

[ 초보자 및 학생들을 위한 천체 관측 교육 애플리케이션 ]

---

## 빌드 개발 계획서

---

2025년 11월 11일

문서번호 : Byeoldori\_Doc\_06

소 속 : 충북대학교 천문우주학과

팀 명 : 천문소프트

팀 원 : 서범수, 윤태영, 김채영

교 수 : 조오현 교수님

## 목 차

1. 시스템 개요 -----	1
1.1 시스템명 -----	1
1.2 시스템 설명 -----	1
2. 요구기능 요약표 -----	2
3. 1차 빌드 개발 대상 -----	3
4. 2차 빌드 개발 대상 -----	6
5. 3차 빌드 개발 대상 -----	10
6. 4차 빌드 개발 대상 -----	13
7. 5차 빌드 개발 대상 -----	17
8. 6차 빌드 개발 대상 -----	21
9. 7차 빌드 개발 대상 -----	24
10. 개발 환경 -----	27
11. 기타 사항 -----	28

# 1. 시스템 개요

## 1.1 소프트웨어명

본 시스템의 명칭은 '초보자 및 학생들을 위한 천체 관측 교육 애플리케이션' 이다.

## 1.2 시스템 설명

본 시스템은 사용자에게 신뢰할 수 있는 천문학 정보를 기반으로 천체 관측의 흥미와 교육적 가치를 동시에 제공하는 모바일 애플리케이션이다. 특히, 천체 관측을 한번도 해보지 않거나 익숙하지 않은 초보자 및 학생들을 대상으로 하여금 천체 관측에 대한 두려움을 해소시키고, 천문학에 쉽게 입문하도록 지원하는 것이 핵심 목표이다.

기존에 존재하는 Star Walk 2, SkySafari, Stellarium, 별해는 밤과 같은 천체 관측 앱의 기능을 참고하였으며, '별도리'라는 마스코트 캐릭터가 관측을 지도해주어 사용자에게 흥미를 유발시키고, 천체에 대한 기초 지식을 스토리텔링 방식으로 전달함으로써 교육적인 요소를 추가하여 기존 앱과의 차별성을 두었다.

개발 대상 소프트웨어가 처한 환경과 상호작용은 다음과 같다.

### - 환경

운영 환경: 모바일 플랫폼 Android 환경에서 실행

네트워크 환경: 실시간으로 천체 데이터를 동기화하기 위해 안정적인 인터넷 연결(최소 3Mbps) 필요

### - 외부 시스템과의 상호작용

지도 연동: Naver Map API를 통해 주변의 관측 장소를 시각화하고, 기상청 API를 통해 해당 장소의 정확한 날씨 정보를 가져와 관측 적합도를 판단하고 최적의 관측 장소를 사용자에게 제공

SNS 연동: 구글 계정으로 연동이 가능하고, 사용자가 관측하여 작성한 관측 기록을 소셜 미디어 (Instagram 등)에 공유

사용자 위치: GPS, 자이로스코프, 가속도계를 사용하여 사용자의 방향과 위치 감지

천체 데이터 연동: 센서 데이터와 NASA API, OpenSky API를 통해 최신 천체 정보를 실시간 반영

## 2. 요구기능 요약표

식별자	요구 기능	중요도	우선순위	크기	메핑된 NF식별자
FR-100	관측 교육 프로그램	상	6	4	NF-102, NF-103, NF-202
FR-200	마스코트 캐릭터	중	4	5	NF-102, NF-303
FR-300	별지도	상	5	5	NF-101, NF-103, NF-106, NF-301, NF-303
FR-400	관측지 정보 제공	중	1	3	NF-102, NF-103, NF-304
FR-500	관측 커뮤니티	하	2	2	NF-201, NF-202
FR-600	마이페이지	하	3	1	NF-104, NF-201, NF-202

### 3. 1차 빌드 개발 대상

#### 3.1 선정 기능명

선정 기능명은 다음과 같다.

**FR-400: 관측지 정보**

**UIR-400: 관측지 화면**

**UIR-4100: 관측지 정보 제공 화면**

##### 3.1.1 선정 기능 요구사항

###### **FR-400: 관측지 정보 제공**

FR-401: 정보의 고려 사항 - 정보 제공 시 주변의 지리정보, 온도, 습도, 광량, 날씨를 고려해야 한다.

FR-402: 관측 장소 제공 - 주변 환경이나 지리정보를 고려하여 관측 장소를 제시해야 한다. 이때 관측 장소는 유명 장소-이미 검증되어 있는곳 ex) 천문대, 안반데기-와 일반 장소- ex) 주변의 공원- 두가지가 있다.

FR-403: 기상청 API 호출 - 기상청에서 초단기, 단기, 중기 예보를 불러와 실시간 날씨와 10일 후 까지의 온도, 습도, 강수 확률, 하늘 상태 예보를 제공해주어야 한다.

