

# 빼빼로 공유 플랫폼: 상부상조

3학년 4반 3번 이름: 김석준

## ■ 요약 ■

지인들과의 관계를 확인하고 신뢰를 쌓는 기념일, 소위 ‘빼빼로데이’와 같은 행사의 규모는 날이 갈수록 커지는 중이다. 이에 따라 상대적 박탈감을 느끼는 사람의 수도 늘고 있으며, 문제 해결의 필요성이 점점 높아지고 있다. 본 프로그램은 빼빼로 과자를 공유할 수 있는 플랫폼으로, 메모를 붙인 빼빼로를 서로 주고받으며 외로움에서 벗어날 수 있다. 이는 미시적으로 사람들의 화합과 소통을 증진시킬 뿐 아니라, 공유 플랫폼의 활용 가능성을 넓히고 발전 방향을 제시하는 프로그램으로 기능할 것이다.

주제어: 공유플랫폼, 빼빼로

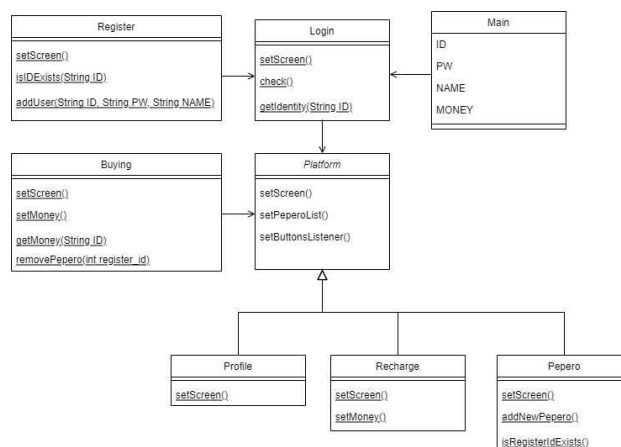
### I. 프로그램 개발의 동기 및 목적

최근 지인들과의 관계를 확인하고 돈독히 다지는 기념일들이 중시되고 있다. 특히, 기존의 ‘화이트데이’ 및 ‘밸런타인데이’ 외에 소위 ‘빼빼로데이’라는 기념일도 존재한다. 이는 길쭉한 빼빼로의 모양을 형상화한 것으로, 날짜도 11월 11일로 이를 상징하고 있다. 그러나, 이러한 빼빼로데이는 적지 않은 비판도 함께 받고 있다. ‘상대적 박탈감’에 대한 것이다. 물론 빼빼로데이는 선물을 주고받은 사람들간의 정을 쌓을 수 있는 의미있는 시간이 될 수 있으나, 선물을 받지 못하는 사람들도 생기기 마련이다. 또한, 각자가 받은 선물의 양과 내용을 서로 비교하게 되면서 상대적 열등감도 느낄 수 있게 된다. 따라서, 본 개발자는 이러한 사태의 해결의 필요성을 인지하였으며, 해결책으로 본 프로그램을 제작하였다. 이 프로그램을 통해 박탈감을 느끼는 사람의 수를 줄이고, 빼빼로데이가 모두가 행복한 기념일로 거듭날 수 있도록 하고자 한다.

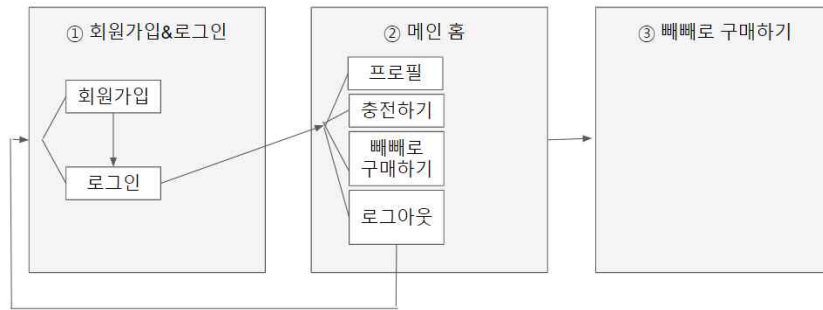
### II. 프로그램 소개

본 프로그램은 빼빼로를 공유하는 플랫폼 형태로 이루어져 있다. 프로그램을 실행하면 로그인창이 나타나며, 회원가입 또는 로그인이 가능하다. 회원가입을 진행하면 데이터베이스상에 회원 정보가 저장된다. 회원으로 등록되면 로그인이 가능하다. 로그인 이후에는 플랫폼 내로 들어오게 된다. 메뉴바 내에서는 자신의 프로필을 확인할 수 있고, 돈을 충전할 수 있으며, 빼빼로를 등록할 수 있다. 특히, 빼빼로를 등록하면 플랫폼에 자신이 등록한 빼빼로가 자신의 아이디와 함께 나타나며, 누군가가 이를 구매하면 돈의 이동이 이루어지며 플랫폼에서 빼빼로가 삭제된다.

