

YMV ERP 개발 백업 보고서 - V08 규칙 준수

Generated: 2024-09-26 | Session: Step 8 완료 - expense_management.py 분리 + 구매품 관리 개선

1. 시스템 현황

1.1 프로젝트 정보

- 프로젝트명: YMV 관리 프로그램 v4.0
- 위치: D:\ymv-business-system
- 현재 단계: Step 8 완료 (expense_management.py 컴포넌트 분리)
- 전체 완성도: 약 70%
- 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 ✓ (400줄, 50% 단축됨)
│   └── components/
│       ├── __init__.py ✓
│       ├── code_management.py ✓ (정상 작동)
│       ├── multilingual_input.py ✓ (정상 작동)
│       ├── quotation_management.py ✓ (Step 7에서 완전 수정됨)
│       └── expense_management.py ✓ (Step 8에서 분리 완료)
├── database/
│   ├── upgrade_v4.sql ✓ (실행됨)
│   ├── additional_schema_fix.sql ✓ (실행됨)
│   ├── fix_expenses_schema_safe.sql ✓ (실행됨)
│   └── quotation_schema.sql ⚠ (생성됨, 실행 불필요)
└── requirements.txt ✓
```

└── .streamlit/config.toml ✓
└── .env ✓ (Supabase 연결 정보)

2.2 파일 상태 분석

- main.py: 800줄 → 400줄 (50% 단축 완료)
- expense_management.py: 400줄 분리 완료 (Step 8)
- components/: 컴포넌트 구조 완전 구축

3. 함수 목록 및 역할

3.1 main.py 주요 함수 (Step 8 리팩토링 완료)

3.1.1 초기화 및 설정 (최상단)

```
python

# 1. 페이지 설정 (최우선 실행)
st.set_page_config() - 페이지 구성 설정
└── 위치: import 직후, 최상단
└── 매개변수: title, icon, layout, sidebar_state

# 2. 전역 초기화
@st.cache_resource
init_supabase() - Supabase 연결 초기화
└── 호출 위치: 전역 변수 설정
└── 반환: supabase client object
```

3.1.2 데이터베이스 연결 함수 (utils.py로 이동 예정)

