

# 정산 시스템 전체 점검 보고서

작성일: 2026년 2월 7일 | 프로젝트: 탑셀 주문관리 시스템

## 1. DB 스키마 현황

### 1-1. members 테이블 잔액 컬럼

컬럼명	데이터 타입	기본값	필수	설명
deposit	integer	0	YES (NOT NULL)	예치금 잔액
point	integer	0	YES (NOT NULL)	포인트 잔액

▣ DB 확인 완료 - 두 컬럼 모두 정상 존재

### 1-2. settlement\_history 테이블 (정산 이력)

컬럼명	데이터 타입	필수	기본값	설명
id	varchar	YES	gen_random_uuid()	PK
member_id	varchar	YES	-	회원 ID (FK: members.id)
order_id	varchar	YES	-	주문 ID
settlement_type	text	YES	'auto'	정산 유형 (auto/manual)
pointer_amount	integer	YES	0	포인트 차감액
deposit_amount	integer	YES	0	예치금 차감액
total_amount	integer	YES	0	정산 총액
pointer_balance	integer	YES	0	정산 후 포인트 잔액
deposit_balance	integer	YES	0	정산 후 예치금 잔액
description	text	NO	-	설명
created_at	timestamp	YES	now()	생성일시

인덱스: member\_id, order\_id, created\_at

### 1-3. deposit\_history 테이블 (예치금 변동 이력)

컬럼명	데이터 타입	필수	기본값	설명
id	varchar	YES	gen_random_uuid()	PK
member_id	varchar	YES	-	회원 ID (FK: members.id)
type	text	YES	-	유형 (charge/refund/deduct)
amount	integer	YES	-	변동 금액
balance_after	integer	YES	-	변동 후 잔액
description	text	NO	-	설명
admin_id	varchar	NO	-	처리 관리자 ID
related_order_id	varchar	NO	-	관련 주문 ID
created_at	timestamp	YES	now()	생성일시

인덱스: member\_id, created\_at, type

#### 1-4. pointer\_history 테이블 (포인터 변동 이력)

컬럼명	데이터 타입	필수	기본값	설명
id	varchar	YES	gen_random_uuid()	PK
member_id	varchar	YES	-	회원 ID (FK: members.id)
type	text	YES	-	유형 (grant/deduct)
amount	integer	YES	-	변동 금액
balance_after	integer	YES	-	변동 후 잔액
description	text	NO	-	설명
admin_id	varchar	NO	-	처리 관리자 ID
related_order_id	varchar	NO	-	관련 주문 ID
created_at	timestamp	YES	now()	생성일시

인덱스: member\_id, created\_at, type

## 2. API 목록 및 상세

### 2-1. 관리자 API

#### [API 1] 예치금 충전

```
HTTP POST /api/admin/members/:memberId/deposit/charge
파라미터: server/routes.ts (9363)
요청 헤더: 
요청 바디: { amount: number, description?: string }
응답 코드: 
SELECT FOR UPDATE FROM : (tx.select().for('update'))
deposit_history WHERE : (type: "charge")
응답 코드:
- 예상 오류 401
- 예상 오류 403
- amount <= 0 400
- 예상 오류 404
```

#### [API 2] 예치금 환급

```
HTTP POST /api/admin/members/:memberId/deposit/refund
파라미터: server/routes.ts (9402)
요청 헤더: 
요청 바디: { amount: number, description?: string }
응답 코드: 
SELECT FOR UPDATE FROM : 
deposit_history WHERE : (type: "refund")
응답 코드:
- 예상 오류 400 ("예상 오류 예상 오류")
- 예상 오류 404
```

#### [API 3] 포인터 지급

```
HTTP POST /api/admin/members/:memberId/pointer/grant
파라미터: server/routes.ts (9445)
요청 헤더: 
요청 바디: { amount: number, description?: string }
응답 코드: 
SELECT FOR UPDATE FROM : 
pointer_history WHERE : (type: "grant")
응답 코드: 예상 오류 404, amount <= 0 400
```

#### [API 4] 정산 이력 조회

```
□□: GET /api/admin/settlements
□□ □□: server/routes.ts (9484□)
□□: □□ □□ □□ □□ (□□□□ □□, □□□)
□□ □□□□: memberId, startDate, endDate, type, page, limit
□□: { records, total, page, limit }
KST □□ □□: □ (parseDateRangeKST □□)
```

#### [API 5] 예치금 이력 조회

```
□□: GET /api/admin/deposit-history
□□ □□: server/routes.ts (9547□)
□□: □□ □□□ □□ □□ □□
□□ □□□□: memberId, startDate, endDate, type, page, limit
□□: { records, total, page, limit }
```

