팀 프로젝트 결과보고서

4조

15011037 김동익

17013141 봉수연

15011059 이세영

16013083 이주형

1. **제목**

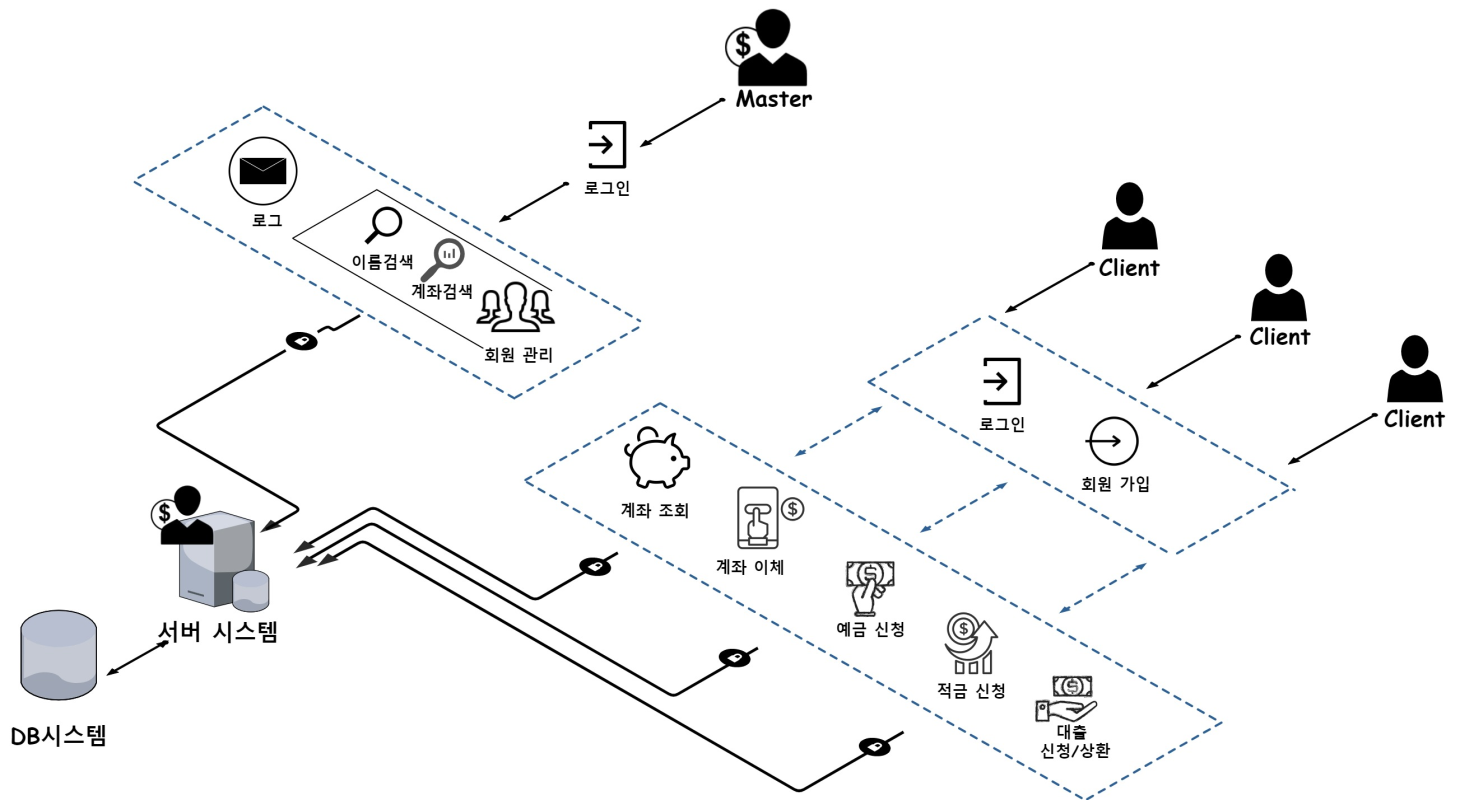
은행 전산 시스템

1. **프로그램 개요 및 설명**

데이터베이스를 이용하여 CIient들의 잔액정보 및 계좌 정보 등을 확인할 수 있고 서버를 통해 어떤 정보들이 오고 갔는지 볼 수 있고, 자신의 계좌를 조회하거나 다른 사람에게 이체 할 수 있고 예금,적금을 신청 및 해지 가능하며 마지막으로 대출을 신청하거나 대출금을 상환은행 전산 프로그래밍을 개발했습니다.