### III. 프로그램 구성도(UML 다이어그램)



#### IV. 사용자 메뉴 구성도



#### IV. 주요 실행 결과

##### 가. 회원가입&로그인

프로그램을 실행하면 로그인 창이 나타난다. 회원으로 등록되어 있지 않으면 새로 회원가입이 가능하며, 이름, 아이디, 비밀번호를 각각 입력하고 회원가입을 할 수 있다. 이후 로그인을 통해 메인 화면으로 들어갈 수 있다.

##### 나. 메인 화면

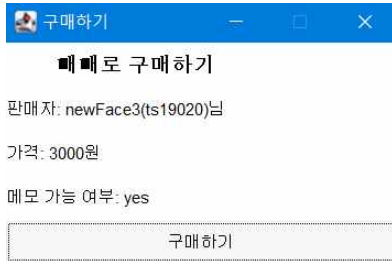


로그인을 하면 메인 화면으로 들어갈 수 있다. 메인화면 상에서는 빼빼로 화면이 나타나며(현재 빼빼로가 등록되어있지 않은 상태이다), 메뉴를 통해 원하는 기능을 수행할 수 있다. 프로필을 누르면 자신의 이름과 아이디, 비밀번호와 잔액을 빠르게 확인할 수 있다. 충전하기를 누르면 원하는 금액을 충전할 수 있다. 빼빼로 등록하기를 누르면 빼빼로를 등록할 수 있으며, 로그아웃을 누르면 로그아웃이 이루어진다.

##### 다. 빼빼로 등록하기

빼빼로 등록하기를 누르면 다음과 같은 창이 나타난다. 종류와 메모 유무, 추가 가격을 입력한 후 예상가격 보기를 누르면 예상 가격이 출력된다. 이후 등록하기를 누르면 본인의 빼빼로가 플랫폼상에 나타난다.

라. 뽀뽀로 구매하기



플랫폼에 나타난 버튼을 누르면 그 뽀뽀로를 구매할 수 있다. 이때 판매자의 이름과 아이디, 가격, 배송 가능 여부가 출력되며, 구매자는 이를 기반으로 구매를 결정할 수 있다. 이때, 자신의 잔액보다 가격이 높으면 구매가 불가능하며, 구매가 완료되면 돈의 이동이 이루어진다. 이후 그 뽀뽀로는 플랫폼상에서 삭제된다.

## V. 기술적 요소

회원 정보 및 등록된 뽀뽀로를 저장하는 데이터베이스가 가장 어려웠다. 본인이 사용한 것은 MySQL이고, user\_database에 users와 pepero 테이블을 만들어 각각 회원 정보와 뽀뽀로를 저장했다. 특히, 이클립스에서 쿼리를 실행해 테이블 내의 값들을 수정하는 작업이 코드의 양과 조작의 측면에서 만만하지 않았다. 다음은 해당 코드이다.

```
public String getMoney(String ID) {
    Connection conn = null;
    Statement stmt = null;
    ResultSet rs = null;
    String money = "0";

    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        String url = "jdbc:mysql://localhost/user_database";
        conn = DriverManager.getConnection(url, "manager", "1234");
        stmt = conn.createStatement();

        String sql = "SELECT * FROM users";
        rs = stmt.executeQuery(sql);

        while(rs.next()) {
            if(ID.equals(rs.getString(1))) {
                money = rs.getString(4);
                break;
            }
        }
    } catch(ClassNotFoundException e) { System.out.println("드라이버 로딩 실패"); }

    catch(SQLException e) { System.out.println("에러 " + e); }
    finally {
        try{
            if( conn != null && !conn.isClosed()){ conn.close(); }
            if( stmt != null && !stmt.isClosed()){ stmt.close(); }
        }
    }
}
```

```

        if( rs != null && !rs.isClosed()){ rs.close(); }
    } catch(SQLException e){ e.printStackTrace(); }
    }
    return money;
}

```

이 코드는 Buying 클래스에서 해당 ID의 잔액을 얻는 코드이다. 이처럼 쿼리를 실행하는 예시 코드를 적절히 변형하여 원하는 기능을 얻도록 하는 작업이 어려운 면이 있었다. 다음은 cmd창에서 데이터베이스를 활용한 결과를 캡처한 사진이다.

```

mysql> use user_database
Database changed
mysql> SELECT * FROM users;
+----+-----+-----+-----+
| id | pw    | name  | money |
+----+-----+-----+-----+
| ts19018 | 19018 | newFace1 | 0 |
| ts19019 | 19019 | newFace2 | 12000 |
| ts19020 | 19020 | newFace3 | 0 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