```
python
```

Step 8에서 유지, Step 10에서 utils.py로 이동 예정

`load_data_from_supabase(table, columns="*", filters=None)`

└─ 호출 위치: 모든 페이지 함수에서 데이터 로드시

└─ 반환: list of dictionaries

`save_data_to_supabase(table, data)`

└─ 호출 위치: 품 제출시 (지출요청서, 구매품, 견적서)

└─ 반환: boolean

`update_data_in_supabase(table, data, id_field="id")`  Step 7 수정됨

└─ 호출 위치: 데이터 수정시

└─ 수정 내용: data.copy()로 원본 보호, 응답 데이터 확인

`delete_data_from_supabase(table, item_id, id_field="id")`

└─ 호출 위치: 삭제 버튼 클릭시

└─ 반환: boolean

3.1.3 사용자 관리 및 유틸리티 (유지)

python

인증 시스템

`login_user(username, password)`

`logout_user()`

`get_current_user()`

유틸리티 함수들 (Step 10에서 utils.py로 이동 예정)

`get_approval_status_info(status)`

`calculate_expense_statistics(expenses)`

`create_csv_download(expenses, employees)`

`render_print_form(expense)` - CSS 방식으로 개선 필요

3.1.4 페이지 함수들 (호출 구조)

python

main() 호출 흐름 (단순화됨)

main()

└── 로그인 체크

└── show_login_page() (로그인 안됨시)

└── 메뉴별 페이지 호출

└── show_dashboard() (100줄, Step 9에서 분리 예정)

└── show_expense_management_page() → components.expense_management

└── show_purchase_management() (개선 완료)

└── show_quotation_management() → components.quotation_management

└── 기타 메뉴들

각 페이지 함수 상세

show_dashboard() - 대시보드 (100줄) ← Step 9 분리 대상

└── 내부 호출: 안전한 필드 체크, 6개 DB 조회

show_purchase_management() - 구매품 관리 (완전 개선됨)

└── 기능: USD/VND/KRW 통화 지원, 수정/삭제, 권한 관리

└── 탭 구조: 2개 탭 (등록/목록)

show_expense_management_page() - 지출관리 래퍼 (5줄)

└── 호출: components.expense_management.show_expense_management()

└── 매개변수 전달: 9개 함수 전달

3.2 components/expense_management.py 함수 (Step 8에서 분리됨)

3.2.1 메인 함수 및 구조

python

show_expense_management(9개 매개변수) - 메인 컨테이너

- └─ 호출 위치: main.py show_expense_management_page()에서
- └─ 내부 구조: 4개 탭 (작성/목록/통계/CEO승인)
- └─ 매개변수: 모든 DB 및 유틸리티 함수들

render_expense_form(load_data_func, save_data_func, get_current_user_func)

- └─ 기능: employee_id 필드 사용, 안전한 폼 처리
- └─ 검증: 금액 > 0, 지출내역 필수

render_expense_list(7개 매개변수)

- └─ 기능: 필터링, 정렬, CRUD 작업
- └─ 권한: 관리자는 전체, 일반사용자는 본인만

render_expense_statistics(load_data_func, calculate_expense_statistics_func)

- └─ 기능: 상태별, 카테고리별, 월별 통계
- └─ 차트: 월별 건수/금액 바 차트

render_ceo_approval(4개 매개변수)

- └─ 권한: role='manager'만 접근
- └─ 기능: 승인/거부 처리, 승인의견 기록

3.3 components/quotation_management.py (Step 7 완성)

python

show_quotation_management(4개 DB 함수)

- └─ 기능: customers, products 테이블 완전 연동
- └─ total_amount 자동 계산 및 저장

render_quotation_form() - 견적서 작성

render_quotation_list() - 견적서 목록

show_quotation_statistics() - 견적서 통계

3.4 구매품 관리 개선 (Step 8에서 완전 개선)

python

show_purchase_management() - 완전 개선된 구매품 관리

- └─ 통화: USD/VND/KRW 완전 지원
- └─ 기능: 등록/수정/삭제 (권한 관리)
- └─ 필터: 상태/카테고리/통화별
- └─ DB 연동: purchases 테이블 완전 매칭

4. 데이터베이스 구조 및 변경사항

4.1 Step 8에서 수정된 테이블

4.1.1 employees 테이블 (중요한 변경)

sql

-- Step 8에서 추가된 컬럼

