

고급자바프로그래밍 및 실습 중간고사 보고서

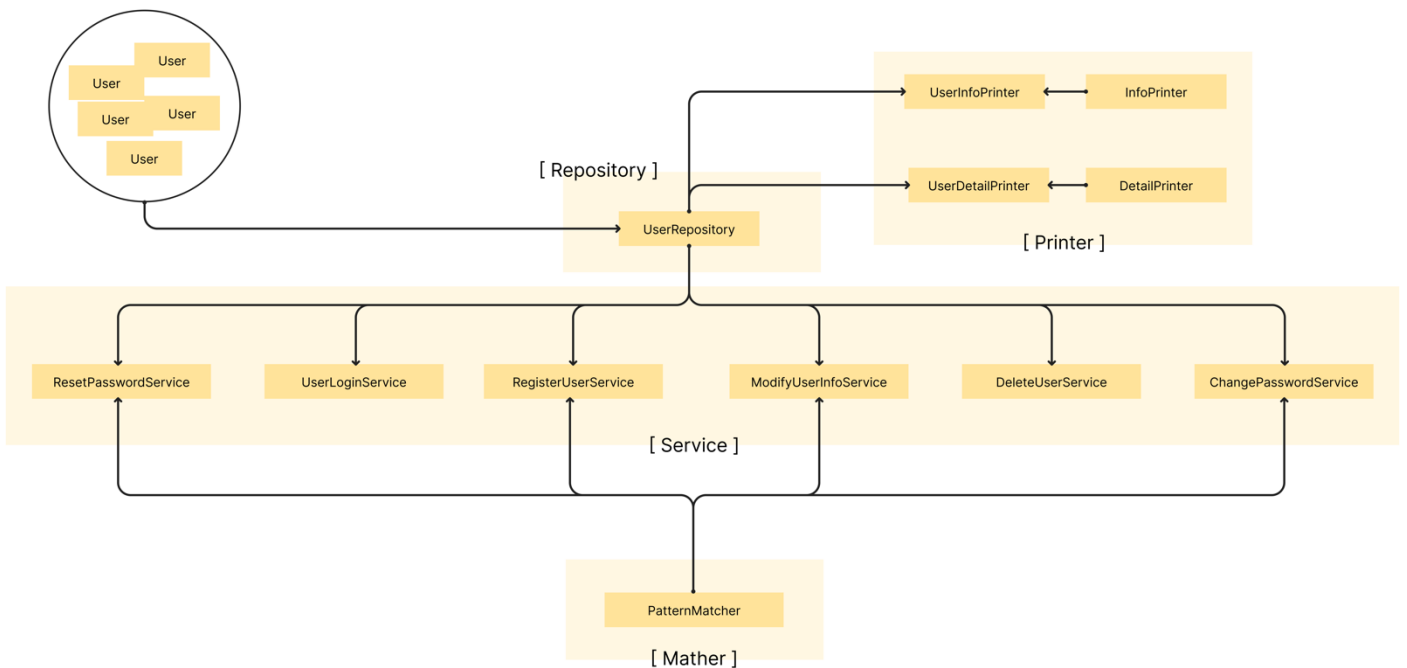
주제	스프링 프레임워크를 사용한 회원관리 기능 설계 및 구현		제출 일자	2022.04.26	
학부	글로벌미디어학부	학번	20192666	이름	김윤서

환경

intelliJ, springframework 5.3.18, java17 사용

설계도

: 프로그램 중 일부만 표현하였습니다.



프로그램 시작

: 콘솔을 통해 0~8사이 숫자를 입력받는다. 0~8 이외의 키를 입력할 경우 프로그램을 계속할지 말지에 대해 결정할 수 있다.

어떤 서비스를 원하세요?

