사용자 로그인 화면 (USR-SCR-001)

사용 역할자 : 일반 사용자, 강사, 관리자 (로그인이 필요한 모든 사용자)

관련 메뉴 : (해당 없음 - 시스템 진입점)

기능 1: 사용자 로그인 처리 (ID/PW)

- 관련 요구사항: USR-REQ-003 (사용자 로그인), POR-REQ-046 (포털 로그인)
- 입력 정보:
  - 아이디: 사용자가 입력하는 텍스트 필드. 필수 입력. (프로그램 설계서 4.3 User.login_id 또는 User.email 참조)
  - 비밀번호: 사용자가 입력하는 비밀번호 필드. 마스킹 처리. 필수 입력. (프로그램 설계서 4.3 User.password_hash 참조)
  - 자동 로그인: 사용자가 선택 가능한 체크박스. (선택 사항)
- 출력 정보:
  - 로그인 성공 시: 사용자의 역할에 따라 해당 대시보드 또는 이전 요청 페이지로 리다이렉트. 사용자 정보(이름, 역할 등)를 세션/토큰에 저장.
  - 로그인 실패 시 (ID/PW 불일치): "아이디 또는 비밀번호가 일치하지 않습니다." 오류 메시지를 화면에 표시.
  - 로그인 실패 시 (계정 상태 오류 - 예: 비활성, 잠김): 해당 상태에 맞는 오류 메시지 표시 (예: "사용이 중지된 계정입니다.", "로그인 시도 횟수 초과로 계정이 잠겼습니다.").
  - 네트워크 또는 서버 오류 시: "서비스 접속이 원활하지 않습니다. 잠시 후 다시 시도해주세요." 메시지 표시.
- 프로세스 설명:
  1. 사용자가 아이디와 비밀번호를 입력하고 '로그인' 버튼을 클릭한다.
  2. 클라이언트 측에서 입력값 유효성 검사 (빈 값 여부).
  3. 입력값(아이디, 비밀번호, 자동 로그인 여부)을 서버 로그인 API로 전송한다.
  4. 서버는 전달받은 정보로 사용자 인증을 시도한다 (DB 조회 및 비밀번호 검증).
  5. 인증 성공 시: 사용자 정보와 JWT(토큰)를 생성하여 클라이언트에 반환한다. '자동 로그인' 선택 시 Refresh Token을 추가 발급/관리할 수 있다.
  6. 인증 실패 시: 실패 사유(ID/PW 불일치, 계정 상태 등)를 포함한 응답을 클라이언트에 반환한다.
  7. 클라이언트는 서버 응답에 따라 처리한다. 성공 시 토큰 저장 후 페이지 이동, 실패 시 오류 메시지 표시.
- 호출 API 정보:
  - HTTP 메서드: POST
  - API 엔드포인트: `/api/v1/auth/login`
  - API 목적: 사용자 ID/PW 기반 인증 및 세션/토큰 발급
  - 요청 파라미터:

```
{
  "loginId": "string", // 또는 "email": "string"
  "password": "string",
  "autoLogin": boolean // 자동 로그인 체크 여부
}
```

  - 응답 데이터 (성공 시 예시):

```
{
  "success": true,
  "accessToken": "eyJhbGciOiJIUzI1NiIsIn...",
  "refreshToken": "optional_eyJhbGciOiJIUzI1NiIsIn...", // 자동 로
그인 시
  "userName": "홍길동",
  "userRole": "ROLE_USER" // 또는 ROLE_INSTRUCTOR, ROLE_ADMIN 등
}
```

- 응답 데이터 (실패 시 예시):

```
{
  "success": false,
  "errorCode": "AUTH_INVALID_CREDENTIALS", // 또는
AUTH_ACCOUNT_LOCKED 등
  "message": "아이디 또는 비밀번호가 일치하지 않습니다."
}
```

- Mockup 데이터 예시:
  - 입력:

```
{
  "loginId": "learner01",
  "password": "password123!",
  "autoLogin": false
}
```

  - 출력 (성공): 상기 응답 데이터 성공 예시 참조
  - 출력 (실패): 상기 응답 데이터 실패 예시 참조

기능 2: 소셜 로그인 처리 (네이버)

- 관련 요구사항: USR-REQ-006 (소셜 로그인 - 네이버)
- 입력 정보:
  - '네이버 아이디로 로그인' 버튼 클릭.
- 출력 정보:
  - 네이버 인증 페이지로 리다이렉트.
  - 인증 후 콜백 처리 성공 시: 로그인 성공 처리 (기능 1과 유사하게 대시보드 이동 및 토큰 저장).
  - 인증 후 콜백 처리 실패 시: 오류 메시지 표시 (예: "네이버 로그인에 실패했습니다.", "시스템에 등록되지 않은 사용자입니다.").
  - 최초 소셜 로그인 시: 추가 정보 입력 화면으로 이동하거나, 필수 정보가 충분하면 자동 회원가입 후 로그인 처리 될 수 있음 (정책 결정 필요).
- 프로세스 설명:
  1. 사용자가 '네이버 아이디로 로그인' 버튼을 클릭한다.

2. 클라이언트는 백엔드에 네이버 인증 URL 생성을 요청하거나, 미리 정의된 URL로 네이버 인증 페이지로 리다이렉트한다.
3. 사용자는 네이버 페이지에서 로그인 및 정보 제공 동의를 진행한다.
4. 네이버는 인증 완료 후, 사전에 등록된 콜백 URL로 사용자를 리다이렉트시키며 인증 코드(Authorization Code)를 전달한다.
5. 콜백 URL을 받은 클라이언트(또는 서버)는 해당 코드를 사용하여 백엔드 API를 호출한다.
6. 백엔드는 전달받은 코드로 네이버 API를 호출하여 Access Token을 받고, 이 토큰으로 사용자 프로필 정보를 조회한다.
7. 조회된 네이버 사용자 정보(고유 ID 등)를 기반으로 시스템 DB에서 기존 사용자인지 확인한다.
8. 기존 사용자일 경우: 로그인 처리 후 토큰 발급 (기능 1의 성공 프로세스와 유사).
9. 신규 사용자일 경우:
    - (정책1) 필수 정보가 충분하면 자동 회원가입 후 로그인 처리 및 토큰 발급.
    - (정책2) 추가 정보 입력 화면으로 이동시켜 가입 절차 마무리 유도.
10. 처리 결과를 클라이언트에 반환하여 페이지 이동 또는 오류 메시지를 표시한다.

