

ACCOUNT_INFO

통합계좌관리시스템

요구사항명세서&기능설명서

2060340016 박천호

주제 설명을 위한 및 서비스 구상과 가정사항

- 가정 사항 :

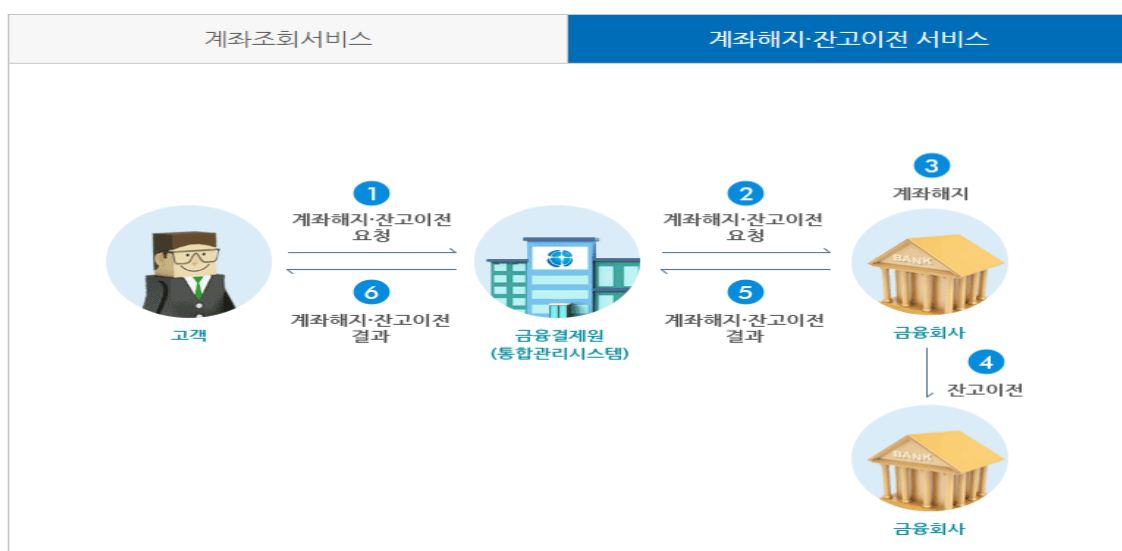
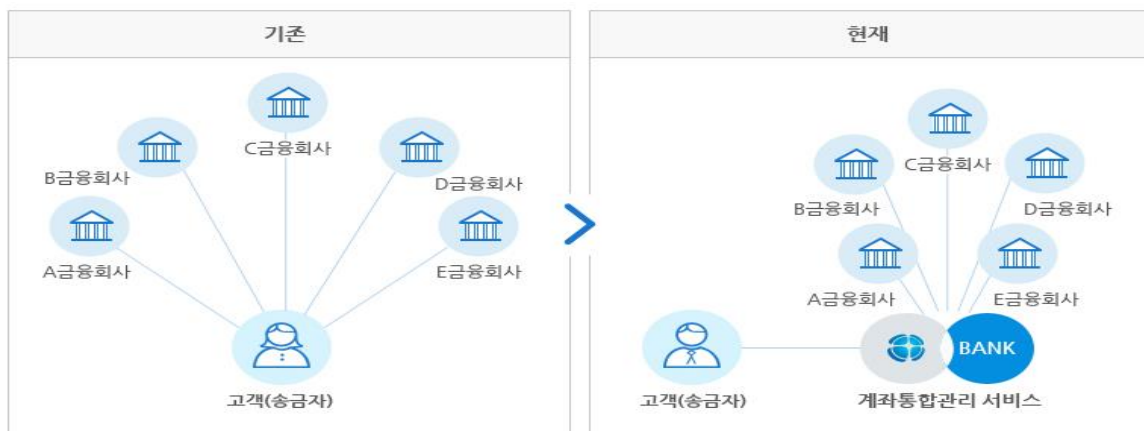
금융 서비스 기관의 접근이 아닌 범국민적 통합계좌 관리 시스템을 새로 구상하는 상황