0: 프로그램 종료

1: 회원가입

2: 로그인

3: 암호 재설정

4: 암호 변경

5: 회원 정보 조회

6: 회원 정보 수정

7: 회원 탈퇴

8: 서비스 메뉴 다시 보기

회원가입 RegisterUserService

1) 입력 조건

- 아이디의 경우 2글자에서 10글자 사이, 숫자와 영어만 사용 가능하다.
- 비밀번호의 경우 영어, 숫자, 특수문자를 포함하여 8자 이상만 사용 가능하다.
- 이메일 형식의 경우 올바른 이메일 형식만 사용이 가능하다.
- 전화번호의 경우 "-" 문자를 포함하여 작성해야한다.
- 이름의 경우 빈문자만 아니면 입력가능하다.

2) 필드

- UserRepository userRepository
 - 유저 정보를 가지고 있는 레포지토리를 필드로 선언한다.
- PatternMatcher matcher
 - 유효성 검사를 위한 matcher를 필드로 선언한다.

3) 메서드

- @Autowired
public void setUserList(UserRepository userRepository)
 - UserRepository 객체를 자동 주입받는다.
- @Autowired
public void setMatcher(PatternMatcher matcher)
 - PatternMatcher 객체를 자동 주입 받는다.
- public String register(RegisterRequest req)
 - RegisterRequest 객체를 인자로 받아 회원가입한다.
 - 성공시 해당 유저의 아이디를 반환한다.
- public void isValidId(String id)
 - 아이디를 검사한다
 - 이미 가입된 아이디인 경우 중복된 유저에 대한 예외(DuplicateUserException)를 발생시킨다.
 - 형식에 맞지 않는 아이디인 경우 예외(InvalidIdException)를 발생시킨다.
- public void isValidEmail(String email)
 - 이메일을 검사한다.
 - 형식에 맞지 않는 이메일인 경우 예외(InvalidEmailException)를 발생시킨다.
- public void isValidPassword(String password)
 - 패스워드를 검사한다.
 - 형식에 맞지 않는 패스워드일 경우 예외(InvalidPasswordException)를 발생시킨다.
- public void isPasswordEqualToConfirmPassword(String password, String confirmPassword)
 - 입력된 두가지의 암호가 일치하는 지 검사한다.
 - 일치하지 않은 경우 예외(WrongPasswordException)를 발생시킨다.
- public void isValidPhoneNumber(String phone)
 - 전화번호를 검사한다.
 - 형식에 맞지 않는 전화번호일 경우 예외(InvalidPhoneNumberException)를 발생시킨다.

4) 실행 화면

```
=====
회원가입
=====
```

```
아이디 >> yo
```

```
아이디는 2글자에서 10글자 사이, 숫자와 영어만 사용 가능합니다.
```

```
=====
아이디 >> yo26on
```

```
암호 입력 >> yoon11
```

```
암호는 영어, 숫자, 특수기호를 포함하여 8자 이상으로 입력해주세요.
```

```

=====
암호 입력 >> yoon11**
암호 확인 >> yoon11
암호가 일치하지 않습니다.
>>>> 계속하려면 "0"을 입력하세요.
>>>> 서비스 종료를 원하시면 "0" 이외의 키를 입력해주세요.
>>>> 0

=====
암호 입력 >> yoon11**
암호 확인 >> yoon11**
이메일 입력 >> yo26on@n
올바른 이메일 형식이 아닙니다.
>>>> 계속하려면 "0"을 입력하세요.
>>>> 서비스 종료를 원하시면 "0" 이외의 키를 입력해주세요.
>>>> 0

=====
이메일 입력 >> yo26on@naver.com
전화번호 입력 ("- " 을 포함하여 입력) >> 010
올바른 휴대전화 형식이 아닙니다.
>>>> 계속하려면 "0"을 입력하세요.
>>>> 서비스 종료를 원하시면 "0" 이외의 키를 입력해주세요.
>>>> 0

=====
전화번호 입력 ("- " 을 포함하여 입력) >> 010-6515-6030
이름 입력 >> ㄱ
yo26on님 회원가입되었습니다.
=====

```

로그인 UserLoginService

- 1) 입력조건
 - 사용자로부터 아이디와 패스워드를 입력받는다.
- 2) 필드
 - UserRepository userRepository
 - 유저 정보를 가지고 있는 레포지토리를 필드로 선언한다.

