

# 01\_overview\_table\_order\_portfolio

## 테이블오더 재현 시스템 포트폴리오 – 1차 역기획 문서

저장 위치(권장): /docs/01\_1st\_reverse\_planning.md

문서 성격: **프로젝트 기준 문서(1차 역기획)** — 이후 모든 세부 기획/구현 논의의 기준으로 사용

### 1. 문서 목적

본 문서는 테이블오더 시스템을 포트폴리오로 재현하기 위한 1차 역기획 문서이다. 실제 상용 환경의 모든 요소를 구현하는 것이 아니라, **포트폴리오에서 재현 가능한 범위**에서 핵심 구조, 역할 분리, 주요 흐름, 그리고 설계 판단의 이유를 명확히 남기는 것을 목표로 한다.

본 문서는 이후 단계(세부 기획, ERD 확정, 구현 논의)의 기준 문서로 사용된다.

### 2. 프로젝트 핵심 주제

본 포트폴리오는 다음 세 가지 핵심 주제를 중심으로 구성된다.

#### 1. 테이블오더 흐름 재현

테이블에서 주문이 발생하고, 주방(KDS)에서 이를 확인·처리하는 전체 흐름을 재현한다.

#### 2. 토스페이먼츠 결제 연동 재현

테이블 단말에서 결제가 이루어지고, 결제 결과가 주문 상태에 반영되는 구조를 구현한다.

#### 3. 점포별 운영 세팅 재현

점주가 가입 후 각 지점(Store)마다 서로 다른 메뉴, 가격, 테이블 수를 설정·관리할 수 있는 구조를 설계한다.

### 3. 용어 정의 및 제품 구성(운영 주체 기준)

본 포트폴리오에서는 화면/권한의 혼선을 막기 위해 아래 용어를 먼저 정의한다.

- 점주(Owner):** 사업자(사업자등록번호) 단위의 운영 주체. 하나의 점주가 여러 지점(Store)을 가질 수 있다.
- 주방(KDS 사용자):** 주문을 접수/조리/완료로 처리하는 현장 운영 주체. 본 포트폴리오에서는 '점주 앱' 내부의 KDS 화면으로 재현한다.
- 테이블 메뉴 화면(손님용 UI):** 테이블 태블릿에서 주문·결제를 수행하는 손님용 화면. 본 포트폴리오에서는 별도 단말/앱을 강제하지 않고, 점주 앱에서 미리보기/테스트 가능한 형태로도 접근 할 수 있게 재현한다.

- **테이블오더 관리자(서비스 제공자)**: 여러 점주의 지점들을 서비스하는 운영 주체(플랫폼 관리자). 점주 데이터의 온보딩/검증/운영 정책을 관리한다.

### 3-1. 점주 및 주방용 앱 (점주 앱)

대상 사용자: 점주(사업자), 주방(KDS 사용자)

점주 앱에서 제공되는 화면(상위 3분류)

#### 1. 점주 회원가입 및 로그인

- 인증/인가가 통과되어야만 아래 운영 화면에 진입 가능

#### 2. 주방용 KDS 화면

- 신규 주문 확인, 주문 상태 처리(접수/조리중/완료)
- 실시간 간신은 MQTT 이벤트 전파를 통해 재현

#### 3. 매장 테이블용 메뉴 화면

- 테이블 태블릿에서 보여지는 메뉴/주문/결제 흐름을 재현
- 포트폴리오에서는 별도 손님용 웹 제공 범위를 포함하지 않고, 점주 앱 내에서 **미리보기/테스트 가능한 형태로 재현**한다.

포트폴리오에서 재현하는 기능 범위(요약)

- 점주 회원가입 / 로그인(사업자번호 기반)
- 점주 1명당 여러 지점(Store) 관리
- 지점별 테이블 구성 설정
- 지점별 메뉴판 구성(메뉴/가격 관리)
- 주문 목록 및 주문 상태 처리(KDS)
- 결제 결과 및 운영 리포트 조회(가상 정산 수준)

### 3-2. 테이블오더 관리자용 콘솔 (서비스 제공자용)

참고: 관리자 콘솔은 웹 또는 앱 중 하나로 제공할 수 있으나, 현 단계에서는 구현 형태(웹/앱)를 확정하지 않는다. 대상 사용자: 테이블오더 서비스 운영자(플랫폼 관리자)

역할: 여러 점주의 지점들을 서비스하기 위한 관리/운영 기능 제공

포트폴리오에서 재현하는 기능 범위:

- 점주/지점 목록 관리(플랫폼 관점)
- 사업자번호 검증 버튼 및 검증 결과 관리(가입 시점은 형식 검증, 콘솔에서 API 검증)
- 운영 정책 관점의 모니터링(필요 시): 지점별 주문/결제 이벤트 상태 확인

주의: 본 콘솔은 '손님용 테이블 UI'가 아니라, 서비스 제공자(플랫폼 관리자) 관점의 관리 화면이다.