#### [API 6] 포인트 이력 조회

```
□□: GET /api/admin/pointer-history
□□ □□: server/routes.ts (9607□)
□□: □□ □□□ □□ □□ □□
□□ □□□□: memberId, startDate, endDate, type, page, limit
□□: { records, total, page, limit }
```

#### [API 7] 회원 잔액 목록 조회

```
□□: GET /api/admin/members-balance
□□ □□: server/routes.ts (9667□)
□□: □□ □□□ □□□/□□□ □□ □□ □□
□□: [{ id, companyName, grade, deposit, point, username }]
```

#### [API 8] 특정 회원 잔액 상세 조회

```
□□: GET /api/admin/members/:memberId/balance
□□ □□: server/routes.ts (9338□)
□□: □□ □□□ □□ □□ □□ □□ □□
□□: { memberId, companyName, grade, deposit, point, totalBalance, pendingOrdersTotal,
availableBalance }
```

### 2-2. 회원 API

#### [API 1] 내 잔액 조회

```
□□: GET /api/member/my-balance
□□ □□: server/routes.ts (9696□)
```

```
    deposit: { deposit, point, pendingOrdersTotal, availableBalance }  
    calculateAvailableBalance deposit: deposit
```

## [API 2] 내 정산 이력 조회

```
    00: GET /api/member/my-settlements
    00  00: server/routes.ts (97150)
    00  000: startDate, endDate, page, limit
    00: { records, total, page, limit }
    00  00: 00  00 ID00  00
```

### [API 3] 내 예치금 이력 조회

```
□: GET /api/member/my-deposit-history  
□: server/routes.ts (9755□)  
□: { records, total, page, limit }
```

#### [API 4] 내 포인터 이력 조회

```
□□: GET /api/member/my-pointer-history  
□□ □□: server/routes.ts (9795□)  
□□: { records, total, page, limit }
```

### 3. 잔액 검증 로직 상세

#### 3-1. "사용 가능 잔액" 계산

```
○○ ○○: server/routes.ts (5828○) - calculateAvailableBalance ○○  
  
○○ ○○:  
    ○○ ○○ ○○ = (○○○ + ○○○) - ○○ ○○ ○○  
  
○○ ○○ ○○ ○○: ["○○", "○○○○", "○○○○○"]  
○○ ○○ ○○ ○○: ["○○", "○○", "○○○", "○○○"]  
  
○○ ○○ ○○:  
    1. priceConfirmed=true○○ supplyPrice ○○ -> ○○○ supplyPrice ○○  
    2. supplyPrice○○ -> ○○○ supplyPrice ○○  
    3. supplyPrice ○○ -> current_products ○○○○ ○○ ○○ ○○ ○○  
        (getSupplyPriceByGrade ○○: START->startPrice, DRIVING->drivingPrice, TOP->topPrice)  
  
○○ ○○ ○○ ○○: ○
```

#### 3-2. 엑셀 업로드 시 잔액 검증

```
○○ ○○: server/routes.ts (6524○)  
  
○○ ○○ (○○ ○○ ○):  
    ① SHA-256 ○○ ○○ ○○  
    ○○ ○○ ○○  
    ○○ ○○ ○○ ○○ (current_products)  
    ○○○○ ○○ ○○  
    ○○ ○○ ○○  
    -> [○○ ○○ ○○ ○○] <- ○○/○○ ○○ ○, ○○ ○○ ○○  
    ○○ ○○ (JUSO API + AI)  
    ○○ ○○  
  
○○○ ○○ ○○ ○○: ○  
    - pendingValidationRows (○○ ○○ ○○ ○○)○ ○○○○ ○○  
    - getSupplyPriceByGrade(currentProduct, member.grade) ○ ○○  
  
○○ ○○ ○○ ○○:  
    status: 'insufficient_balance'  
    total: ○○ ○○ ○○  
    validCount: ○○○ ○○  
    errorCount: ○○○ ○○  
    totalOrderAmount: ○○○ ○○○○  
    balanceInfo: { deposit, point, pendingOrdersTotal, availableBalance, shortage }  
    errors: ○○○ ○○ (○○ ○○ ○○ ○○)
```

订单状态未通过验证：失败（余额不足）

validation\_failed 订单金额未通过：

totalOrderAmount, balanceInfo, balanceSufficient 余额不足

## 4. 배송중 전환 자동 정산 상세

파일 경로: server/routes.ts (7781行)  
API 경로: POST /api/admin/orders/to-shipping

수정 내용:

