

# Test Case Documentation

소프트웨어 공학 프로젝트: POS기 시스템

2018060646 이한승

2019031858 김재명

2021030873 이동오

2019008813 문원찬

2021051321 가제고 달린

# Table Of Content

## **1.Introduction**

- 1.1 Purpose of Testing
- 1.2 Scope of Testing
- 1.3 Testing Objectives

## **2. Server Test Cases**

- 2.1 Connection
- 2.2 Login
- 2.3 Product

## **3. Login Page Test Cases**

- 3.1 Register
- 3.2 Login
- 3.3 Security

## **4. Business Page Test Cases**

- 4.1 Add (Product, Discount)
- 4.2 Fix (Product, Discount)
- 4.3 Delete (Product, Discount)
- 4.4 Table Paging

## **5. Transaction Page Test Cases**

- 5.1 Purchase List
- 5.2 Calculate
- 5.3 Completing Transaction

## **6. Sales Report Page Test Cases**

- 6.1 Chart and Table
- 6.2 Navigation

## **1. Introduction**

### **1.1 Purpose of Testing**

POS(Point of Sale) 시스템을 테스트하는 목적은 해당 시스템의 기능, 신뢰성, 그리고 보안이 명시된 요구사항을 충족하는지 확인하는 것입니다. 테스트는 거래 프로세스, 재고 관리, 그리고 전반적인 사용자 상호작용 중 시스템의 성능에 영향을 미칠 수 있는 결함이나 문제를 식별하고 해결하는 데 도움이 됩니다.

### **1.2 Scope of Testing**

테스트 범위는 상품 거래, 재고 관리, 직원 기능 및 사용자 인터페이스를 포함한 POS 시스템의 모든 기능적 측면을 포함합니다.

### **1.3 Testing Objectives**

POS 시스템을 테스트하는 주요 목표는 다음과 같습니다:

- 모든 기능이 명시된 요구사항에 따라 작동하는지 확인합니다.
- 정확한 실시간 재고 관리를 보장합니다.
- 직원 관리 기능의 효과를 확인합니다.
- 잠재적인 보안 취약점을 확인하고 해결합니다.

## 2. Server Test Cases

### 2.1 Connection

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/ 실패
ST01	유효한 데이터로 서버 연결이 성공적으로 연결되는지 확인하기	1. 터미널에서 다음 명령을 사용합니다. “node server”	Mongodb link	터미널은 다음을 출력해야 합니다: Using collection: products  Server started on port 8000 Connected Succesfully...	터미널은 예상대로 출력됩니다.	통과
ST02	유효하지 않은 데이터를 서버가 정상적으로 처리하는지 확인하기	1. 터미널에서 다음 명령을 사용합니다. “node server”	Invalid Mongodb link	터미널은 다음을 출력해야 합니다: Using collection: products  Server started on port 8000  MongoServerError: bad auth : authentication failed	터미널은 예상대로 출력됩니다.	통과

## 2.2 Login

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/ 실패
ST03	서버가 이미 등록된 사용자 목록에 액세스할 수 있는지 확인하기	1. 터미널에서 다음 명령을 사용합니다. “curl http://localhost:8000/users”	-	List of users in the format  {ID: “”  password: “”  Name: “”  Role: “”}	터미널은 예상대로 출력됩니다.	통과
ST04	유효한 정보로 새 사용자 추가 확인하기	1. 터미널에서 다음 명령을 사용합니다. curl -X POST -H "Content-Type: application/json" -d '{"ID": "your_user_id", "password": "your_password", "name": "your_name", "role": "your_role"}' http://localhost:8000/register	ID: abc  password:1234  Name: abcd  Role: Manager	{"message": "Registration successful"}	터미널은 예상대로 출력됩니다.	통과

ST05	반복된 정보로 새 사용자 추가 확인하기	1. 터미널에서 다음 명령을 사용합니다.  curl -X POST -H "Content-Type: application/json" -d '{"ID": "your_user_id", "password": "your_password", "name": "your_name", "role": "your_role"}' http://localhost:8000/re gister	ID: abc  password:1234  Name: abcd  Role: Manager	{"error": "ID already in use"}	터미널은 예상대로 출력됩니다.	통과
ST06	유효한 정보로 로그인 확인하기	1. 터미널에서 다음 명령을 사용합니다.  curl -X POST -H "Content-Type: application/json" -d '{"ID": "your_user_id", "password": "your_password"}' http://localhost:8000/log in	ID: your_user_id  password: Your_password	The terminal should send back a token in the format of '{"token": " _"}'	터미널은 예상대로 출력됩니다.	통과

