 합니다.

확장성:

시스템은 사용자 수와 거래량의 증가에 따라 쉽게 확장될 수 있어야 합니다.

보안:

모든 거래 기록은 안전하게 저장되어야 합니다.

거래 데이터는 정기적인 백업과 함께 암호화되어야 합니다.

사용성:

사용자 인터페이스는 직관적이고 사용하기 쉬워야 합니다.

사용자는 최소한의 훈련으로 시스템을 사용할 수 있어야 합니다.

### 3.2.2 매출 분석 및 보고 시스템

- 기능적 요구 사항

매출 데이터 수집:

시스템은 모든 거래에서 발생하는 매출 데이터를 수집하고 기록합니다.

매출 보고서 생성:

사용자는 시스템을 통해 매출 보고서를 생성할 수 있습니다.

보고서에는 일별 매출 통계, 인기 제품 등의 정보가 포함됩니다.

- 비기능적 요구 사항

사용성:

매출 보고서는 사용자가 쉽게 이해하고 해석할 수 있도록 직관적이어야 합니다.

성능:

매출 데이터의 처리와 보고서 생성은 신속하게 이루어져야 합니다.

### 3.2.3 재고 관리 시스템

- 기능적 요구 사항

재고 상태 모니터링:

시스템은 모든 제품에 대한 재고 수준을 실시간으로 모니터링합니다.

사용자는 시스템을 통해 재고 수준을 조회하고, 재고가 부족한 제품을 식별할 수 있습니다.

재고 업데이트:

거래가 완료될 때마다 시스템은 자동으로 관련 제품의 재고 수준을 감소시킵니다.

사용자는 수동으로 재고를 추가하거나 조정할 수 있습니다.

재고 경고 및 알림:

특정 제품의 재고 수준이 임계값 이하로 떨어지면, 시스템은 사용자에게 경고를 발송할 수 있습니다.

- 비기능적 요구 사항

확장성:

시스템은 다양한 제품과 증가하는 재고 수준을 관리할 수 있도록 설계되어야 합니다.

정확성:

재고 데이터는 정확해야 하며, 모든 거래 후 실시간으로 업데이트되어야 합니다.

### 3.2.4 (선택적) 영수증 생성 기능

- 기능적 요구 사항

영수증 생성:

거래 완료 후, 시스템은 선택적으로 영수증을 생성할 수 있는 기능을 제공합니다.

영수증에는 거래 날짜, 시간, 제품 목록, 총액, 받은 금액, 거스름돈 등의 정보가 포함됩니다.

영수증은 간단한 텍스트 형태로 화면에 표시되며, 필요한 경우 인쇄할 수 있도록 설계됩니다.

- 비기능적 요구 사항

유연성:

영수증 생성 기능은 시스템 설정을 통해 활성화 또는 비활성화할 수 있어야 합니다.

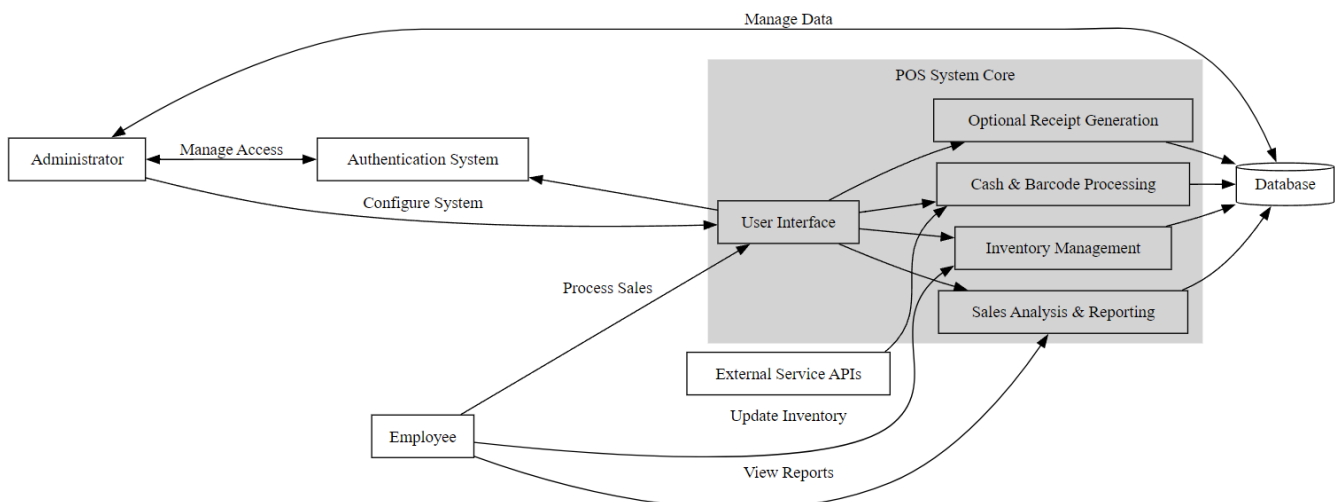
영수증은 사용자가 요청할 경우에만 생성되어야 합니다.

