

Software Requirements Specification

소프트웨어 공학 프로젝트: **POS**기 시스템

2018060646 이한승

2019031858 김재명

2021030873 이동오

2019008813 문원찬

2021051321 가제고 달린

Table of Contents

1. Introduction

- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Project Scope
- 1.4 References

2. System Description

3. Functional Requirements

- 3.1 System Features
 - 3.1.1 System Feature 1
 - 3.1.2 System Feature 2
- 3.2 Use Cases
 - 3.2.1 Use Case Diagrams
 - 3.2.2 Use Case 1
 - 3.2.3 Use Case 2
- 3.3 Entity Relationship Diagram
- 3.4 Sequence Diagrams
 - 3.4.1 Case 1
 - 3.4.2 Case 2
 - 3.4.3 Case 3
- 3.5 Data Dictionary

4. External Interface Requirements

5. Technical Requirements (Non functional)

- 5.1 Performance
- 5.2 Scalability
- 5.3 Security
- 5.4 Maintainability
- 5.5 Usability
- 5.6 Multi lingual Support
- 5.7 Auditing and Logging
- 5.8 Availability

6. Open Issues

1 Introduction

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 소프트웨어 요구사항 명세서(SRS) 문서의 주요 목표는 Point of Sale (POS) 시스템 개발을 위한 요구사항을 정의하는 것입니다. 이 문서는 개발 팀, 이해관계자 및 프로젝트에 참여하는 기타 당사자들을 위한 포괄적인 가이드 역할을 합니다. 이 문서의 범위는 POS 시스템의 주요 기능과 특징을 명확하게 기술하는 것입니다. 이는 시스템 설계, 개발, 테스트 및 배포를 위한 기초 역할을 합니다.

이 문서는 다음을 대상으로 합니다.

개발 팀: 명시된 요구사항을 이해하고 구현하는 데 사용됩니다.

이해관계자: POS 시스템의 기능 및 능력에 대한 통찰력을 얻습니다.

프로젝트 매니저: 개발 프로세스를 감독하고 관리합니다.

1. 직원과 고객 모두에게 효율성 향상

POS 시스템은 직원과 고객 모두에게 효율성을 향상시키는데 중요한 역할을 합니다. 직원은 거래를 빠르게 처리할 수 있도록 간편한 사용자 인터페이스와 효율적인 기능을 활용할 수 있습니다. 이로써 결제 및 거래 처리 시간이 단축되어 고객들은 금방 결제를 완료할 수 있습니다.

2. 재고 관리 향상

POS 시스템은 실시간으로 재고를 관리합니다. 제품이 판매될 때마다 시스템은 재고에서 해당 제품을 자동으로 차감합니다. 따라서 제품의 과잉 재고나 부족을 방지할 수 있습니다. 결과적으로, 재고 관리 향상은 관리 비용 절감과 제품 유통의 효율성을 증가시킬 수 있습니다.

3. 거래 처리 단순화

POS 시스템은 거래 처리를 단순화하여 직원과 고객 모두에게 편의를 제공합니다. 예를 들어, 고객은 다양한 결제 수단을 편리하게 사용할 수 있으며, 직원은 시스템이 자동으로 총액과 잔돈을 계산하여 계산 오류를 줄여줍니다.

1.2 Document Conventions

2.1 Terminology (용어)

AWS	Amazon Web Services
DBMS	Database Management System
ERM	Entity Relationship Diagram
IDE	Integrated Development Environment
POS	Point of Sale
SRS	Software Requirements Specification
VCS	Version Control System
VSC	Visual Studio Code

2.2 Format (형식)

- **Bold Text**(굵기): 중요 용어 및 강조에 사용됩니다.
- **Text Size (Point)**: 글씨 크기로 제목을 구분할 때 사용됩니다.
- **Indent** (들여쓰기): 주변의 문장과 구분할 때 사용됩니다.

1.3 Project Scope

POS 시스템은 다음과 같은 주요 기능을 포함할 것 입니다.

제품 데이터베이스 및 재고관리:

- 제품의 중앙 데이터베이스 유지.
- 실시간 재고 추적 및 관리 기능.

현금 처리 및 잔돈 계산:

- 현금 거래 안전하게 처리.
- 정확한 잔돈 계산 및 제공.

매출 보고:

- 매출 거래에 대한 종합 보고서 생성.
- 제품 성과 및 수익에 대한 통찰력 제공.

영수증 만들기:

- 각 거래에 대한 디지털 또는 인쇄 영수증 생성.

1.4 References

개발 팀은 프로젝트 중 다음 문서 및 자료를 참고할 것 입니다.

POS 시스템 설계 지침서

POS 시스템에 대한 권장 설계 원칙에 관한 문서

<https://ecommerce-platforms.com/ko/glossary/point-sale>

프로그래밍 언어 문서

선택한 프로그래밍 언어에 대한 참고 자료.