ST07	유효하지 않은 정보로 로그인 확인하기	1. 터미널에서 다음 명령을 사용합니다. curl -X POST -H "Content-Type: application/json" -d '{"ID": "your_user_id", "password": "your_password"}' http://localhost:8000/login	ID: your_user  password: password	{"error": "Incorrect credentials"}	터미널은 예상대로 출력됩니다.	통과
ST08	데이터베이스에서 비밀번호가 안전한지 확인하기.	1. MongoDB user collection 확인합니다.	-	Plain password가 아닌 salt로 암호화된 password를 보여줘야 합니다.	예상대로 표시됩니다.	통과

## 2.3 Product

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/실패
ST09	데이터베이스의 상품 확인하기	1. 터미널에서 다음 명령을 사용합니다. curl http://localhost:8000/products	—	{ "ID": "_", "name": "_", "price": , "counts": }	터미널은 예상대로 출력됩니다.	통과
ST10	유효한 정보로 새 상품 추가하기	1. 터미널에서 다음 명령을 사용합니다. curl curl -X POST -H "Content-Type: application/json" -d '{"ID": "your_product_id", "name": "your_product_name", "price": your_price, "counts": your_counts, "category": "your_category", "event": "your_event"}' http://localhost:8000/products	ID: 1412 Name: 콜라 Price: 2000 Counts:3 Category: "음료" Event: ""	Its should return the product info in the format { "ID": "1412", "name": "콜라", "price": 2000, "counts": 3, "event": "", "category": "음료수", "_id": "656843efbc79dc52770d607b", "__v": 0 }	터미널은 예상대로 출력됩니다.	통과



ST11	유효하지 않은 정보로 새 상품 추가하기(ID가 이미 사용 중이며, 가격 또는 수량이 정수가 아닌 경우)	1. 터미널에서 다음 명령을 사용합니다.  curl -X POST -H "Content-Type: application/json" -d '{"ID": "your_product_id", "name": "your_product_name", "price": your_price, "counts": your_counts,  "category": "your_category" "event": "your_event"}' http://localhost:8000/products	ID: 1412  Name: 콜라  Price: abs  Counts: as  Category: “음료”  Event: “”	{"error": "Error the product could not be added..."}	터미널은 예상대로 출력됩니다.	통과
ST12	유효한 정보로 상품 삭제하기	1. 터미널에서 다음 명령을 사용합니다.  curl -X DELETE http://localhost:8000/products/ name/your_product_name	Your_product-name : 콜라	{"message": "Product deleted successfully"}	Ass expected	통과

ST13	유효하지 않은 정보로 상품 삭제하기(존재 하지 않는 이름)	1. 터미널에서 다음 명령을 사용합니다.  curl -X DELETE http://localhost:8000/products/na me/your_product_name	Your_product-name : 콜라	{"error": "Product not found"}		
ST14	유효한 정보로 상품 이름으로 검색하기	1. 터미널에서 다음 명령을 사용합니다.  curl http://localhost:8000/products/na me/your_product_name	Your_product_name : 새우깡	Show the info of the product in the format { "_id": "", "ID": "", "name": "", "price": , "counts": , "__v": 0, "category": "", "event": "" }	터미널은 예상대로 출력됩니다.	통과
ST15	유효하지 않은 정보로 상품 이름으로 검색하기(존재 하지 않는 이름)	1. 터미널에서 다음 명령을 사용합니다.  curl http://localhost:8000/products/na me/your_product_name	Your_product_name : 콜라	{"error": "Product not found"}	터미널은 예상대로 출력됩니다.	통과

ST16	유효한 정보로 상품 이름으로 업데이트	1. 터미널에서 다음 명령을 사용합니다.  curl -X PUT -H "Content-Type: application/json" -d '{"newName": "new_product_name", "newPrice": new_price, "newCounts": new_counts}' http://localhost:8000/products/na me/your_product_name	newName: 망고  newPrice: 1000  newCounts:30  your_product_name :사과	Show the new info of the product in the format  {"_id": "", "ID": "", "name": "", "price": , "counts": , "__v": 0, "category": "", "event": ""}	터미널은 예상대로 출력됩니다.	통과
ST17	유효하지 않은 정보로 상품 이름으로 업데이트(잘못 된 이름, 가격 또는 수량)	1. 터미널에서 다음 명령을 사용합니다.  curl -X PUT -H "Content-Type: application/json" -d '{"newName": "new_product_name", "newPrice": new_price, "newCounts": new_counts}' http://localhost:8000/products/na me/your_product_name	newName: 망고  newPrice: 1000  newCounts:30  your_product_name :사과	{"error": "Product not found"}	터미널은 예상대로 출력됩니다.	통과