- 호출 API 정보:
    - 클라이언트 -> 네이버: (리다이렉트) 네이버 인증 요청 URL
    - 네이버 -> 클라이언트/서버: (리다이렉트) 콜백 URL + `code`, `state`
    - 클라이언트/서버 -> 백엔드:
        - HTTP 메서드: GET (또는 POST)
        - API 엔드포인트: `/api/v1/auth/social/naver/callback`
        - API 목적: 네이버 인증 콜백 처리, 사용자 식별/가입/로그인 및 토큰 발급
        - 요청 파라미터: `code` (네이버 발급 인증 코드), `state` (CSRF 방지용)
    - 백엔드 -> 네이버 API: 토큰 요청, 사용자 정보 요청 (백엔드 내부 처리)
    - 응답 데이터: 기능 1의 로그인 성공/실패 응답과 유사. 추가 정보 필요시 별도 응답 정의.
- Mockup 데이터 예시:
    - 입력: (네이버 인증 후 콜백 URL 호출 시 파라미터) `?code=NAVER_AUTH_CODE&state=CSRF_TOKEN`
    - 출력 (성공): 기능 1의 로그인 성공 응답과 유사
    - 출력 (실패): 기능 1의 로그인 실패 응답과 유사 (오류 코드/메시지 상이)

기능 3: 소셜 로그인 처리 (카카오)

- 관련 요구사항: USR-REQ-007 (소셜 로그인 - 카카오톡)
- 입력 정보:
    - '카카오 로그인' 버튼 클릭.
- 출력 정보: 기능 2 (네이버)와 동일 (카카오 인증 페이지, 성공/실패 처리).
- 프로세스 설명: 기능 2 (네이버)와 동일한 플로우, 단 인증 주체 및 API 호출 대상이 카카오로 변경됨.
- 호출 API 정보:
    - 클라이언트 -> 카카오: (리다이렉트) 카카오 인증 요청 URL
    - 카카오 -> 클라이언트/서버: (리다이렉트) 콜백 URL + `code`
    - 클라이언트/서버 -> 백엔드:
        - HTTP 메서드: GET (또는 POST)
        - API 엔드포인트: `/api/v1/auth/social/kakao/callback`
        - API 목적: 카카오 인증 콜백 처리, 사용자 식별/가입/로그인 및 토큰 발급
        - 요청 파라미터: `code` (카카오 발급 인증 코드)
    - 백엔드 -> 카카오 API: 토큰 요청, 사용자 정보 요청 (백엔드 내부 처리)
    - 응답 데이터: 기능 1의 로그인 성공/실패 응답과 유사.

- Mockup 데이터 예시:
  - 입력: (카카오 인증 후 콜백 URL 호출 시 파라미터) `?code=KAKAO_AUTH_CODE`
  - 출력: 기능 2 (네이버)와 동일.

기능 4: 아이디 찾기 화면 이동

- 관련 요구사항: USR-REQ-004 (아이디 찾기 기능으로 연결)
- 입력 정보:
  - "아이디 찾기" 링크 클릭.
- 출력 정보:
  - 아이디 찾기 화면 (USR-SCR-006)으로 페이지 전환.
- 프로세스 설명:
  1. 사용자가 로그인 폼 하단의 "아이디 찾기" 링크를 클릭한다.
  2. 클라이언트 라우터가 `/find-id` 경로로 네비게이션을 수행한다.
- 호출 API: 없음 (클라이언트 사이드 라우팅)
- Mockup 데이터: 없음

기능 5: 비밀번호 찾기 화면 이동

- 관련 요구사항: USR-REQ-005 (패스워드 찾기 기능으로 연결)
- 입력 정보:
  - "비밀번호 찾기" 링크 클릭.
- 출력 정보:
  - 비밀번호 찾기 화면 (USR-SCR-007)으로 페이지 전환.
- 프로세스 설명:
  1. 사용자가 로그인 폼 하단의 "비밀번호 찾기" 링크를 클릭한다.
  2. 클라이언트 라우터가 `/find-password` 경로로 네비게이션을 수행한다.
- 호출 API: 없음 (클라이언트 사이드 라우팅)
- Mockup 데이터: 없음

기능 6: 회원가입 화면 이동

- 관련 요구사항: USR-REQ-002 (회원 가입 기능으로 연결), POR-REQ-047 (포털 회원 가입)
- 입력 정보:
  - "회원가입" 버튼/링크 클릭.
- 출력 정보:
  - 회원가입 약관 동의 화면 (USR-SCR-002)으로 페이지 전환.
- 프로세스 설명:
  1. 사용자가 로그인 폼 하단 또는 별도의 "회원가입" 버튼/링크를 클릭한다.
  2. 클라이언트 라우터가 `/signup/terms` (또는 `/signup`) 경로로 네비게이션을 수행한다.
- 호출 API: 없음 (클라이언트 사이드 라우팅)
- Mockup 데이터: 없음

예외 처리 및 유의 사항

- 입력값 유효성 검사: 아이디 또는 비밀번호 미입력 시, '로그인' 버튼 비활성화 또는 클릭 시 입력 요청 메시지 표시.
- 로그인 시도 제한: 일정 횟수(예: 5회) 이상 로그인 실패 시 계정 잠금 처리 및 안내 메시지 표시. 잠금 해제 절차 필요 (예: 비밀번호 재설정).
- 자동 로그인: 체크 시 발급된 Refresh Token을 안전하게 저장(예: HttpOnly 쿠키)하고, 다음 방문 시 Access Token 재발급에 사용. 미체크 시 브라우저 종료 시 로그아웃.
- 소셜 로그인 연동 해제: 마이페이지 등에서 소셜 로그인 연동을 해제하는 기능 필요 (별도 화면 설계).
- 소셜 로그인 계정 통합: 동일인(CI 기준)이 ID/PW 가입 후 소셜 로그인을 시도하거나, 여러 소셜 계정으로 로그인 시도 시 계정 통합/연동 처리 방안 필요 (정책 및 추가 화면 설계 필요).