---

## 4. 전체 시스템 구조 (개념 아키텍처)

### 구조 요약

- 서버(DB)가 모든 데이터의 기준(**Source of Truth**)
- 실시간성은 MQTT를 통해 이벤트 전파 방식으로 처리

개념 구조:

- 테이블 UI → 서버: 주문/결제 요청
- 서버 → DB: 주문·결제·상태 저장
- 서버 → MQTT: 주문/상태 변경 이벤트 전파
- 주방(KDS) / 점주 앱: MQTT 이벤트 수신 후 화면 갱신

### 설계 이유

- 데이터 저장과 실시간 전달을 분리함으로써 시스템 책임을 명확히 하기 위함
  - 네트워크 단절, 앱 종료 등 상황에서도 데이터 유실 없이 복구 가능한 구조를 만들기 위함
  - 다수 클라이언트(테이블, 주방, 점주 화면)에 동일 이벤트를 효율적으로 전달하기 위함
- 

## 5. 실무 환경 대비 포트폴리오 적용 범위

- 실무 환경에서는 주방 프린터와 KDS를 병행하여 주문 안정성을 확보한다.
- 본 포트폴리오에서는 재현 가능한 범위에서 하드웨어 복잡도를 줄이기 위해 주방 태블릿(KDS) 하나로 해당 역할을 대체한다.

이는 실무 구조를 이해한 상태에서 의도적으로 범위를 조정한 결정이다.

---

## 6. 핵심 워크플로우 (레벨 0)

### 6-1. 점포 세팅 흐름 (점주 앱)

1. 점주 회원가입 / 로그인
  2. 지점(Store) 생성 또는 선택
  3. 지점별 테이블 수 및 구성 설정
  4. 지점별 메뉴판 및 가격 설정
  5. 운영 시작
-

## 6-2. 주문 흐름 (테이블 ↔ 주방)

1. 테이블 UI에서 주문 생성
  2. 서버가 주문 데이터를 저장
  3. MQTT를 통해 “신규 주문” 이벤트 전파
  4. 주방(KDS)에서 주문 확인 및 상태 변경
  5. 서버가 상태 변경 저장
  6. MQTT를 통해 상태 변경 이벤트 전파
  7. 테이블 UI에 주문 진행 상태 반영
- 

## 6-3. 결제 흐름 (테이블 UI)

1. 테이블 UI에서 결제 요청
  2. 토스페이먼츠 결제 흐름 진행(샌드박스)
  3. 결제 결과를 서버에 반영
  4. 주문 상태를 결제 완료 상태로 전환
  5. 점주 앱에서 결제 결과 확인 가능
- 

# 7. 계정 구조 기준 및 사업자번호 정책

## 기본 계정 구조 기준

본 포트폴리오에서는 **점포(사업자)당 하나의 계정**을 사용하는 구조를 기본 전제로 한다.

이는 국내 상용 테이블오더 서비스(예: 티오더)의 실제 운영 방식과, 한국의 사업자 등록 제도를 반영한 결정이다.

## 왜 점포당 1 계정 구조인가

### 1) 법·제도적 이유 (가장 핵심적인 배경)

- 사업자등록번호는 **사업장 주소 단위**로 발급된다.
- 매장이 서로 다른 경우:
  - 사업자등록번호가 서로 다르거나
  - 최소한 지점 사업자 형태로 별도 등록이 필요하다.
- 카드 가맹, 결제 정산, 세금 신고는 모두 **사업자 단위**로 귀속된다.

따라서,

한 사업자 = 한 점포 = 한 계정

이라는 구조가 가장 단순하며, 운영 및 책임 분리 측면에서도 분쟁 소지가 적다.

## 확장 시나리오에 대한 위치 정의

- 프랜차이즈 본사 또는 다점포 사업자와 같이 여러 점포를 운영하는 경우, 실제 서비스 환경에서는 **점포별로 별도 계정을 생성하는** 방식이 일반적이다.
- 본 포트폴리오에서는 이러한 다점포 통합 관리를 **확장 시나리오**로만 정의하고, 기본 구현 범위에는 포함하지 않는다.

## 사업자번호 가입 및 검증 정책

- 점주는 사업자등록번호를 기반으로 가입한다.
- **가입 시점**에는 형식 검증을 통해 입력 오류만 최소화한다.
- **관리 화면**에는 “사업자 검증” 버튼을 제공하며, 해당 시점에 공식 API를 통한 검증을 수행한다.
- 검증 미통과 시, 서비스 신뢰성 확보를 위해 제한 조치가 가능하도록 정책적 여지를 남긴다.

구체적인 검증 단계와 제한 수위는 이후 단계에서 확정한다.

## 8. 데이터 모델 및 ERD 접근 방향

1차 역기획 단계에서는 세부 DB 필드 설계는 진행하지 않는다.

다만 다음 원칙만을 합의한다:

- 모든 운영 데이터는 지점(Store) 단위로 분리된다.
- 주문과 결제는 개념적으로 분리된다.
- 실시간 이벤트(MQTT)는 DB 모델을 그대로 복제하지 않는다.

ERD는 이후 단계에서 본 문서의 흐름을 만족하는 최소 구조로 설계한다.

## 9. 다음 단계 기획을 위한 체크포인트

다음 단계(2차 문서)에서는 아래 항목을 중심으로 세부 기획을 확정한다.

- 테이블 식별 방식 결정
- MQTT 이벤트 탑재 최소 세트 정의
- 주방(KDS) UI를 점주 앱에 포함할지 여부
- 가상 정산 집계 기준
- 최소 ERD(엔티티 수준) 확정

본 문서는 테이블오더 시스템의 **큰 흐름과 설계 판단을 공유하기 위한 기준 문서**이며, 이후의 모든 세부 기획은 본 문서를 기준으로 확장한다.