DBMS 문서

선택한 데이터베이스 관리 시스템에 대한 문서.

클라우드 서비스 제공자 문서

클라우드 서비스를 효율적으로 사용하기 위한 자료.

Apple UI 디자인 기본 원칙

기본 디자인 개념을 고려하여 다양한 사용자를 위한 깔끔하고 효율적인 UI를 위한 자료.

<https://developer.apple.com/kr/design/tips/>

2 System Description

POS 시스템은 다음 도구 및 자원을 사용하여 개발될 것 입니다.

1. Development Environment

Visual Studio Code

소스 코드 편집 및 개발 환경으로 VSC를 사용할 예정입니다. 이 편집기는 다양한 언어와 환경을 지원하는 확장 기능을 제공하여 본 프로젝트의 다양한 요구사항을 만족시키기에 적합합니다.

2. Version Control

Git

소스 코드의 버전 관리를 위해 Git 을 사용합니다. 이를 통해 코드 변경 사항의 추적 및 여러 팀원 간의 협업이 용이합니다.

GitHub

Git 리포지토리 호스팅 및 코드 리뷰, 이슈 트래킹 등의 기능을 제공하는 온라인 플랫폼으로, 프로젝트의 중앙 리포지토리로 사용될 예정입니다.

3. Programming Language

JavaScript

프로젝트의 주요 프로그래밍 언어로서, 웹 기반 애플리케이션 개발에 사용됩니다.

4. Database

MongoDB

NoSQL 데이터베이스로서, 유연한 스키마와 확장성을 제공합니다. 프로젝트 데이터 관리 및 저장에 사용될 예정입니다.

5. Communication Tools

Zoom

원격 회의 및 팀원 간의 커뮤니케이션 도구로 사용됩니다.

KakaoTalk

실시간 메시지 전송 및 간편한 알림을 위한 메신저로 사용됩니다.

6. Cloud Services

Amazon Web Services

클라우드 호스팅 및 기타 서비스 제공을 위한 플랫폼으로, 프로젝트의 배포 및 운영에 사용될 예정입니다.

3 Functional Requirements

3.1 System Features

3.1.1 Product Database and Inventory Management (제품 데이터베이스 및 재고 관리)

이 시스템에는 제품 데이터베이스와 실시간 재고 관리 기능이 포함되어 있습니다. 제품에 대한 종합적인 정보를 저장하고 실시간으로 정확한 재고 추적을 보장합니다.

기능

- 제품을 추가, 업데이트 및 제거합니다.
- 재고 수준을 실시간으로 모니터링하고 조정합니다.
- 상품의 다양성과 가용성을 관리하여 고객에게 실시간 정보를 제공합니다.

3.1.2 Cash Handling and Change Calculation (현금 처리 및 거스름돈 계산)

현금 거래 시스템은 정확한 금액을 계산해 잔돈을 반환할 수 있도록 도와줍니다. 이로써 빠르고 정확한 현금 거래가 가능하며 오류를 최소화 합니다.

기능

- 현금결제시 정확한 금액을 계산합니다.
- 자동으로 잔돈을 확인하고 반환합니다.
- 오류를 최소화하여 빠르고 정확한 현금 거래를 보장합니다.

3.1.3 Sales Reporting (매출 보고)

POS 시스템은 매출과 관련된 정보를 자동으로 기록할 수 있습니다. 이를 통해 사업 주인은 판매 통계를 쉽게 확인하고 사업 운영을 최적화할 수 있습니다.

기능

- 매출 정보를 기록하고 저장합니다.
- 사업 분석을 위한 매출 보고서를 생성할 수 있습니다.
- 사업 운영을 최적화하기 위해 데이터 기반 의사 결정을 지원합니다.