계좌 조회, 이체 등의 기본적 관리 업무도 지원하지만 본인이 거래하는 금융기관의 계좌를 한 눈에 조회하고 잔고가 적은 소액 비활동성 계좌를 이체를 통해 잔고이전 및 등록삭제(=해지)하도록 돕는 서비스

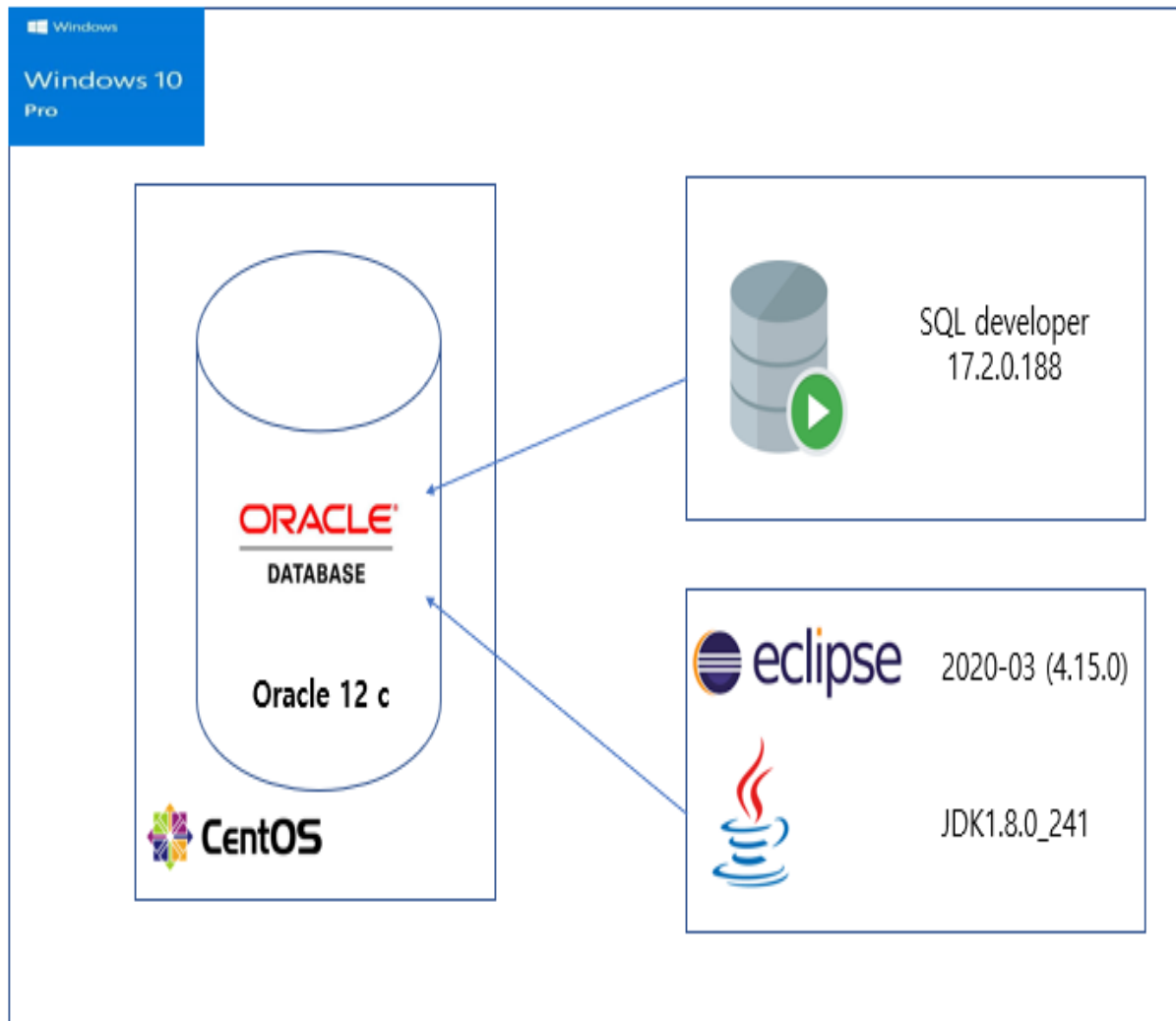
- 서비스 개요 : 통합 계좌 관리 시스템

현재 본인이 보유하고 있는 계좌를 금융감독원 계좌 통합관리 서비스(ACCOUNT_INFO)를 통해 조회 및 관리하고 입출금, 계좌 해지를 지원하는 시스템으로 구상

- 서비스 도식화



개발 사양 설명서 (개발 환경)



Entity 관계와 구조

SCOTT.ACCOUNT	
NAME	VARCHAR2 (10 BYTE)
ID	VARCHAR2 (20 BYTE)
ACCOUNT_NUMBER	VARCHAR2 (20 BYTE)
BANK_CODE	VARCHAR2 (3 BYTE)
BALANCE	NUMBER
BANK_NAME	VARCHAR2 (30 BYTE)
NICKNAME	VARCHAR2 (50 BYTE)

SCOTT.USERS	
NAME	VARCHAR2 (10 BYTE)
P * ID	VARCHAR2 (20 BYTE)
PWD	VARCHAR2 (20 BYTE)
SEQ	NUMBER
REG_DATE	DATE
↳ USERS_PK (ID)	

