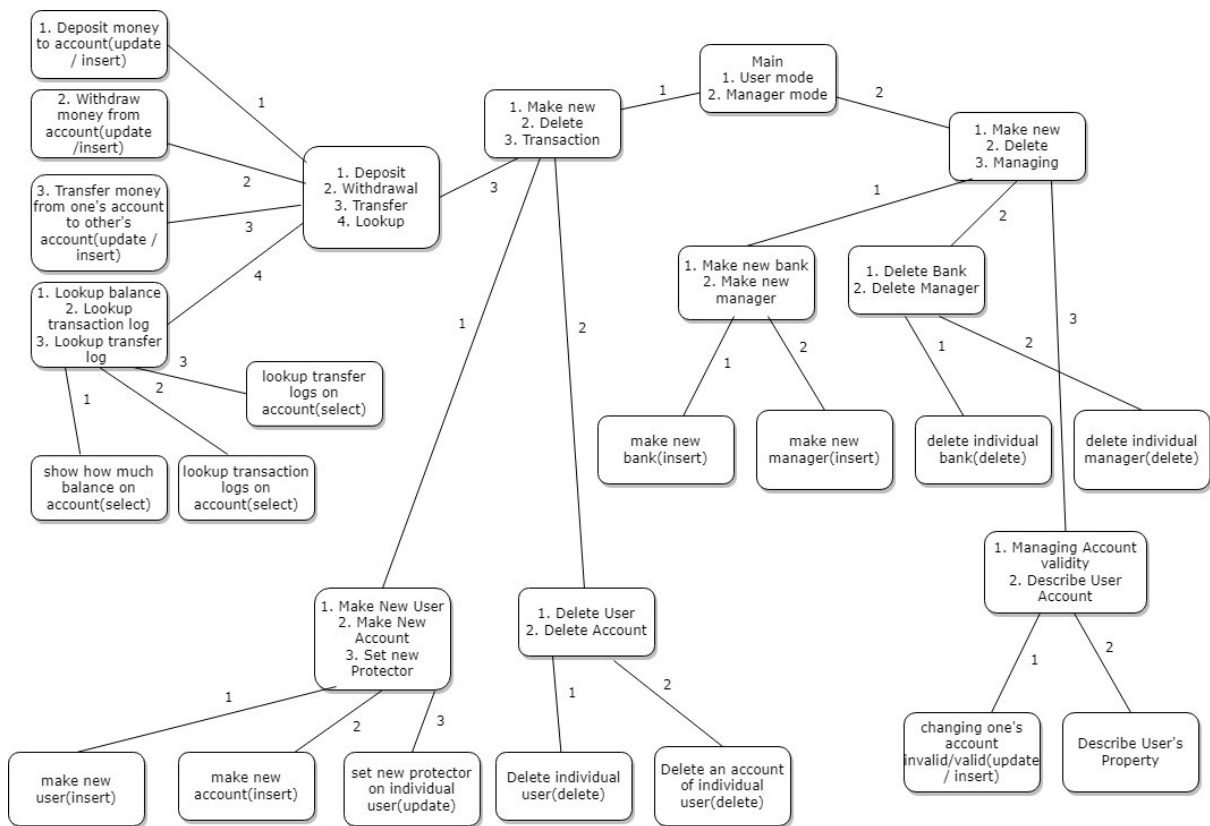


## 1. 개요

은행 DB와 이를 관리하는 Python 프로그램이다.

관리자용 관리 페이지와 사용자용 조회 페이지가 하나로 구현되어 있다.

## 2. 메뉴 구성



User mode와 Manager mode로 나누어지는 메뉴 구성이다.

User mode에서는 신규 사용자를 생성하거나, 신규 계좌를 개설하고, 사용자에게 보호자를 지정하는 Make New 항목, 사용자나 계좌를 삭제하는 Delete 항목, 그리고 거래에 관한 Transaction 항목이 존재한다.

Transaction항목은 또한 입금, 출금, 송금, 조회로 나뉘며 조회는 잔고 조회, 입출금 내역 조회, 이체 내역 조회로 나뉘어진다.