3.1.4 Create a Receipt (영수증 만들기)

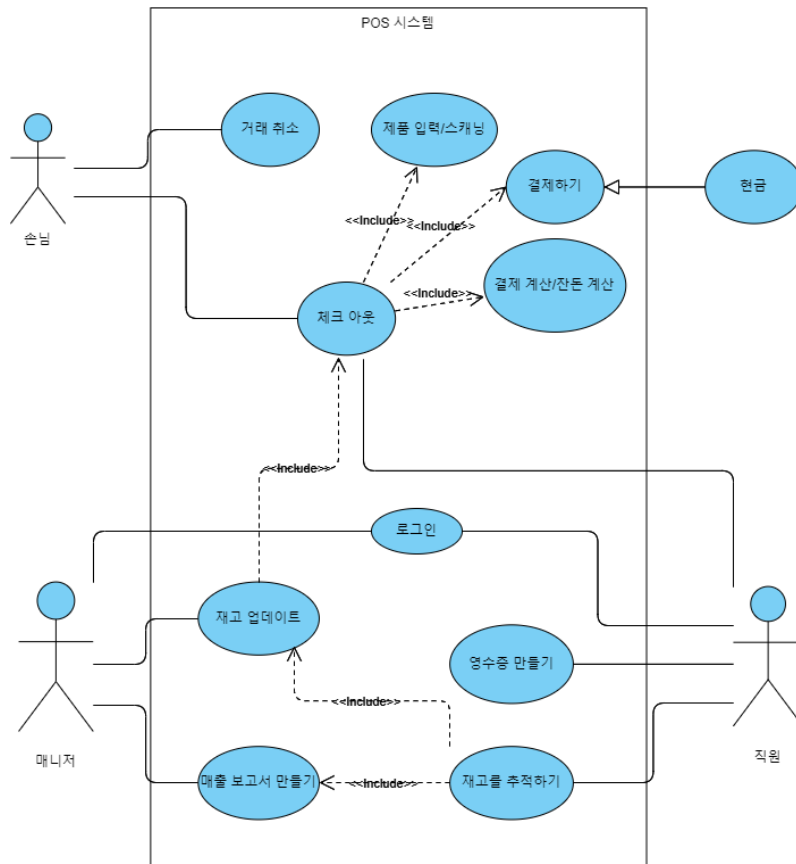
시스템은 각 거래를 기록하고 고객에게 명확하고 유용한 영수증을 생성합니다.

기능

- 거래 세부정보를 기록합니다.
- 영수증은 거래 내역을 명확하게 보여주어 고객에게 투명성을 제공합니다.
- 고객에게 구매에 대한 자세한 증거를 제공합니다.

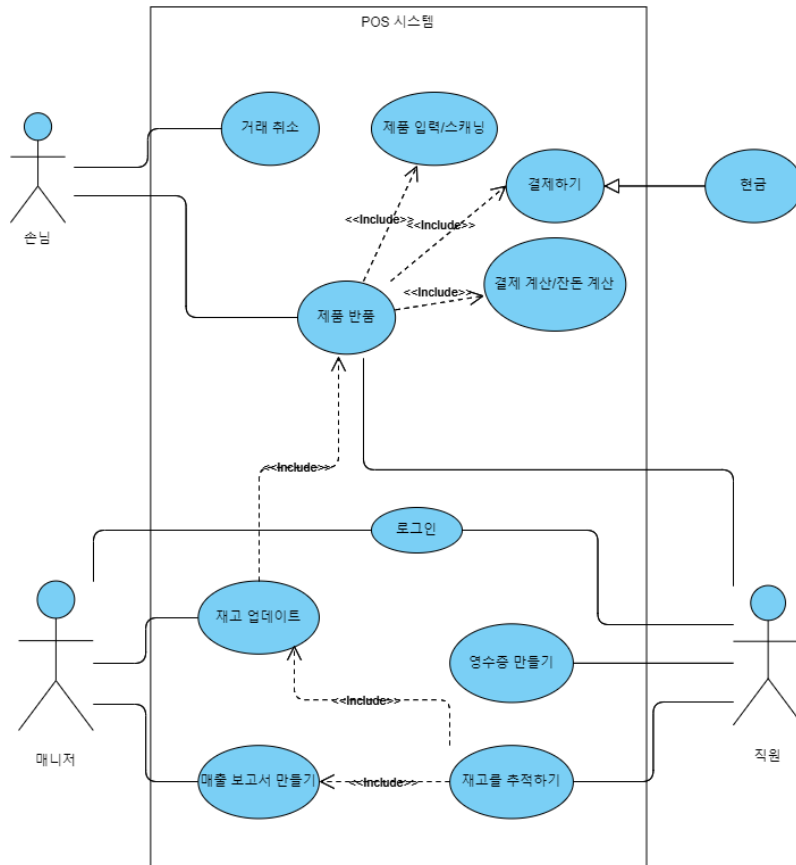
3.2.1 제품 구매

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.