간소화된 디자인:

영수증은 필수적인 정보만을 포함하여 간소화되어야 합니다.

디자인은 사용자가 쉽게 읽고 이해할 수 있도록 명확해야 합니다.

## 3.3 Architectural Context Diagram



### **POS 시스템 코어:**

시스템의 중심이 되는 핵심 모듈로, 모든 주요 기능을 처리합니다. 이는 현금 결제 처리, 바코드 이미지 처리, 매출 분석 및 보고, 재고 관리 기능을 포함합니다.

### **데이터베이스:**

모든 거래 데이터, 제품 정보, 재고 수준, 사용자 정보 등을 저장합니다. **POS** 시스템 코어와 직접 통신하여 데이터를 저장하고 검색합니다.

### **사용자 인터페이스:**

직원이 고객의 거래를 처리하고, 매출 보고서를 조회하며, 재고를 업데이트할 수 있는 인터페이스입니다. 관리자는 이 인터페이스를 통해 시스템 설정을 구성하고 데이터베이스를 관리할 수 있습니다.

### **인증 시스템:**

사용자가 시스템에 접근하기 전에 인증을 수행합니다. 관리자와 직원의 역할에 따라 다른 수준의 접근 권한을 부여합니다.

### **외부 서비스 API:**

현금 결제 처리기와 바코드 스캐너와 같은 외부 장치와 통신하기 위한 인터페이스입니다. 이 **API**를 통해 시스템은 외부 장치로부터 데이터를 받고, 필요한 경우 외부 서비스와 통신할 수 있습니다.

### **관리자:**

시스템의 설정을 구성하고, 데이터베이스를 관리하는 역할을 합니다. 제품의 신규 등록, 삭제, 프로모션 등록 등의 관리 작업을 수행할 수 있습니다.

