Self-Check Kiosk 기능 명세서 (Functional Specification)

문서 버전	작성일	작성자	변경 이력
v1.0	2025.11.XX	유채영	최초 작성 (손님용 결제 프로세스 중심)
v2.0	2026.02.09	유채영	관리자 로그인, 백엔드 API 도입, 데이터 보안 강화

0. 개요 (Overview)

0.1 프로젝트 정의

본 프로젝트는 웹캠 기반의 상품 스캔 기능과 주류 구매 시 3 단계 성인 인증 프로세스를 탑재한 무인 키오스크 시스템이다. v2.0 에서는 보안 강화를 위해 백엔드 API 서버를 도입하여, 데이터(상품, 로그)의 직접 접근을 차단하고 관리자 인증(로그인)을 통해서만 매출 내역을 조회할 수 있도록 고도화한다.

0.2 시스템 아키텍처 (v2.0 변경)

- Frontend: HTML/CSS/JS (키오스크 UI, 웹캠 제어)
- **Backend [New]**: Node.js Express 또는 Python FastAPI (데이터 제공, 인증 처리, 로그 저장)
- Database: JSON 파일 또는 SQLite (상춤 정보, 결제 로그, **관리자 계정 [New]**)

1. 화면 구성 (UI Structure)

화면 ID	화면 명	접근 권한	설명
P-01	**메인 스캔 화면**	손님	가장 기본이 되는 대기 화면. 웹캠 영상을 통해 상품 바코드를 스캔하고, 우측에는 인식된 상품 목록(장바구니)과 총 결제 금액을 표시함.
P-02	**1 차 성인 인증 팝업**	손님	장바구니에 주류가 포함된 상태로 결제 버튼 클릭 시 가장 먼저 뜨는 모달 창. 사용자의 성인 여부를 자가 확인하는 단순 질문을 던짐.
P-03	**2 차 법적 책임 동의 팝업**	손님	1 차 팝업에서 '예'를 선택했을 때 이어서 뜨는 강력한 경고 창. 허위 정보 제공에 대한 법적/경제적 책임을 구매자 본인이 지겠다는 명시적 동의를 구함.
P-04	**3 차 신분증 인식 화면 (시뮬레이션)**	손님	2 차 팝업 동의 후, 실제 신분증 인식을 요청하는 단계. MVP 에서는 실제 OCR 기능을 '인식 성공' 테스트 버튼으로 대체하여 시나리오를 진행함.
P-05	**최종 결제 확인 팝업**	손님	모든 인증 단계(1~3 차)가 성공적으로 완료된 후, 최종적으로 결제할 상품 목록과 금액을 확인하고 결제를 확정하는 팝업.
P-06	**관리자 로그인 팝업**	**[v2.0] 관리자**	P-01 히든 버튼 클릭 시 노출. 아이디/비밀번호 입력 모달
P-07	**관리자 대시보드**	**[v2.0] 관리자**	로그인 성공 시 진입. 매출/결제 로그 조회 및 로그아웃 기능.

2. 상세 기능 명세 (Detailed Features)

A. 일반 사용자(손님) 모드

A-1. 상품 스캔 및 장바구니

- F-01 (상시 스캔): 웹캠은 항상 활성화되어 바코드를 인식함.
- F-02 (상품 정보 조회) [v2.0 변경]:
    - (기존) 프론트엔드 JS 가 로컬 상품 DB 파일을 직접 읽음.
    - (변경) 프론트엔드가 백엔드 API 를 호출하여 정보를 받아옴.

A-2. 성인 인증 프로세스 (v1.0 유지)

- F-04 (결제 진입 판별): '결제하기' 클릭 시 장바구니 내 주류 포함 여부를 확인.
- F-05 (상품 정보 조회): "19 세 이상입니까?" 팝업. '아니오' 시 주류 삭제 후 복귀.
- F-06 (2 차 인증): 법적 책임 동의 경고. '동의' 시 다음 단계, '취소' 시 주류 삭제
- F-07 (3 차 인증): 신분증 인식 시뮬레이션 버튼 제공

A-3. 결제 및 로그 저장 [v2.0 변경]:

- (기존) 프론트엔드가 로컬 로그 파일에 직접 기록.
- (v2.0 변경) 프론트엔드가 백엔드 API 로 결제 데이터를 전송. 백엔드가 이를 DB 에 저장.

