**‘종오뱅크(JongOhBank)’ 프로젝트 관련 문서**

**팀 명 :**

**7조(종오뱅크(JongOhBank))**

**제 출 인 : 김현영**

**권도연**

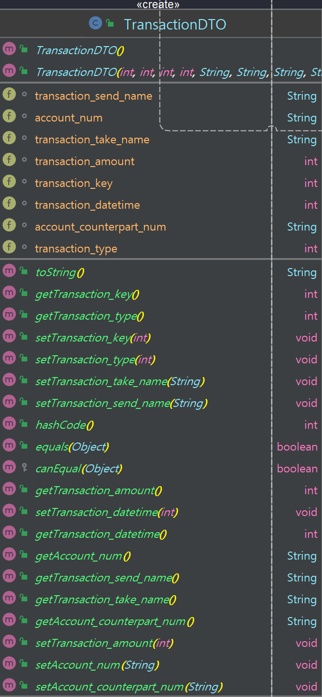
**정종오**

**전희주**

**07. Class Diagram**

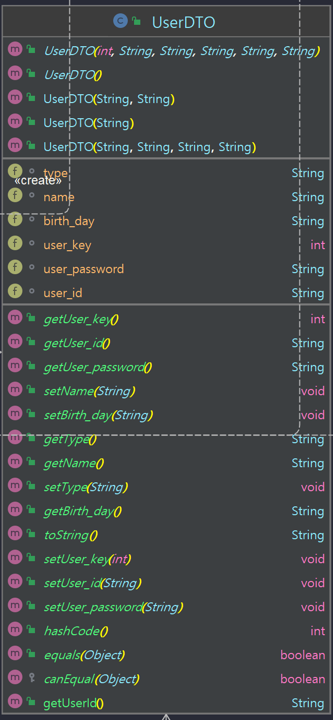
**상세 명세서**

## TransactionDTO



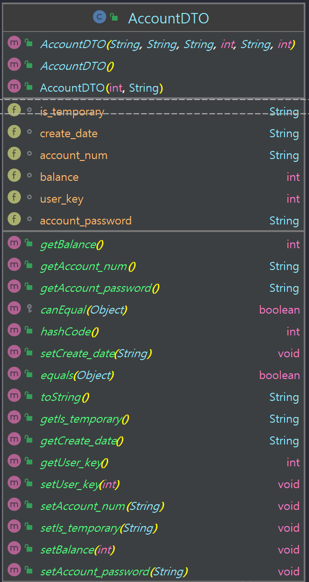
| **Method Summary** | |
| --- | --- |
| string | **toString()**  // 모든 멤버 변수의 값을 String으로 반환하는 toString() 메서드가 Lombok을 통해 자동 생성 |
| get/set | **getter/setter**  //한 Getter/Setter 메서드가 Lombok을 통해 자동 생성 |

## UserDTO



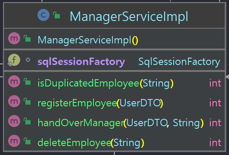
| **Method Summary** | |
| --- | --- |
| string | **toString()**  // 모든 멤버 변수의 값을 String으로 반환하는 toString() 메서드가 Lombok을 통해 자동 생성 |
| get/set | **getter/setter**  //한 Getter/Setter 메서드가 Lombok을 통해 자동 생성 |

## AccountDTO



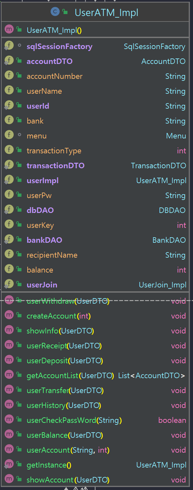
| **Method Summary** | |
| --- | --- |
| string | **toString()**  // 모든 멤버 변수의 값을 String으로 반환하는 toString() 메서드가 Lombok을 통해 자동 생성 |
| get/set | **getter/setter**  //한 Getter/Setter 메서드가 Lombok을 통해 자동 생성 |