ACCOUNT

Table logical name

PK	FK	Null	Logical Name	Name	Type
		✓	사용자 이름	NAME	VARCHAR2(10)
		✓	사용자 아이디	ID	VARCHAR2(20)
			사용자의 계좌번호	ACCOUNT_NUM	VARCHAR2(20)
		✓	은행별 코드 세자리수	BANK_CODE	VARCHAR2(3)
			잔액	BALANCE	NUMBER
			은행의 이름	BANK_NAME	VARCHAR2(30)
			계좌의 별명	NICKNAME	VARCHAR2(50)

SQL / Menu

USERS

Table logical name

PK	FK	Null	Logical Name	Name	Type
✓		✓	사용자의 이름	NAME	VARCHAR2(10)
		✓	사용자의 아이디	ID	VARCHAR2(20)
			사용자의 비밀번호	PWD	VARCHAR2(20)
			사용자의 시퀀스	SEQ	NUMBER
			사용자 등록일자	REG_DATE	DATE

SQL / Menu

실제 테이블은 은행 테이블도 있었으나, 활용하지는 못하여서 2개의 테이블이 존재합니다.

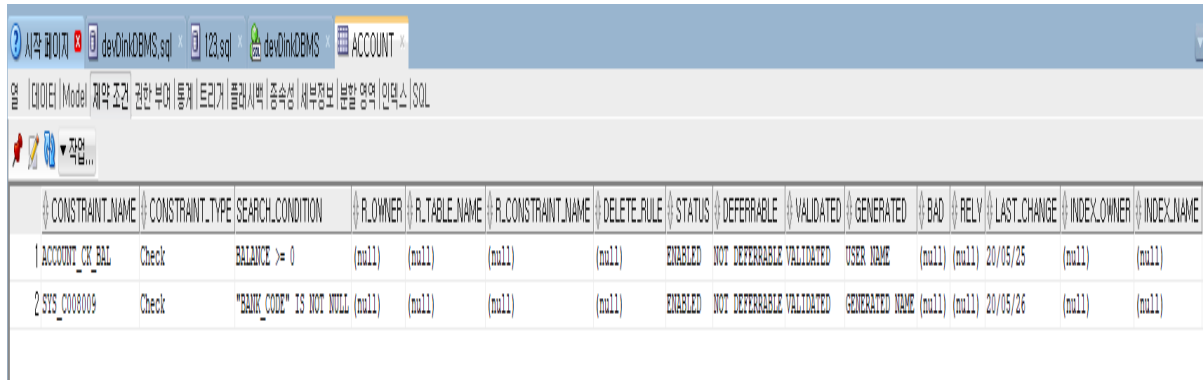
USERS 테이블의 시퀀스와 등록일자도 실질적으로 사용하지는 못했습니다.