3) 메서드

- @Autowired
public void setUserList(UserRepository userRepository)
 - UserRepository 객체를 자동 주입받는다.
- public User login(String id, String password)
 - 아이디와 암호를 입력받아 로그인을 한다.
 - 유저 레포지토리에 없는 아이디일 경우 예외(NoUserException)을 발생시킨다.
 - 유저 레포지토리에 있는 유저 정보의 암호와 일치하지 않을 경우 예외(WrongPasswordException)을 발생시킨다.
 - 성공적으로 로그인 했을 경우 User 객체를 반환하고, main에서는 로그인 된 객체를 loggedInUser 객체에 저장한다.

4) 실행 화면

```
=====
로그인
=====

아이디 입력 >> yo
비밀번호 입력 >> 11
가입된 아이디가 존재하지 않습니다.
>>>> 계속하려면 "0"을 입력하세요.
>>>> 서비스를 종료하려면 "0" 이외의 키를 입력해주세요.
>>>> 0

=====

아이디 입력 >> yo26on
비밀번호 입력 >> yoon
암호가 일치하지 않습니다.
>>>> 계속하려면 "0"을 입력하세요.
>>>> 서비스를 종료하려면 "0" 이외의 키를 입력해주세요.
>>>> 0

=====

아이디 입력 >> yo26on
비밀번호 입력 >> yoon11**
yo26on님 반갑습니다. 로그인되었습니다.
```

암호 재설정 ResetPasswordService

1) 입력 조건

- 유효성 검사를 위해 아이디, 이메일, 핸드폰 번호를 입력받는다.
- 유효성 검사가 통과되면 암호가 초기화 되며 새로운 암호로 재설정 할 수 있다.

2) 필드

- UserRepository userRepository
 - 유저 정보를 가지고 있는 레포지토리를 필드로 선언한다.
- PatternMatcher matcher
 - 유효성 검사를 위한 matcher를 필드로 선언한다.

3) 메서드

- @Autowired
public void setUserList(UserRepository userRepository)
 - UserRepository 객체를 자동 주입받는다.
- @Autowired
public void setMatcher(PatternMatcher matcher)
 - PatternMatcher 객체를 자동 주입 받는다.
- public void resetPassword(String id, String email, String phone)
 - 아이디, 이메일, 암호를 입력받아 비밀번호를 초기화 한다.
 - 유저 레포지토리에 없는 아이디일 경우 예외(NoUserException)을 발생시킨다.
 - 유저 레포지토리에 있는 정보와 일치하지 않는 경우 예외(WrongInfoException)을 발생시킨다.
 - 위 정의한 예외에 해당하지 않으면 정상적으로 암호가 초기화 된다.
- public void registerNewPassword(String id, String newPassword, String confirmPassword)
 - 아이디와 새로운 비밀번호, 확인 비밀번호를 입력받아 새로운 비밀번호를 설정한다.
 - matcher.isValidPassword(newPassword);
matcher.isPasswordEqualToConfirmPassword(newPassword, confirmPassword); 를 통해 패스워드 유효성 검사를 한다.
 - 형식에 맞지 않는 패스워드일 경우 예외(InvalidPasswordException)를 발생시킨다.
 - 입력된 두가지의 암호가 일치하지 않은 경우 예외(WrongPasswordException)를 발생시킨다.
 - 위 정의한 예외에 해당하지 않으면 정상적으로 암호가 변경된다.

