

# YMV ERP 개발 백업 보고서

Generated: 2024-12-19 | Session: Step 6 견적서 관리 시스템 개발

## 1. 시스템 현황

### 1.1 프로젝트 정보

- 프로젝트명: YMV 관리 프로그램 v4.0
- 위치: D:\ymv-business-system
- 현재 단계: Step 6 완료 (견적서 관리 시스템)
- 전체 완성도: 약 60%
- 배포 URL: <https://ymv-business-system.streamlit.app>
- 테스트 계정: Master / 1023

### 1.2 개발 환경

- 프레임워크: Streamlit
- 데이터베이스: Supabase (PostgreSQL)
- 언어: Python
- 배포: Streamlit Cloud

## 2. 파일 구조 현황

### 2.1 현재 디렉토리 구조

```
D:\ymv-business-system/  
├── app/  
│   ├── main.py ✓ (800줄, 리팩토링 필요)  
│   └── components/  
│       ├── __init__.py ✓  
│       ├── code_management.py ✓ (정상 작동)  
│       ├── multilingual_input.py ✓ (정상 작동)  
│       └── quotation_management.py ✓ (Step 6에서 생성, 테스트 필요)  
├── database/  
│   ├── upgrade_v4.sql ✓ (실행됨)  
│   ├── additional_schema_fix.sql ✓ (실행됨)  
│   ├── fix_expenses_schema_safe.sql ✓ (실행됨)  
│   └── quotation_schema.sql ⚠ (생성됨, 실행 불필요 - 테이블 이미 존재)  
├── requirements.txt ✓  
├── .streamlit/config.toml ✓  
└── .env ✓ (Supabase 연결 정보)
```

## 2.2 파일 상태 분석

- **main.py**: 800줄로 과도하게 길어서 리팩토링 필요
- **components/**: 컴포넌트 구조는 잘 구축됨
- **quotation\_management.py**: 이번 세션에서 생성, 테스트 미완료

## 3. 함수 목록 및 역할

### 3.1 main.py 주요 함수

```
python

# 초기화 및 유틸리티
init_supabase() - Supabase 연결 초기화
generate_unique_key(prefix) - 고유 위젯 키 생성

# 데이터베이스 연결 (공통 함수 - 리팩토링 대상)
load_data_from_supabase(table, columns, filters) - 데이터 로드
save_data_to_supabase(table, data) - 데이터 저장
update_data_in_supabase(table, data, id_field) - 데이터 수정
delete_data_from_supabase(table, item_id, id_field) - 데이터 삭제

# 사용자 관리
login_user(username, password) - 로그인 처리
logout_user() - 로그아웃 처리
get_current_user() - 현재 사용자 정보

# 통계 및 유틸리티
get_approval_status_info(status) - 승인 상태 정보
calculate_expense_statistics(expenses) - 지출 통계 계산
create_csv_download(expenses, employees) - CSV 다운로드
render_print_form(expense) - 프린트 양식 렌더링

# 페이지 함수들 (리팩토링 대상)
show_login_page() - 로그인 페이지 (50줄)
show_dashboard() - 대시보드 (100줄)
show_expense_management() - 지출 관리 (400줄) ← 최우선 리팩토링 대상
show_purchase_management() - 구매품 관리 (100줄)
show_quotation_management() - 견적서 관리 (임시 래퍼)

# 메인 함수
main() - 메인 애플리케이션 로직
```

### 3.2 components/quotation\_management.py 함수

```
python
```

```
# 데이터 함수 (main.py와 중복 - 향후 utils.py로 이동)
load_data_from_supabase(conn, table, columns, filters)
save_data_to_supabase(conn, table, data)
update_data_in_supabase(conn, table, data, id_field)
delete_data_from_supabase(conn, table, item_id, id_field)

# 견적서 전용 함수
generate_quote_number() - 견적서 번호 생성 (YMV-Q-2024-XXXX)
get_quotation_status_info(status) - 견적서 상태 정보
render_quotation_form(conn, quotation_data, is_edit) - 견적서 작성/수정 폼
render_quotation_list(conn) - 견적서 목록 표시
render_quotation_print(quotation, items, customer) - 견적서 출력
show_quotation_statistics(conn) - 견적서 통계

# 메인 함수
show_quotation_management(conn) - 견적서 관리 메인 함수 (3개 탭)
```