### **직원:**

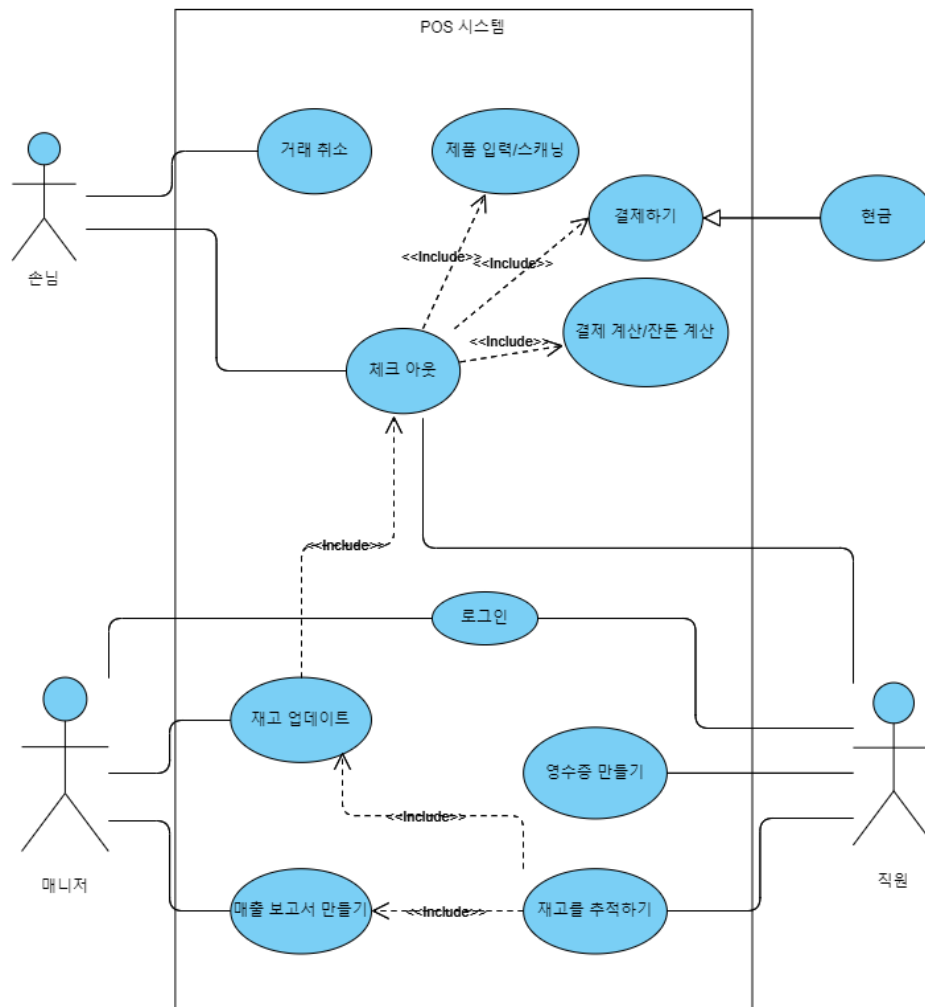
일상적인 판매 거래를 처리하고, 매출 보고서를 조회하며, 재고를 업데이트하는 역할을 합니다.

## 4. Interface Design

### 4.1 Use case

#### 4.1.1 제품 구매