USERS 테이블에서 아이디는 PRIMARY KEY, NOT NULL 비밀번호를 NOT NULL로 지정하였습니다.

ACCOUNT 테이블에서 요구사항에 있는 지원하는 은행들만 서비스를 이용할 수 있다는 점을 반영하기 위해 은행 코드를 NOT NULL로 지정하고 계좌 등록시에 입력하도록 하였습니다.

제약조건 CONSTRAINT

ACCOUNT table의 CONSTRAINT

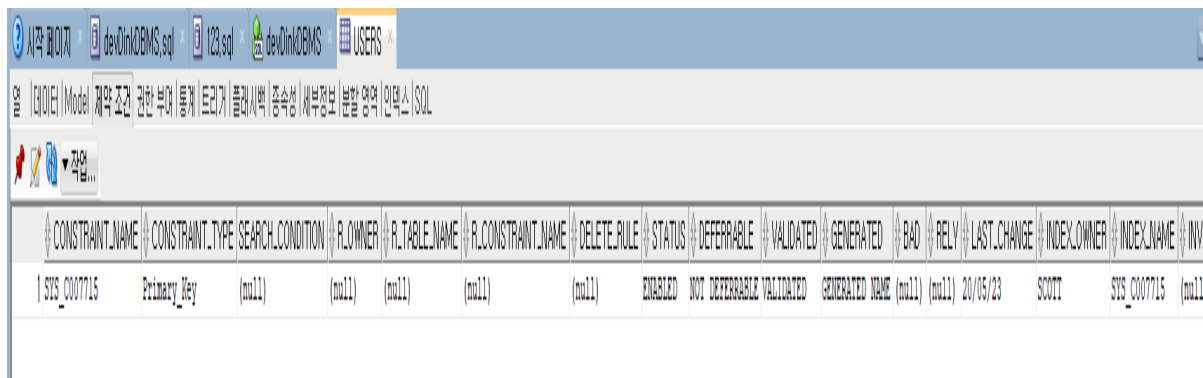


CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE	VALIDATED	GENERATED	BAD	RELY	LAST_CHANGE	INDEX_OWNER	INDEX_NAME
ACCOUNT_CK_BAL	Check	BALANCE >= 0	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED	USER NAME	(null)	(null)	20/05/25	(null)	(null)
SYS_C008009	Check	"BANK_CODE" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED	GENERATED NAME	(null)	(null)	20/05/26	(null)	(null)

1 잔고(BALANCE)가 >= 0 인 제약조건

2 서비스를 제공하는 은행만 시스템에서 사용이 가능하도록 하기 위해 계좌 등록시에 은행코드 입력하도록 안내 후 NOT NULL 제약조건 지정

USERS table 의 CONSTRAINT



CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE	VALIDATED	GENERATED	BAD	RELY	LAST_CHANGE	INDEX_OWNER	INDEX_NAME
SYS_C007715	Primary Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED	GENERATED NAME	(null)	(null)	20/05/23	SCOTT	SYS_C007715