3.3 components/code\_management.py

```
python

class CodeManagementComponent:
    __init__(self, supabase) - 초기화
    render_code_management_page(self) - 제품 코드 관리 페이지
    # 기타 제품 코드 관련 함수들
```

4. 데이터베이스 구조

4.1 완전 구현된 테이블 (28개)

테이블명	기능	레코드 수	상태
employees	직원 관리/로그인	다수	정상 작동
expenses	지출 요청서	다수	100% 완성
customers	고객 정보	1건	요구사항 충족
products	제품 정보	2건	요구사항 충족
company_info	회사 정보	1건	요구사항 충족
quotations	견적서 마스터	0건	테이블 존재, 컴포넌트 구현됨
quotation_items	견적서 상세	0건	테이블 존재, 컴포넌트 구현됨
product_codes	제품 코드	다수	v4.0 기능, 정상 작동
purchases	구매품 관리	다수	기본 CRUD 구현

4.2 고급 기능 테이블 (이미 존재)

테이블명	기능	활용도
products_multilingual	다국어 제품명	견적서에서 활용 가능
exchange_rates	환율 정보	다중 통화 지원
translations	다국어 번역	시스템 전반
audit_logs	감사 로그	시스템 추적
user_permissions	사용자 권한	Phase 2에서 구현 예정

4.3 데이터베이스 연결 해결 이슈

- expenses 테이블: business\_purpose 컬럼 NOT NULL 제약조건 해결됨
- quotations/quotation\_items: 테이블 구조 완성됨
- 모든 기본 데이터: customers(1), products(2), company\_info(1) 준비됨

5. 이번 세션 완료 내용

5.1 주요 성과

1. quotation\_management.py 컴포넌트 완성
  - 3개 탭 구성 (작성/목록/통계)
  - 완전한 CRUD 기능
  - 다국어 견적서 출력
  - 자동 견적번호 생성
  - 실시간 제품 추가 및 합계 계산
2. 시스템 분석 완료
  - 전체 업무 프로세스 맵 작성
  - 데이터 연결 관계 분석
  - 권한 시스템 설계 방향 제시
3. 개발 규칙 체계화
  - 백업/연속성 규칙 추가
  - 한글/영어 병행 문서화
  - AI 이해도 향상을 위한 구조화

5.2 해결된 문제점

- ModuleNotFoundError: quotation\_management 모듈 누락 해결
- Form Submit Button: form 구조 문제 해결
- 변수 참조 오류: 'name' 오류 등 변수 초기화 문제 해결

- DB 상태 확인: 실제 테이블 존재 여부 확인 완료

## 6. 현재 미해결 이슈

### 6.1 테스트 필요 사항

- quotation\_management.py 실제 동작 테스트
- 견적서 작성/수정/삭제 기능 확인
- 출력 기능 정상 작동 확인
- 데이터베이스 연결 안정성 테스트

### 6.2 구조적 문제

- main.py 파일 크기: 800줄로 과도하게 큼
- 중복 함수: DB 연결 함수가 여러 파일에 중복
- 컴포넌트 연결 방식 불일치: 각기 다른 방식으로 연결

## 7. 다음 개발 계획 (승인됨)

### Phase 1: 시스템 안정화 (1-2주)

#### 1. quotation\_management.py 테스트 및 버그 수정

- 실제 동작 확인
- 오류 발생시 수정
- 기능 완성도 100% 달성

#### 2. main.py 리팩토링

- expense\_management.py 컴포넌트 분리 (400줄 → 컴포넌트)
- dashboard.py 컴포넌트 분리 (100줄 → 컴포넌트)
- utils.py 공통 함수 분리 (200줄 → 유틸리티)
- main.py 최종 크기: 800줄 → 100줄