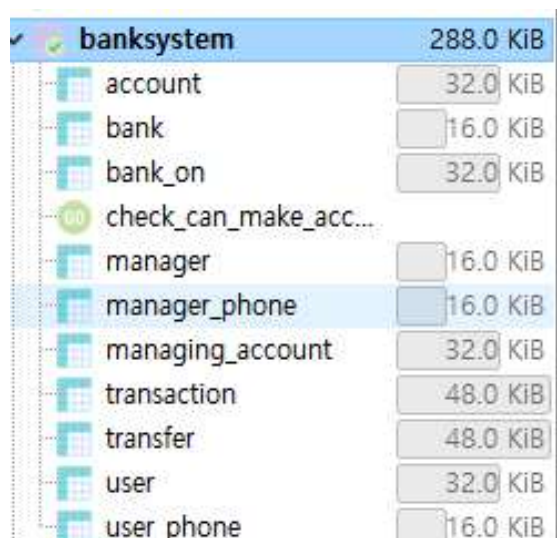
Manager mode는 Manager의 ssn을 입력한 후에 접속 가능해지며 Make new, Delete, Managing 항목으로 이루어진다. 초기에 아무런 Manager도 없는 상태로 접근하면 root 관리자 계정이 생성

된다.

Make new와 Delete에서는 각각 은행과 매니저를 새롭게 생성하거나 삭제할 수 있다. 은행의 삭제의 경우, 거래 내역이 존재할 경우 삭제가 불가능해진다.

Managing 항목에서는 User의 Account의 validity를 설정하여 계좌를 정지하거나 복구할 수 있고 User들의 정보와 계좌, 거래 내역 등에 대한 통계 자료를 열람할 수 있다.

### 3. DB Table 구조



banksystem	288.0 KiB
account	32.0 KiB
bank	16.0 KiB
bank_on	32.0 KiB
check_can_make_acc...	
manager	16.0 KiB
manager_phone	16.0 KiB
managing_account	32.0 KiB
transaction	48.0 KiB
transfer	48.0 KiB
user	32.0 KiB
user phone	16.0 KiB