```
ALTER TABLE employees ADD COLUMN employee_id VARCHAR(20) UNIQUE;
```

```
UPDATE employees SET employee_id = 'EMP' || LPAD(id::text, 3, '0');
```

-- 현재 구조 (14개 컬럼)

- id (integer, PK)
- name (varchar)
- username (varchar)
- password (varchar)
- department (varchar)
- position (varchar)
- email (varchar)
- phone (varchar)
- is_admin (boolean)
- is_active (boolean)
- notes (text)
- created_at (timestamp)
- updated_at (timestamp)
- employee_id (varchar) ← Step 8에서 추가 (EMP001, EMP002 형태)

4.1.2 purchases 테이블 스키마 확인됨

sql

-- 총 16개 컬럼, Generated Column 포함

- id (integer, PK)
- category (varchar, NOT NULL)
- item_name (varchar, NOT NULL)
- quantity (integer, NOT NULL)
- unit (varchar, DEFAULT '개')
- unit_price (numeric, NOT NULL)
- total_price (numeric, GENERATED ALWAYS AS quantity * unit_price)
- supplier (varchar)
- request_date (date, NOT NULL)
- urgency (varchar, DEFAULT '보통')
- status (varchar, DEFAULT '대기중')
- notes (text)
- requester (integer, FK to employees)
- created_at (timestamp, DEFAULT now())
- updated_at (timestamp, DEFAULT now())
- currency (varchar, DEFAULT 'KRW') ← Step 8에서 활용

4.2 DB 연동 최적화

python

Step 8에서 해결된 DB 연동 이슈들

1. total_price Generated Column 문제 해결
2. employee_id 필드 추가로 KeyError 해결
3. currency 필드 완전 활용 (USD/VND/KRW)
4. urgency 기본값 '보통' 매칭
5. status 기본값 '대기중' 지원

5. 이번 세션 완료 내용 (Step 8)

5.1 주요 성과

1. expense_management.py 컴포넌트 분리

- main.py 800줄 → 400줄 (50% 단축)
- 400줄 지출관리 코드를 완전히 분리
- 9개 매개변수로 깔끔한 인터페이스

2. 구매품 관리 완전 개선

- USD/VND/KRW 다중 통화 완전 지원
- 수정/삭제 기능 추가 (권한 관리 포함)
- 통화별 단계값 자동 조정
- DB 스키마 완전 매칭

3. employees 테이블 employee_id 추가
- EMP001, EMP002 형태로 직원 ID 생성
 - KeyError 완전 해결
 - 지출요청서 시스템 정상화

4. 오류 해결 완료
- set_page_config 위치 수정
 - Generated Column 문제 해결
 - dashboard 안전한 필드 체크 추가

5.2 해결된 오류들

오류명	발생 상황	원인	해결 방법	참고 위치
KeyError: employee_id	지출요청서 접근시	employees 테이블에 employee_id 없음	DB에 employee_id 컬럼 추가	employees 테이블
set_page_config 중복 호출	앱 실행시	main() 함수 내 위치 문제	파일 최상단으로 이동	main.py 상단
Generated Column 오류	구매품 등록시	total_price에 값 전송	total_price 데이터 전송 제거	purchases 테이블
NameError: render_expense_list	지출관리 접근시	함수 정의 누락	완전한 expense_management.py 생성	컴포넌트 파일
통화 표시 오류	구매품 등록시	f"단가 ({currency})" 표시 문제	"단가"로 단순화	폼 라벨

6. 현재 완전 작동 기능

6.1 100% 완성 기능

- 로그인/로그아웃 시스템: 완전 작동
- 대시보드: 안전한 필드 체크로 안정화
- 지출요청서 관리: 4개 탭 모든 기능 완성
 - 지출요청서 작성 (employee_id 연동)
 - 목록 관리 (필터링, 정렬, 삭제)
 - 통계 (상태별, 카테고리별, 월별)
 - CEO 승인 (승인/거부, 의견 기록)
- 견적서 관리: customers/products 완전 연동
- 구매품 관리: USD/VND/KRW 통화, 수정/삭제 완성
- 코드 관리 시스템: 7단계 제품코드 완전 작동

6.2 90% 완성 기능

- **프린트 기능:** HTML/텍스트 다운로드는 작동, CSS 방식 개선 필요

7. 다음 개발 계획 (우선순위)

Phase 1: 시스템 최적화 완료 (1주)

Step 9: dashboard.py 분리

- **목표:** 100줄 대시보드 코드를 컴포넌트로 분리
- **결과:** main.py 400줄 → 300줄
- **개선:** dashboard 독립성 향상

Step 10: utils.py 공통 함수 분리

- **목표:** 200줄 공통 함수들을 유틸리티로 분리
- **결과:** main.py 300줄 → 100줄 (최종 목표)
- **ConnectionWrapper 완전 구현:** 데이터베이스 접근 표준화

Phase 2: 프린트 시스템 완성 (3일)

- CSS @media print 규칙 적용
- 브라우저 프린트 창 자동 실행
- 견적서 + 지출요청서 전용 프린트 레이아웃

Phase 3: 기능 확장 (2주)

- 고객 관리 시스템 완전 구현
- 제품 관리 시스템 완전 구현
- 견적서-주문서-인보이스 연동

8. 기술적 고려사항 및 주의점

8.1 Step 8에서 도입된 패턴

python

컴포넌트 분리 패턴

```
def show_expense_management_page():  
    """지출 관리 페이지 (컴포넌트 호출)"""  
    show_expense_management(  
        # DB 함수들  
        load_data_from_supabase,  
        save_data_to_supabase,  
        update_data_in_supabase,  
        delete_data_from_supabase,  
        # 유틸리티 함수들  
        get_current_user,  
        get_approval_status_info,  
        calculate_expense_statistics,  
        create_csv_download,  
        render_print_form  
    )
```

8.2 다중 통화 지원 패턴

python

통화별 처리 표준화

```
currency_symbol = {'USD': '$', 'VND': '₫', 'KRW': '원'}  
currency_steps = {'USD': 10, 'VND': 10000, 'KRW': 1000}
```

통화별 필터링 및 총계 계산