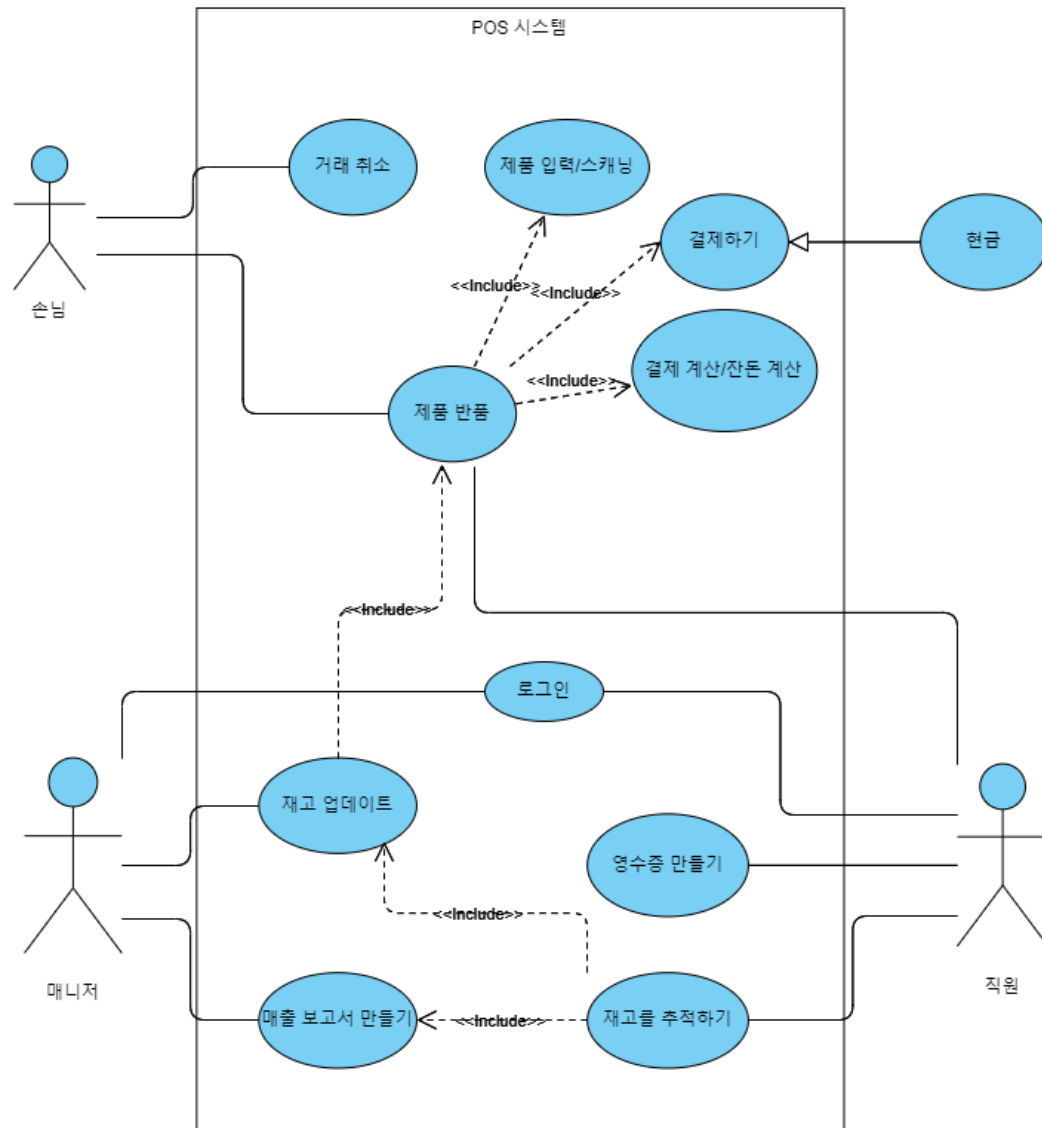
이 사용 사례는 고객이 POS 시스템을 사용하여 소매점에서 구매하는 과정을 나타냅니다. 직원은 POS 시스템을 이용해 고객이 구매하려는 상품을 입력/스캐닝 하고 기록하고 총액을 계산한 뒤 결제를 처리하고 영수증을 발급합니다.