account : 계좌에 대한 정보를 저장하는 Table

bank : 은행(지점)에 대한 정보를 기록하는 Table

bank\_on : 은행과 매니저에 대한 관계를 기록하는 Table

check\_can\_make\_account : 계좌를 개설할 수 있는지에 대한 여부를 보여주는 function

manager : 매니저에 대한 정보를 기록하는 Table

manager\_phone : 매니저의 연락처 정보를 기록하는 Table

transaction : 입출금 내역에 대한 Log가 기록되는 Table

transfer : 이체 내역에 대한 Log가 기록되는 Table

user : 사용자에 대한 정보가 기록되는 Table

user\_phone : 사용자의 연락처 정보를 기록하는 Table

### 4. 요구사항 만족 정도

## 은행 서비스 DBMS 요구사항 분석

요구사항_id	Entity	분류	요구사항 명	요구사항 내용	비고	구현 여부
M_1_1	Manager	계좌 관리	계좌 입출금 내역 조회	관리자는 각 사용자의 개인 정보와 계좌 번호를 대조하여 계좌 입출금 내역을 조회할 수 있다.		△
M_1_2			계좌 잔고 조회	관리자는 각 사용자의 개인 정보와 계좌 번호를 대조하여 계좌의 잔고를 조회할 수 있다.		○
M_1_3			계좌 사용자 조회	관리자는 계좌 번호를 통해 계좌의 사용자를 조회할 수 있다.		○
M_1_4			계좌 개설일 조회	관리자는 계좌 번호를 통해 계좌의 개설일을 조회할 수 있다.		○
M_1_5			계좌 조회 불가능 속성	관리자는 각 계좌의 비밀번호를 볼 수 없다.		○
M_1_6			계좌 정지	관리자는 부정한 사용 내역이 있는 계좌의 입출금을 발견하면 계좌를 정지시킬 수 있다.	v1.1	○
M_1_7			계좌 정지 해제	관리자는 계좌 정지를 해제할 수 있다.	v1.1	○
M_1_8			계좌 삭제	관리자는 정지된 계좌가 위법되게 사용되었다고 판단하면 해당 계좌를 삭제할 수 있다.	v1.1	X
M_1_9			관리 가능한 계좌 수	관리자는 1개 이상의 계좌를 관리할 수 있다. (관리하지 않는 계좌가 있을 수 있다.)	v1.1	○
M_2_1		은행 관리	은행 소속	관리자는 반드시 하나의 은행에 소속되어 있다. 둘 이상의 은행에 소속된 관리자나 은행에 소속되지 않은 관리자는 없다.		○
M_3_1	관리자 속성		관리자 식별	관리자는 유일한 관리자id로 식별한다. 또한 유일한 관리자 주민번호를 가지고 있다.	v1.3	○
M_3_2			관리자 필수 속성	반드시 이름과 출생연월이 작성되어야 하며, 최소 1개 이상의 연락처가 있어야 한다.		○
			관리자 조건	관리자는 반드시 19세 이상이어야 한다.	v1.3	○
U_1_1	User	계좌 보유 조건	계좌 최소 보유 조건	계좌가 없는 사용자도 존재할 수 있다.		○
U_1_2			계좌 최대 보유 조건	사용자는 최대 10개까지의 계좌를 보유할 수 있다.		X
U_2_1		계좌 개설	계좌 개설 조건	사용자의 연령이 19세 이상인 경우에는 조건 없이 계좌를 개설할 수 있다.		○
U_2_2			19세 미만 조건	단, 19세 미만인 경우 최소 1명 이상의 19세 이상의 보호자 등록이 필요하다.	v1.3	○
U_2_3			연속 계좌 생성 불가능	추가로 하나의 계좌가 개설되지 20일이 지나지 않았으면 새로운 계좌를 개설할 수 없다.		X
U_3_1		계좌 입금	계좌 입금	사용자는 보유한 계좌가 정지되지 않았다면 돈을 입금할 수 있다.	v1.1	○
U_4_1		계좌 출금	계좌 출금	사용자는 자신이 개설한 계좌가 정지되지 않았다면 최대 잔고만큼의 돈을 출금할 수 있다.	v1.1	○
U_4_2			계좌 출금 취소 조건	출금을 요청한 계좌 잔고가 부족하면 출금이 취소될 수 있다.	v1.3	X
U_5_1		계좌 삭제	계좌 삭제	사용자는 계좌를 삭제할 수 있다. 삭제할 시 계좌에 남아있던 잔고는 사용자에게 출금된다.		X
U_5_2			계좌 삭제 취소 조건	만약 계좌 출금이 불가능하여 취소된다면 계좌 삭제가 취소된다.		X
U_6_1		공동 명의	공동 명의 계좌 생성	사용자는 다른 사용자와 공동명의를 만들어 계좌를 생성할 수 있다. (공동명의를 존재하지 않을 수 있다.)		X
U_6_2			공동 명의 계좌 최대 개수	사용자는 최대 5개의 공동 명의 계좌를 생성할 수 있다. 생성된 공동 명의 계좌는 사용자의 개인 계좌 최대 개수와는 별개로 계수된다.		X
U_6_3			공동 명의 계좌 생성 제한 조건	사용자의 나이가 19세 미만이면 공동 명의 계좌를 생성할 수 없다.		X
U_7_1		사용자 속성	사용자 식별	사용자는 유일한 사용자id로 식별할 수 있다.		○
U_7_2			사용자 필수 속성	사용자는 반드시 이름, 출생연월, 지역, 신용등급 정보가 있어야 하며, 최소 1개 이상의 연락처가 있어야 한다.		○
A_1_1	Account	계좌 생성 조건	사용자 계좌 생성	계좌는 개인 사용자의 명의로 개설될 수 있다. 단 공동 명의 계좌일 경우 대표자의 명의로 개설된다.	v1.1	△
A_1_2			사용자 계좌 관리	계좌는 반드시 한 명 이상의 관리자의 관리를 받아야 한다	v1.1	○
A_1_2			공동 명의 계좌 생성	계좌는 공동 명의로 개설될 수 있다.	제거	X
A_1_3			계좌 생성 조건	계좌는 반드시 하나의 사용자 또는 공동 명의에 속해 있다.	제거	X
A_2_1		계좌 정지	계좌 정지	계좌는 정지될 수 있다.	v1.1	○
A_2_2			계좌 정지 기간	정지된 계좌는 사용자가 정지 해제되기 전 또는 계좌 삭제 시까지 유지된다.	v1.1	○
A_2_3			계좌 정지 해제	관리자는 계좌 정지를 해제할 수 있다.	v1.1	○
A_3		입출금 내역 관리	계좌 입출금 내역 저장	계좌에 입출금이 이루어질 때 해당 내역을 Log로 저장한다.		○
A_3_1			로그 관리	해당 Log는 보유 사용자 또는 공동 명의의 대표자와 관리자만 관리할 수 있다.		△
A_3_2			로그 최대 저장 길이	로그는 최대 100개까지 저장할 수 있다. 이보다 오래된 로그는 별도로 저장한다.		X
A_4_1		계좌 속성	계좌 식별	계좌는 유일한 계좌번호로 식별한다.	v1.3	○
A_4_2			비밀번호 설정	계좌에는 반드시 비밀번호를 설정해야 한다.	보호조치 필요	○
A_4_3			계좌 필수 속성	계좌 종류와 계좌 개설일이 필요하다.		○