## ManagerServiceImpl



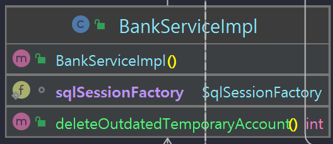
| **Method Summary** | |
| --- | --- |
| int | **isDupicatedEmployee(String)**  // 모든 멤버 변수의 값을 String으로 반환하는 toString() 메서드가 Lombok을 통해 자동 생성 |

## UserAtmImpl



| **Method Summary** | |
| --- | --- |
| void | **userWithdraw(UserDTO)**  // UserDTO를 매개 변수로 받아 고객이 출금 요청시 실행 |
| void | **createAccount(int)**  // 새로운 계좌를 데이터 베이스에 등록 |
| void | **showInfo(UserDTO)**  //UserDTO를 매개 변수로 받아 고객 정보 조회 |
| void | **userReceipt(UserDTO)**  // UserDTO를 매개 변수로 받아 고객 정보에 해당하는 영수증 출력 |

## BankServiceImpl



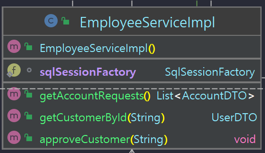
| **Method Summary** | |
| --- | --- |
| int | **deleteOutdatedTemporaryAccount()**  // 승인 기간 지난 계좌는 삭제 |

## UserJoinImpl



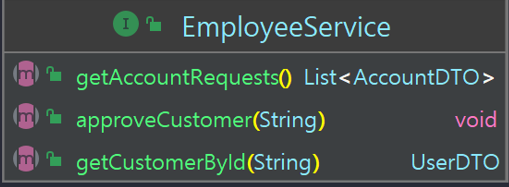
| **Method Summary** | |
| --- | --- |
| void | **userLogin(UserDTO)**  // 매개 변수를 UserDTO로 받아 사용자가 로그인할 때 실행 |
| void | **userSignup()**  // 사용자 회원 가입을 위해 호출되는 메서드 |

## EmployeeServiceImpl



| **Method Summary** | |
| --- | --- |
| List<AcountDTO> | **getAccountRequest()**  // 계좌 생성 요청 조회 |
| UserDTO | **getCustomerById(String)**  // 고객 정보 조회 |
| void | **approveCustomer(String)**  // 고객 계좌 승인, is\_temporary 변수를 1로 변경 |

## EmployeeService



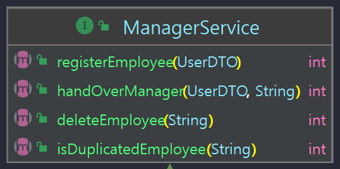
| **Method Summary** | |
| --- | --- |
| List<AcountDTO> | **getAccountRequest()**  // 계좌 생성 요청 조회 |
| UserDTO | **getCustomerById(String)**  // 고객 정보 조회 |
| void | **approveCustomer(String)**  // 고객 계좌 승인, is\_temporary 변수를 1로 변경 |

## BankService



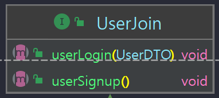
| **Method Summary** | |
| --- | --- |
| int | **deleteOutdatedTemporaryAccount()**  // 승인 기간 지난 계좌는 삭제 |

