# 4주차 과제2 (회원가입) 기능 명세

|  |  |  |
| --- | --- | --- |
| 기능 번호 | 세부 기능 설명 | 추가기능여부 |
| FR1 | 아이디 중복 체크가 되어야 합니다. | N |
| FR2 | 비밀번호에 본인만의 규칙을 만들어주세요.  (비밀번호 정상성 체크 기능 있어야 함) | N |
| FR3 | 이름, 아이디, 휴대폰 번호를 입력할 때 규칙에 맞게 입력해야 합니다. | Y |
| FR4 | 비밀번호 확인을 해야 합니다. | Y |
| FR5 | 회원가입을 진행한 후 입력한 내용을 다음 화면에서 확인할 수 있어야 합니다. | Y |
| FR6 | 화면을 껐다 켰다 할 수 있어야 합니다. | Y |

# 구현

## 구현 사항

|  |  |  |
| --- | --- | --- |
| 기능 번호 | 완료여부  (O,X) | 세부사항 |
| FR1 | O | 아이디 입력 후 중복 버튼을 클릭하면 중복 여부를 확인할 수 있습니다. 중복일 시 아이디 입력란에 에러 문구가 뜨고 중복이 아닐 시 중복 버튼이 체크 버튼으로 바뀝니다.  중복 여부가 체크되어 있어야만 회원가입을 할 수 있습니다. |
| FR2 | O | 비밀번호를 8~16글자 사이로 입력해야 하며,  영문, 숫자, 특수문자가 포함되어야 합니다.  해당 조건을 충족하지 못할 시 비밀번호 입력란에 에러 문구가 뜨며 회원가입을 할 수 없습니다. |
| FR3 | O | 이름, 아이디, 휴대번호 또한 규칙이 있으며 다음과 같습니다.  이름: 2글자 이상, 한글만 가능  아이디: 4글자 이상, 숫자와 영어로 가능  휴대번호: 000-0000-0000 형태로만 가능  마찬가지로 해당 조건을 충족하지 못하면 각각 입력란에 에러 문구가 뜨며 회원가입을 할 수 없습니다. |
| FR4 | O | 비밀번호 입력란과 동일한 비밀번호를 입력해야 합니다. 동일하지 않을 시 에러문구가 표기되며 회원가입을 진행할 수 없습니다. |
| FR5 | O | 모든 조건을 완수하고 회원가입을 진행하면 화면이 홈 화면으로 이동합니다. 해당 화면에서는 인사 문구와 함께 사용자가 회원가입 화면에서 입력했던 내용을 확인할 수 있습니다. 또 돌아가기 버튼을 통해 다시 회원가입 화면으로 돌아갈 수 있습니다. |
| FR6 | O | 홈 버튼을 통해 화면을 껐다 킬 수 있습니다. |

## 웹 화면 구성

제목 : 회원가입 화면 (기본)

|  |  |
| --- | --- |
| 화면 | |
| 배지 3 단색으로 채워진배지 단색으로 채워진 배지 1 단색으로 채워진 | |
| 설명 | ① 회원가입 양식을 기입하는 곳입니다. 위에서부터 차례로 이름, 아이디, 비밀번호, 비밀번호 확인, 휴대폰 번호를 입력합니다. 각각 입력 양식이 존재하며 입력된 양식에 맞게 기입해야 합니다.  ② 입력을 성공적으로 마친 뒤 가입하기 버튼을 통해 다음 화면으로 넘어갈 수 있습니다. 입력란이 비어 있는 곳이 있거나 한 곳이라도 오류 문구가 출력되어 있으면 화면은 넘어가지 않습니다.  ③ 아이디 중복 검사를 진행할 수 있는 버튼입니다. 중복 클릭 시 에러 문구 표기 혹은 체크 표시가 됩니다. |

제목 : 회원가입 화면 (에러표기1)

|  |  |
| --- | --- |
| 화면 | |
| 배지 단색으로 채워진배지 1 단색으로 채워진 | |
| 설명 | ① 각 입력란 별로 특정 길이 조건을 충족하지 않을 시 표기되는 오류 화면입니다.  ② 각 입력란 별로 특정 문자 조건을 충족하지 않을 시 표기되는 오류 화면입니다. |