A_5_1		계좌 이체	계좌 이체	계좌 사이에서 돈을 주고받을 수 있다.	v1.2	○
A_5_2			계좌 이체 발생	계좌 이체는 송신자의 요청에 의해 발생한다.	v1.2	○
A_5_3			계좌 이체 성공/실패 1	계좌 이체는 송신자의 잔액이 요청 금액보다 크면 성공하고 적으면 실패한다.	v1.2	○
A_5_4			계좌 이체 성공/실패 2	계좌 이체는 송신자와 수신자의 계좌가 모두 활성화 상태일 때만 성공하고 아니라면 실패한다	v1.2	○
B_1_1	Bank	은행 관리	은행 관리	은행은 최소 1명 이상의 관리자가 필요하다. (관리자가 없는 은행은 없다.)		X
B_2_1		잔고 관리	잔고 관리	사용자의 출금 요청이 있을 경우 보유 잔고량에서 출금을 한다.	v1.3-제거	X
B_3_1		은행 속성	은행 식별	은행은 고유한 지점코드로 식별된다. 은행은 소속된 은행 코드가 부여되어 있다.	v1.3	○
B_3_2			은행 필수 속성	은행 지정명과 위치가 필요하다.	v1.3	○
J_1_1	Joint	공동 명의 성립 조건	공동명의 성립 인원 조건	공동 명의는 최소 2명 이상의 사람이 있어야 성립한다.	v1.3 구현 실패	X
J_1_2			공동명의 성립 대표자 조건	공동 명의는 19세 이상의 한 명의 대표자가 있어야 성립한다. (다른 구성원은 반드시 19세 이상일 필요는 없다.)	v1.4 구현 실패	X
J_1_3			공동명의 구별	구성원이 동일하더라도 대표자가 다른 경우 또는 이름이 다른 경우는 다른 공동 명의로 본다.	v1.5 구현 실패	X
J_2_1		공동 명의 계좌 성립	공동명의 계좌 최소 보유 조건	계좌가 없는 공동명의는 존재할 수 없다.	v1.6 구현 실패	X
J_2_2			계좌 최대 보유 조건	하나의 공동 명의로 하나의 계좌만 개설할 수 있다.	v1.7 구현 실패	X
J_3_1		공동 명의 계좌 입금	계좌 입금	공동 명의의 대표자는 보유한 공동 명의 계좌에 돈을 입금할 수 있다.	v1.8 구현 실패	X
J_4_1		공동 명의 계좌 출금	계좌 출금	공동 명의의 대표자는 보유한 공동 명의 계좌에서 돈을 출금을 할 수 있다.	v1.9 구현 실패	X
J_4_2			계좌 출금 취소 조건	단, 출금을 요청한 은행의 잔고가 부족하면 출금이 취소될 수 있다.	v1.10 구현 실패	X
J_5_1		공동 명의 계좌 삭제	계좌 삭제	공동명의 대표자는 계좌를 삭제할 수 있다. 삭제할 시 계좌에 남아있던 잔고는 대표자에게 출금된다.	v1.11 구현 실패	X
J_5_2			계좌 삭제 취소 조건	만약 계좌 출금이 불가능하여 취소된다면 계좌 삭제가 취소된다.	v1.12 구현 실패	X
J_6_1		공동 명의 계좌 소유	공동 명의 계좌 소유자	공동 명의 계좌는 공동 명의 대표자의 소유이며 대표자의 보유한 계좌와 동일하게 이용할 수 있다.	v1.13 구현 실패	X
J_7_1		공동 명의 속성	공동 명의 식별	공동 명의는 공동명의 대표자의 id로 구별한다. 같은 공동 명의 대표자를 가진 공동명의의 경우 공동명의 이름으로 구별할 수 있다.	v1.14 구현 실패	X
J_7_2			공동명의 필수 속성	공동명의 이름, 1명 이상의 소속 구성원 사용자 명단이 필요하다.	v1.15 구현 실패	X
구현 완료율						54.03225806

## 5. 주요 변경점

공동 명의 계좌를 구현하지 못하였다. (낮은 구현 완료율에 가장 큰 영향을 미침)

은행 지점의 잔고 내역을 제거하였다.

관리자의 사용자 계좌 삭제에 대한 부분을 미처 구현하지 못하였다.

그 밖에 자잘한 제약 조건을 만족하지 못하였다.

## 6. Specification of Testing

OS : Windows 10 Home (x64)

Language version : Python 3.7.0, Mysql 8.0.27