### 3. Login Page Test Cases

#### 3.1 Register

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/ 실패
LT01	회원가입 페이지가 잘 슬라이드되는지 확인하기.	1. Sign Up 버튼을 누릅니다.	-	페이지가 올바르게 슬라이드되고 회원가입 페이지가 표시되어야 합니다.	예상대로 표시됩니다.	통과
LT02	회원가입 페이지에서 문제없이 빈칸에 정보를 입력할 수 있는지 확인하기.	1. Sign Up 버튼을 누릅니다. 2. 빈칸에 정보를 채웁니다. (StoreName, ID, Name, Password)	StoreName: 한양 왕십리1호점 ID: 123 Name: 김재명 Password: Password	빈칸에 있는 정보를 볼 수 있어야 합니다.	예상대로 표시됩니다.	통과
LT03	역할 버튼이 올바르게 작동하는지 확인하기.	1. 역할버튼을 누릅니다	-	버튼을 누른 후 버튼에 색깔이 바뀌는 것을 볼 수 있어야 합니다.	예상대로 표시됩니다.	통과
LT04	Sign up 버튼이 올바르게 작동하는지 확인하기.	1. 빈칸에 정보를 채웁니다. 2. 역할버튼을 누릅니다 3. Sign up 버튼을 누릅니다	ID: 123 Name: 김재명 Password: Password	mongodb에서 데이터베이스의 정보를 볼 수 있어야 합니다.	예상대로 표시됩니다.	통과

### 3.2 Login

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/ 실패
LT05	로그인 페이지에서 문제없이 빈칸에 정보를 입력할 수 있는지 확인하기.	1. 빈칸에 정보를 채웁니다. (ID, Password)	ID:123 Password: Password	빈칸의 ID는 확인할 수 있고, Password는 가려져야 합니다.	예상대로 표시됩니다.	통과
LT06	Sign in 버튼이 제대로 작동하는지 확인하기.	1. Sign in 버튼을 누릅니다.	-	버튼을 누른 후 버튼의 색깔이 바뀌는 것을 볼 수 있어야 합니다.	예상대로 표시됩니다.	통과
LT07	유효한 데이터로 올바르게 로그인되는지 확인하기.	1. 빈칸에 정보를 채웁니다. (ID, Password) 2. Sign in 버튼을 누릅니다.	ID:123 Password: password	POS 시스템의 메인 페이지로 리디렉션되어야 합니다.	예상대로 표시됩니다.	통과
LT09	유효하지 않은 데이터를 올바르게 처리하는지 확인하기.	1. 빈칸에 정보를 채웁니다. (ID, Password) 2. Sign in 버튼을 누릅니다.	ID: 121 Password: password	정보가 잘못되었다는 메시지가 표시됩니다.	예상대로 표시됩니다.	통과

### 3.3 Security

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/ 실패
LT10	로그인/Sign in 페이지에서 password의 보안이 잘 되는지 확인하기.	1.로그인 페이지에서 password 빈칸을 채웁니다. 2. Sign in 페이지에서 password 빈칸을 채웁니다.	Password: password	Password 빈칸의 글자를 ‘*’로만 보여줘야 합니다.	예상대로 표시됩니다.	통과

#### 4. Business Page Test Cases

##### 4.1 Add (Product, Discount)

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/실패
IT01	유효한 정보로 상품이 성공적으로 추가되는지 확인하기	1. 영업 관리 페이지에서 상품 추가 버튼을 클릭합니다. 2. 빈 칸을 유효한 정보로 채우고 추가 버튼을 클릭합니다. 3. 재고 목록을 확인해 상품이 추가되는지 확인합니다.	카테고리: 과일상품명: 오렌지  단가: 1000  수량: 10	재고 목록 표에 항목에 맞게 데이터가 추가됩니다.	상품은 예상대로 추가됩니다.	통과
IT02	유효하지 않은 정보로는 상품이 추가되지 않는지 확인하기	1. 영업 관리 페이지에서 상품 추가 버튼을 클릭합니다. 2. 빈 칸을 유효하지 않은 정보로 채우고 추가 버튼을 클릭합니다. 3. 재고 목록을 확인해 상품이 추가되는지 확인합니다.	카테고리: [공백]  상품명: [공백]  단가: [공백]  수량: [공백]	재고 목록 표에 데이터가 추가되지 않습니다.	상품은 예상대로 추가되지 않습니다.	통과