제목 : 회원가입 화면 (에러표기2)

|  |  |
| --- | --- |
| 화면 | |
| 배지 4 단색으로 채워진배지 단색으로 채워진배지 3 단색으로 채워진배지 1 단색으로 채워진 | |
| 설명 | ① 아이디 중복 체크 시 중복이 있다면 화면과 같은 오류가 표기됩니다.  ② 아이디 중복 체크 시 중복이 아니라면 화면과 같이 v가 표기됩니다.  ③ 비밀번호 확인 입력란이 비밀변호와 동일하지 않다면 화면과 같은 오류가 표기됩니다.  ④ 내용이 비어 있는 상태에서 가입하기 버튼을 누르면 화면과 같은 오류가 표기됩니다. |

제목 : 회원가입 성공 후 홈 화면 이동

|  |  |
| --- | --- |
| 화면 | |
| 배지 3 단색으로 채워진배지 단색으로 채워진배지 1 단색으로 채워진 | |
| 설명 | ① 성공적으로 입력란에 입력을 마쳤을 때의 화면입니다. 가입하기 버튼을 통해 회원가입을 진행할 수 있습니다.  ② 회원가입 성공 시 홈 화면으로 이동되어 인사 문구와 함께 입력된 내용을 확인할 수 있습니다.  ③ 돌아가기 버튼을 통해 다시 회원가입 화면으로 돌아갈 수 있습니다. |

제목 : 홈 버튼을 통한 화면 껐다 키기

|  |  |
| --- | --- |
| 화면 | |
| 배지 단색으로 채워진배지 1 단색으로 채워진 텍스트, 스크린샷, 휴대 전화, 모바일 기기이(가) 표시된 사진  자동 생성된 설명 | |
| 설명 | ① 홈 화면을 통해 켜져 있는 화면을 끌 수 있습니다.  ② 화면이 종료된 모습입니다. 다시 홈 버튼을 누르면 화면을 킬 수 있습니다. |

## 구현 내용

### HTML

|  |  |
| --- | --- |
| 파일명 | 구현 내용 |
| index.html | 회원가입 화면을 구성하고 css와 js 파일을 적용합니다. |

### CSS

|  |  |
| --- | --- |
| 파일명 | 구현 내용 |
| base/\_reset.scss | 브라우저의 기본 스타일을 초기화합니다. |
| base/\_global.scss | 공통적으로 적용되는 스타일을 적용합니다. |
| abstracts/\_mixins.scss | 호출해서 사용할 mixin 스타일들을 정의합니다. |
| abstracts/\_variables.scss | 자주 사용하는 색상을 변수로 정의합니다.   * $white, $black, $background: 기본 색상 정의 * $phone-case: 핸드폰 케이스 색상 정의 * $button, $button--normal: 버튼 색상 정의 * $input, input-border: input 색상을 정의 * $error, $clear: 에러, 성공에 따른 색상을 정의 |
| components/\_components.scss | 자주 사용하는 요소들의 스타일을 정의합니다.   * input-form, input-container, id-container: 입력 양식의 스타일 정의 * input, label, button: 각 요소 스타일 정의 |
| pages/\_page.scss | 화면을 구성하는 세부 스타일을 정의합니다.   * phone-container: 핸드폰 전체 스타일 정의 * phone-top, lighter, camera, speaker: 핸드폰 상단 스타일 정의 * phone-middle, title: 핸드폰 중단 스타일 정의 * power—off/on, off/on-ani: 전원 상태에 따른 스타일 및 애니메이션 정의 * phone-bottom, home-button: 핸드폰 하단 및 홈 버튼 스타일 정의 |
| main.scss | \_reset.scss, \_global.scss, \_page.scss, \_components.scss 파일을 통합적으로 가져옵니다. |
| main.css | scss 파일을 css로 전환하여 화면에 스타일을 적용합니다. |