또한 관리자모드를 만들어 은행을 이용하는 고객들의 정보를 Search하여 확인 할 수 있고, 고객들의 정보들을 로그 창을 통해 확인해 볼 수 있습니다.

1. **시스템 구조도**



1. **Client 기능(일반 유저)**
2. Login 기능

서버와 통신을 통해 ID와 PW가 일치하는 데이터가 있으면 로그인에 성공하며 MENU 창으로 넘어간다. 실패 시 오류 메시지 띄운다.

1. 회원 가입 기능

6개의 Edit박스로 구성되어 있으며, 각 Edit 박스는 ID, 비밀번호, 비밀번호 확인, 이름, 계좌번호, 휴대폰 번호로 구성 되어 있다.

먼저, 중복확인 버튼을 통해 ID값을 서버통신을 통해 DB와의 연동을 통해 중복 여부를 검사 후, 중복 되는 ID값이 없는 경우 비밀번호 칸이 열리게 되며 비밀번호와 비밀번호 확인 칸에 입력된 값이 같고, 회원 가입 정보가 비워져 있으면, 회원가입에 성공하며 DB에 추가된다. 그 외의 실패 메시지 박스를 띄운다.

1. 계좌 조회 기능

MENU 창에서 로그인 한 회원이 가지고 있는 모든 계좌의 번호와 현재 잔액을 출력한다.

1. 계좌 이체 기능

계좌 이체 정보 창을 띄워 송금할 계좌 번호, 보낼 금액을 입력 후 수행 서버와 통신 후 계좌 잔액을 확인한다.

보낼 금액만큼 DB에서 보낸 계좌 잔액 와 수령할 계좌 잔액 수정되며, 부족 시 처리되지 않는다.

1. 예금 신청/해지 기능

예금 / 적금 신청 시, 계좌 개설에 관한 이용약관 동의서를 체크 해야 한다.

위의 4개의 체크박스를 모두 선택하면, 마지막 체크박스가 선택 가능해진다.

(1개의 체크박스라도 해제하면 마지막 체크박스는 선택 불가능 해진다)

예금 신청 할 시 서버 쪽에서 회원에게 임의의 계좌를 지정해 주어서 계좌번호와 계좌 정보를 회원에 맞는 DB에 추가 시켜준다.

해지 시 해지 처리 창을 띄워, 남은 금액 처음 회원가입 할 때 등록한 계좌로 전송되며, 그 후 서버 쪽 DB에서 선택 계좌를 삭제 시켜준다.

1. 적금 신청/상환 기능

적금 신청 시 적금 신청 창을 띄워, 적금 신청 금액, 이자율을 선택 가능하며 매달(매주) 등록한 계좌에서 이자 금액만큼 잔액이 증가되며 임의의 계좌를 지정해주어 DB에 추가 된다.

해지 시 예금 과 같이 회원가입 할 때 등록한 계좌로 잔액이 전송 되며 서버 쪽 DB에서 선택 계좌를 삭제된다

1. 대출 신청/ 상환

대출 신청의 경우, 대출 신청 금액을 입력 후 대출을 신청하면 회원가입시 등록한 계좌로 돈이 입금된다.

대출 상환의 경우, 상환할 금액은 입력 후 회원가입 할 때 등록한 계좌에서 잔액이 상환 금액만큼 감소 하지만, 서버를 통해 계좌의 잔액을

확인 후 계좌에서 입력한 상환금만큼 계좌에서 빠져나가며, 계좌의 잔액보다 상환 금액이 많을 시 실행 되지않는다.