## ManagerService



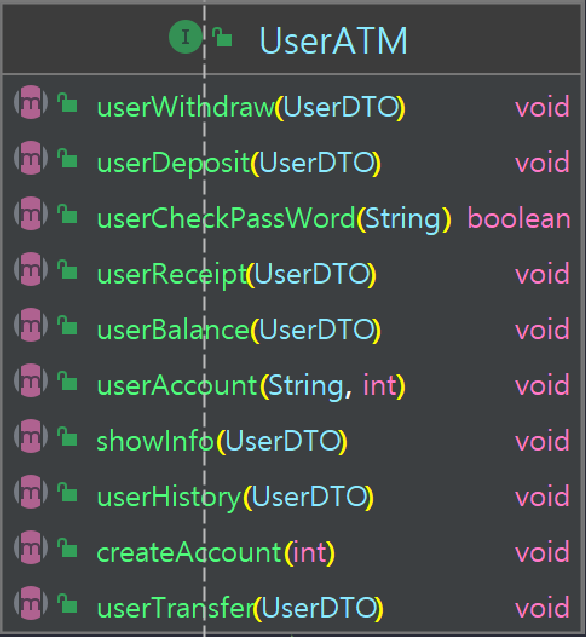
| **Method Summary** | |
| --- | --- |
| int | **isDupicatedEmployee(String)**  // 모든 멤버 변수의 값을 String으로 반환하는 toString() 메서드가 Lombok을 통해 자동 생성 |

## UserJoinService



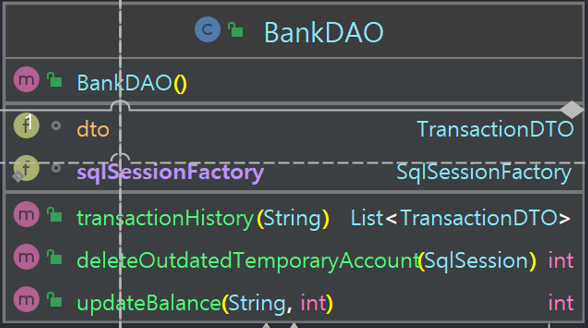
| **Method Summary** | |
| --- | --- |
| void | **userLogin(UserDTO)**  // 매개 변수를 UserDTO로 받아 사용자가 로그인할 때 실행 |
| void | **userSignup()**  // 사용자 회원 가입을 위해 호출되는 메서드 |

## UserAtmService



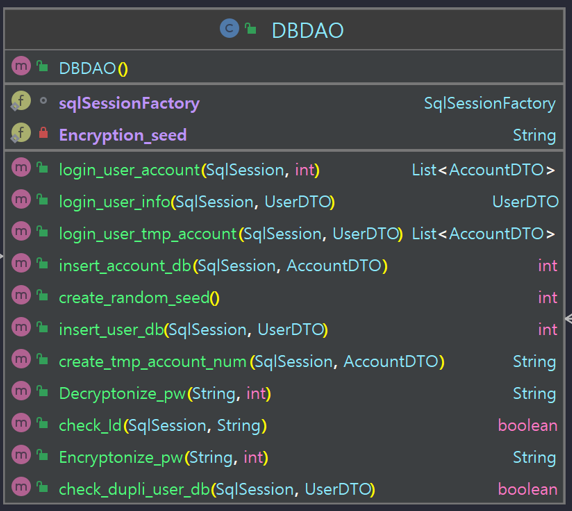
| **Method Summary** | |
| --- | --- |
| void | **userWithdraw(UserDTO)**  // UserDTO를 매개 변수로 받아 고객이 출금 요청시 실행 |
| void | **createAccount(int)**  // 새로운 계좌를 데이터 베이스에 등록 |
| void | **showInfo(UserDTO)**  //UserDTO를 매개 변수로 받아 고객 정보 조회 |
| void | **userReceipt(UserDTO)**  // UserDTO를 매개 변수로 받아 고객 정보에 해당하는 영수증 출력 |