### JavaScript: 변수/요소, 함수, 클래스/속성

파일 명 : main.js

|  |  |
| --- | --- |
| 함수명 | 구현 내용 |
| main() | 함수 인자 설명: 없음  리턴값  addUser, getUserIdArray: initUserArray를 통해 받은 함수  동작 설명  **사용자 정보 배열을 초기화 및 선언하고 최초 화면과 이벤트를 등록합니다. 배열을 조작할 수 있는 함수들을 반환합니다.**  initUserArray()를 통해 사용자 정보 배열을 생성한 뒤 배열을 조작할 수 있는 함수들을 반환합니다.  addPowerEvent()를 통해 전원 이벤트를 등록합니다.  render()를 통해 최초로 회원가입 화면을 띄우고 화면의 각 요소들에 이벤트를 추가가합니다.  에러가 발생할 시 pageErrorHandler를 실행합니다.  외부에서도도 호출이 가능하도록 변수 app을 통해 선언하고 export 해줍니다. |
| render(userData) | 함수 인자 설명  userData: 사용자 정보를 담은 변수  리턴값: 없음  동작 설명  **사용자 정보가 없다면 회원가입 화면을, 있다면 홈 화면을 띄웁니다. 그리고 각 요소별 이벤트를 추가합니다.**  userData가 있다면 moveToHomePage()를 통해 홈화면을 띄우고 없다면 moveToSignPage()를 통해 회원가입 화면을 띄웁니다.  addEvents()를 통해 각 요소별 이벤트를 추가합니다. |
| initUserArray() | 함수 인자 설명: 없음  리턴값  addUser: 사용자 추가 함수  getUserIdArray: 등록된 사용자 아이디들을 가져오는 함수  동작 설명  **사용자 정보를 담는 배열을 생성하고 배열을 조작할 수 있는 함수를 생성하여 반환합니다.**  빈 배열인 userArray를 생성합니다. 이 배열을 조작할 수 있는 addUser()함수와 getUserIdArray()함수를 선언하여 반환합니다. |
| addUser(user) | 함수 인자 설명  user: 사용자 정보를 담은 객체  리턴값: 없음  동작 설명  **전달 받은 사용자 정보를 배열에 추가합니다.**  사용자 정보를 담은 객체 user를 userArray 배열에 추가합니다. |
| getUserIdArray() | 함수 인자 설명: 없음  리턴값  userIdArray: 사용자 아이디를 담은 배열  동작 설명  **사용자 정보 아이디를 담은 배열을 반환합니다.**  map함수를 통해 userArray로부터 사용자 아이디를 추출하여 배열로 담아 반환합니다. |

파일 명 : event/addEvent.js

|  |  |
| --- | --- |
| 함수명 | 구현 내용 |
| addEvents() | 함수 인자 설명: 없음  리턴값: 없음  동작 설명  **전체적인 이벤트를 추가합니다.**  다음 이벤트를 추가합니다.   * addInputEvent() * addSubmitEvent() * addCheckIdEvent() * addMoveBackEvent() |
| addInputEvent() | 함수 인자 설명: 없음  리턴값: 없음  동작 설명  **모든 입력란에 대한 변화 이벤트를 등록합니다.**  모든 input 요소를 담은 inputArray배열에 forEach문을 통해 변화 시 발생하는 이벤트인 handleInputChange()를 등록합니다. |
| addSubmitEvent() | 함수 인자 설명: 없음  리턴값: 없음  동작 설명  **회원가입 양식에 대한 전송 이벤트를 등록합니다.**  form 요소를 가져와 전송 시 발생하는 이벤트인 handleSubmit()을 등록합니다. |
| addCheckIdEvent() | 함수 인자 설명: 없음  리턴값: 없음  동작 설명  **중복 버튼에 중복 확인 이벤트를 등록합니다.**  중복 버튼 요소를 가져와 클릭 시 발생하는 이벤트인 handleIDcheck()를 등록합니다. |
| addMoveBackEvent() | 함수 인자 설명: 없음  리턴값: 없음  동작 설명  **돌아가기 버튼에 돌아가기 이벤트를 등록합니다.**  돌아가기 버튼 요소를 가져와 클릭 시 발생하는 이벤트인 handleMoveBack()을 등록합니다. |
| addPowerEvent() | 함수 인자 설명: 없음  리턴값: 없음  동작 설명  **전원 버튼에 전원 이벤트를 등록합니다.**  전원 버튼 요소를 가져와 클릭 시 발생하는 이벤트인 handlePowerClick()를 등록합니다. |