IT03	유효한 정보로 할인이 성공적으로 추가되는지 확인하기	1. 영업 관리 페이지에서 할인 추가 버튼을 클릭합니다.  2. 빈 칸을 유효한 정보로 채우고 추가 버튼을 클릭합니다.  3. 할인 목록을 확인해 할인이 추가되는지 확인합니다.	상품명 : 오렌지  할인(%) : 20  기한 : 2023-12-1	할인 목록 표에 항목에 맞게 데이터가 추가됩니다.	할인은 예상대로 추가됩니다.	통과
IT04	유효하지 않은 정보로는 할인이 추가되지 않는지 확인하기	1. 영업 관리 페이지에서 할인 추가 버튼을 클릭합니다.  2. 빈 칸을 유효하지 않은 정보로 채우고 추가 버튼을 클릭합니다.  3. 할인 목록을 확인해 할인이 추가되는지 확인합니다.	상품명 : [재고 목록에 존재하지 않는 상품명]  할인(%) : [공백]  기한 : [공백]	할인 목록 표에 데이터가 추가되지 않습니다.	할인은 예상대로 추가되지 않습니다.	통과

\* 유효하지 않은 정보의 Test Case는 하나라도 유효하지 않은 정보를 가지는 경우를 모두 Test한 결과입니다.

#### 4.2 Fix (Product, Discount)

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/ 실패
--------------	--------------	------------	-----------	-------	-------	-----------



IT05	유효한 정보로 상품이 성공적으로 수정되는지 확인하기	1. 영업 관리 페이지에서 상품 수정 버튼을 클릭합니다. 2. 빈 칸을 유효한 정보로 채우고 수정 버튼을 클릭합니다. 3. 재고 목록을 확인해 상품이 수정되는지 확인합니다.	현재 상품명: 오렌지 카테고리 수정: 과일 상품명 수정: 오렌지 단가 수정: 2000 수량 수정: 5	재고 목록에 항목에 맞게 데이터가 수정됩니다.	상품은 예상대로 수정됩니다.	통과
IT06	유효하지 않은 정보로는 상품을 수정할 수 없는지 확인하기	1. 영업 관리 페이지에서 상품 수정 버튼을 클릭합니다. 2. 빈 칸을 유효하지 않은 정보로 채우고 수정 버튼을 클릭합니다. 3. 재고 목록을 확인해 상품이 수정되는지 확인합니다.	현재 상품명: [표에 없는 상품명] 카테고리 수정: [공백] 상품명 수정: [공백] 단가 수정: [공백] 수량 수정: [공백]	재고 목록에 데이터가 수정되지 않습니다.	상품은 예상대로 수정되지 않습니다.	통과

IT07	유효한 정보로 할인이 성공적으로 수정되는지 확인하기	1. 영업 관리 페이지에서 할인 수정 버튼을 클릭합니다. 2. 빈 칸을 유효한 정보로 채우고 수정 버튼을 클릭합니다. 3. 할인 목록을 확인해 할인이 수정되는지 확인합니다.	상품명: 오렌지  할인 수정: 40  기한 수정: 2023-12-2	할인 목록에 항목에 맞게 데이터가 수정됩니다.	할인은 예상대로 수정됩니다.	통과
IT08	유효하지 않은 정보로는 할인을 수정할 수 없는지 확인하기	1. 영업 관리 페이지에서 할인 수정 버튼을 클릭합니다. 2. 빈 칸을 유효하지 않은 정보로 채우고 수정 버튼을 클릭합니다. 3. 할인 목록을 확인해 상품이 수정되는지 확인합니다.	상품명: [표에 없는 상품명]  할인 수정: [공백]  기한 수정: [공백]	할인 목록에 데이터가 수정되지 않습니다.	할인은 예상대로 수정되지 않습니다.	통과

\* 유효하지 않은 정보의 Test Case는 하나라도 유효하지 않은 정보를 가지는 경우를 모두 Test한 결과입니다.

