

# 내일의 집

이정진 김만선 김은진

### 테스트 항목 실행결과(회원가입)

## Form Validation

[illegible]

- 1.

localhost:8080 내용:  
인증번호가 발송되었습니다.

- 2.

☐ ☆ **내일의집**
[내일의집] 이메일 인증 - [이메일 인증번호 업로드]

- 3.

『내일의집』 이예일 연중 (문학평론가)

내 필명 : syonparkhynafn@naver.com

[이메일 인증번호입니다.]

이해 원금번호를 인증번호 확인란에 입력 후 인증번호 화면을 누르시면 아래를 인증이 완료됩니다.  
인증번호 : PYNQGWJ30

```
//인증번호 확인 후 로그인 서버에서 번호 확인후 값 리턴 받음
function matchAuthNumber(){
    var auth_input = document.getElementById('auth_input');
    var authNumber = document.getElementById('auth_input').value;

    //ajax로 인증번호 보내기
    $.ajax({
        type: "post",
        async: false,
        url: "/email/authMatch",
        data: {
            authNumber: authNumber,
        },
        success: function(data){
            if(data.trim()=="success"){
                alert("인증완료");
                auth_input.disabled = true;
                clearInterval(interval);
                document.getElementById('password').focus();
            }
            if(data.trim()=="fail"){
                alert("인증실패. 인증번호를 다시 입력해주세요");
                auth_input.focus();
                authNumber = "";
            }
        }
    }); //end ajax
}
```

```
//인증메일 전송
@PostMapping("/auth")
@ResponseBody
public String emailAuth(MemberDTO memberDTO, HttpSession session) {
    //메일 전송
    String email = memberDTO.getEmail1() + "@" + memberDTO.getEmail2();
    String authKey = mailSendService.sendAuthMail(email);
    System.out.println("authKey>>>>>>>" + authKey);

    //인증키 암호화 해서 세션에 붙임
    String encodedAuthKey = bcryptPasswordEncoder.encode(authKey);
    System.out.println("encodedAuthKey>>>>>>>>>>>" + encodedAuthKey);
    session.setAttribute("authKey", encodedAuthKey);

    deleteSessionAfterTenMinutes(session);

    return "success";
}

static ExecutorService executorService = Executors.newFixedThreadPool(5);

//10분 뒤에 세션삭제
public void deleteSessionAfterTenMinutes(HttpSession session){
    executorService.submit(new Runnable() {
        @Override
        public void run() {
            System.out.println("deleteSession>>10분뒤 세션삭제됨~~");
            try {
                TimeUnit.MILLISECONDS.sleep(600000); //10분 슬립
                session.invalidate(); //세션 삭제
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    });
}
```

4.

08:53

5.

localhost:8080 내용:

인증완료

```
display = document.getElementById('timer');
let interval;
//인증번호 발송 누르면 타이머 시작
function sendMailAndStartTimer() {
    let auth_minutes = 60 * 10; //10분
    var regist_form = document.getElementById('regist_form');
    var email1 = document.getElementById('email1').value;
    var email2 = document.getElementById('email2').value;
    //ajax로 이메일 인증 보내기
    $.ajax({
        type : "post",
        async : false,
        url : "/email/auth",
        data : {
            email1: email1,
            email2: email2
        },
        success : function(data) {
            if(data.trim()=='success'){
                alert("인증번호가 발송되었습니다.");
                document.getElementById('auth_input').focus();
                document.getElementById('email1').readOnly = true;
                document.getElementById('email2').readOnly = true;
            }
        }
    }); //end ajax
};
```

```
//인증번호 확인 누르면 서버에서 번호 확인후 값 리턴 받음
function matchAuthNumber(){
    var auth_input = document.getElementById('auth_input');
    var authNumber = document.getElementById('auth_input').value;

    //ajax로 인증번호 보내기
    $.ajax({
        type : "post",
        async : false,
        url : "/email/authMatch",
        data : {
            authNumber: authNumber,
        },
        success : function(data) {
            if(data.trim()=='success'){
                alert("인증완료");
                auth_input.disabled = true;
                clearInterval(interval);
                document.getElementById('password').focus();
            }
            if(data.trim()=='fail'){
                alert("인증실패. 인증번호를 다시 입력해주세요");
                auth_input.focus();
                authNumber = "";
            }
        }
    }); //end ajax
};
```

6.

localhost:8080 내용:

동일한 비밀번호를 입력해주세요.