회원가입 - 약관 동의 화면 (USR-SCR-002)

사용 역할자 : 비회원 (회원가입 시도 사용자)

관련 메뉴 : (회원가입 프로세스 시작)

기능 1: 약관 내용 조회

- 관련 요구사항: POR-REQ-009 (서비스 약관 조회), POR-REQ-010 (개인 정보 처리 방침 조회), SYS-REQ-033 (약관 관리 - 조회 기능 연계)
- 입력 정보:
  - '서비스 이용약관' 보기 링크/버튼 클릭.
  - '개인정보 수집 및 이용 동의' 보기 링크/버튼 클릭.
  - (선택) '마케팅 정보 수신 동의' 보기 링크/버튼 클릭.
- 출력 정보:
  - 해당 약관의 전체 내용이 모달 팝업 또는 별도 영역에 표시됨. (내용은 관리자 시스템에서 관리된 데이터를 불러옴)
- 프로세스 설명:
  1. 사용자가 특정 약관의 '보기' 링크/버튼을 클릭한다.
  2. 클라이언트는 해당 약관 내용을 표시하기 위한 UI(모달 등)를 활성화한다.
  3. 약관 내용은 정적으로 포함하거나, 서버 API를 통해 동적으로 로드하여 표시한다.
  4. 사용자는 내용을 확인하고 팝업/영역을 닫을 수 있다.
- 호출 API 정보: (약관 내용을 동적으로 로드할 경우)
  - HTTP 메서드: GET
  - API 엔드포인트: `/api/v1/terms/{termType}` (예: `/api/v1/terms/service`, `/api/v1/terms/privacy`)
  - API 목적: 지정된 타입의 최신 약관 내용 조회
  - 요청 파라미터: `termType` (Path Variable: service, privacy, marketing 등)
  - 응답 데이터:

```
{
  "termType": "service",
  "title": "서비스 이용약관",
  "content": "제 1조 (목적)...(약관 내용 HTML 또는 Text)...",
  "effectiveDate": "2024-01-01"
}
```

- Mockup 데이터 예시:
  - 입력: '서비스 이용약관' 보기 클릭
  - 출력: 약관 내용이 담긴 모달 팝업 표시

기능 2: 약관 동의 처리

- 관련 요구사항: USR-REQ-002 (회원 가입 - 약관 동의는 가입의 필수 절차)
- 입력 정보:
  - 전체 동의 체크박스 클릭.
  - 서비스 이용약관 (필수) 동의 체크박스 클릭.
  - 개인정보 수집 및 이용 동의 (필수) 동의 체크박스 클릭.
  - (선택) 마케팅 정보 수신 동의 체크박스 클릭.
  - '다음' 또는 '동의하고 가입 계속' 버튼 클릭.
- 출력 정보:
  - 필수 약관 모두 동의 시: 본인 인증 화면 (USR-SCR-003)으로 페이지 전환.
  - 필수 약관 미동의 시: '필수 약관에 동의해주세요.' 안내 메시지 표시 및 진행 차단.
- 프로세스 설명:
  1. 사용자가 각 약관 항목의 체크박스를 선택/해제한다. '전체 동의' 체크 시 하위 필수/선택 항목이 연동되어 변경될 수 있다.
  2. 사용자가 '다음' 버튼을 클릭한다.
  3. 클라이언트는 필수 약관(서비스 이용약관, 개인정보 수집 및 이용) 동의 여부를 확인한다.
  4. 모든 필수 약관에 동의한 경우, 다음 단계인 본인 인증 화면으로 이동한다. 동의한 약관 정보 (특히 선택 약관)는 다음 단계로 전달하거나 임시 저장한다.
  5. 필수 약관 중 하나라도 미동의 시, 사용자에게 알림 메시지를 표시하고 진행을 막는다.
- 호출 API: 없음 (클라이언트 사이드 처리 및 다음 단계로 진행)
- Mockup 데이터: 없음

예외 처리 및 유의 사항

- 필수 약관과 선택 약관(마케팅 등)을 명확히 구분하여 표시해야 한다.
- '전체 동의' 체크박스 기능은 사용 편의성을 위해 제공하며, 하위 항목과의 연동 로직이 정확해야 한다.
- 약관 내용은 최신 버전이 표시되어야 하며, 변경 시 사용자에게 고지하는 절차가 필요할 수 있다 (별도 정책).
- 사용자가 동의한 약관 버전 및 동의 일시는 회원 정보와 함께 저장되어야 한다 (백엔드 처리).

회원가입 - 본인 인증 화면 (USR-SCR-003)

사용 역할자 : 비회원 (회원가입 중 약관 동의 완료 사용자)

관련 메뉴 : (회원가입 프로세스 중)