4) 실행 화면

```
=====
메뉴를 입력하세요. (0~8) >> 3
=====

암호 재설정
=====

가입된 아이디 입력 >> yo26on
가입된 이메일 입력 >> yo26on
가입된 전화번호 입력 >>010
입력하신 정보가 가입된 정보와 일치하지 않습니다.
>>>> 계속하려면 "0"을 입력하세요.
>>>> 서비스 종료를 원하시면 "0" 이외의 키를 입력해주세요.
>>>> 0
=====
=====

새로운 비밀 번호 입력 >> yoon1126
새로운 비밀 번호 확인 >> yoon1126
암호는 영어, 숫자, 특수기호를 포함하여 8자 이상으로 입력해주세요.
>>>> 계속하려면 "0"을 입력하세요.
>>>> 서비스 종료를 원하시면 "0" 이외의 키를 입력해주세요.
>>>> 0
```

```
=====
새로운 비밀번호 입력 >> yoon1126**
새로운 비밀번호 확인 >> yoon11
암호가 일치하지 않습니다.
>>>> 계속하려면 "0"을 입력하세요.
>>>> 서비스 종료를 원하시면 "0" 이외의 키를 입력해주세요.
>>>> 0
=====
```

```
=====
새로운 비밀번호 입력 >> yoon1126**
새로운 비밀번호 확인 >> yoon1126**
암호가 변경되었습니다.
=====
```

변경된 비밀번호로 로그인됨

로그인

```
=====
아이디 입력 >> yo26on
비밀번호 입력 >> yoon1126**
yo26on님 반갑습니다. 로그인되었습니다.
=====
```

암호 변경 ChangePasswordService

1) 입력 조건

- 아이디와 기존 비밀번호, 새로운 비밀번호를 입력받는다.
- 예외가 발생하지 않는다면 비밀번호가 초기화된다.

2) 필드

- UserRepository userRepository
 - 유저 정보를 가지고 있는 레포지토리를 필드로 선언한다.
- PatternMatcher matcher
 - 유효성 검사를 위한 mathcher를 필드로 선언한다.

3) 메서드

- @Autowired
public void setUserList(UserRepository userRepository)
 - UserRepository 객체를 자동 주입받는다.
- @Autowired
public void setMatcher(PatternMatcher matcher)
 - PatternMatcher 객체를 자동 주입 받는다.

- `public void changePassword(String id, String oldPassword, String newPassword)`
 - 유저 레포지토리에 없는 아이디일 경우 예외(NoUserException)을 발생시킨다.
 - 유저 레포지토리에 있는 정보와 일치하지 않는 경우 예외(WrongInfoException)을 발생시킨다.
 - 새로운 암호가 올바른 형식이 않으면 예외(InvalidPasswordException)을 발생시킨다.
 - 위 정의한 예외에 해당하지 않으면 정상적으로 암호가 변경된다.

암호 변경

```
=====
아이디 입력 >> yo
기존 비밀번호 입력 >> yoonn
새로운 비밀번호 입력 >> oooooonn
가입된 아이디가 존재하지 않습니다.
```

```
=====
아이디 입력 >> yo26on
기존 비밀번호 입력 >> yoon11**
새로운 비밀번호 입력 >> yoon112626
암호가 일치하지 않습니다.
```

```
=====
아이디 입력 >> yo26on
기존 비밀번호 입력 >> yoon1126**
새로운 비밀번호 입력 >> yoon11**
비밀번호가 변경되었습니다.
```

회원정보 조회 (UserInfoPrinter, UserListPrinter)

일부 회원 조회 조회 UserInfoPrinter

- 1) 입력 조건
 - 조회할 회원의 id를 입력한다.
- 2) 필드
 - UserRepository userRepository
 - 유저 정보를 가지고 있는 레포지토리를 필드로 선언한다.
 - InfoPrinter printer;
 - InfoPrinter 를 상속받은 DetailInfoPrinter를 주입받기 위해 세부 정보를 프린트하는 printer를 필드로 선언한다.
- 3) 메서드
 - `@Autowired`
`public void setUserList(UserRepository userRepository)`
 - UserRepository 객체를 자동 주입받는다.
 - `(@Autowired`
`@Qualifier("detail")`
`public void setPrinter(InfoPrinter printer)`
 - InfoPrinter 를 상속받은 DetailInfoPrinter객체를 자동 주입 받는다.