1 ID 를 PRIMARY KEY 로 지정하도록 제약 조건 지정

DB와 테이블 구조

ACCOUNT table의 구조

ACCOUNT

데이터 | Model | 제약 조건 | 권한 부여 | 통계 | 트리거 | 플래시백 | 종속성 | 세부정보 | 분할 영역 | 인덱스 | SQL




작업...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	NAME	VARCHAR2 (10 BYTE)	Yes	(null)	1 (null)	
2	ID	VARCHAR2 (20 BYTE)	Yes	(null)	2 (null)	
3	ACCOUNT_NUMBER	VARCHAR2 (20 BYTE)	Yes	(null)	3 (null)	
4	BANK_CODE	VARCHAR2 (3 BYTE)	Yes	(null)	4 (null)	
5	BALANCE	NUMBER	Yes	(null)	5 (null)	
6	BANK_NAME	VARCHAR2 (30 BYTE)	Yes	(null)	6 (null)	
7	NICKNAME	VARCHAR2 (50 BYTE)	Yes	(null)	7 (null)	

USERS table 의 구조

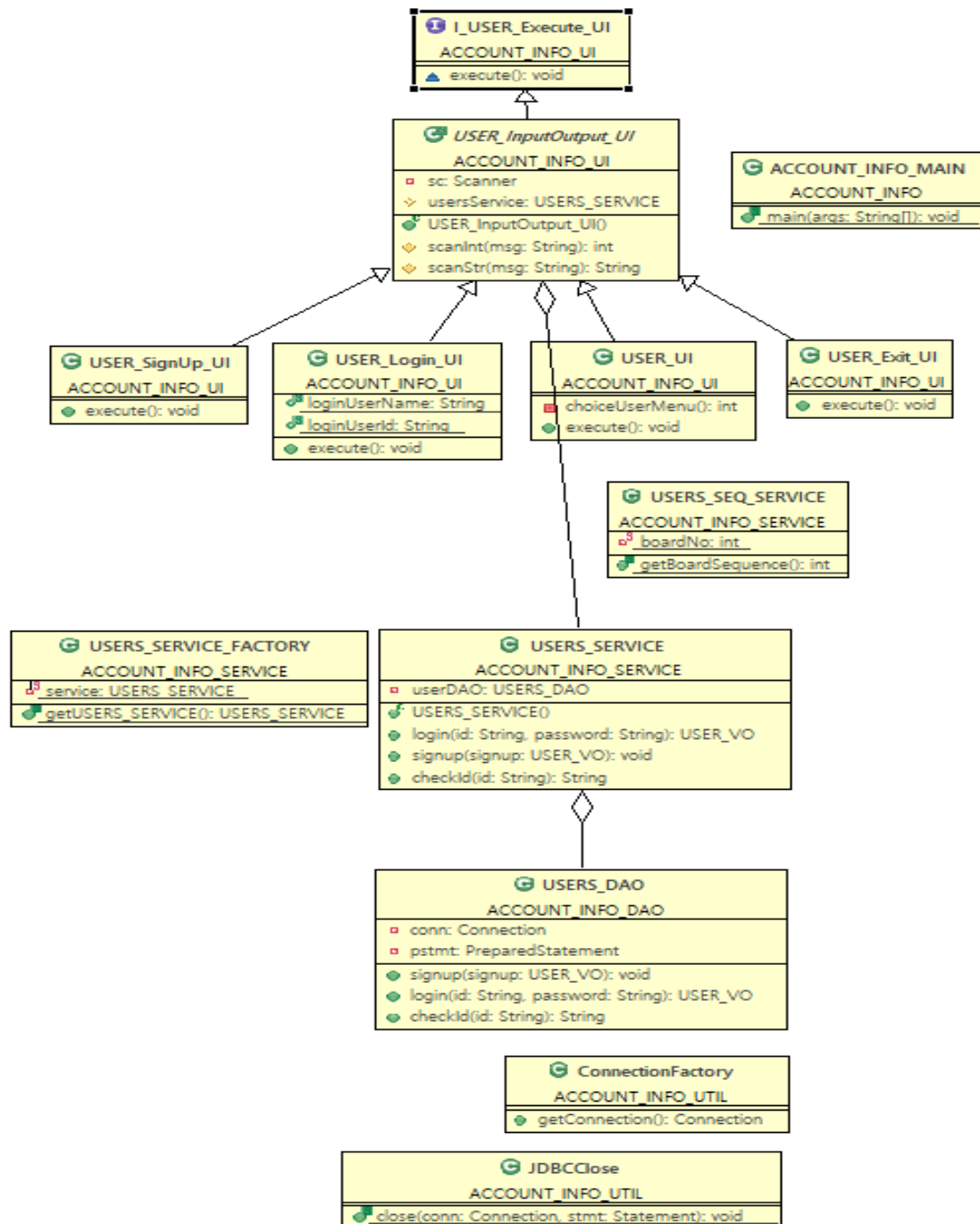
USERS

열 데이터 | Model | 제약 조건 | 권한 부여 | 통계 | 트리거 | 플래시백 | 종속성 | 세부정보 | 분할 영역 | 인덱스 | SQL

 ▼ 작업...

	↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS
1	NAME	VARCHAR2 (10 BYTE)	Yes	(null)	1	(null)
2	ID	VARCHAR2 (20 BYTE)	No	(null)	2	(null)
3	PWD	VARCHAR2 (20 BYTE)	Yes	(null)	3	(null)
4	SEQ	NUMBER	Yes	(null)	4	(null)
5	REG_DATE	DATE	Yes	(null)	5	(null)