기능 1: 본인 인증 수행

- 관련 요구사항: USR-REQ-001 (회원가입 시 본인 인증)
- 입력 정보:
  - 인증 수단 선택 (예: 휴대폰 인증 - 기본값 또는 유일 옵션일 수 있음).
  - 이름: 사용자 입력 텍스트 필드. 필수.
  - 생년월일: 사용자 입력 필드 (YYYYMMDD 형식 또는 Date Picker). 필수.
  - 성별: 라디오 버튼 또는 드롭다운 선택 (남/여). 필수.
  - 통신사: 드롭다운 선택 (SKT, KT, LGU+, 알뜰폰). 필수.
  - 휴대폰 번호: 사용자 입력 텍스트 필드 ('-' 제외 숫자만 입력). 필수.
  - 인증번호 요청 버튼 클릭.
  - 인증번호: 사용자가 수신한 SMS 인증번호를 입력하는 텍스트 필드. 필수.
  - '확인' 또는 '인증 완료' 버튼 클릭.
- 출력 정보:
  - 인증번호 요청 시: 사용자의 휴대폰으로 SMS 인증번호 발송. 화면에는 인증번호 입력 필드 및 유효 시간 타이머 표시.
  - 인증 성공 시: "본인 인증이 완료되었습니다." 메시지 표시 후, 회원 정보 입력 화면 (USR-SCR-004)으로 자동 전환 또는 '다음' 버튼 활성화. 인증된 사용자 정보(이름, 생년월일, 성별, CI, DI 등)를 안전하게 다음 단계로 전달.
  - 인증 실패 시 (정보 불일치): "입력하신 정보와 일치하는 사용자가 없습니다." 메시지 표시.
  - 인증 실패 시 (인증번호 불일치): "인증번호가 올바르지 않습니다." 메시지 표시.
  - 인증 실패 시 (유효 시간 초과): "인증 유효 시간이 초과되었습니다. 다시 시도해주세요." 메시지 표시.
  - 이미 가입된 사용자일 경우: "이미 가입된 사용자입니다. 아이디/비밀번호 찾기를 이용해주세요." 메시지 표시 및 가입 중단.
- 프로세스 설명:
  1. 사용자가 본인 명의의 정보(이름, 생년월일, 성별, 통신사, 휴대폰 번호)를 입력하고 '인증번호 요청' 버튼을 클릭한다.
  2. 클라이언트는 입력 정보를 백엔드 본인 인증 요청 API로 전송한다.
  3. 백엔드는 외부 본인 인증 서비스(예: PG사 제공 서비스, PASS 앱 등)와 연동하여 사용자 정보 검증 및 인증번호 발송을 요청한다.
  4. 외부 서비스는 사용자에게 SMS로 인증번호를 발송한다. 백엔드는 발송 성공 여부 및 인증 트랜잭션 ID 등을 클라이언트에 응답할 수 있다.
  5. 클라이언트는 인증번호 입력 필드와 유효 시간 타이머를 표시한다.
  6. 사용자가 수신한 인증번호를 입력하고 '확인' 버튼을 클릭한다.
  7. 클라이언트는 입력된 인증번호와 트랜잭션 ID(필요시)를 백엔드 본인 인증 확인 API로 전송한다.
  8. 백엔드는 외부 본인 인증 서비스에 인증번호 검증을 요청한다.
  9. 검증 성공 시, 외부 서비스는 CI(연계 정보), DI(중복가입 확인 정보) 등 고유 식별 정보를 백엔드에 반환한다.
  10. 백엔드는 반환된 CI/DI로 시스템 DB를 조회하여 기가입 여부를 확인한다.
  11. 기가입 사용자가 아닐 경우, 인증 성공 응답과 함께 CI/DI 및 확인된 본인 정보(이름 등)를 클라이언트에 전달한다 (또는 서버 세션에 저장).
  12. 기가입 사용자일 경우, 가입 불가 응답을 전달한다.
  13. 인증 실패(번호 불일치, 시간 초과 등) 시, 해당 오류 응답을 전달한다.
  14. 클라이언트는 응답에 따라 성공 시 다음 화면으로 이동하거나, 실패/오류 메시지를 표시한다.
- 호출 API 정보:
  - 본인 인증 요청:
    - HTTP 메서드: POST
    - API 엔드포인트: `/api/v1/auth/verify/request` (또는 `/api/v1/verification/request-code`)

        - API 목적: 외부 본인 인증 서비스 연동하여 인증번호 발송 요청
        - 요청 파라미터: `name`, `birthDate`, `gender`, `telecom`, `phoneNumber`
        - 응답 데이터: `{"success": true, "transactionId": "...", "expiresIn": 180}` 또는 `{"success": false, "message": "..."}`
    - 본인 인증 확인:
        - HTTP 메서드: POST
        - API 엔드포인트: `/api/v1/auth/verify/confirm` (또는 `/api/v1/verification/confirm-code`)
        - API 목적: 입력된 인증번호 검증 및 사용자 식별 정보(CI/DI) 획득
        - 요청 파라미터: `transactionId` (요청 시 받은 ID), `authCode` (사용자 입력 인증번호)
        - 응답 데이터 (성공): `{"success": true, "ci": "...", "di": "...", "name": "홍길동", ...}`
        - 응답 데이터 (실패): `{"success": false, "message": "인증번호 불일치"}` 또는 `{"success": false, "message": "이미 가입된 사용자"}`
- Mockup 데이터 예시:
    - 입력 (요청 시): `{"name": "홍길동", "birthDate": "19900101", "gender": "M", "telecom": "SKT", "phoneNumber": "01012345678"}`
    - 입력 (확인 시): `{"transactionId": "...", "authCode": "123456"}`
    - 출력: 상기 API 응답 데이터 예시 참조

예외 처리 및 유의 사항

- 본인 인증 서비스는 외부 유료 서비스를 사용하는 경우가 많으므로, 연동 규격 및 비용 정책을 확인해야 한다.
- 인증 유효 시간(보통 3분)을 명확히 표시하고, 시간 초과 시 재시도 안내를 해야 한다.
- 인증 시도 횟수 제한을 두어 비정상적인 요청을 방지할 수 있다.
- 통신 장애 또는 외부 서비스 장애 시 예외 처리 방안이 필요하다.
- CI/DI 등 개인 식별 정보는 암호화하여 안전하게 관리되어야 한다.

회원가입 - 정보 입력 화면 (USR-SCR-004)

사용 역할자 : 비회원 (회원가입 중 본인 인증 완료 사용자)

관련 메뉴 : (회원가입 프로세스 중)

기능 1: 회원 정보 입력 및 가입 처리

- 관련 요구사항: USR-REQ-002 (회원 가입 완료), SFR-003 (회원 관련 기능 - 기본 정보 입력)
- 입력 정보:
    - 아이디: 사용자 입력 텍스트 필드. 필수. (영문/숫자 조합, 길이 제한 등 규칙 필요) + '중복 확인' 버튼.
    - 비밀번호: 사용자 입력 비밀번호 필드. 필수. (보안 강도 규칙 적용: 길이, 특수문자 포함 등)
    - 비밀번호 확인: 사용자 입력 비밀번호 필드. 필수. (비밀번호와 일치 여부 확인)
    - 이름: 본인 인증 결과로 자동 입력 (수정 불가 처리). 필수.
    - 이메일 주소: 사용자 입력 텍스트 필드. 필수. (형식 검증 필요) + '인증' 버튼 (선택 사항, 이메일 인증 추가 시).
    - 휴대폰 번호: 본인 인증 결과로 자동 입력 (수정 불가 처리). 필수.