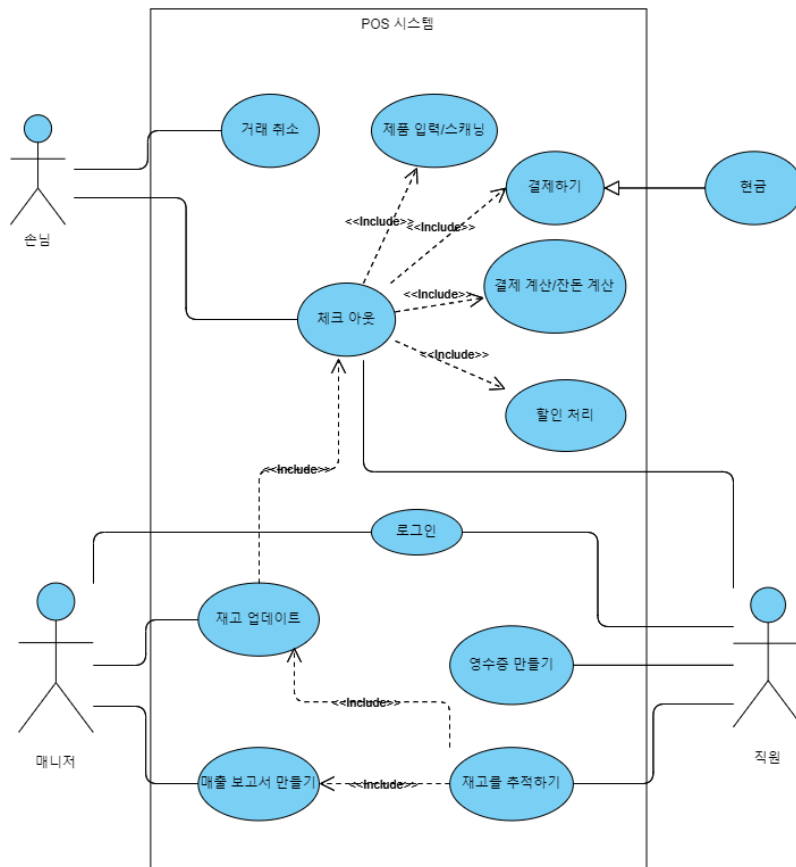
3.2.2 제품 반품

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

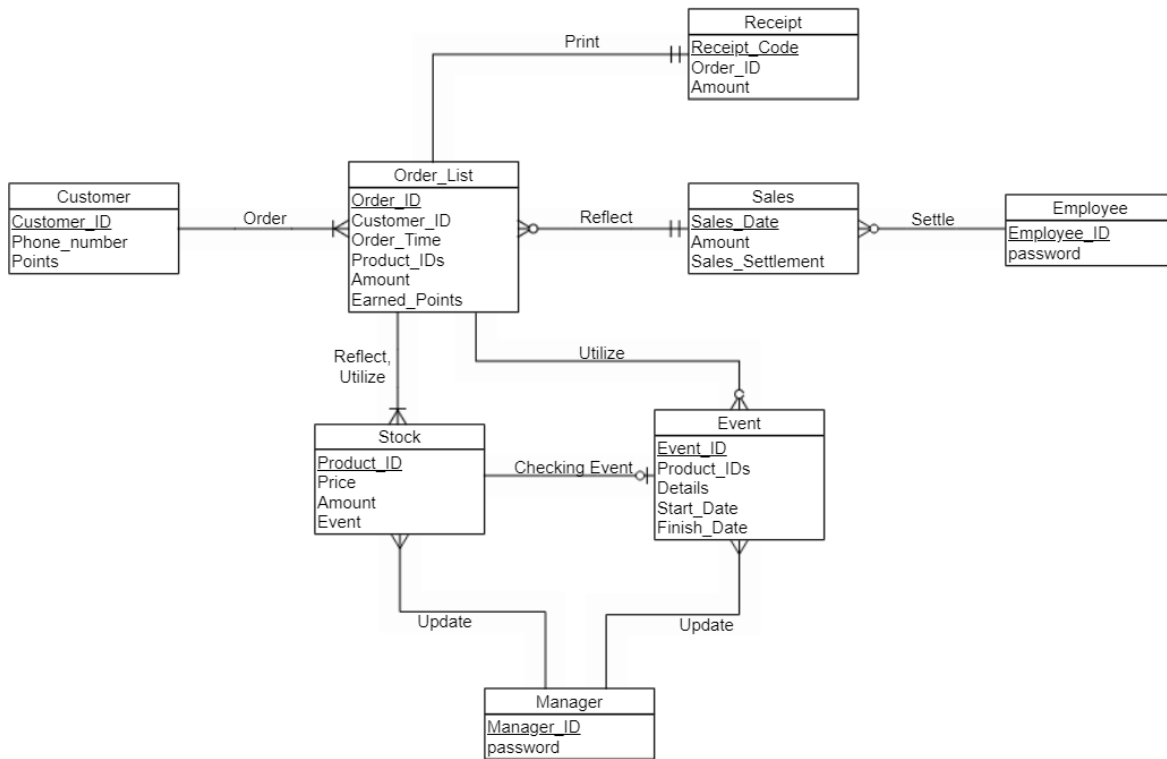


3.2.3 할인 적용

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



3.3 Entity Relationship Diagram (ERM)

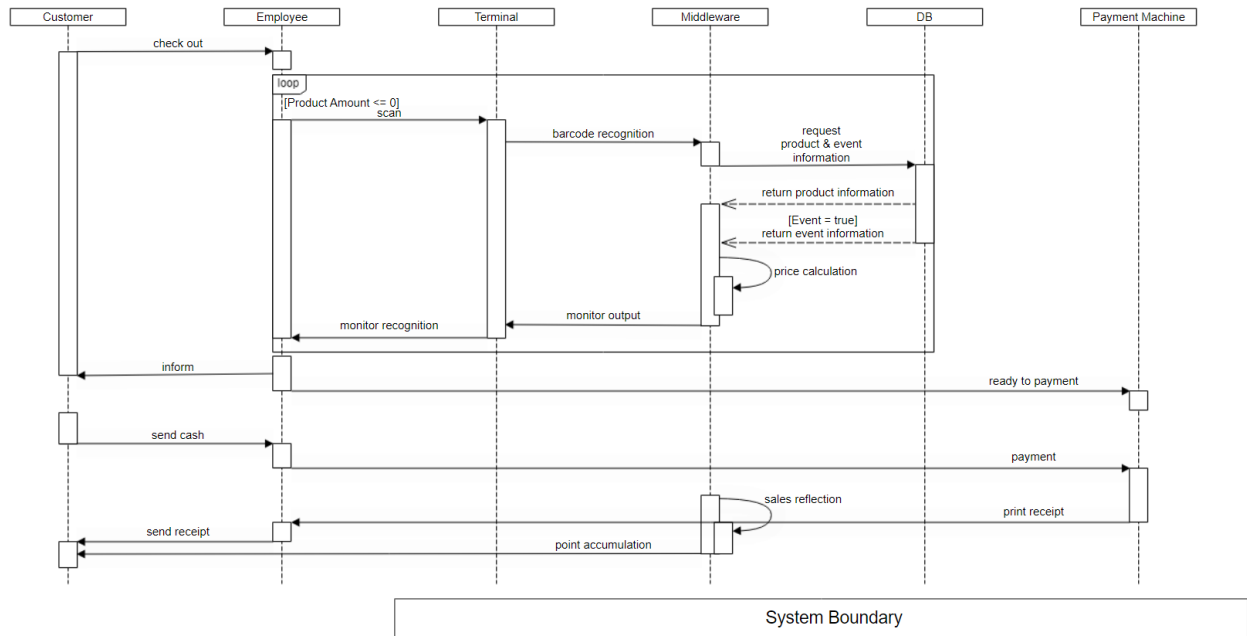


이 ERM에서는 POS 시스템의 데이터 처리 과정을 중심으로 각 객체와 객체들 간의 관계를 까마귀 발 표기법으로 나타냈습니다.

고객이 구매한 상품에 대한 정보들은 **Order_List** 객체에 모아서 관리하고 이 데이터들은 **Stock** 객체와 **Event** 객체에서 가져오는 데이터를 활용합니다. 연산이 완료되고 구매 처리 과정으로 넘어가면 **Sales** 객체와 **Stock** 객체에 이를 반영하고 **Receipt** 객체에 데이터를 넘겨 영수증을 출력할 수 있게 합니다. 그리고 이 전체 구조에 고객과 직원, 매니저가 관여하는 모습입니다.