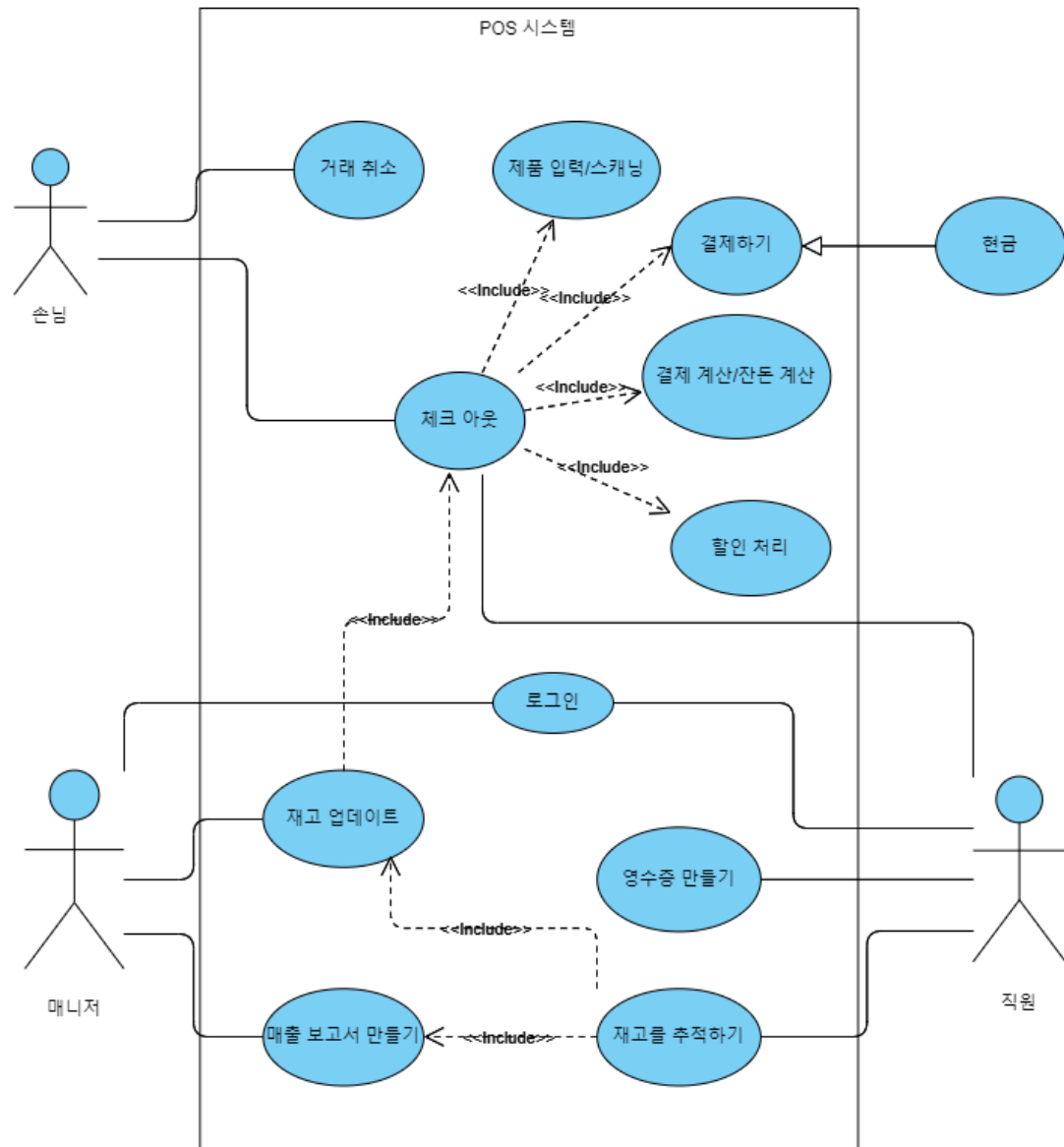


#### 4.1.2 제품 반품

이 사용 사례는 고객이 소매점에 제품을 반품하고 해당 제품에 대해 환불 또는 교환을 받는 과정을 나타냅니다.



이 사용 사례는 POS 시스템에서 고객 구매에 할인을 적용하는 과정을 나타냅니다.  
할인은 1+1 행사, 우대 프로그램 또는 기타 기준에 따라 달라질 수 있습니다.



## 4.2 User Function Categories

### 제품 추가

사용자가 시스템에 새 제품을 추가할 수 있습니다.

**Input:** 제품 세부정보(이름, 가격, 카테고리 등)

**Process:** 입력을 확인하고 제품 데이터베이스를 업데이트합니다.

**Output:** 확인 메시지.

### 판매 거래 처리

사용자가 판매 거래를 시작할 수 있습니다.

**Input:** 스캔하거나 수동으로 입력한 제품 코드, 수량.

**Process:** 총 가격을 계산하고 재고를 업데이트합니다.

**Output:** 판매 영수증.

### 판매 보고서 생성

사용자에게 판매 보고서를 생성하는 기능을 제공합니다.

**Input:** 날짜 범위, 필터.

**Process:** 판매 데이터를 쿼리하고, 필터를 적용하고, 보고서를 생성합니다.

**Output:** 판매 보고서를 표시하거나 내보냅니다.

### 반품/환불 처리

사용자가 반품 또는 환불을 처리할 수 있습니다.

**Input:** 원래 거래 내역, 반품 사유.

**Process:** 반품 조건을 확인하고, 재고를 업데이트하고, 환불을 처리합니다.

**Output:** 반품/환불 확인.

제품 세부정보 보기

사용자가 특정 제품에 대한 자세한 정보를 볼 수 있도록 허용합니다.

**Input:** 제품 식별자.

**Process:** 제품 세부정보를 검색하고 표시합니다.

**Output:** 제품 정보.

사용자 역할 관리

관리자가 사용자 역할 및 권한을 정의할 수 있습니다.