    - (선택) 사용자 유형: 드롭다운 또는 라디오 버튼 (선수, 지도자, 심판, 행정가, 일반 체육인 등 - POR-REQ-006 분류 참조).
    - (선택) 관심 분야: 체크박스 또는 다중 선택 목록 (스포츠 어학, 소양 과정 등 - POR-REQ-007, POR-REQ-020 참조).
    - '가입 완료' 버튼 클릭.
- 출력 정보:
    - 아이디 중복 확인 시: "사용 가능한 아이디입니다." 또는 "이미 사용 중인 아이디입니다." 메시지 표시.
    - 비밀번호 규칙 미준수 시: 규칙 안내 메시지 표시 (예: "8~16자 영문, 숫자, 특수문자를 조합해주세요.").
    - 비밀번호 불일치 시: "비밀번호가 일치하지 않습니다." 메시지 표시.
    - 이메일 형식 오류 시: "올바른 이메일 형식이 아닙니다." 메시지 표시.
    - (이메일 인증 시) 인증번호 발송/확인 결과 메시지.
    - 가입 성공 시: 회원가입 완료 화면 (USR-SCR-005)으로 페이지 전환.
    - 가입 실패 시 (DB 오류 등): "회원가입 중 오류가 발생했습니다. 다시 시도해주세요." 메시지 표시.
- 프로세스 설명:
    1. 사용자는 아이디를 입력하고 '중복 확인' 버튼을 클릭한다. 클라이언트는 API를 호출하여 아이디 사용 가능 여부를 확인하고 결과를 표시한다.
    2. 사용자는 비밀번호와 비밀번호 확인 값을 입력한다. 클라이언트는 실시간으로 두 값의 일치 여부 및 비밀번호 규칙 준수 여부를 검사하여 피드백을 준다.
    3. 이름과 휴대폰 번호는 이전 단계에서 인증된 정보로 자동 채워진다 (수정 불가).
    4. 사용자는 이메일 주소를 입력한다. 클라이언트는 형식 유효성을 검사한다. (선택적 이메일 인증 절차 추가 가능).
    5. 사용자는 선택 항목(사용자 유형, 관심 분야 등)을 입력/선택한다.
    6. 모든 필수 정보 입력 및 검증(아이디 중복 확인 등)이 완료되면 '가입 완료' 버튼이 활성화된다.
    7. 사용자가 '가입 완료' 버튼을 클릭한다.
    8. 클라이언트는 모든 입력 정보와 함께 본인 인증 결과(CI/DI 등)를 회원가입 API로 전송한다.
    9. 백엔드는 최종 유효성 검사 후 사용자 정보를 DB에 저장한다 (비밀번호는 안전하게 해싱하여 저장). 약관 동의 정보(버전, 일시)도 함께 저장한다.
    10. 가입 성공 시 성공 응답을, 실패 시 실패 응답을 클라이언트에 반환한다.
    11. 클라이언트는 응답에 따라 완료 화면으로 이동하거나 오류 메시지를 표시한다.
- 호출 API 정보:
    - 아이디 중복 확인:
        - HTTP 메서드: GET (또는 POST)
        - API 엔드포인트: `/api/v1/auth/check-id/{loginId}` (또는 `/api/v1/users/check-id`)
        - API 목적: 입력된 아이디의 사용 가능 여부 확인
        - 요청 파라미터: `loginId` (Path Variable 또는 Request Parameter)
        - 응답 데이터: `{"available": true}` 또는 `{"available": false}`
    - 회원가입 처리:
        - HTTP 메서드: POST
        - API 엔드포인트: `/api/v1/auth/signup`
        - API 목적: 최종 사용자 정보 등록
        - 요청 파라미터:

```
{
  "loginId": "string",
  "password": "string",
```

```
  "email": "string",
  "name": "string", // 인증된 정보
  "phoneNumber": "string", // 인증된 정보
  "ci": "string", // 인증된 정보
  "di": "string", // 인증된 정보
  "userType": "string", // 선택
  "interests": ["string"], // 선택
  "agreedTerms": [ // 동의한 약관 정보
    {"termType": "service", "agreed": true, "agreedAt":
"timestamp"},
    {"termType": "privacy", "agreed": true, "agreedAt":
"timestamp"},
    {"termType": "marketing", "agreed": false, "agreedAt":
"timestamp"} // 선택 약관 예시
  ]
}
```

    - 응답 데이터: `{"success": true}` 또는 `{"success": false, "message": "..."}`
- Mockup 데이터 예시:
    - 입력 (아이디 중복 확인): `testuser01`
    - 출력 (아이디 중복 확인): `{"available": false}`
    - 입력 (회원가입 처리): 상기 API 요청 파라미터 예시 참조
    - 출력 (회원가입 처리): `{"success": true}`

예외 처리 및 유의 사항

- 아이디, 비밀번호, 이메일 등 각 필드별 유효성 검사 규칙을 명확히 정의하고 사용자에게 안내해야 한다.
- 비밀번호는 보안 강도(복잡도) 요구사항을 충족해야 하며, 클라이언트/서버 양쪽에서 검증해야 한다.
- 비밀번호는 평문으로 전송/저장되지 않도록 HTTPS 사용 및 서버측 해싱(Salt 사용 권장)이 필수이다.
- 본인 인증 정보(이름, 휴대폰)는 수정할 수 없도록 처리해야 한다.
- 선택 입력 항목(사용자 유형, 관심 분야)은 추후 마이페이지에서 수정 가능하도록 하는 것이 일반적이다.