3.4 Sequence Diagram

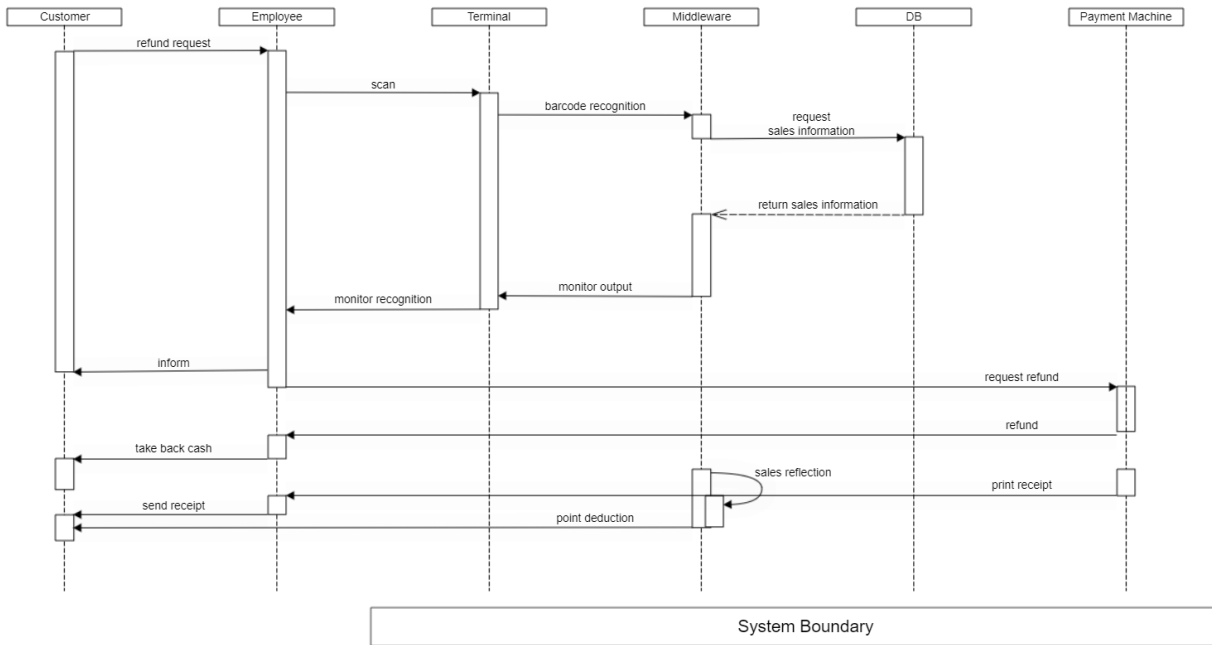
3.4.1 Case 1



이 Sequence Diagram은 고객이 상품을 구매하는 경우의 diagram입니다.

고객이 체크 아웃하면, 직원이 모든 물건을 스캔할 때까지 물건을 시스템에 등록하는 일련의 과정을 반복합니다. 모든 물건이 스캔되면 고객에게 이를 알리고 결제 준비를 합니다. 고객이 현금을 건네면 결제 후 영수증을 출력하고 고객에게 넘겨주면서 이를 매출에 반영합니다.

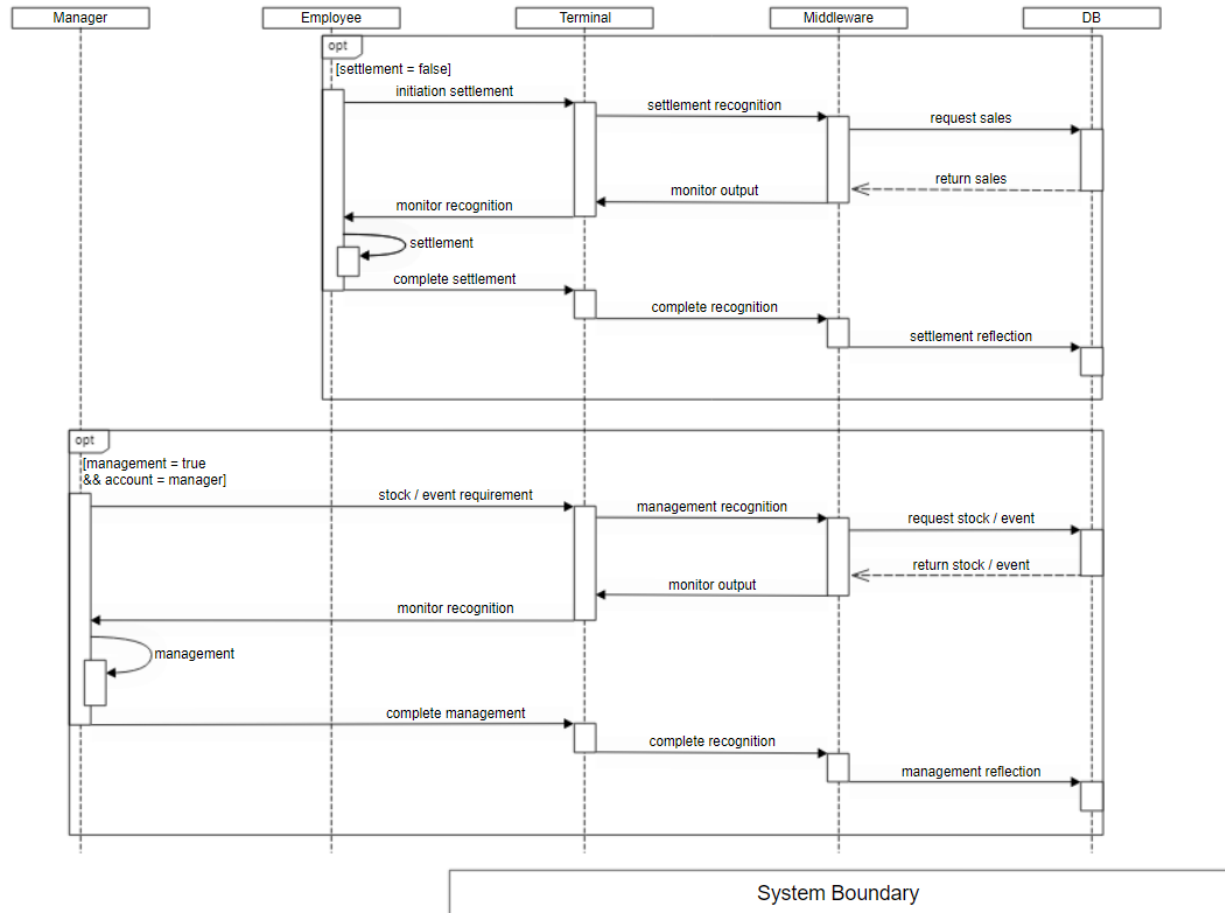
3.4.2 Case 2



이 Sequence Diagram은 고객이 상품을 반품하는 경우의 diagram입니다.