```

mysql> SELECT * FROM users;
+----+-----+-----+-----+
| id | pw    | name  | money |
+----+-----+-----+-----+
| ts19018 | 19018 | newFace1 | 0 |
| ts19019 | 19019 | newFace2 | 12000 |
| ts19020 | 19020 | newFace3 | 5000 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

```

```

mysql> SELECT * FROM pepero;
+----+-----+-----+-----+-----+
| id | name | price | isMemoPossible | register_id |
+----+-----+-----+-----+-----+
| ts19020 | newFace3 | 3000 | yes | 9017 |
+----+-----+-----+-----+-----+
1 row in set (0.01 sec)

```

```

mysql> SELECT * FROM pepero;
Empty set (0.00 sec)

```

좌측 상단부터 users 테이블에서 회원 목록을 확인한 사진, users 테이블에서 newFace3에 5000원을 충전한 결과를 확인한 사진, pepero 테이블에서 등록한 뽀빠로를 확인한 사진, 뽀빠로 구매 후 pepero 테이블을 다시 확인한 사진이다.

또한, 뽀빠로를 등록할 때 고유의 아이디(register\_id)를 부여하였다. 등록한 뽀빠로별로 구별할 고유의 키가 필요했기 때문에 random 메소드를 이용하여 키를 부여하였다. 이때, 이미 존재하는 키는 부여하면 안되었기에, while문을 돌리며 중복되지 않은 키를 얻을 때까지 random 메소드를 실행하였다. 다음은 해당 코드이다.

```

while(true) {
    register_id = (int) (Math.random()*10000);
    if(!isRegisterIdExists(register_id)) {
        break;
    }
    System.out.println(register_id);
}
pstmt.setInt(5, register_id);

```

이 코드는 Pepero 클래스에서 뽀빠로를 등록할 때 뽀빠로에 register\_id를 부여하는 코드이다. 이처럼 뽀빠로 각각에 고유 키를 부여하기 위해 따로 random 메소드로 키를 생성해 부여한 점이 기억에 남았다.

## VI. 참고 문헌 및 웹 사이트

- <https://blog.naver.com/lghlove0509/221026795913> (mysql 설치 및 기초)
- <https://victorydntmd.tistory.com/145> (JDBC, 쿼리 실행)

## VII. 기대효과 및 개발 소감

본 프로그램은 공유 플랫폼의 새로운 방향성을 제시한 프로그램이다. 공유 플랫폼의 새로운 주제를 제시하여

추후 발전 가능성이 높다는 점에서 의의가 있다. 저가형 제품을 대상으로 플랫폼을 구현함으로써, 기존의 차량 및 주택과 같이 가격대가 높은 상품만을 대상으로 하는 관행에서 벗어나 활용 및 접근 가능성을 높였다는 점에서 기대효과가 크다고 볼 수 있다. 또한, 뽀빠로데이등의 기념 행사시 소외받는 사람의 수를 줄여 상대적 박탈감의 발생 가능성을 낮춘다는 점에서 사회적 이익도 존재한다고 볼 수 있다. 본 프로젝트를 수행하며 awt를 통해 실제 활용 가치가 있는 프로그램을 구현할 수 있다는 것을 알게 되어 흥미로웠고, MySQL을 직접 실행하고 활용해보므로 데이터베이스의 개념과 활용법을 체험하여 의미있었다. 물론 데이터베이스에서 쿼리를 실행해 원하는 정보를 얻고 수정하는 코드를 작성하는 과정에서 어려움이 없지 않아 있었다. 또한, 뽀빠로 등록시 각각을 구별할만한 데이터가 따로 없어 새로 만들어야 했던 것도 어려움이라고 할 수 있었다. 이는 타 개발자의 코드를 참조해 나의 목적에 맞게 수정하고, 뽀빠로마다 random 메소드로 고유 키를 만들어 제공함으로써 해결할 수 있었다. 본 프로젝트를 수행하며 코드를 간결하고 가독성있게 작성하는 것이 어렵다는 것을 느낄 수 있었다. 막상 작성한 코드를 되돌아보니 가독성이 떨어지고 길이가 매우 길어 읽기가 불편하였다. 이러한 아쉬운 점을 토대로 이후 작업시에는 코드의 가독성 측면에서 조금 더 생각하며 개발에 임해야겠다고 생각했다.