- public void print(String id)
 - 유저 레포지토리에 없는 아이디일 경우 예외(NoUserException)을 발생시킨다.
 - 예외에 해당하지 않을 경우 조회할 회원의 id를 입력받아 정보를 프린트한다.

회원 정보 조회

```
=====
회원 리스트 조회 : 0, 특정 회원 조회 : 1, 서비스 종료 : 2 >> 1
=====
회원 아이디 입력 >> test3
test3님의 회원정보를 조회합니다.
이름 : 강호동
연락처 : 010-4444-2553
이메일 : test3@naver.com
=====
회원 아이디 입력 >> 111tfg
회원이 존재하지 않습니다.
```

회원 리스트 조회 UserListPrinter

- 1) 입력 조건
 - 별다른 입력 조건은 없으며 호출 즉시 리스트를 프린트한다.
- 2) 필드
 - UserRepository userRepository
 - 유저 정보를 가지고 있는 레포지토리를 필드로 선언한다.
 - InfoPrinter printer;
 - 간단 정보를 프린트하는 printer를 필드로 선언한다.
- 3) 메서드
 - @Autowired
public void setUserList(UserRepository userRepository)
 - UserRepository 객체를 자동 주입받는다.
 - (@Autowired
@Qualifier("detail"))
public void setPrinter(InfoPrinter printer)
 - InfoPrinter 를 상속받은 DetailInfoPrinter객체를 자동 주입 받는다.
 - public void print()
 - 유저 레포지토리에 회원이 한명도 존재하지 않을 경우 예외(NoUserInfoException)가 발생한다.
 - 예외에 해당되지 않을 경우 유저 레포지토리에 존재하는 회원리스트를 전부 다 프린트한다.

```
회원 리스트 조회 : 0, 특정 회원 조회 : 1, 서비스 종료 : 2 >> 0
=====
-----[ 회원 리스트 조회 ]-----
아이디 : test2 이메일 : test2@naver.com 이름 : 유재석 전화번호 : 010-2344-2344
아이디 : test3 이메일 : test3@naver.com 이름 : 강호동 전화번호 : 010-4444-2553
아이디 : test 이메일 : test@naver.com 이름 : 곽두팔 전화번호 : 010-2222-3333
아이디 : yo26on 이메일 : yo26on2@naver.com 이름 : 홍길동 전화번호 : 010-1234-5678
```


회원 정보 수정 (ModifyUserInfoService)

1) 입력 조건

- 가입된 아이디, 이메일, 전화번호를 받아 인증을 받는다.
- 인증을 통과하면 가입된 정보를 수정할 수 있으며 엔터키를 입력하면 변경사항이 없는걸로 간주하여 정보를 수정하지 않는다.

2) 필드

- UserRepository userRepository
 - 유저 정보를 가지고 있는 레포지토리를 필드로 선언한다.
- PatternMatcher matcher
 - 유효성 검사를 위한 matcher를 필드로 선언한다.