파일 명 : event/eventHandler.js

|  |  |
| --- | --- |
| 함수명 | 구현 내용 |
| handleInputChange(target) | 함수 인자 설명  target: 이벤트가 발생한 요소 (input 요소)  리턴값: 없음  동작 설명  **변화가 발생한 입력란에 대하여 검증 검사를 실시합니다.**  변화가 발생한 input 요소를 가져와 validator()를 통해 올바른 입력을 수행했는지 확인합니다. |
| handleSubmit(form) | 함수 인자 설명  form: 제줄된 form 요소  리턴값: 없음  동작 설명  **입력받은 사용자 정보를 확인하고 최종적으로 오류가 있는지 확인하고 사용자 정보를 사용자 정보 배열에 등록한 뒤 화면을 전환합니다. 오류가 있다면 오류를 표기하고 마칩니다.**  전달받은 form 요소 정보를 분해해 userData 객체에 담습니다. 그 후 input 요소를 모두 가져와 오류가 없는지 확인합니다. 다음으로 중복 체크를 했는지 확인합니다.  여기까지 오류가 없을 시 userData 객체를 userArray 배열에 addUser()를 통해 등록합니다. 그리고 render()를 통해 다음 화면으로 이동합니다.  오류가 있다면 validateErrorHandler()를 통해 오류에 대한 처리를 하고 기능을 마칩니다. |
| handleIDCheck() | 함수 인자 설명: 없음  리턴값: 없음  동작 설명  **아이디 중복 체크를 확인합니다. 중복이 아닐 시 체크를 하고, 중복일 시 오류를 발생합니다.**  input 요소에 올바르게 입력된 아이디 정보를 가져와 checkIDDuplicate()를 통해 중복 여부를 확인합니다.    중복이 아닐 시 버튼에 체크표시가 뜨며, 중복일 시 validateErrorHandler()를 통해 오류에 대한 처리를 합니다. |
| handleMoveBack() | 함수 인자 설명: 없음  리턴값: 없음  동작 설명  **홈 화면에서 회원가입 화면으로 돌아갑니다.**  render()를 통해 회원가입 화면으로 이동합니다.  에러 발생 시 pageErrorHandler()를 통해 오류를 처리합니다. |
| handlePowerClick() | 함수 인자 설명: 없음  리턴값: 없음  동작 설명  **전원을 끄거나 킵니다.**  핸드폰 몸체 요소를 가져와 power—off 클래스가 있다면 제거한 뒤 power—on 클래스를 추가합니다. 반대로 없다면 power—on 클래스를 제거하고 power—off 클래스를 추가합니다. |

파일 명 : event/errorHandler.js

|  |  |
| --- | --- |
| 함수명 | 구현 내용 |
| validateErrorHandler(target, errorMsg) | 함수 인자 설명  target: 오류를 표기할 요소  errorMsg: 표기할 오류 메시지  리턴값: 없음  동작 설명  **전달받은 요소에 오류문구를 표기하고 요소를 포커스 시킵니다.**  전달받은 target 요소의 값을 비운 뒤 input—error 클래스를 추가하고 placeholder 속성에 errorMsg를 값으로 할당합니다. 그리고 target 요소에 마우스 포커스를 줍니다. |
| pageErrorHandler(error) | 함수 인자 설명  error: 전달받은 에러  리턴값: 없음  동작 설명  **제목란에 오류문구를 표기합니다.**  제목 요소를 가져와 error.message를 값으로 줍니다. |

파일 명 : page/signup.js

|  |  |
| --- | --- |
| 함수명 | 구현 내용 |
| moveToSignupPage() | 함수 인자 설명: 없음  리턴값: 없음  동작 설명  **회원가입 화면을 구성하는 요소를 생성하여 핸드폰 화면에 새롭게 띄웁니다. 만약 기존에 띄어진 화면이 있었다면 그 화면은 제거합니다.**  회원가입 화면을 Html 요소로 구성한 뒤 핸드폰 화면 요소에 insertAdjacentHTML()을 통하여 삽입합니다. 제목도 화면에 맞게 수정합니다.  그리고 기존에 존재했던 화면 요소를 removeChild()를 통해 제거합니다. 기존 화면에 등록되어 있던 화면이 없었다면 오류를 던집니다. |