B. 관리자 모드 [v2.0 신규]

B-1. 관리자 접근 및 인증

- F-09 (히든 엔트리): 메인 화면(P-01)의 특정 영역(우측 로고)을 롱프레스 시 로그인 팝업(P-06)을 호출한다.
- F-10 (로그인 처리):
    - 관리자 아이디와 비밀번호를 입력받아 백엔드 API 로 전송.
    - 백엔드는 DB 의 암호화된 정보와 대조 후, 유효하면 JWT 액세스 토큰을 발급한다.
    - 프론트엔드는 발급받은 토큰을 localStorage 에 저장하고 관리자 대시보드(P-07)로 화면을 전환한다.

B-2. 보안 로그 조회

- F-11 (로그 데이터 로딩):
  - 대시보드 진입 시 자동으로 백엔드 API 를 호출.
  - [보안 핵심] 이때 HTTP 요청 헤더(Authorization)에 저장된 JWT 토큰을 반드시 포함해야 한다.
  - 토큰이 없거나 유효하지 않으면 백엔드는 401/403 에러를 리턴하고, 프론트엔드는 로그인 팝업으로 튕겨낸다.

B-3. 세션 관리

- F-12 (로그아웃): 대시보드의 '로그아웃' 버튼 클릭 시 localStorage 의 토큰을 삭제하고 메인 스캔 화면(P-01)으로 복귀한다.

3. 데이터베이스 스키마 (Database Schema)

백엔드에서 관리하는 데이터 구조 정의

3.1. Admins (관리자 계정)

Field	Type	Description
id	Integer	Primary Key
username	String	관리자 로그인 ID
password	String	Bcrypt 로 암호화된 해시값 (평문 저장 금지)

3.2. Products (상품 정보)

Field	Type	Description
barcode	String	상품 바코드 (Key)
name	String	상품명
price	Integer	가격
is_alcohol	Boolean	주류 여부 (true/false)

4. API 인터페이스 명세 (API Specification)

4.1. 인증 (Auth)

- **POST /api/login**
  - 설명: 관리자 로그인 및 토큰 발급
  - **Request:** `{ "username": "admin", "password": "1234" }`
  - **Response:** `{ "success": true, "token": "eyJhbGciOiJIUz..." }`
  -

4.2. 상품 (Products)

- **GET /api/products/{barcode}**
  - 설명: 바코드로 상품 단건 조회 (인증 불필요)
  - **Response:** { "name": "참이슬", "price": 1500, "is_alcohol": true }

4.3. 로그 (Logs)

- **POST /api/logs**
  - 설명: 손님 결제 내역 저장 (인증 불필요)
  - **Request:** `{ "items": [...], "total": 5000, "timestamp": "..." }`

- **Response:** `{ "success": true }`

- **GET /api/logs**
  - **설명: [보안]** 전체 결제 로그 조회 (관리자 토큰 필수)
  - **Header:** `Authorization: Bearer <JWT_TOKEN>`
  - **Response:** [{ "date": "...", "items": "소주 2 병", "total": 3000 }, ...]
  - **Error:** 토큰 누락 시 `401 Unauthorized`

5. 시나리오 흐름 (User Flow)

Scenario A: 손님 주류 결제 (v1.0)

1. 손님: 상품 스캔 -> [시스템] 백엔드에서 상품 정보 조회.
2. 손님: 결제 버튼 클릭 -> [시스템] 주류 감지.
3. [시스템] 1 차/2 차/3 차 인증 팝업 진행.
4. 손님: 최종 결제 -> [시스템] 백엔드로 로그 데이터 전송 및 저장.

Scenario B: 관리자 로그 확인 (v2.0)

1. 관리자: 메인 화면 로고 5 회 터치 (히든 버튼).
2. [시스템] 로그인 팝업 노출.
3. 관리자: 아이디/비번 입력 후 로그인 클릭.
4. [시스템] 백엔드 검증 -> 성공 시 토큰 발급 -> 대시보드 진입.
5. [시스템] (자동) 토큰을 실어 로그 조회 API 호출 -> 리스트 출력.
6. 관리자: 확인 후 '로그아웃' 클릭 -> 메인 화면 복귀.