|  |
| --- |
| WinForm 은행 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **개발 인원** | 4명 | **본인 역할** | 팀장 | **개발 기간** | 1주 |
| **개발 언어** | C# | | | | |
| **사용 툴** | Visual Studio 2022, Oracle DB | | | | |

1. **개발 목적**

WinForm과 서버, DB를 연결하여 은행을 만들어 보았습니다.

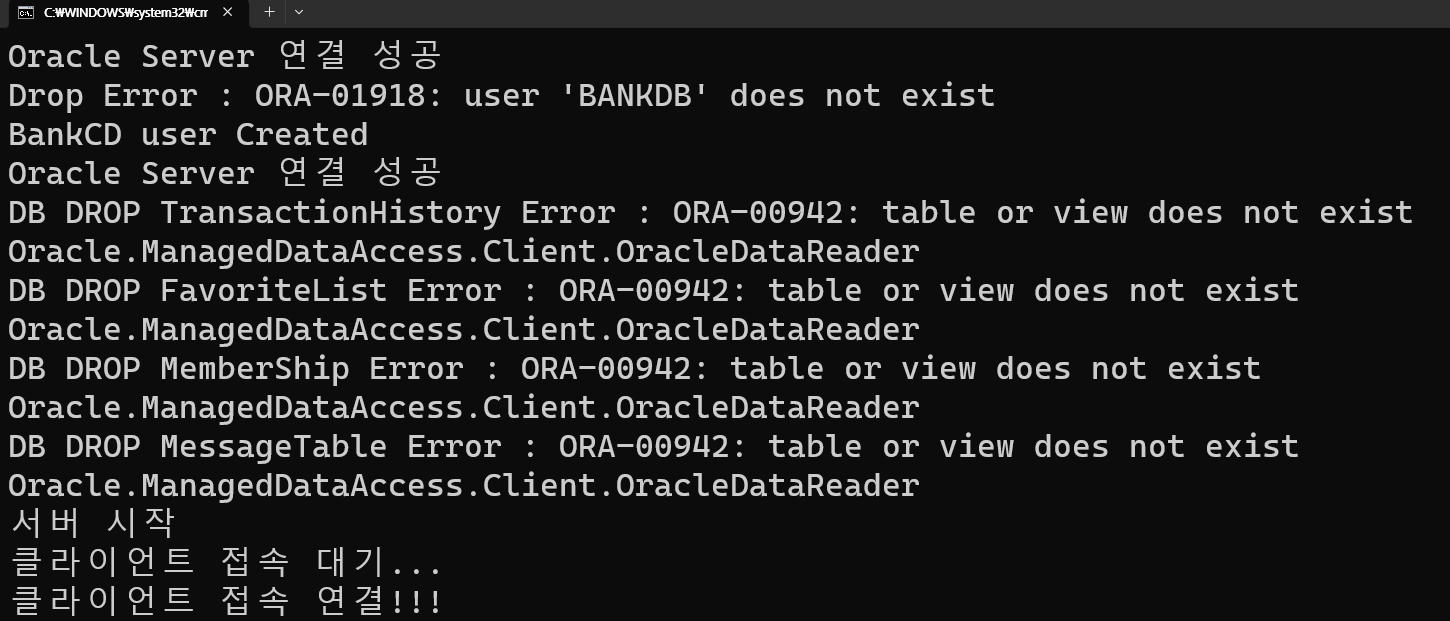
서버의 Packet은 DB 명령어와 명령어가 DB에 데이터를 넣는 Input인지 값을 받아와야하는 Output인지를 같이 보내 처리하도록 만들었습니다.

1. **아키텍처**



1. **서버**

서버는 C#으로 구현하였습니다. 클라이언트가 접속할때까지 대기 후 명령어 처리를 하도록 멀티스레드를 사용해 구현하였습니다.

서버 측에서 DB Table 및 User를 관리하도록 만들었습니다. 

1. **클라이언트**
   1. **로그인**

로그인 시 서버로 로그인 데이터를 보내고 서버에서 DB의 데이터를 확인하여 알맞은 정보가 입력되었다면 클라이언트로 성공/실패 여부를 보내도록 만들었습니다.

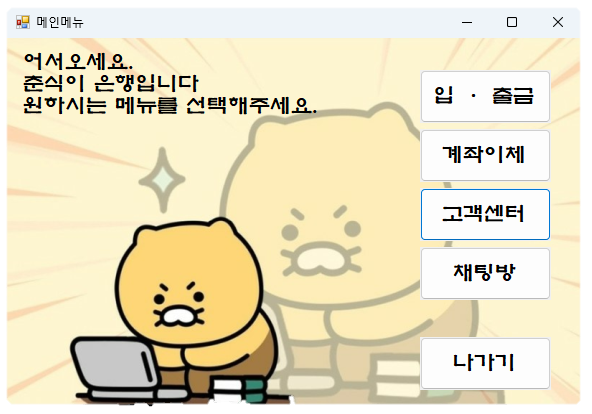
****

* 1. **회원가입**

클라이언트에서 서버로 정보를 보내 확인 후 같은 계좌가 없다면 생성할 수 있도록 만들었습니다.