파일 명 : page/home.js

|  |  |
| --- | --- |
| 함수명 | 구현 내용 |
| moveToHomePage(userData) | 함수 인자 설명  userData: 사용자 정보 객체  리턴값: 없음  동작 설명  **홈 화면을 구성하는 요소를 생성하여 핸드폰 화면에 새롭게 띄웁니다. 만약 기존에 띄어진 화면이 있었다면 그 화면은 제거합니다.**  홈 화면을 Html 요소로 구성한 뒤 userData를 통해 사용자가 입력했던 정보를 담아 핸드폰 화면 요소에 insertAdjacentHTML()을 통하여 삽입합니다. 제목도 화면에 맞게 수정합니다.  그리고 기존에 존재했던 화면 요소를 removeChild()를 통해 제거합니다. 기존 화면에 등록되어 있던 화면이 없었다면 오류를 던집니다. |

파일 명 : utils/validator.js

|  |  |
| --- | --- |
| 함수명 | 구현 내용 |
| validator(target) | 함수 인자 설명  target: 변화가 발생한 입력 요소  리턴값: 없음  동작 설명  **입력 요소에 대한 타입에 따라 검증을 실시합니다. 만약 오류가 없다면 올바른 표기를 하고, 오류가 있다면 오류 표기를 합니다.**  target을 type과 value로 나눕니다. type의 종류 (이름, 아이디, 비밀번호, 비밀번호 확인, 전화번호)에 따라 그에 해당하는 type+validator()를 실행합니다.  만약 오류가 없다면 target에 “nput—clear 클래스를 추가해주고 오류가 있다면 validateErrorHander()를 실행하여 오류 처리를 합니다. |
| nameValidator(value) | 함수 인자 설명  value: 입력된 값  리턴값: 없음  동작 설명  **이름에 대한 검증을 실시합니다. 조건이 맞으면 통과를, 맞지 않으면 오류를 발생합니다.**  정규표현식을 통해 이름에 대한 조건을 선언합니다.  조건은 가-힣 사이의 글자만 가능하며, 2글자 이상이어야 합니다. 조건이 안 맞을 시 특정 오류문구를 던집니다. |
| userIdValidator(value) | 함수 인자 설명  value: 입력된 값  리턴값: 없음  동작 설명  **아이디에 대한 검증을 실시합니다. 조건이 맞으면 통과를, 맞지 않으면 오류를 발생합니다.**  정규표현식을 통해 아이디에 대한 조건을 선언합니다.  조건은 a-z, A-Z, 0-9 사이의 글자만 가능하며, 4글자 이상이어야 합니다. 조건이 안 맞을 시 특정 오류문구를 던집니다. |
| passwordValidator(value) | 함수 인자 설명  value: 입력된 값  리턴값: 없음  동작 설명  **비밀번호에 대한 검증을 실시합니다. 조건이 맞으면 통과를, 맞지 않으면 오류를 발생합니다.**  정규표현식을 통해 이름에 대한 조건을 선언합니다.  조건은 각 a-Z, 0-9, !@#$%^&\* 의 글자를 반드시 하나씩 포함해야 하며, 8글자 이상 16글자 이하여야 합니다. 조건이 안 맞을 시 특정 오류문구를 던집니다. |
| passwordConfirmValidator(value) | 함수 인자 설명  value: 입력된 값  리턴값: 없음  동작 설명  **비밀번호 확인에 대한 검증을 실시합니다. 조건이 맞으면 통과를, 맞지 않으면 오류를 발생합니다.**  비밀번호 입력 요소의 값을 가져와 value와 일치한지 확인합니다. 입력된 비밀번호가 없거나 일치하지 않는다면 특정 오류문구를 던집니다. |
| phonenumberValidator(value) | 함수 인자 설명  value: 입력된 값  리턴값: 없음  동작 설명  **전화번호에 대한 검증을 실시합니다. 조건이 맞으면 통과를, 맞지 않으면 오류를 발생합니다.**  정규표현식을 통해 이름에 대한 조건을 선언합니다.  조건은 숫자로 01x-xxx-xxxx이거나 01x-xxxx-xxxx 이어야 합니다. 조건이 안 맞을 시 특정 오류문구를 던집니다. |

파일 명 : utils/duplicate.js

|  |  |
| --- | --- |
| 함수명 | 구현 내용 |
| checkIDDuplicate(value) | 함수 인자 설명  value: 입력 받은 아이디  리턴값: 없음  동작 설명  **입력받은 아이디가 사용자 정보 배열에 등록되어 있는지 확인합니다.**  value가 getUserIdArray()를 통해 반환된 아이디 배열에 있는지 확인합니다. 있다면 에러 문구를 던집니다. |