## BankDAO



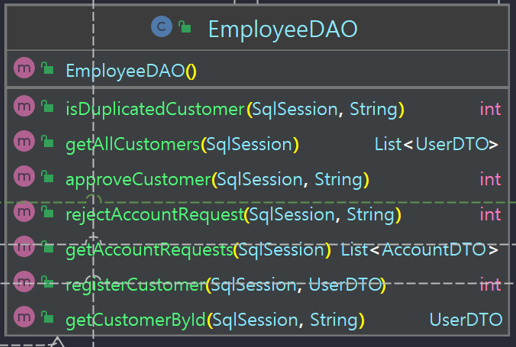
| **Method Summary** | |
| --- | --- |
| List<TransactionDTO> | **transactionHistory(String)**  // 파라미터로 받은 String형의 이름을 사용하여 MyBatis를 통해 SQL 쿼리를 실행하여 해당 이름으로 등록된 모든 거래 내역을 DB에서 조회하여 리스트로 반환 |
| int | **deleteOutdatedTemporaryAccount(SqlSession)**  // SqlSession 객체를 사용하여 MyBatis를 통해 SQL 쿼리를 실행하여 DB에서 임시 계좌 중 만료일이 지난 계좌를 삭제 |
| int | **updateBalance(String account, int balance)**  //파라미터로 받은 계좌 번호와 잔액을 사용하여 MyBatis를 통해 SQL 쿼리를 실행하여 해당 계좌의 잔액을 DB에서 업데이트 |

## DBDAO



| **Method Summary** | |
| --- | --- |
| boolean | **check\_Id(SqlSession session, String id)**  // 파라미터로 받은 아이디와 동일한 아이디가 이미 존재하는지 확인하여 boolean 타입으로 반환 |
| boolean | **check\_dupli\_user\_db(SqlSession session, UserDTO dto)**  // 파라미터로 받은 UserDTO의 로그인 정보가 이미 존재하는지 확인하여 boolean 타입으로 반환 |
| int | **insert\_user\_db(SqlSession session, UserDTO dto)**  //파라미터로 받은 UserDTO 객체의 회원가입 정보를 추가하고, 결과값으로 추가된 row의 수를 반환 |
| UserDTO | **login\_user\_info(SqlSession session, UserDTO dto)**  //파라미터로 받은 UserDTO 객체의 회원가입 정보를 추가하고, 결과값으로 추가된 row의 수를 반환 |
| List<AccountDTO> | **login\_user\_account(SqlSession session, int user\_key)**  //파라미터로 받은 user\_key를 가지는 사용자의 계좌 정보를 조회하여 List<AccountDTO> 형태로 반환 |
| List<AccountDTO> | **login\_user\_tmp\_account(SqlSession session, UserDTO dto)**  //파라미터로 받은 UserDTO 객체의 임시 계좌 정보를 조회하여 List<AccountDTO> 형태로 반환 |

## EmployeeDAO



| **Method Summary** | |
| --- | --- |
| int | **isDuplicatedCustomer(SqlSession, String)**  // 중복 여부를 확인하는 메서드 |
| List<AccountDTO> | **getAccountRequests(SqlSession)**  // 파라미터로 받은 UserDTO의 로그인 정보가 이미 존재하는지 확인하여 boolean 타입으로 반환 |
| int | **approveCustomer(SqlSession, String)**  //고객을 승인하는 메서드 |
| List<AccountDTO> | **getCustomerById**  //고객 정보를 조회하는 메서드 |

## ManagerDAO



| **Method Summary** | |
| --- | --- |
| int | **deleteEmployee(SqlSession, String)**  // 파라미터로 받은 직원 정보를 DB에서 찾아 삭제 수행 |
| int | **registerEmployee(SqlSession session, UserDTO dto)**  // 파라미터로 받은 UserDTO 객체에 직원 정보 추가 |
| int | **makeManager(SqlSession, String)**  //파라미터로 받은 관리자 정보를 DB에 저장 |
| int | **isDuplicatedEmployee(SqlSession, String)**  // 중복 여부를 확인하는 메서드 |
| int | **verifyEmployee(SqlSession, String)**  // 파라미터로 받은 직원 정보를 DB에서 찾아 검색 |
| int | **giveupManager(SqlSession, String)**  // 파라미터로 받은 관리자 정보를 DB에서 찾아 권한 인계 수행 |