고객이 구매 기록이 있는 영수증과 함께 반품을 신청하면 직원이 영수증의 바코드를 스캔 후 해당 기록을 확인합니다. 그리고 환불 절차를 진행하고 현금을 돌려줍니다. 이후 환불 확인 영수증을 출력 후 고객에게 건네주면서 환불 내역을 매출에 반영합니다.

3.4.3 Case 3



이 **Sequence Diagram**은 직원이 매출을 정산할 때와 매니저가 재고 및 이벤트를 관리하는 경우의 **diagram**입니다.

직원의 정산의 경우 정산을 아직 진행하지 않은 상태에서 정산을 **POS** 시스템에 요청하면 관련 정보를 모니터에 보여줍니다. 해당 정보를 토대로 정산을 진행 후 완료하면 이를 **POS** 시스템에 반영합니다.

매니저의 재고 및 이벤트 관리의 경우 관리가 가능하면서 로그인 계정이 매니저 계정인 상태일 때 **POS** 시스템에 요청하면 해당 정보를 모니터에 보여줍니다. 이를 토대로 재고를 확인하거나 이벤트를 변경하는 등의 관리를 하면 이를 **POS** 시스템에 반영합니다.

3.5 Data Dictionary

In 3.3

Sales_Settlement = [true | false]

Stock_Event = [true | false]

Event_Details = (function calculating actual price)

Event 객체의 Details 정보는 event가 존재하지 않으면 생략 가능하다. Event가 존재하면 product_ID와 Price 정보를 입력받아 실제 가격을 return하는 일종의 function으로 동작한다.

Order_List_Amount = {Event_Details -> Stock_Product_ID, Stock_Price}

Order_List 객체의 Amount 정보는 Stock 객체에 있는 상품의 Price 정보에 Event 객체의 정보를 더하여 계산한 실제 가격의 총합이다.

In 3.4.1

Product Amount

loop의 조건에 포함되는 Product Amount 정보는 시스템에 저장되는 정보가 아니라 고객이 체크아웃에 들고 온 물건의 남은 개수를 의미한다.

Sales Reflection

Middleware에서 진행되는 sales reflection은 고객이 구매한 가격을 매출에 반영하는 과정과 이에 대한 로그를 남기는 등의 과정을 포함한다. (3.4.2에서도 동일)

In 3.4.3

Stock / Event Requirement

매니저의 요청은 재고 관리와 이벤트 처리를 동시에 하는 것이 아니라 실제로는 둘 중 하나를 선택 후 추후 과정을 진행한다.

4. External Interface Requirements

4.1 User Interfaces

홈 화면

사용자는 웹 기반 POS 시스템의 홈 화면에서 주요 기능에 빠르게 접근할 수 있어야 합니다.

로그인/회원가입 화면

사용자는 개인 정보와 함께 회원가입을 할 수 있으며, 로그인 시에는 아이디와 비밀번호를 입력하여 접속할 수 있습니다.

카테고리 등록 화면

사용자는 원하는 카테고리를 등록할 수 있습니다.

물품 등록 화면

관리자는 등록된 카테고리에 판매하려는 물품의 상세 정보(이름, 가격, 설명, 사진 등)를 입력하거나 수정할 수 있습니다.

판매 기록 화면

사용자는 특정 기간 동안의 판매 기록을 조회할 수 있습니다. (차트나 그래프 형태의 판매 추이는 **Challenge** 요소로 고려 중)

바코드 입력 화면

사용자는 바코드 이미지를 입력받아 해당 물품의 정보를 가져옵니다.

반품 화면

사용자는 판매된 물품을 반품 처리할 수 있습니다. 반품 사유, 날짜, 수량 등을 입력할 수 있어야 합니다.

이벤트 관리 화면

관리자는 할인, 1+1 행사 등의 이벤트를 생성, 수정, 삭제할 수 있습니다. 이벤트의 유형, 기간, 대상 물품 등을 설정할 수 있어야 합니다.

4.2 Hardware Interfaces

클라우드 서버

웹 기반 POS 시스템은 클라우드 서비스에서 호스팅됩니다.

5. Technical Requirements (Non-functional)

5.1 Performance

응답 시간

사용자의 모든 요청에 대한 시스템의 응답 시간은 2초 이내여야 합니다.

동시 사용자 처리

시스템은 동시에 여러 사용자의 요청을 2초 내에 처리할 수 있어야 합니다.

5.2 Scalability

데이터베이스 확장성

사용자 수나 물품의 종류가 증가할 경우 데이터베이스는 성능 저하 없이 확장될 수 있어야 합니다.

서버 확장성

트래픽이 증가할 경우 추가 서버를 쉽게 추가하여 시스템을 확장할 수 있어야 합니다.