Class Diagram

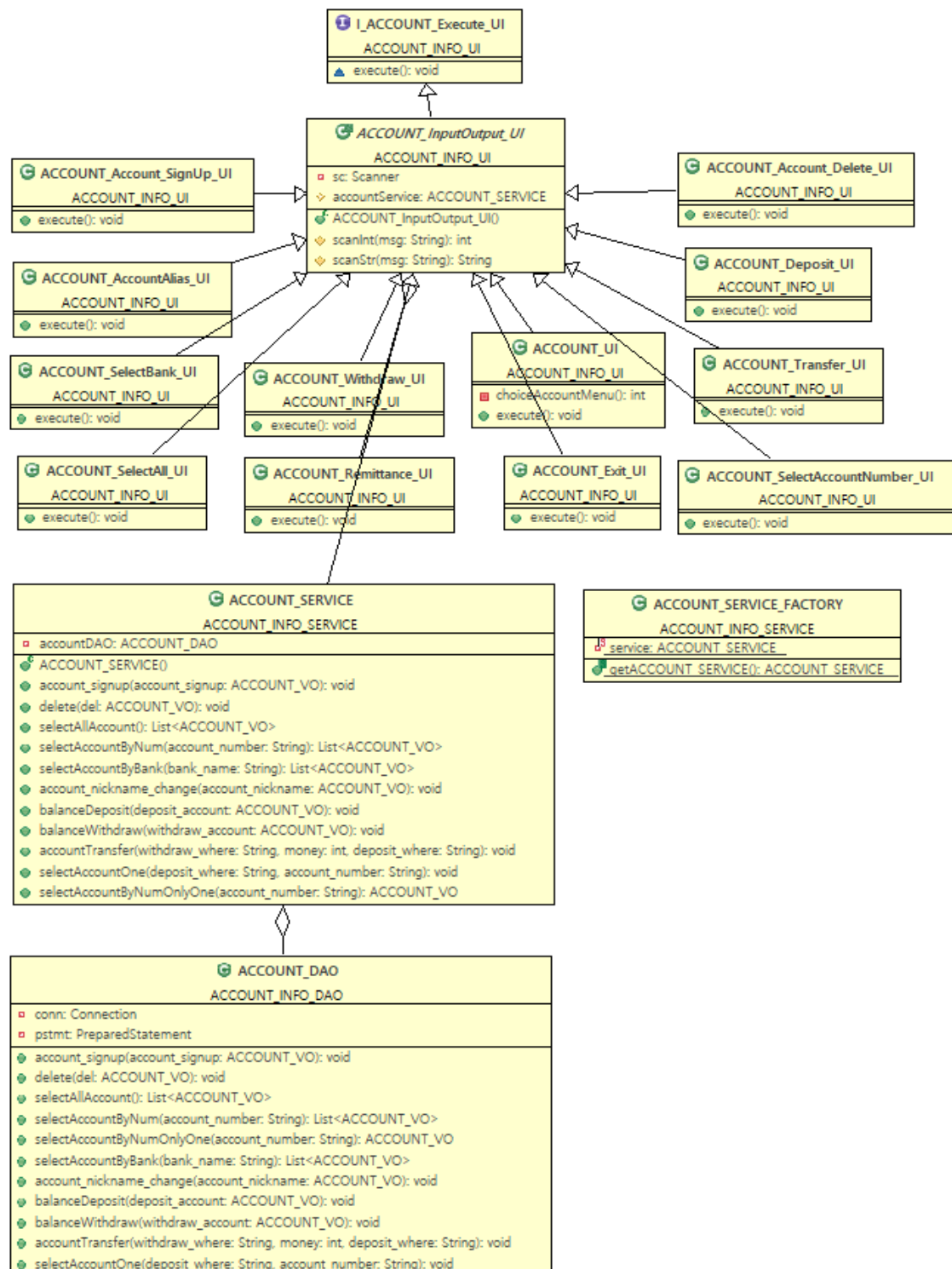


UI에서 입력 받은 변수들을 SERVICE로 전달하고 DAO로 전달하여 쿼리를 보내는 흐름입니다.

로그인한 사용자의 아이디, 이름을 static으로 지정하여 추후 ACCOUNT UI에서도 사용

InputOutput을 추상클래스로 하여 상속하고 EXECUTE를 인터페이스 화 하여 각각의 기능 UI에 다중 상속의 효과를 구상했습니다.