확인

7.

localhost:8080 내용:

중복된 닉네임 입니다. 다른 닉네임을 입력해주세요.

확인


//입력한 비밀번호와 비밀번호 체크가 동일한지 확인

```
if (document.getElementById('password').value != document.getElementById('passwordConfirm').value) {  
    alert("동일한 비밀번호를 입력해주세요.");  
    document.getElementById('password').focus();  
    document.getElementById('password').value = "";  
    document.getElementById('passwordConfirm').value = "";  
    return;  
}
```

//중복된 닉네임 있는지 ajax로 값 넘겨서 확인 후 데이터 받아오기

```
var nickname = document.getElementById('nickname').value;  
$.ajax({  
    type : "post",  
    async : false,  
    url : "${contextPath }/member/nicknameSearch",  
    data : {  
        nickname: nickname  
    },  
    success : function(data) {  
        if(data.trim()=="overlap"){  
            alert("중복된 닉네임 입니다. 다른 닉네임을 입력해주세요.");  
            document.getElementById('nickname').focus();  
            document.getElementById('nickname').value = "";  
        }  
    }  
}); //end ajax
```

```
document.getElementById('regist_form').action = "/member/registForm";  
document.getElementById('regist_form').submit();
```



내일의집




이메일

비밀번호

로그인

비밀번호 재설정    회원가입

SNS계정으로 간편하게 로그인/회원가입

로그인에 문제가 있으신가요?

1.

localhost:8080 내용:  
이메일을 입력해주세요.

확인

2.

localhost:8080 내용:  
비밀번호를 입력해주세요.

확인

```
//null체크
if (login_email == '' || login_email == null) {
    alert("이메일을 입력해주세요.");
    login_email.focus();
    return;
}
if (login_password == '' || login_password == null) {
    alert("비밀번호를 입력해주세요.");
    login_password.focus();
    return;
}
```

3.

localhost:8080 내용:

로그인 성공

확인

```
// 로그인 정보 입력후 접근하면 로그인 후 세션에 정보 붙이기
@PostMapping("/login")
@ResponseBody
public String login(MemberDTO memberDTO, HttpSession session) {
    System.out.println("membercontroller>>login>>memberDTO>>>>>>>" + memberDTO);
    AuthUserDTO authUser = memberService.login(memberDTO);
    System.out.println("membercontroller>>authUser>>>" + authUser);
    if (authUser == null) {
        return "fail";
    } else {
        System.out.println("membercontroller>>authUser>>>>" + authUser);
        session.setAttribute("authUser", authUser);
        return "success";
    }
}
```

```
//ajax로 아이디 비밀번호 보내고 리턴값 받기
$.ajax({
    type : "post",
    async : false,
    url : "/member/login",
    data : {
        email1: email1,
        email2: email2,
        password: login_password
    },
    success : function(data) {
        if(data.trim()=='success'){
            alert("로그인 성공");
            let url = '/';
            location.replace(url);
        }
        if(data.trim()=='fail'){
            alert("아이디 또는 비밀번호가 맞지 않습니다. 다시 시도해주세요.");
            document.getElementById('login_email').focus();
            login_email = "";
            login_password = "";
        }
    }
}); //end ajax
```

## 테스트 항목 실행결과 (회원정보 수정)

## Form Validation

회원정보수정

이메일 \*

이메일을 변경하시려면 운영자에게 이메일을 보내주세요.

별명 \*

테스트123

홈페이지

https://tunaorowhome.com


생년월일

연도-월-일

프로필 이미지

파일 선택

선택한 파일 업로드



활동 소개

\* 표시는 필수적으로 입력해야 하는 사항이 가능합니다.

회원 정보 수정

1.

localhost:8080 내용:

중복된 닉네임 입니다. 다른 닉네임을 입력해주세요.