### Phase 2: 권한 시스템 구축 (1주)

#### 3. 직원 권한 관리 시스템

- user\_permissions 테이블 활용
- 메뉴별 접근 권한 제어
- 기능별 권한 체크 함수

### Phase 3: 기본 마스터 데이터 완성 (2주)

#### 4. 고객 관리 시스템 구축

- customers 테이블 CRUD

- 고객 정보 관리 컴포넌트

## 5. 제품 관리 시스템 구축

- products 테이블과 product\_codes 연결
- 다국어 제품명 관리
- 제품 정보 완전 관리

## Phase 4: 영업 프로세스 완성 (2-3주)

### 6. 견적서-주문서 연결 시스템

- orders 테이블 생성 및 관리
- 견적서 승인 → 주문서 자동 생성

### 7. 인보이스 시스템

- invoices 테이블 및 관리 기능

## 8. 기술적 고려사항

### 8.1 컴포넌트 설계 패턴

- 일관된 인터페이스: 모든 컴포넌트가 동일한 방식으로 DB 연결
- ConnectionWrapper 클래스: quotation\_management.py에서 사용 중
- 공통 유틸리티: utils.py로 중복 함수 집약

### 8.2 권한 시스템 구현 방향

- user\_permissions 테이블 활용
- 모듈별 권한 체크 함수
- 메뉴 표시 제어 로직

### 8.3 성능 최적화 방향

- 컴포넌트별 캐싱
- DB 쿼리 최적화
- 세션 상태 관리 개선

## 9. 주의사항 및 알려진 제약

### 9.1 Supabase 연결 제약

- 연결 객체 래핑 필요: quotation\_management.py 방식 참조
- 테이블별 권한 설정: RLS(Row Level Security) 고려

## 9.2 Streamlit 제약사항

- 세션 상태 관리: 위젯 키 중복 방지 필수
- 폼 구조: 복잡한 폼은 제품 추가 부분을 폼 외부로 분리
- 리로딩: st.rerun() 사용 시 상태 초기화 주의

## 9.3 개발 규칙 준수

- 전체 파일 단위 코드 제공
- Step 단위 진행
- 사용자 확인 후 다음 단계

## 10. 빠른 시작 가이드

### 10.1 시스템 실행

```
bash
cd D:\ymv-business-system
streamlit run app/main.py
```

### 10.2 테스트 계정

- 사용자명: Master
- 비밀번호: 1023
- 권한: 관리자 (모든 기능 접근 가능)

### 10.3 현재 작동 기능

- ☒ 로그인/로그아웃
- ☒ 대시보드
- ☒ 지출 요청서 관리 (100% 완성)
- ☒ 코드 관리 시스템
- ☒ 구매품 관리 (기본 기능)
- ☐ 견적서 관리 (테스트 필요)

### 10.4 다음 세션 시작 방법

1. 이 백업 파일을 새 채팅창에 업로드
2. "이 백업 파일을 기반으로 YMV 개발을 계속해줘" 요청
3. Phase 1부터 순차적으로 진행

## 11. 개발자 참고 정보

## 11.1 중요 파일 위치

- 메인 애플리케이션: `app/main.py`
- 견적서 컴포넌트: `app/components/quotation_management.py`
- 개발 규칙: 별도 텍스트 파일로 관리
- DB 스키마: `database/` 폴더의 .sql 파일들

## 11.2 개발 환경 요구사항

- Python 3.8+
- Streamlit
- Supabase Python Client
- pandas, datetime 등 기본 라이브러리

## 11.3 배포 정보

- 배포 플랫폼: Streamlit Cloud
- 자동 배포: GitHub 연동
- 환경 변수: .env 파일 또는 Streamlit Secrets

---

백업 생성일: 2024-12-19

세션 ID: Step 6 - 견적서 관리 시스템 개발 완료

다음 예정 작업: Phase 1 - quotation\_management.py 테스트 및 main.py 리팩토링