#### 4.3 Delete (Product, Discount)

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/실패
--------------	--------------	------------	-----------	-------	-------	-------

IT09	유효한 정보로 상품이 성공적으로 제거되는지 확인하기	1. 영업 관리 페이지에서 상품 제거 버튼을 클릭합니다. 2. 빈 칸을 유효한 정보로 채우고 제거 버튼을 클릭합니다. 3. 재고 목록을 확인해 상품이 제거되는지 확인합니다.	상품명: 오렌지	재고 목록에서 상품명에 해당하는 행이 제거됩니다.	상품은 예상대로 제거됩니다.	통과
IT10	유효하지 않은 정보로는 상품을 제거할 수 없는지 확인하기	1. 영업 관리 페이지에서 상품 제거 버튼을 클릭합니다. 2. 빈 칸을 유효하지 않은 정보로 채우고 제거 버튼을 클릭합니다. 3. 재고 목록을 확인해 상품이 제거되는지 확인합니다.	상품명: [표에 없는 임의의 상품명]	재고 목록에 데이터에 영향을 미치지 못합니다.	재고 목록은 예상대로 변화하지 않습니다.	통과
IT11	유효한 정보로 할인이 성공적으로 제거되는지 확인하기	1. 영업 관리 페이지에서 할인 제거 버튼을 클릭합니다. 2. 빈 칸을 유효한 정보로 채우고 제거 버튼을 클릭합니다. 3. 할인 목록을 확인해 상품이 제거되는지 확인합니다.	상품명: 오렌지	할인 목록에서 상품명에 해당하는 행이 제거됩니다.	할인은 예상대로 제거됩니다.	통과

IT12	유효하지 않은 정보로는 할인을 제거할 수 없는지 확인하기	1. 영업 관리 페이지에서 할인 제거 버튼을 클릭합니다. 2. 빈 칸을 유효하지 않은 정보로 채우고 제거 버튼을 클릭합니다. 3. 할인 목록을 확인해 할인이 제거되는지 확인합니다.	상품명: [표에 없는 임의의 상품명]	할인 목록에 데이터에 영향을 미치지 못합니다.	할인 목록은 예상대로 변화하지 않습니다.	통과
------	---------------------------------	--	----------------------	---------------------------	------------------------	----

#### 4.3 Table Paging

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/실패
IT13	재고 목록 표가 한 페이지에 5개의 상품만 표시되는지 확인하기	1. 영업 관리 페이지에서 재고 목록에 상품을 6개 이상 추가합니다. 2. 재고 목록 1 페이지에 5개의 상품만 표시되는지 확인합니다.	6개 이상의 임의의 상품을 추가한 표	재고 목록 표가 한페이지에 5개의 상품만 표시합니다.	표는 예상대로 5개의 상품만 나타납니다.	통과

IT14	재고 목록 표의 이전, 다음 버튼이 정상 작동하는지 확인하기	1. 영업 관리 페이지에서 재고 목록의 이전, 다음 버튼을 클릭합니다.  2. 표가 제대로 표시되는지 확인합니다.	이전, 다음 버튼 클릭	이전 버튼을 누르면 현재 페이지의 전 상품들을 표시하고 다음 버튼을 누르면 현재 페이지의 다음 상품들을 표시합니다.	버튼은 예상대로 작동합니다.	통과
IT15	할인 목록 표가 한 페이지에 5개의 상품만 표시되는지 확인하기	1. 영업 관리 페이지에서 할인 목록에 할인을 6개 이상 추가합니다.  2. 할인 목록 1 페이지에 5개의 할인만 표시되는지 확인합니다.	6개 이상의 임의의 할인을 추가한 표	할인 목록 표가 한페이지에 5개의 상품만 표시합니다.	표는 예상대로 5개의 할인만 나타납니다.	통과
IT16	할인 목록 표의 이전, 다음 버튼이 정상 작동하는지 확인하기	1. 영업 관리 페이지에서 할인 목록의 이전, 다음 버튼을 클릭합니다.  2. 표가 제대로 표시되는지 확인합니다.	이전, 다음 버튼 클릭	이전 버튼을 누르면 현재 페이지의 전 할인들을 표시하고 다음 버튼을 누르면 현재 페이지의 다음 할인들을 표시합니다.	버튼은 예상대로 작동합니다.	통과