확인

```
//닉네임 변경하면 중복된 닉네임 있는지 ajax로 값 넘겨서 확인 후 데이터 받아오기
let nickname = "<:out value=${memberInfo.nickname}' />";
let changeWickname = document.getElementById('nickname').value;

if (nickname != changeNickname) { //닉네임 값이 바뀌었는지 확인
$.ajax({
  type : "post",
  async : false,
  url : "${contextPath }/member/nicknameSearch", //닉네임 중복체크
  data : {
    nickname : changeNickname
  },
  success : function(data) {
    if (data.trim() == 'overlap') {
      console.log('overlap');
      alert("중복된 닉네임 입니다. 다른 닉네임을 입력해주세요.");
      document.getElementById('nickname').focus();
      document.getElementById('nickname').value = "";
      return;
    }
  } //end if (overlap)
  if(data.trim() == 'unused'){
    console.log('unused');
    modify_submit();
  } //end if (unused)
} // end success
}); //end ajax
} //end if
```

## 테스트 항목 실행결과 (비밀번호 변경)

## Form Validation

### 비밀번호변경

새 비밀번호

영문, 숫자를 포함한 8자 이상의 비밀번호를 입력해주세요.

새 비밀번호 확인

비밀번호 변경

1.

localhost:8080 내용:  
비밀번호를 입력해주세요.

확인

2.

localhost:8080 내용:  
비밀번호 확인을 입력해주세요.

확인

3.

localhost:8080 내용:  
동일한 비밀번호를 입력해주세요.

확인



```
function password_change_click() {

    let password = document.getElementById('password').value
    let password_confirm = document.getElementById('password_confirm').value
    let memberId = "<<:out value=${memberInfo.memberId} />";

    if(password == ""){
        alert("비밀번호를 입력해주세요.");
        document.getElementById('password').focus();
        return;
    }
    if(password_confirm == ""){
        alert("비밀번호 확인을 입력해주세요.");
        document.getElementById('password_confirm').focus();
        return;
    }

    //입력한 비밀번호와 비밀번호 체크가 동일한지 확인
    if (password != password_confirm) {
        alert("동일한 비밀번호를 입력해주세요.");
        document.getElementById('password').focus();
        document.getElementById('password').value = "";
        document.getElementById('password_confirm').value = "";
        return;
    }

    $.ajax({
        type : "post",
        async : false,
        url : "${contextPath }/member/changePassword",
        data : {
            password : password,
            memberId : memberId
        },
        success : function(data) {
            if (data.trim() == 'success') {
                alert("비밀번호 변경 완료");
            }
        }
    }); //end ajax
}
```

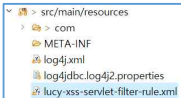
## 1.pox.xml

```
<!-- XSS 방어 - naver / lucy -->
<dependency>
<groupId>com.navercorp.lucy</groupId>
<artifactId>lucy-xss-servlet</artifactId>
<version>2.0.0</version>
</dependency>
</dependencies>
```

## 2.web.xml

```
<!-- XSS 방어 - naver / lucy -->
<filter>
  <filter-name>xssEscapeServletFilter</filter-name>
  <filter-class>com.navercorp.lucy.security.xss.servletfilter.XssEscapeServletFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>xssEscapeServletFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

## 3.lucy-xss-servlet-filter-rule.xml



```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="http://www.navercorp.com/lucy-xss-servlet">
  <defenders>
    <!-- XssPreventer 등록 -->
    <defender>
      <name>xssPreventerDefender</name>
      <class>com.navercorp.lucy.security.xss.servletfilter.defender.XssPreventerDefender</class>
    </defender>

    <!-- XssSaxFilter 등록 -->
    <defender>
      <name>xssSaxFilterDefender</name>
      <class>com.navercorp.lucy.security.xss.servletfilter.defender.XssSaxFilterDefender</class>
      <init-param>
        <param-value>lucy-xss-sax.xml</param-value>    <!-- lucy-xss-filter의 sax을 설정파일 -->
        <param-value>>false</param-value>              <!-- 필터링된 코멘트를 남김지 여부, 성능 효율상 false 추천 -->
      </init-param>
    </defender>