3) 메서드

- @Autowired
public void setUserList(UserRepository userRepository)
 - UserRepository 객체를 자동 주입받는다.
- @Autowired
public void setMatcher(PatternMatcher matcher)
 - PatternMatcher 객체를 자동 주입 받는다.
- public void isValidUser(String id, String email, String phone)
 - 유효한 사용자인지 확인한다.
 - 유저 레포지토리에 없는 아이디일 경우 예외(NoUserException)을 발생시킨다.
 - 유저 레포지토리에 저장되어있는 유저의 정보와 다른 정보일 경우 예외(WrongInfoException)을 발생시킨다.
 - 예외에 해당하지 않을 경우 유효한 사용자로 판단한다.
- public void isValidEmail(String email)
 - 이메일을 검사한다.
 - 형식에 맞지 않는 이메일인 경우 예외(InvalidEmailException)를 발생시킨다.
- public void isValidPhoneNumber(String phone)
 - 전화번호를 검사한다.
 - 형식에 맞지 않는 전화번호일 경우 예외(InvalidPhoneNumberException)를 발생시킨다.
- public void modifyInfo(String id, ModifyRequest req)
 - Email, Phone, Name 정보를 담고있는 ModifyRequest와 아이디를 넘겨받는다.
 - 입력된 정보가 있을 경우에만 (null이 아닌 경우) 정보를 수정한다.

```
=====
메뉴를 입력하세요. (0~8) >> 6
=====
회원 정보 수정
=====
가입된 아이디 입력 >> test
가입된 이메일 입력 >> test@naver.com
가입된 전화번호 입력 >> 010-2222-3333
인증되었습니다.
=====
가입 정보를 변경합니다. 변경할 사항이 없으면 엔터키를 입력하세요.
새로운 이메일 입력 >>
이메일 정보를 변경하지 않습니다.
새로운 전화번호 입력 ("- " 을 포함하여 입력) >>
핸드폰 번호 정보를 변경하지 않습니다.
새로운 이름 입력 >>이수근
회원 정보 변경을 완료했습니다.
```

회원 탈퇴 DeleteUserService

1) 입력 조건

- 현재 로그인된 상태에 한해서 탈퇴를 할 수 있으며, 로그인된 상태가 아닌경우 2번메뉴를 통해 로그인을 먼저 해야한다.

2) 필드

- UserRepository userRepository
 - 유저 정보를 가지고 있는 레포지토리를 필드로 선언한다.

3) 메서드

- @Autowired
public void setUserList(UserRepository userRepository)
 - UserRepository 객체를 자동 주입받는다.
- public void deleteUser(User user)
 - 현재 로그인된 User 객체를 넘겨 받아 탈퇴 처리하며, 로그인된 상태가 아니어서 null이 넘겨져 온 경우 예외(NoUserException)를 발생시킨다.
 - 예외에 해당되지 않으면 탈퇴처리 된다.

```
=====
메뉴를 입력하세요. (0~8) >> 7
=====
회원 탈퇴
=====
로그인 상태가 아닙니다.
로그인을 먼저 해주세요.
메뉴를 입력하세요. (0~8) >> 2
=====
로그인
=====
아이디 입력 >> yo26on
비밀번호 입력 >> yoon11**
yo26on님 반갑습니다. 로그인되었습니다.
=====
메뉴를 입력하세요. (0~8) >> 7
=====
회원 탈퇴
=====
로그인된 계정 : yo26on
탈퇴하시겠습니까? (YES:0 NO:1) >> 1
회원 정보 변경 서비스를 종료합니다.
메뉴를 입력하세요. (0~8) >> 7
=====
회원 탈퇴
=====
로그인된 계정 : yo26on
탈퇴하시겠습니까? (YES:0 NO:1) >> 0
탈퇴되었습니다.
```

프로그램 실행 시나리오

어떤 서비스를 원하세요?

0: 프로그램 종료

1: 회원가입

2: 로그인

3: 암호 재설정

4: 암호 변경

5: 회원 정보 조회

6: 회원 정보 수정

7: 회원 탈퇴

8: 서비스 메뉴 다시 보기

메뉴를 입력하세요. (0~8) >> 1

=====

회원가입

=====

아이디 >> test

암호 입력 >> test11**

암호 확인 >> test11**

이메일 입력 >> test@naver.com

전화번호 입력 ("-" 을 포함하여 입력) >> 010-1234-1234

이름 입력 >> 테스트

test님 회원가입되었습니다.