**Input:** 사용자 역할, 권한.

**Process:** 사용자 역할을 검증하고 업데이트합니다.

**Output:** 역할 업데이트 확인.

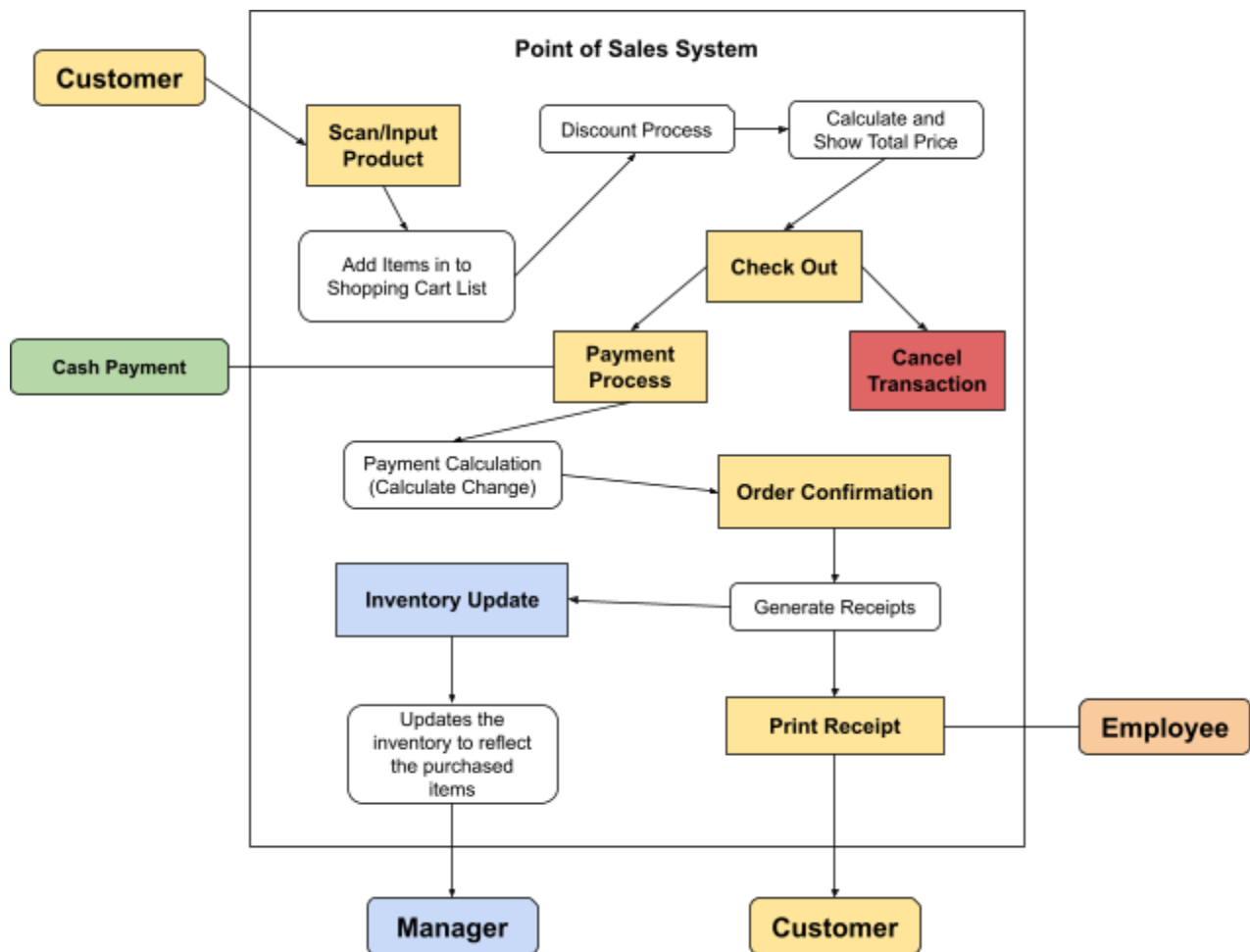
## 5. Procedural Design

### 5.1 Activity Diagram

이 활동 다이어그램의 절차적 설계는 특정 프로세스 또는 기능과 관련된 단계 및 작업의 순서를 설명합니다. POS 시스템에서는 3가지의 활동 다이어그램이 존재합니다. 고객, 직원, 그리고 매니저의 절차적 설계가 있습니다.

#### 5.1.1 Activity: Process Sale Transaction (Customer)

이 활동 다이어그램은 고객이 POS 시스템의 프로세스 판매 거래 활동에 대한 절차적 설계를 설명합니다.



### 5.1.2 Activity: Generate Sales Report (Manager) & Process Customer Purchase (Employee)

이 활동 다이어그램은 관리자가 POS 시스템에서 판매 보고서를 생성하기 위해 수행하는 단계와 직원이 POS 시스템에서 고객의 구매를 처리하기 위해 수행하는 단계를 간략하게 설명합니다.