****

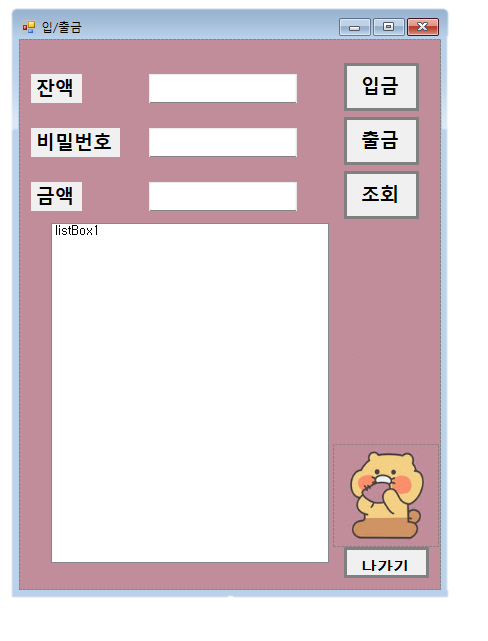
* 1. **메인화면**

****

* 1. **입/출금**

자신의 계좌에 돈을 입/출금 할 수 있도록 구현하였습니다.

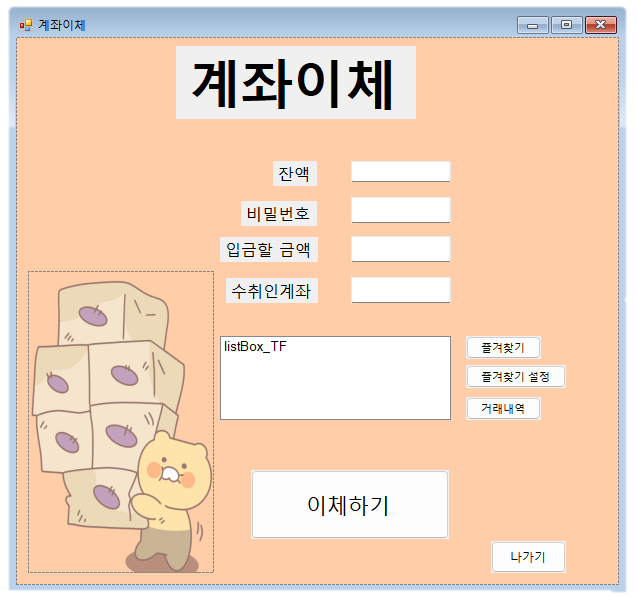
돈이 이동한 정보는 TransactionHistory 테이블에 저장됩니다.

****

* 1. **계좌이체**

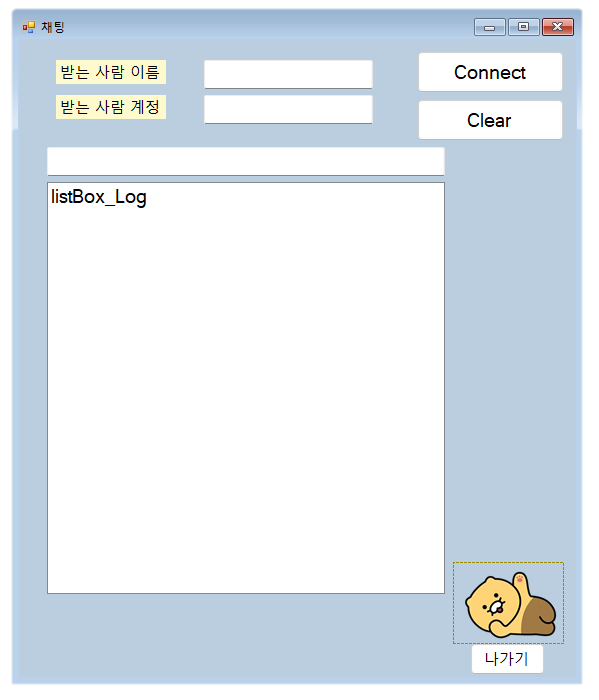
상대 계좌에 돈을 보낼 수 있습니다.

자주 보내는 계좌번호를 즐겨찾기 지정 할 수 있도록 구현했습니다.

****

* 1. **고객센터**

관리자 아이디를 사용하여 로그인 하게 된다면 다른 사용자들이 보낸 요구사항을 볼 수 있도록 구현하였습니다. 관리자 아이디가 아닌 일반 사용자는 관리자에게만 메시지를 보낼 수 있습니다.

****

* 1. **채팅방**

모든 사용자들이 들어와 채팅을 칠 수 있습니다. 서버에서는 받은 메세지를 접속한 사람들에게 BroadCast 해주고 클라이언트는 0.5초마다 데이터를 받아 새로운 데이터가 있다면 화면에 출력해줍니다.

****