회원가입 - 완료 화면 (USR-SCR-005)

사용 역할자 : 신규 회원 (회원가입 완료 직후)

관련 메뉴 : (회원가입 프로세스 종료)

기능 1: 가입 완료 안내 및 네비게이션

- 관련 요구사항: USR-REQ-002 (회원 가입 완료 후 상태)
- 입력 정보: 없음 (이전 단계에서 가입 성공 시 자동으로 이동됨).
- 출력 정보:
    - "회원가입이 성공적으로 완료되었습니다." 와 같은 환영 메시지.
    - 가입된 사용자 이름 표시 (예: "홍길동님, 환영합니다!").
    - '로그인 하러 가기' 버튼.

    - '메인 페이지로 이동' 버튼/링크.
- 프로세스 설명:
    1. 회원 정보 입력 화면에서 가입 처리가 성공하면 이 화면으로 전환된다.
    2. 화면에는 가입 완료를 알리는 정적 메시지와 로그인 또는 메인 페이지로 이동할 수 있는 버튼/링크가 표시된다.
    3. 사용자는 버튼/링크를 클릭하여 다음 행동을 선택한다.
- 호출 API: 없음
- Mockup 데이터: 없음

기능 2: 로그인 화면 이동

- 입력 정보: '로그인 하러 가기' 버튼 클릭.
- 출력 정보: 사용자 로그인 화면 (USR-SCR-001)으로 페이지 전환.
- 프로세스 설명: 사용자가 버튼을 클릭하면 클라이언트 라우터가 로그인 페이지 경로로 이동시킨다.
- 호출 API: 없음
- Mockup 데이터: 없음

기능 3: 메인 페이지 이동

- 입력 정보: '메인 페이지로 이동' 버튼/링크 클릭.
- 출력 정보: 시스템의 메인 페이지(대시보드)로 페이지 전환. (로그인 상태는 아님)
- 프로세스 설명: 사용자가 버튼/링크를 클릭하면 클라이언트 라우터가 메인 페이지 경로로 이동시킨다.
- 호출 API: 없음
- Mockup 데이터: 없음

예외 처리 및 유의 사항

- 이 화면은 가입 절차가 성공적으로 끝났음을 명확히 인지시키는 역할을 한다.
- 사용자가 다음 단계로 쉽게 이동할 수 있도록 명확한 CTA(Call to Action) 버튼을 제공해야 한다.

아이디 찾기 화면 (USR-SCR-006)

사용 역할자 : 아이디를 분실한 회원

관련 메뉴 : 로그인 화면 내 '아이디 찾기' 링크

기능 1: 아이디 찾기 (본인 인증 기반)

- 관련 요구사항: USR-REQ-004 (아이디 찾기)
- 입력 정보:
    - 이름: 사용자 입력 텍스트 필드. 필수.
    - 생년월일: 사용자 입력 필드 (YYYYMMDD). 필수.
    - 성별: 라디오 버튼 (남/여). 필수.
    - 통신사: 드롭다운 선택. 필수.

    - 휴대폰 번호: 사용자 입력 텍스트 필드. 필수.
    - '인증번호 요청' 버튼 클릭.
    - 인증번호: SMS 수신 번호 입력 필드. 필수.
    - '아이디 찾기' 또는 '확인' 버튼 클릭.
- 출력 정보:
    - 인증 성공 및 아이디 존재 시: "회원님의 아이디는 [user_id] 입니다." 메시지 표시. 아이디 일부 마스킹 처리 가능 (예: `test****`). '로그인 하러 가기' 버튼 표시.
    - 인증 성공 했으나 해당 정보로 가입된 아이디가 없을 시: "입력하신 정보로 가입된 아이디가 없습니다." 메시지 표시. '회원가입' 버튼 표시 가능.
    - 인증 실패 시: 본인 인증 화면(USR-SCR-003)의 실패 메시지와 동일하게 표시 (정보 불일치, 번호 불일치, 시간 초과 등).
- 프로세스 설명:
    1. 사용자가 아이디를 찾기 위해 자신의 본인 인증 정보(이름, 생년월일, 성별, 통신사, 휴대폰 번호)를 입력한다.
    2. '인증번호 요청' 버튼 클릭 시, 회원가입 시의 본인 인증 프로세스(USR-SCR-003 기능 1의 2~5단계)와 동일하게 인증번호 발송 및 입력 대기 상태가 된다.
    3. 사용자가 인증번호를 입력하고 '아이디 찾기' 버튼을 클릭한다.
    4. 클라이언트는 인증번호와 관련 정보를 아이디 찾기 API로 전송한다.
    5. 백엔드는 외부 본인 인증 서비스로 인증번호를 검증하고 성공 시 CI/DI 값을 얻는다.
    6. 백엔드는 얻은 CI/DI 값으로 시스템 DB의 User 테이블을 조회하여 일치하는 사용자가 있는지 확인한다.
    7. 일치하는 사용자가 있으면 해당 사용자의 아이디(loginId 또는 email)를 포함한 성공 응답을 보낸다.
    8. 일치하는 사용자가 없으면 아이디 없음 응답을 보낸다.
    9. 인증 자체가 실패하면 해당 오류 응답을 보낸다.
    10. 클라이언트는 응답에 따라 아이디를 표시하거나, 아이디 없음 메시지, 또는 인증 실패 메시지를 보여준다.
- 호출 API 정보:
    - (인증 요청/확인 API는 USR-SCR-003과 동일하거나 유사한 API 사용 가능, 목적만 분리)
    - 아이디 찾기 확인 (인증 확인 후 처리):
        - HTTP 메서드: POST
        - API 엔드포인트: `/api/v1/auth/find-id`
        - API 목적: 본인 인증 정보(CI/DI)를 기반으로 사용자 아이디 조회
        - 요청 파라미터: (본인 인증 성공 후 획득한 정보 전달 방식 필요 - 예: 서버 세션 또는 암호화된 토큰)

```
{
  "verificationToken": "..." // 본인 인증 성공 후 백엔드가 발급한 임시 토큰 or CI/DI 직접 전달 (보안 고려)
  // 또는 CI/DI 값을 직접 포함
  // "ci": "..."
}
```

        - 응답 데이터 (성공): `{"success": true, "loginId": "masked_user_id"}`
        - 응답 데이터 (아이디 없음): `{"success": false, "errorCode": "USER_NOT_FOUND", "message": "가입된 아이디 없음"}`
        - 응답 데이터 (인증 실패): `{"success": false, "errorCode": "VERIFICATION_FAILED", "message": "인증 실패"}`
- Mockup 데이터 예시:
    - 입력: 본인 인증 정보 입력 및 인증번호 입력
    - 출력 (성공): "회원님의 아이디는 test**** 입니다."