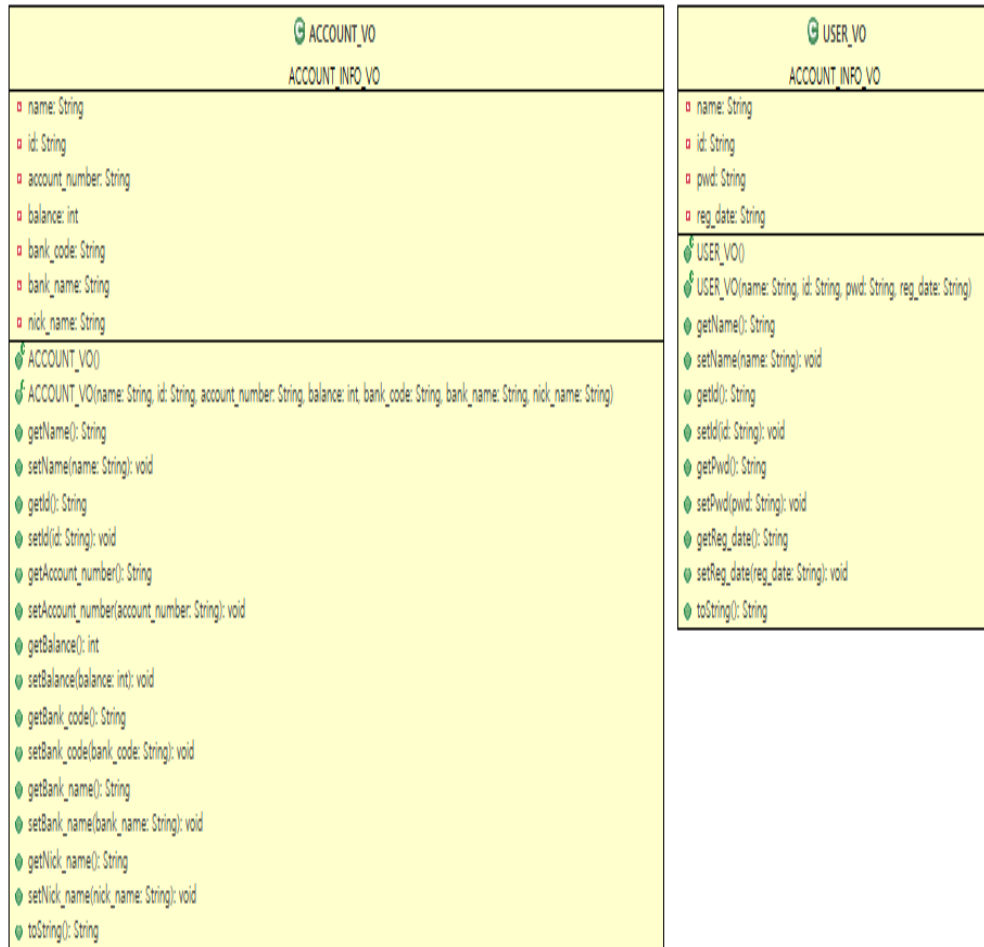
Class Diagram



UI에서 입력 받은 변수들을 SERVICE로 전달하고 DAO로 전달하여 쿼리를 보내는 흐름입니다.

InputOutput을 추상클래스로 하여 상속하고 EXECUTE를 인터페이스 화 하여 각각의 기능 UI에 다중 상속의 효과를 구상했습니다.

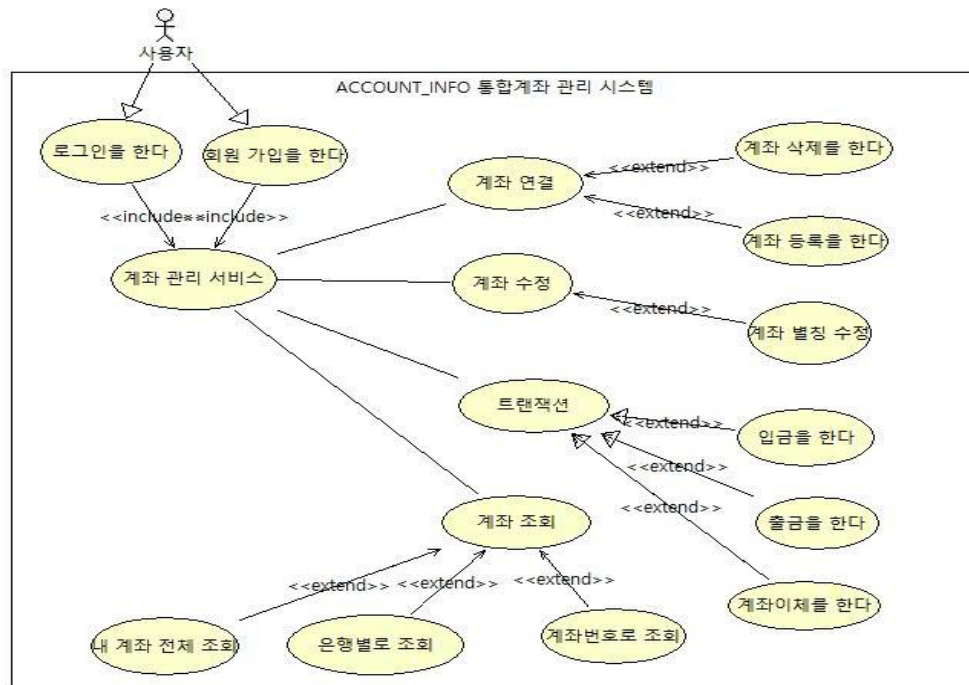
Class Diagram



VO table 은 멤버변수, 생성자(기본생성자와 모든 변수가 들어간 생성자), getter&setter 를 미리 지정해 두는 table 입니다.

Value Object 로 값을 저장해서 운반하는 역할을 합니다.

Use case



시스템과 사용자의 상호작용을 다이어그램으로 표현하고자 사용자의 관점에서 시스템의 서비스 혹은 기능을 나타내는 유스케이스로 나타냈습니다.