## 5. Transaction Page Test Cases

### 5.1 Purchase List

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/ 실패
PT01	상품 목록에 상품을 추가할 수 있는지 확인하기	1.추가하려는 상품명이 있는 버튼을 누릅니다 2. 상품 목록에 해당 상품이 표시되는지 확인합니다.	-	상품 목록에 상품이 잘 표시합니다.	예상대로 표시됩니다	통과
PT02	+, - 버튼을 확인하여 상품 수량을 정확하게 추가 및 감소시키는지 확인하기	1. 목록에 상품을 추가합니다 2. + 버튼을 누릅니다. 3. - 버튼을 누릅니다.	-	버튼을 누른 후 상품의 수량이 감소 및 증가해야 합니다.	예상대로 표시됩니다.	통과
PT03	위쪽 및 아래쪽 버튼을 사용하여 목록에서 이동할 수 있는지 확인하기.	1. 목록에 상품을 추가합니다 2. 위쪽 버튼을 누릅니다. 3.아래쪽 버튼을 누릅니다.	-	목록에서 이동할 수 있어야 합니다.	예상대로 표시됩니다.	
PT04	선택 취소 버튼이 올바르게 작동하는지 확인하기.	1. 목록에 상품을 추가합니다 2. 위쪽 및 아래쪽 버튼을 사용하여 상품을 선택합니다. 3. 선택 취소	-	선택한 상품은 목록에서 삭제되어야 합니다.	예상대로 표시됩니다.	

		버튼을 누릅니다.				
PT05	전체 최소 버튼이 올바르게 작동하는지 확인하기.	1. 전체 최소 버튼을 누릅니다.	-	모든 상품이 목록에서 삭제되어야 합니다.	예상대로 표시됩니다.	통과

## 5.2 Calculate

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/ 실패
PT06	넘버패드가 제대로 작동하는지 확인하기.	1. 넘버 패드의 숫자를 누릅니다. 2. 엔터를 누릅니다.	1000 누릅니다.	값은 받은금액 부분에 나타나야 합니다.	예상대로 표시됩니다.	통과
PT07	총금액이 올바르게 계산되는지 확인하기.	1. 목록에 상품을 추가합니다. 2. 총금액을 확인합니다.	상품을 누릅니다.	상품 목록의 총금액이 표시되어야 합니다.	예상대로 표시됩니다.	통과
PT09	할인이 올바르게 계산되는지 확인하기.	1. 목록에 상품을 추가합니다. 2. 할인금액을 확인합니다.	할인이 있는 상품을 누릅니다.	할인 가격이 표시되어야 합니다.	예상대로 표시됩니다.	통과
PT10	받을 금액이 올바르게 계산되는지 확인하기.	1. 목록에 상품을 추가합니다. 2. 받을 금액을 확인합니다.	상품을 누릅니다.	할인 적용 후 전체 구매 가격을 표시해야 합니다.	예상대로 표시됩니다.	통과

PT11	거스름돈이 올바르게 계산되는지 확인하기.	1. 목록에 상품을 추가합니다. 2. 넘버패드로 지불하는 금액을 추가합니다.	상품을 누릅니다. 지불하는 금액을 누릅니다.	거스름돈 금액이 표시되어야 합니다.	예상대로 표시됩니다.	통과
------	------------------------	---	-----------------------------	---------------------	-------------	----

### 5.3 Completing Transaction

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/ 실패
PT12	결제 버튼이 제대로 작동하는지 확인하기	1. 목록에 상품을 추가합니다. 2. 넘버패드로 지불하는 금액을 추가합니다. 3. 결제 버튼을 누릅니다	상품을 누릅니다. 지불하는 금액을 누릅니다.	거래가 완료되고 목록의 상품이 삭제되어야 합니다.	예상대로 표시됩니다.	통과
PT13	X 버튼이 제대로 작동하는지 확인하기.	1. X 버튼을 누릅니다.	-	영업관리 페이지로 이동할 수 있어야 합니다.	예상대로 표시됩니다.	통과



## 6. Sales Report Page Test Cases

### 6.1 Chart and table

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/ 실패
RT01	차트가 올바르게 표시되는지 확인하기.	1.매출현황 버튼을 누릅니다.	-	판매된 상위 5개 상품이 포함된 차트가 보여야 합니다.	예상대로 표시됩니다.	통과
RT02	판매 상품 목록이 올바르게 표시되는지 확인하세요.	1.매출현황 버튼을 누릅니다.	-	판매 상품 목록이 잘 표시되어야 합니다. 형식은 날짜, 상품명, 단가, 수량, 총액이 포함됩니다.	예상대로 표시됩니다.	통과

### 6.2 Navigation

Test Case ID	Test Case 설명	Test Steps	Test Data	예상 결과	실제 결과	통과/ 실패
RT03	다음 및 이전 버튼이 올바르게 작동하는지 확인하기.	1.다음 버튼이 누릅니다. 2.이전 버튼이 누릅니다	-	문제 없이 페이지를 이동할 수 있어야 합니다.	예상대로 표시됩니다.	통과