- 출력 (실패): "입력하신 정보로 가입된 아이디가 없습니다."

기능 2: 로그인 화면 이동

- 입력 정보: '로그인 하러 가기' 버튼 클릭 (아이디 찾기 성공 후 표시됨).
- 출력 정보: 사용자 로그인 화면 (USR-SCR-001)으로 페이지 전환.
- 프로세스 설명: 사용자가 버튼을 클릭하면 클라이언트 라우터가 로그인 페이지 경로로 이동시킨다.
- 호출 API: 없음
- Mockup 데이터: 없음

예외 처리 및 유의 사항

- 아이디 찾기 역시 본인 인증 절차를 거치므로, USR-SCR-003의 유의사항이 동일하게 적용된다.
- 조회된 아이디를 전체 다 보여줄지, 일부 마스킹 처리할지는 보안 정책에 따라 결정한다.
- 아이디 찾기 시도 횟수 제한을 두는 것이 좋다.

비밀번호 찾기 (재설정) - 정보 입력 화면 (USR-SCR-007)

사용 역할자 : 비밀번호를 분실한 회원

관련 메뉴 : 로그인 화면 내 '비밀번호 찾기' 링크

기능 1: 비밀번호 재설정을 위한 사용자 확인 (본인 인증)

- 관련 요구사항: USR-REQ-005 (패스워드 찾기)
- 입력 정보:
  - 아이디: 사용자 입력 텍스트 필드. 필수.
  - 이름: 사용자 입력 텍스트 필드. 필수.
  - 생년월일: 사용자 입력 필드 (YYYYMMDD). 필수.
  - 성별: 라디오 버튼 (남/여). 필수.
  - 통신사: 드롭다운 선택. 필수.
  - 휴대폰 번호: 사용자 입력 텍스트 필드. 필수.
  - '인증번호 요청' 버튼 클릭.
  - 인증번호: SMS 수신 번호 입력 필드. 필수.
  - '확인' 또는 '다음' 버튼 클릭.
- 출력 정보:
  - 인증 성공 시: 비밀번호 재설정 화면 (USR-SCR-008)으로 페이지 전환. 사용자를 식별할 수 있는 임시 토큰 또는 정보가 다음 단계로 전달되어야 함.
  - 입력한 아이디와 인증된 사용자 정보(CI/DI 기준)가 일치하지 않을 시: "입력하신 아이디와 사용자 정보가 일치하지 않습니다." 메시지 표시.
  - 아이디 자체가 존재하지 않을 시: "존재하지 않는 아이디입니다." 메시지 표시.
  - 인증 실패 시: 아이디 찾기 화면(USR-SCR-006)의 실패 메시지와 동일하게 표시.
- 프로세스 설명:
  1. 사용자가 비밀번호를 재설정할 아이디와 자신의 본인 인증 정보를 입력한다.

2. '인증번호 요청' -> 인증번호 입력 -> '확인' 과정은 아이디 찾기(USR-SCR-006 기능 1)와 동일하게 진행하여 본인 인증을 완료하고 CI/DI 값을 얻는다.
3. 백엔드는 얻어진 CI/DI 값과 입력된 아이디(loginId)가 DB에서 실제로 매칭되는 사용자인지 검증한다.
4. 검증 성공 시: 비밀번호 재설정을 진행할 수 있음을 확인하는 성공 응답과 함께, 다음 단계(비밀번호 변경)에서 사용자를 식별할 임시 토큰(Password Reset Token)을 발급하여 클라이언트에 전달한다.
5. 아이디와 인증 정보 불일치, 아이디 부재, 인증 실패 등의 경우 해당 오류 응답을 전달한다.
6. 클라이언트는 성공 시 발급받은 토큰을 가지고 비밀번호 재설정 화면으로 이동하거나, 실패 시 오류 메시지를 표시한다.

- 호출 API 정보:
  - (인증 요청/확인 API는 USR-SCR-003/006과 동일/유사 API 사용 가능)
  - 비밀번호 재설정 요청 (사용자 확인 및 토큰 발급):
    - HTTP 메서드: POST
    - API 엔드포인트: `/api/v1/auth/request-password-reset`
    - API 목적: 아이디와 본인 인증 정보 일치 여부 확인 후 비밀번호 재설정 토큰 발급
    - 요청 파라미터:

```
{
  "loginId": "string",
  "verificationToken": "..." // 본인 인증 성공 후 받은 임시 토큰 or CI/DI 직접 전달
}
```

    - 응답 데이터 (성공): `{"success": true, "passwordResetToken": "..."}` (짧은 유효 기간 가짐)
    - 응답 데이터 (실패): `{"success": false, "errorCode": "USER_MISMATCH", "message": "아이디와 사용자 정보 불일치"}` 또는 `{"success": false, "errorCode": "USER_NOT_FOUND", ...}` 등
- Mockup 데이터 예시:
  - 입력: 아이디, 본인 인증 정보, 인증번호
  - 출력 (성공): `{"success": true, "passwordResetToken": "SOME_SECURE_TOKEN"}`
  - 출력 (실패): "입력하신 아이디와 사용자 정보가 일치하지 않습니다."

예외 처리 및 유의 사항

- 비밀번호 재설정은 민감한 기능이므로 반드시 본인 인증 절차를 거쳐야 한다. 이메일 기반 재설정도 가능하나, 본인 인증이 더 안전하다.
- 재설정 토큰은 짧은 유효 시간(예: 10분)을 가져야 하며, 한 번 사용되면 만료되어야 한다.