1. **Server 기능(master)**
2. 이름/계좌번호 검색

이름 조회의 경우, DB에 접근하여 같은 이름 값이 있다면 해당하는 회원의 정보 값을 가져오고, 일치하는 회원 정보가 없다면 알림 창을 띄우고 값을 가져오지않습니다. 계좌번호로 조회의 경우도 이와 같습니다**.**

1. **Server-Client 구조**
2. 패킷(256byte 배열 사용)

|  |  |
| --- | --- |
| Header | 내용 |

헤더 – iType(데이터 타입), Length(데이터 길이 표현시 2byte씩 사용), 데이터 개수

내용 – id, password , 계좌 번호, 이름 , 휴대폰 번호, 금액, client가 요청한 데이터(계좌 번호, 계좌 잔액, 대출 금액)

1. 서버 – SocCom, SocServer 클래스 이용 OnAccept, OnReceive 오버 라이딩 사용
2. 클라이언트 – SocCom 클래스 이용 OnReceive 오버라이딩 사용
3. 클라이언트 데이터 전송 시 서버에서는 데이터 베이스에 접근하여서, 삽입,삭제,갱신 기능을 수행 후, 수행 시 1 실패 시 0으로 결과 전송
4. 클라이언트 데이터 요구 시 서버는 DB에 접근하여, 데이터를 패킷에 담아 클라이언트에 전송
5. iType

SOC\_LOGIN 1

SOC\_REGISTER 2

SOC\_IDCHECK 3

SOC\_ACCOUNT\_INFO 4

SOC\_ACCOUNT\_MOVE 5

SOC\_Deposit\_Application 6

SOC\_Deposit\_Delete 7

SOC\_Saving\_Application 8

SOC\_Saving\_Delete 9

SOC\_Loan 10

SOC\_Loan\_Delete 11 으로 구성됨

1. **DB 구조**

관리방법 Microsoft Access을 이용하여 ODBC를 통한 관리 총 4개의 table로 이루어짐

1. Member table(회원 정보 관리)

Primary key – id, Password, ClientName, PhoneNum, AccountNum, Money 튜플로 구성됨

1. Deposit Account(예금 계좌 정보 관리)

Primary key – id, Password, DepositDate, DepositAccount, Money 튜플로 구성됨

1. SavingAccount(적금 계좌 정보 관리)

Primary key – id , SavingDate, SavingAccount, Money 튜플로 구성됨

1. Loan(대출 계좌 정보 관리)

Primart key – id, Loadmoney 튜플로 구성됨

1. **역할 분담**

* 김동익: 제안서 작성, 프로그램 기능 구현 코딩(서버), 서버-클라이언트 연동, 최종 보고서 작성
* 봉수현: 시스템 구조도 작성, 서버 클라이언트 UI 제작, PPT작성
* 이세영: 역할 분담 조정, 프로그램 기능 구현 코딩(클라이언트), 발표
* 이주형:, UI 기능 정의서 작성, DB제작, DB-서버 연동

1. **개인 의견**

김동익 – 지금까지 수업 시간에 정해진 과제를 한 학기 동안 하였지만, 직접 설계 및 구현을 찾아가며, 8일이라는 짧은 시간 내에 한 부분이 놀라왔다. Mfc 라는 프로그램을 배우면서 프로그램의 여러 방향이 있다는 것을 확인 하여 뿌듯하였습니다.

이세영 – MFC에서 클라이언트 부분을 제작하면서 소켓 통신에 대한 개념을 다시 한 번 이해할 수 있었다. MFC를 배우면서 C++에서도 객체지향과 UI 설정이 가능하다는 것을 깨달았다.

봉수연 – 어렵게만 생각되었던 MFC를 8일동안 공부해보며 생각보다 많은 부분을 다뤄보게 되었습니다. 실제로 코드를 짜는 부분들도 많지만, 지원해주는 함수나 속성 기능들이 많아 신기하기도 했고 덕분에 C++언어에 대해 좀 더 공부를 해보고 싶어졌습니다. 실제로 프로젝트를 진행한 것은 5일밖에 되지않아 아쉬운 부분들도 있지만 짧은 기간동안 할 일을 잘 나누어 다들 열심히 한 덕분에 좋은 성과를 얻게 된 것 같습니다. 덕분에 mfc뿐 아니라 DB연동, 소켓이 어떻게 되는지 알게 된 부분들도 많았습니다. 현재 만든 프로젝트에 대해 만족하고 뿌듯합니다.

