Loyalty point exchange smart contract

```
pragma solidity 0.5.1;

contract LoyaltyPointExchangeSystem {
    struct User {
        uint points;
        bool exists;
    }

    mapping (address => User) public users;

    event PointsAccumulated(address indexed user, uint amount);
    event PointsRedeemed(address indexed user, uint amount);
    event PointsTransferred(address indexed from, address indexed to, uint
amount);

    function accumulatePoints(uint amount) public {
        require(amount > 0, "Amount must be greater than 0");

        if (!users[msg.sender].exists) {
            users[msg.sender].exists = true;
        }

        users[msg.sender].points += amount;

        emit PointsAccumulated(msg.sender, amount);
    }

    function redeemPoints(uint amount) public {
        require(amount > 0, "Amount must be greater than 0");
        require(users[msg.sender].points >= amount, "Insufficient points");

        users[msg.sender].points -= amount;

        emit PointsRedeemed(msg.sender, amount);
    }

    function transferPoints(address recipient, uint amount) public {
        require(amount > 0, "Amount must be greater than 0");
        require(users[msg.sender].points >= amount, "Insufficient points");

        if (!users[recipient].exists) {
            users[recipient].exists = true;
        }
```

```
        users[msg.sender].points -= amount;
        users[recipient].points += amount;

        emit PointsTransferred(msg.sender, recipient, amount);
    }
}
```