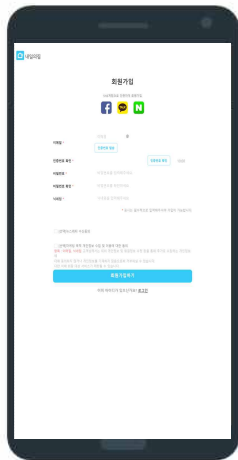
    <!-- XssFilter 등록 -->
    <defender>
      <name>xssFilterDefender</name>
      <class>com.navercorp.lucy.security.xss.servletfilter.defender.XssFilterDefender</class>
      <init-param>
        <param-value>lucy-xss.xml</param-value>        <!-- lucy-xss-filter의 dom을 설정파일 -->
        <param-value>>false</param-value>              <!-- 필터링된 코멘트를 남김지 여부, 성능 효율상 false 추천 -->
      </init-param>
    </defender>
  </defenders>

  <!-- default defender 선언, 별다른 defender 선언이 없으면 default defender를 사용해 필터링 한다. -->
  <default>
    <defender>xssPreventerDefender</defender>
  </default>
</config>
```

## 반응형 UI (회원가입)

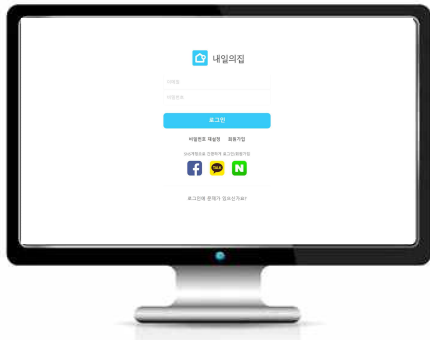


PC화면

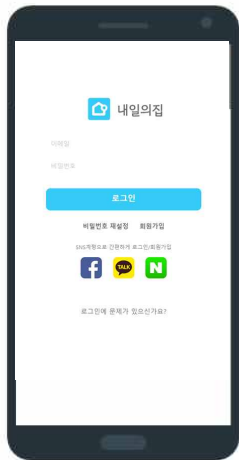


모바일화면

## 반응형 UI (로그인)



PC화면



모바일화면

반응형 UI  
(최의정보 수정)

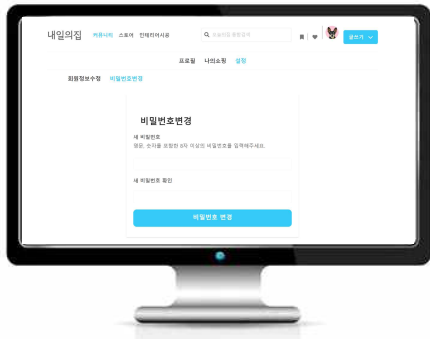


PC화면

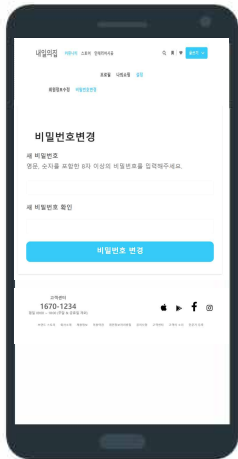


무바일화면

## 반응형 UI (비밀번호 변경)

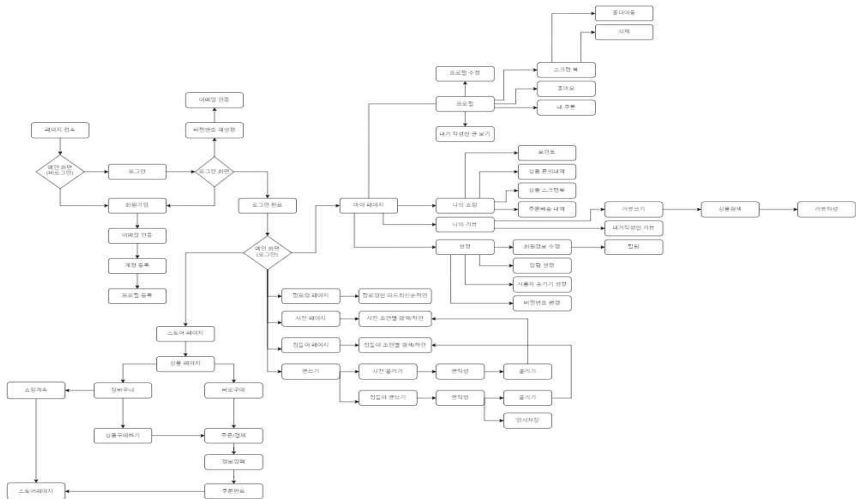


PC화면



모바일화면

기능별 사용자 시나리오
--------------





**감사합니다.**