비밀번호 찾기 (재설정) - 새 비밀번호 입력 화면 (USR-SCR-008)

사용 역할자 : 비밀번호 재설정 중인 회원 (본인 인증 완료)

관련 메뉴 : (비밀번호 찾기 프로세스 중)

기능 1: 새 비밀번호 설정

- 관련 요구사항: USR-REQ-005 (패스워드 찾기 - 최종 재설정 단계)
- 입력 정보:
  - 새 비밀번호: 사용자 입력 비밀번호 필드. 필수. (비밀번호 규칙 적용)
  - 새 비밀번호 확인: 사용자 입력 비밀번호 필드. 필수. (새 비밀번호와 일치 여부 확인)
  - '비밀번호 변경' 또는 '확인' 버튼 클릭.
  - (Hidden Input) 비밀번호 재설정 토큰: 이전 화면(USR-SCR-007)에서 발급받은 토큰.
- 출력 정보:
  - 비밀번호 변경 성공 시: 비밀번호 재설정 완료 화면 (USR-SCR-009)으로 페이지 전환.
  - 비밀번호 규칙 미준수 시: 규칙 안내 메시지 표시.
  - 비밀번호 불일치 시: "비밀번호가 일치하지 않습니다." 메시지 표시.
  - 재설정 토큰 만료/무효 시: "비밀번호를 변경할 수 있는 시간이 초과되었거나 요청이 유효하지 않습니다. 다시 시도해주세요." 메시지 표시 후 로그인 또는 비밀번호 찾기 첫 단계로 이동 유도.
  - 변경 실패 시 (DB 오류 등): "비밀번호 변경 중 오류가 발생했습니다." 메시지 표시.
- 프로세스 설명:
  1. 사용자는 새 비밀번호와 확인용 비밀번호를 입력한다. 클라이언트는 실시간으로 일치 여부 및 규칙 준수 여부를 검사한다.
  2. 사용자가 '비밀번호 변경' 버튼을 클릭한다.
  3. 클라이언트는 새 비밀번호와 함께 이전 단계에서 받은 비밀번호 재설정 토큰을 API로 전송한다.
  4. 백엔드는 전달받은 토큰의 유효성(만료 여부, 위변조 여부)을 검증한다.
  5. 토큰이 유효하면, 백엔드는 해당 토큰과 연결된 사용자의 비밀번호를 입력받은 새 비밀번호로 업데이트한다(DB에 해싱하여 저장). 사용된 토큰은 만료시킨다.
  6. 변경 성공/실패 결과를 클라이언트에 응답한다.
  7. 클라이언트는 응답에 따라 완료 화면으로 이동하거나 오류 메시지를 표시한다.
- 호출 API 정보:
  - HTTP 메서드: POST
  - API 엔드포인트: `/api/v1/auth/reset-password`
  - API 목적: 비밀번호 재설정 토큰 검증 및 새 비밀번호 설정
  - 요청 파라미터:

```
{
  "passwordResetToken": "string",
  "newPassword": "string"
}
```

  - 응답 데이터: `{"success": true}` 또는 `{"success": false, "errorCode": "INVALID_TOKEN", "message": "유효하지 않은 요청"}` 등
- Mockup 데이터 예시:
  - 입력: `{"passwordResetToken": "SOME_SECURE_TOKEN", "newPassword": "newPassword123!"}`
  - 출력 (성공): `{"success": true}`

    - 출력 (실패): `{"success": false, "errorCode": "INVALID_TOKEN", "message": "유효하지 않은 요청입니다."}`

예외 처리 및 유의 사항

- 새 비밀번호는 기존 비밀번호와 동일하지 않도록 검증하는 것이 좋다 (선택 사항).
- 비밀번호 규칙은 회원가입 시와 동일하게 적용해야 한다.

비밀번호 찾기 (재설정) - 완료 화면 (USR-SCR-009)

사용 역할자 : 비밀번호 재설정을 완료한 회원

관련 메뉴 : (비밀번호 찾기 프로세스 종료)

기능 1: 비밀번호 변경 완료 안내 및 네비게이션

- 관련 요구사항: USR-REQ-005 (패스워드 찾기 완료 후 상태)
- 입력 정보: 없음 (이전 단계에서 변경 성공 시 자동으로 이동됨).
- 출력 정보:
    - "비밀번호가 성공적으로 변경되었습니다." 메시지.
    - '로그인 하러 가기' 버튼.
- 프로세스 설명:
    1. 새 비밀번호 입력 화면에서 변경 처리가 성공하면 이 화면으로 전환된다.
    2. 화면에는 변경 완료를 알리는 정적 메시지와 로그인 페이지로 이동할 수 있는 버튼이 표시된다.
    3. 사용자는 버튼을 클릭하여 로그인 화면으로 이동한다.
- 호출 API: 없음
- Mockup 데이터: 없음

기능 2: 로그인 화면 이동

- 입력 정보: '로그인 하러 가기' 버튼 클릭.
- 출력 정보: 사용자 로그인 화면 (USR-SCR-001)으로 페이지 전환.
- 프로세스 설명: 사용자가 버튼을 클릭하면 클라이언트 라우터가 로그인 페이지 경로로 이동시킨다.
- 호출 API: 없음
- Mockup 데이터: 없음

예외 처리 및 유의 사항

- 사용자가 비밀번호 변경이 완료되었음을 명확히 인지하고, 다음 행동(로그인)으로 자연스럽게 이어지도록 안내한다.

USR 모듈의 사용자 측면 핵심 기능(로그인, 회원가입, 아이디/비밀번호 찾기)에 대한 화면 설계를 완료했습니다. 회원 정보 조회/수정, 탈퇴 등은 '마이페이지' 관련 화면 설계 시 USR 요구사항과 연계하여 작성하겠습니다. 관리자 기능(USR-REQ-010 ~ 013)은 별도의 관리자 시스템 화면 설계 시 포함하겠습니다. 다음 모듈 설계를 진행할까요?