=====

메뉴를 입력하세요. (0~8) >> 1

=====

회원가입

=====

아이디 >> test2

암호 입력 >> test22**

암호 확인 >> test22**

이메일 입력 >> test2@naver.com

전화번호 입력 ("-" 을 포함하여 입력) >> 010-2222-2222

이름 입력 >> 테스트2

test2님 회원가입되었습니다.

=====

메뉴를 입력하세요. (0~8) >> 2

=====

로그인

=====

아이디 입력 >> test

비밀번호 입력 >> test11**

test님 반갑습니다. 로그인되었습니다.

=====

```

메뉴를 입력하세요. (0~8) >> 3
=====

암호 재설정
=====

가입된 아이디 입력 >> test
가입된 이메일 입력 >> test@naver.com
가입된 전화번호 입력 >> 010-1234-1234
초기화되었습니다.

=====

새로운 비밀번호 입력 >> test1111**
새로운 비밀번호 확인 >> test1111**
암호가 변경되었습니다.

=====

메뉴를 입력하세요. (0~8) >> 4
=====

암호 변경
=====

아이디 입력 >> test
기존 비밀번호 입력 >> test1111**
새로운 비밀번호 입력 >> test11**
비밀번호가 변경되었습니다.

=====

메뉴를 입력하세요. (0~8) >> 5
=====

회원 정보 조회
=====

회원 리스트 조회 : 0, 특정 회원 조회 : 1, 서비스 종료 : 2 >> 0
=====

-----[ 회원 리스트 조회 ]-----

아이디 : test2 이메일 : test2@naver.com 이름 : 테스트2 전화번호 : 010-2222-2222
아이디 : test 이메일 : test@naver.com 이름 : 테스트 전화번호 : 010-1234-1234
회원 리스트 조회 : 0, 특정 회원 조회 : 1, 서비스 종료 : 2 >> 1
=====

회원 아이디 입력 >> test
test님의 회원정보를 조회합니다.

이름 : 테스트
연락처 : 010-1234-1234
이메일 : test@naver.com
회원 리스트 조회 : 0, 특정 회원 조회 : 1, 서비스 종료 : 2 >> 2
=====

```

```

=====
>>>> 계속하려면 "0"을 입력하세요.
>>>> 서비스 종료를 원하시면 "0" 이외의 키를 입력해주세요.
>>>> j
=====
회원 정보 조회을/를 종료합니다.

메뉴를 입력하세요. (0~8) >> 6
=====
회원 정보 수정
=====
가입된 아이디 입력 >> test
가입된 이메일 입력 >> test@naver.com
가입된 전화번호 입력 >> 010-1234-1234
인증되었습니다.
=====
가입 정보를 변경합니다. 변경할 사항이 없으면 엔터키를 입력하세요.
새로운 이메일 입력 >>
이메일 정보를 변경하지 않습니다.
새로운 전화번호 입력 ("- " 을 포함하여 입력) >>
핸드폰 번호 정보를 변경하지 않습니다.
새로운 이름 입력 >>테스트입니다
회원 정보 변경을 완료했습니다.
메뉴를 입력하세요. (0~8) >> 7
=====
회원 탈퇴
=====
로그인된 계정 : test
탈퇴하시겠습니까? (YES:0 NO:1) >> 0
탈퇴되었습니다.
메뉴를 입력하세요. (0~8) >> 8
=====

```

메뉴를 입력하세요. (0~8) >> 8

=====

0: 프로그램 종료

1: 회원가입

2: 로그인

3: 암호 재설정

4: 암호 변경

5: 회원 정보 조회

6: 회원 정보 수정

7: 회원 탈퇴

8: 서비스 메뉴 다시 보기

=====

메뉴를 입력하세요. (0~8) >> 0

=====

프로그램을 종료합니다.

=====

Process finished with exit code 0