1. 헤더 설정 (요청하는 모드, mode로 설정: all/filtered/selected)
2. 헤더 및 헤더 설정 확인
3. 헤더 설정
4. 헤더 설정 확인 (모든 헤더 설정 확인):
  - a. SELECT FOR UPDATE 헤더 설정
  - b. 헤더 설정 확인 (getSupplyPriceByGrade)
  - c. 헤더 설정
  - d. 헤더 설정 -> 헤더 설정
  - e. 헤더 설정 "정상화" 헤더 설정, supplyPrice 설정, priceConfirmed=true
  - f. settlement\_history 헤더 설정
  - g. pointer\_history 헤더 설정 (정상화 헤더 설정)
  - h. deposit\_history 헤더 설정 (정상화 헤더 설정)
  - i. members 헤더 설정
5. SSE 이벤트 송신 (pending-orders-updated, order-status-changed)

수정 내용: 헤더 설정 -> 헤더 설정

수정 내용: 헤더 설정

수정 내용: 헤더 설정 (정상화 헤더 설정)

SELECT FOR UPDATE 헤더 설정

settlement\_history 헤더 설정

deposit\_history 헤더 설정 (정상화 헤더 설정)

pointer\_history 헤더 설정 (정상화 헤더 설정)

수정 내용: 헤더 설정 (정상화 헤더 설정) (정상화 헤더 설정)

- 헤더 설정, 헤더 설정/정상화 헤더 설정

수정 내용: 헤더 설정

- 헤더 설정: pending\_orders.supplyPrice, pending\_orders.priceConfirmed
- 헤더 설정: current\_products 헤더 설정 (getSupplyPriceByGrade)

## 5. 프론트엔드 현황

### 5-1. 관리자 정산관리 페이지

파일 경로: client/src/pages/admin/settlements.tsx (610줄)  
URL 주소: /admin/settlements  
문서 크기: client/src/App.tsx (174줄)

- 요소 목록:
- "members" - 회원 목록 (이름, 나이, 성별, 등급 정보)
  - "settlements" - 정산 목록 (정산 일자)
  - "deposit" - 예금 목록 (예금/출금/정산 일자)
  - "pointer" - 포인트 목록 (포인트/날짜)

정산/예금/출금 항목별로:  
- deposit-charge: 예금 목록  
- deposit-refund: 출금 목록  
- pointer-grant: 포인트 목록

DateRangeFilter 항목: 날짜 (날짜/월/연도 정산 일자 및 예금/출금)

### 5-2. 회원 대시보드 잔액 표시

파일 경로: client/src/pages/dashboard.tsx

정산 목록: DB 조회 (/api/member/my-balance)  
예금 목록: DB 조회 (/api/member/my-balance)

"정산내역" 템: 표시 (활성 탭 activeTab="deposit"으로 설정)  
정산내역 템: client/src/pages/member/settlement-tab.tsx (369줄)  
- MemberSettlementTab 템으로 표시  
- 날짜, 정산 유형, 정산 금액, 정산 일자 표시

### 5-3. 주문 등록 화면 잔액 배너

파일 경로: client/src/pages/dashboard.tsx (1247줄)

주문 가능 여부: canOrder=true && balanceData 가능 여부  
3개의 경우:  
- >= 10000: 표시 (이상)  
- > 0 & < 5000: 표시 + "0원 이상" 표시 + "5000 원 미만" 표시  
- <= 0: 표시 + "0원 미만" 표시 + "5000 원 이상" 표시

주문 가능 여부: canOrder+balanceData 가능 여부, 표시 가능 여부, 표시 가능 여부

```
data-testid: card-balance-banner, text-balance-total, text-pending-total, text-available-balance
```

## 5-4. 잔액 부족 시 알림 UI (엑셀 업로드)

문서 번호: client/src/pages/dashboard.tsx (1749쪽)

문서 내용: 대화상자 (Dialog)

- 대화상자 + OK: 대화상자 + 대화상자
- 대화상자: 대화상자 대화상자 (max-w-500px)
- 대화상자 + 대화상자: 대화상자 (max-w-560px), "확인 버튼" 대화상자
- 대화상자 대화상자 OK: 대화상자 + 대화상자 대화상자 + "확인 버튼" 대화상자

문서 내용:

- 대화상자, 대화상자 대화상자, 대화상자
- 대화상자
- 대화상자, 대화상자, 대화상자 대화상자, 대화상자 대화상자
- 대화상자 (대화상자 대화상자)

문서 내용: 대화상자 대화상자 대화상자 대화상자 "확인 버튼" 대화상자 대화상자

## 5-5. 회원 이력 조회 화면

문서 번호: client/src/pages/member/settlement-tab.tsx (369쪽)

문서 내용: 대화상자 > "설정" 대화상자 (설정 대화상자)

문서 내용: 대화상자

문서 내용: 대화상자

문서 내용: 대화상자

문서 내용: 대화상자 (DateRangeFilter 설정 대화상자)