# 테스트 결과

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | 테스트 제목 | 테스트 절차 | 관련 FR | 테스트 결과(O,X) |
| 1 | 이름 입력에 대한 유효성 검사 | 1. 입력란에 2글자 미만으로 입력하여 오류가 표기되는지 확인한다.  2. 입력란에 한글이 아닌 문자를 입력하여 오류가 표기되는지 확인한다.  3. 입력란에 2글자 이상, 한글로 입력하여 올바른 입력을 했는지 확인한다. | FR3 | O |
| 2 | 아이디 입력에 대한 유효성 검사 | 1. 입력란에 4글자 미만으로 입력하여 오류가 표기되는지 확인한다.  2. 입력란에 숫자/영어가 아닌 문자를 입력하여 오류가 표기되는지 확인한다.  3. 입력란에 4글자 이상, 숫자/영어를 입력하여 올바른 입력을 했는지 확인한다.  4. 올바르게 입력한 상태에서 중복 버튼을 눌러 중복이 없는지 확인한다. | FR1, FR3 | O |
| 3 | 비밀번호에 대한 유효성 검사 | 1. 입력란에 8글자 미만, 16글자 초과로 입력하여 오류가 표기되는지 확인한다.  2. 입력란에 숫자, 영어, 특수 문자를 각각 빼먹을 경우 오류가 표기되는지 확인한다.  3. 입력란에 8글자 이상, 16글자 이하로 입력하고, 숫자, 영어, 특수문자를 최소 한 문자 씩 포함하여 입력하여 올바른 입력을 했는지 확인한다. | FR2 | O |
| 4 | 비밀번호 확인에 대한 유효성 검사 | 1. 비밀번호가 없는 상태에서 입력 시 오류가 표기되는지 확인한다.  2. 비밀번호가 있는 상태에서 일치하지 않게 입력 시 오류가 표기되는지 확인한다.  3. 비밀번호가 있는 상태에서 일치하게 입력 시 올바른 입력이 되었는지 확인한다. | FR4 | O |
| 5 | 전화번호에 대한 유효성 검사 | 1. 01x-xxx-xxxx, 01x-xxxx-xxxx 양식에 안 맞게 입력 시 오류가 표기되는지 확인한다.  2. 양식에 맞게 입력 시 올바른 입력이 되었는지 확인한다. | FR3 | O |
| 6 | 양식에 대한 최종 유효성 검사 | 1. 가입하기 버튼 클릭 시 비어 있는 곳에 대한 오류를 표기하는지 확인한다.  2. 중복 확인이 안된 상태에서 가입하기 버튼 클릭 시 오류를 표기하는지 확인한다.  3. 양식의 모든 내용이 올바르게 입력되어 있을 시 가입하기 버튼 클릭 시 다음 화면으로 이동하는지 확인한다. | FR1, FR2, FR3, FR4, FR5 | O |
| 7 | 홈 화면 검사 | 1. 입력한 정보에 대하여 잘 표기가 되는지 확인한다  2. 돌아가기 버튼 클릭 시 회원가입 화면으로 돌아가는지 확인한다. | FR5 | O |
| 8 | 전원 검사 | 1. 화면이 켜진 상태에서 홈 버튼 클릭 시 화면이 꺼지는지 확인한다.  꺼진 상태에서는 어떠한 입력도 안되는지 확인한다.  2. 화면이 꺼진 상태에서 홈 버튼 클릭 시 화면이 켜지는지 확인한다.  3. 회원가입, 홈 화면에서 각각 작동하는지 확인한다. | FR6 | O |

# 집중 피드백 받고 싶은 사항

|  |  |
| --- | --- |
| 번호 | 내용 |
| 1 | 먼저 꽤나 내용이 많은 점에 대하여 양해를 구합니다. 기능이 많아짐에 따라 문서와 코드량도 많아져 나름 구조를 나눠보았는데, 잘 되었는지 피드백 부탁드립니다. |
| 2 | 회원가입 화면과 홈 화면으로 서로 전환하고 싶은데 다른 html 문서로 넘기는 방식이 아닌 insertAdjacentHTML을 통하여 계속 요소를 제거하고 추가하는 방식을 선택했습니다. 이 방식이 바람직한 방법인지 아닌지 궁금해서 여쭤봅니다. |