```
currency_totals = {}  
for item in items:  
    currency = item.get('currency', 'KRW')  
    currency_totals[currency] = currency_totals.get(currency, 0) + item['amount']
```

8.3 권한 관리 패턴

python

표준 권한 체크 패턴

```
current_user = get_current_user()  
can_edit = (current_user and  
             (current_user.get('role') == 'manager' or  
              item.get('requester') == current_user['id']))
```

관리자 전용 기능

```
if user_role == 'manager':  
    # 관리자만 접근 가능한 기능
```

9. импорт 및 의존성 관리

9.1 main.py импорт 구조

```
python

# 표준 라이브러리
import streamlit as st
import pandas as pd
from datetime import datetime, date
import uuid, time
from collections import defaultdict
import calendar, io

# 페이지 설정 (최우선)
st.set_page_config(...)

# 내부 모듈
from components.code_management import CodeManagementComponent
from components.multilingual_input import MultilingualInputComponent
from components.quotation_management import show_quotation_management
from components.expense_management import show_expense_management
```

9.2 컴포넌트별 의존성

```
python

# expense_management.py 의존성
- streamlit (UI)
- pandas (데이터 처리)
- datetime (날짜 처리)
- collections.defaultdict (통계 계산)
- io (CSV 생성)

# 외부 의존성 (매개변수로 전달받음)
- 9개 main.py 함수들 (DB 및 유틸리티)
```

10. Step 8 성능 개선 결과

10.1 코드 구조 개선

- main.py 크기: 800줄 → 400줄 (50% 단축)
- 컴포넌트 분리: expense_management.py 독립
- 재사용성: 컴포넌트 패턴 확립

10.2 사용자 경험 개선

- 구매품 관리: 다중 통화, 수정/삭제 기능 완성
- 권한 관리: 관리자/일반사용자 구분 명확
- 오류 처리: 안전한 필드 접근으로 안정성 향상

10.3 DB 연동 최적화

- 스키마 매칭: 모든 테이블과 완전 일치
- Generated Column: 자동 계산 필드 활용
- 데이터 무결성: 제약조건 및 인덱스 활용

11. 빠른 시작 가이드

11.1 시스템 실행

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

11.2 테스트 시나리오

1. Master / 1023 로그인
2. 구매품관리 → USD/VND/KRW 등록/수정/삭제 테스트
3. 지출요청서 → 4개 탭 모든 기능 테스트
4. 견적서관리 → customers/products 연동 확인

11.3 현재 작동 확인된 기능

- ☒ 로그인/로그아웃
- ☒ 대시보드 (안전한 필드 체크)
- ☒ 지출 요청서 관리 (100% 완성, employee_id 연동)
- ☒ 견적서 관리 (customers/products 연동, 프린트 제외)
- ☒ 구매품 관리 (USD/VND/KRW 통화, 수정/삭제 완성)
- ☒ 코드 관리 시스템

12. 다음 세션 시작 방법

1. 이 백업 파일을 새 채팅창에 업로드
2. "이 백업 파일을 기반으로 YMV 개발을 계속해줘" 요청
3. Step 9 (dashboard.py 분리)부터 시작

4. 목표: main.py 400줄 → 300줄 단축

13. 중요 파일 경로 및 설정

13.1 핵심 파일 위치

```
app/main.py (400줄로 단축됨)
app/components/expense_management.py (Step 8에서 분리됨)
app/components/quotation_management.py (Step 7 완성)
```

13.2 데이터베이스 변경사항

```
sql

-- Step 8에서 실행됨
ALTER TABLE employees ADD COLUMN employee_id VARCHAR(20) UNIQUE;
UPDATE employees SET employee_id = 'EMP' || LPAD(id::text, 3, '0');
```

13.3 환경 설정

```
toml

# .streamlit/secrets.toml
SUPABASE_URL = "your_supabase_url"
SUPABASE_ANON_KEY = "your_supabase_key"
```

백업 생성일: 2024-09-26

세션 ID: Step 8 완료 - expense_management.py 분리 + 구매품 관리 완전 개선 다음 예정 작업: Step 9 - dashboard.py 분리 (main.py 400줄 → 300줄)

주요 성과: 컴포넌트 분리 패턴 확립, 다중 통화 지원, 권한 관리 완성 규칙 준수: V08 완전 준수