5.3 Security

데이터 암호화

사용자의 개인 정보 및 판매 기록은 암호화되어 저장되어야 합니다.

접근 제어

사용자는 자신의 판매 기록만 확인할 수 있어야 하며, 다른 사용자의 정보나 기록에는 접근할 수 없어야 합니다.

5.4 Maintainability

모듈화

시스템은 모듈화되어 있어야 하며, 각 모듈은 독립적으로 업데이트 및 유지보수가 가능해야 합니다.

문서화

시스템의 모든 기능과 코드는 충분히 문서화되어야 합니다.

5.5 Usability

사용자는 POS 시스템을 효과적으로, 효율적으로, 만족스럽게 사용할 수 있어야 합니다.

기능

- 메뉴 및 아이콘은 사용자가 쉽게 식별하고 선택할 수 있도록 직관적이어야 합니다.
- 매뉴얼 또는 단축키를 제공해 새로운 사용자가 시스템을 빠르게 익힐 수 있어야 합니다.
- 모든 아이콘의 최소 크기를 규정하여 터치 스크린 환경에서 손가락으로 쉽게 사용할 수 있는 크기 및 간격을 고려해야 합니다.
- 누구나 명확하고 이해하기 쉬운 텍스트 레이블을 사용해야 합니다.
- 텍스트가 선명하게 보일 수 있게 크기, 대비, 간격을 고려해야 합니다.
- 사용자의 이해를 돕기 위한 모든 이미지는 고해상도 버전을 사용하고 왜곡을 방지해야 합니다.
- 사용자의 실수를 방지하기 위해 되돌리기 어려운 행동에 대한 경고 메시지나 확인 절차를 제공해야 합니다.
- “뒤로가기”, “홈 화면” 아이콘을 추가해 주문 수정 및 취소를 간편하게 수행할 수 있는 인터페이스를 제공해야 합니다.
- 할인 또는 추가 상품 입력 시 스크린에 알림 메시지를 통해 사용자에게 관련 정보를 제공해야 합니다.
- 거래 처리 시 속도가 빨라야 하며, 지연 없이 주문을 받고 결제를 처리해야 합니다.

5.6 Multi lingual Support

POS 시스템은 사용자의 편의를 위해 다중 언어를 지원해야 합니다.

(언어 지원 우선 순위 : 한국어, 영어)

기능

- POS 시스템의 텍스트가 다중 언어로 번역되어야 합니다.
- 언어에 따라 텍스트, 글꼴, 정렬 및 레이아웃을 조정해야 합니다.
- 출력될 영수증의 텍스트를 다중 언어로 번역해야 합니다.
- 사용자가 언어 설정을 손쉽게 변경할 수 있는 인터페이스를 제공해야 합니다.

5.7 Auditing and Logging

시스템 운영 및 사용을 추적하기 위해 시스템에 대한 모든 변경 사항을 시간순으로 자세히 기록해야 합니다. 이는 사용자의 부정행위 탐지에 이용될 수 있습니다.

기능

- 사용자의 로그인 및 로그아웃 활동을 기록합니다. 누가 시스템에 로그인했는지, 언제 로그인했는지, 로그아웃 시간 등을 포함합니다. 또한 사용자의 활동, 예를 들어 거래, 할인 부여, 환불, 재고 업데이트 등을 기록합니다.
- 어떤 품목이 판매되었는지, 거래 일자 및 시간, 가격을 기록합니다.
- 시스템에서 발생하는 오류, 경고 및 문제에 대한 로그를 기록합니다.
- 로그 파일의 접근 권한과 보안을 관리하며, 민감한 정보가 안전하게 보호될 수 있도록 조치합니다.
- 저장소를 고려해 로그 파일의 적절한 보존 기간을 설정합니다.

5.8 Availability

검사, 복구, 재시작과 같이 전체 시스템에 대해 정의된 가용성 수준을 보장하는 데 필요한 요소가 지정되어야 한다.

기능

- POS 시스템은 재접속 시 초기화면을 제공함으로써 원활한 재시작을 지원해야 합니다.
- DBMS는 백업 기능을 제공해 예상치 못한 상황에 대비해야 합니다.

6 Open Issues

의문점 또는 논의 사항

제안요청서에서는 결제 처리 방법으로 설명된 신용카드, 모바일 지불, 현금이 기술되었습니다. 하지만 신용카드 및 모바일 지불은 결제 처리 방법에서 제거하기로 결정했습니다. 따라서 현금만이 결제 처리 방법으로써 요구됩니다.

우선순위

높음 카테고리로 분류된 것은 반드시 구현되어야 할 요구사항들입니다. 높음, 낮음 카테고리 안에서도 상위에 존재할수록 우선순위가 높습니다.

*높음	*낮음
로그인 카테고리 등록 물품 등록 재고관리 결제 서비스 매출 보고서	영수증 바코드 (반품) 암호화 이벤트 로그