## 6. 기존 기능 영향 확인

### 6-1. 주문 관련

확인 항목	상태	상세
엑셀 주문 등록 - 기존 검증 흐름	▣ 정상	SHA-256 중복, 필수필드, 상품존재, 공급중지, 주소검증 순서 유지. 잔액 검증은 상품 검증 후, 주소 검증 전에 삽입
검증 오류 시ダイ얼로그	▣ 정상	기존 validation_failed 응답에 balanceInfo, balanceSufficient 필드만 추가. 기존 필드 변경 없음
"정상건만 등록" 기능	▣ 정상	잔액 충분 시 기존과 동일하게 작동. 잔액 부족 + 오류 동시 발생 시에만 버튼 비활성화
오류건 엑셀 자동 다운로드	▣ 정상	errorExcelData 필드 기존 그대로 유지
주문 상태 전환 워크플로우	▣ 정상	대기->상품준비중->배송준비중->배송중 흐름 유지. 배송중 전환에 정산 로직 추가
운송장 전달 기능	▣ 정상	기존 로직 변경 없음
취소마감 기능	▣ 정상	배송중 전환(mode=all) 시 취소마감 초기화 로직 유지
주문 취소 처리	▣ 정상	취소된 주문은 잔액 계산 시 자동 제외 (inProgressStatuses에 미포함)

### 6-2. 회원 관련

확인 항목	상태	상세
PENDING/ASSOCIATE 등급 주문 차단	▣ 정상	canOrder 로직 유지: START/DRIVING/TOP만 주문 가능
등급별 가격체계(3-tier)	▣ 정상	getSupplyPriceByGrade 함수로 START/DRIVING/TOP 가격 매핑
회원 대시보드 기존 기능	▣ 정상	기존 탭 유지 + "예치금충전" 탭 추가
사이드바 메뉴 구조	▣ 정상	기존 메뉴에 "예치금충전" 항목만 추가
로그인/인증	▣ 정	세션 기반 인증 변경 없음

### 6-3. 관리자 관련

확인 항목	상태	상세
관리자 대시보드 기존 기능	▣ 정상	매출/매입/주문관리/상품관리 기존 기능 변경 없음
관리자 사이드바 메뉴	▣ 정상	기존 메뉴 유지 + "정산관리" 메뉴 추가 (admins.tsx에 settlement_management 권한 등록)
메뉴 접근 권한	▣ 정상	SUPER_ADMIN이 ADMIN에게 정산관리 메뉴 권한 설정 가능

### 6-4. 시스템/인프라 관련

확인 항목	상태	상세
SSE 실시간 알림	▣ 정상	배송중 전환 시 pending-orders-updated, order-status-changed 이벤트 발송 유지
엑셀 업로드/다운로드	▣ 정상	기존 파일/다운로드 로직 변경 없음
API 라우팅 충돌	▣ 정상	신규 API 경로(/api/admin/settlements, /api/member/my-* 등) 기존 경로와 충돌 없음
DB 마이그레이션 안전성	▣ 정상	신규 테이블 3개 추가 (settlement_history, deposit_history, pointer_history). members 테이블에 deposit/point 컬럼 추가 (기본값 0, 기존 데이터 보존)
기존 쿼리 정상 작동	▣ 정상	신규 컬럼은 NOT NULL DEFAULT 0이므로 기존 쿼리에 영향 없음

### 6-5. UI/스타일 관련

확인 항목	상태	상세
기존 페이지 레이아웃	▣ 정상	신규 컴포넌트는 기존 레이아웃 내에 탭/카드 형태로 추가. 기존 UI 변경 없음
반응형(모바일) 표시	▣ 정상	잔액 배너, 정산 다이얼로그 모두 반응형 grid/flex 적용

기존 컴포넌트 스타일	<span style="color: green;">정상</span>	DateRangeFilter, 테이블, 토스트 등 공통 컴포넌트 스타일 변경 없음
-------------	---------------------------------------	---

## 7. 종합 결론

### 정산 시스템 구현 상태: 완료

- DB 스키마: members 테이블 잔액 컬럼 + 3개 이력 테이블 모두 정상 생성
- 관리자 API: 8개 엔드포인트 구현 (충전/환급/지급, 이력 조회, 잔액 목록)
- 회원 API: 4개 엔드포인트 구현 (잔액 조회, 정산/예치금/포인트 이력)
- 잔액 검증: 엑셀 업로드 시 정상건 기준 검증, 부족 시 등록 차단
- 자동 정산: 배송중 전환 시 포인터 우선 차감, 트랜잭션/SELECT FOR UPDATE 적용
- 프론트엔드: 관리자 정산관리 페이지, 회원 잔액 배너/이력 조회 구현
- 기존 기능 영향: 모든 항목 정상 확인 (기존 기능 보존)