이주형 – 처음 접해보는 언어와 툴을 통하여 단기간에 프로젝트를 진행해본 것이 큰 경험이 되었다고 생각합니다. 이것저것 알아보고 공부하며 팀원들과 목표했던 부분까지 완성이 되어서 만족스럽고 부족했던 부분들을 많이 알게되어 보람찬 경험이 되었습니다.

1. **발표 질문 정리**
2. **은행 금리는 어떤식으로 적용되는가?**

답변 : 예금 금리 부분을 실제 인터넷 및 은행 사이트에서 가져 왔고, 개월 선택 라디오 리스트를사용하여, 개월 수 선택 후 그 값에 따라 금리 계산 결과를 UI 안에서 바로 확인 가능

1. **이자는 통장에 어떤 방식으로 적용 되는가?**

답변 : 이자가 계산한 후 가능, 정해진 기간이 지난 후 데이터베이스에서 갱신 되는 것을 구현 하지 못하였음 차후 개선 할 수 있는 사항

1. **새로운 계좌 번호를 만들 시 계좌 번호를 동일하면 어떻게 되는가?**

답변 : 계좌 번호를 똑같은 것을 여러 개 만들 수 있게 설정 해 놓았음, 데이터베이스에서 id를 기본 키 값으로 가지고 있기 때문에, 문제 발생이 없을 것으로 생각

하지만, 계좌 이체 시 문제 발생 가능성이 있기 때문에, 서버에서 전역 변수로 계좌 번호를 설정 후, 새로운 값이 들어 올 때마다 그 전역 변수 값을 증가 시켜주는 방식으로 계좌 번호를 다르게 부여 가능

1. **발표시 계좌 번호는 10자리 이상이라고 하였는데, 시연 시 DB안에 3자리 계좌 번호 존재하는가?**

답변 : 이 3자리 계좌 번호는 테스트를 목적으로 하기 위한 것 시연 전 데이터베이스를세팅 하지 못함. 이후에는 회원 가입 UI 안에서 계좌 번호가 10자리 미만 입력 할 시 회원 가입이 될 수 없도록 오류 메시지를 출력함.

1. **회원 가입시 회원이 하나의 계좌만 추가 하지 않고 여러 개의 계좌를 추가 가능한가?**

답변: 회원당 주 계좌는 하나만 가능 하도록 설정함.

1. **은행 관리 업무에서 회원 개인 관리 정보가 암호화 세팅하였나? 항후 보완 할 계획이 있는가?**

답변 : 시연 시 데이터가 잘 들어가고 있는지를 보여주기 위해서 암호화 없이 사용하였음, 차후 개인 정보 보호에 대한 심각성을 인지 하고 있으며, 항후 보안 향상을 위해 개선 할 예정

1. **계좌 이체 시 데이터 처리는 어떤 방식으로 되는가?**

답변 : 계좌 이체 시 로그인한 주 계좌 내에서 계좌 이체 하고자 한 금액 만큼 제거 되어, 금액을 받을 계좌 번호에 그 금액 만큼 추가 해주는 퀴리문을 구현.

보유하고 있는 금액보다 보내고자 하는 금액이 많을 시 서버에서 오류 메시지를 전송해줌.

1. **마스터와 클라이언트에서 서버에 요청 하나? 아니면 DB에 직접 요청 하나?**

답변: 기능들이 요청이 대상이 되는 것이 서버로 전송 된다. 그 이후 서버에서 DB에 접근해 주는 방식을 이용한다. 이와 같은 방식 사용시 시스템 구조도가 잘못 됨. 수정 할 예정

1. **예금과 적금의 차이점을 아는가? 그것이 구현이 되어 있는가?**

답변 : 예금은 한 번에 특정한 금액을 입금하여, 그것을 가지고 이자를 계산 하지만, 적금은 일정한 간격을 가진 시간에 일정한 금액 만큼 입금을 하고, 이를 가지고 이자를 계산함, 이자를 계산하는 방식은 Client UI에서 구현 되어 있지만, DB에 적용 되는 부분은 구현이 안 되어 있음 차후 개선 예정