FR-404: 관측 적합도 - 관측장소, 날씨 정보를 고려하여 시간대 별로 관측 적합도를 제공해주어 사용자는 관측에 적합한 시간이나 날짜를 확인할 수 있어야 한다.

FR-405: 기상 정보 최신화 - 우리나라의 기상정보들은 쉽게 바뀌므로, 10분간격 또는 기상청이 제시하는 시간마다 정보를 최신화 하여야한다.

###### **UIR-400: 관측지 화면**

UIR-401: 지도 - Naver Map에서 지정된 관측지 또는 임의의 장소를 직접 찾아보거나 검색할 수 있다.

UIR-402: 현재 위치 – 현재 위치 버튼을 클릭하여 사용자 현재 위치를 지도에 표시해준다.

UIR-403: 광공해 오버레이 – 현재 위치의 광공해 수치가 얼마나 되는지 표시해주는 기능으로 광공해 지도를 클릭해 광공해 수치를 지도에 오버레이 한다.

UIR-404: NaverMap - 네이버 지도에서 추천 관측지나 인기 관측지 항목들을 마커 형태로 지도에 표시 한다.

###### **UIR-4100: 관측지 정보 제공 화면**

UIR-4101: 관측지 기본 정보 표시 - 사용자가 선택한 관측지의 대표 사진, 이름, 주소, 현재 날씨, 관측 리뷰 수 및 평점, 관측 적합도를 보여준다.

UIR-4102: 관측지 날씨 확인 - 기상청 API를 불러와 실시간으로 화면에 기상 상태를 받아와 사용자에게 실황 및 10일 뒤 예보까지 제공 한다.

UIR-4103: 관측지 관련 후기 네비게이션 - 정보 제공 화면 하단 또는 주요 액션 영역에 해당 관측지의 관측 후기 게시글을 리스트로 보여준다.

### 3.2 태스크 목록

태스크 번호	태스크 명	구현 기술	담당자	소요기간	비고
T-101	기상청 API 호출	DFS 격자 변환(lat/lon→x/y), 스케줄러 타임아웃/재시도/백오프, KST 시간 보정	윤태영	7d	-
T-102	예보 별 관측 적합도 반환	적합도 알고리즘 구현 스코어에 가중치 부여	윤태영	4d	T-101 선행
T-103	관측 장소 제공	Spring MVC + JPA(Hibernate)+MySQL	윤태영	3d	-
T-104	관측지 상세 조회	Spring MVC + JPA(Hibernate)+MySQL	윤태영	5d	T-103 선행
T-105	Naver map API 호출	Jetpack Compose + Naver Maps SDK, ViewModel/Flow 상태관리, 위치 권한	김채영	10d	
T-106	지도에 광공해 오버레이	전지구 광공해 데이터 전처리(GeoTIFF), 대한민국 범위 프로파일 수정(좌표 변환 용도), gamma보정	김채영	4d	
T-107	지도 UI	Navermap Reverse Geocoding으로 현재 위치 이동, 검색 이동, 관측지 API 연동해서 마커 표시	김채영	5d	
T-108	관측지 정보 출력 UI	관측지·예보 API 바인딩, 관측지 정보, 날씨 정보, 관측적합도 표시	김채영	7d	
T-109	관측지 화면 I.A 작성	IA(화면/상태/흐름), Figma 산출물	김채영	7d	
T-110	Observation API 명세서 작성	springdoc-openapi, 요청/응답 예시 JSON, 오류 코드 표준화, DTO 정의	윤태영	7d	
T-111	Observation Controller 연동	Retrofit2 + OkHttp Interceptor, DTO 매핑, 에러 핸들링,	김채영	5d	

### 3.3 종료일

2025년 10월 11일(토)

- 해당 빌드는 수업 일정과 팀의 프로젝트 개발 일정을 고려하여 10월 11일을 종료일로 설정한다.

### 3.4 중간 점검일

2025년 8월 24일

중간 점검에서는 아래와 같은 점검 항목들에 대해 잘 부합하는지에 대해 점검한다.

#### 요구사항 추적성 확인

- 설계 단계에서 정의한 요구사항과의 추적성이 일치하는가?

#### 문서와 기능 연결 확인

- IA 흐름과 지도·관측지 UI 플로우가 일치하는가?
- Observation API 명세서의 요청/응답/상태코드가 구현과 일치하는가?

#### 기능 동작 확인

- 기상청 API(초단기/단기/중기) 선택·호출 로직이 설계와 일치하는가?
- 위도/경도 → DFS 격자 변환이 정확히 적용되는가?
- KST 기준 시간 보정 및 발표/예보 시각 계산이 정확한가?
- 광공해 오버레이(타일/히트맵)가 정확히 표기되고 성능이 적절한가?
- Navermap API 연동시 사용자의 현재 위치에서 정확히 로드가 되는가?
- Navermap 시작 시, 마커 정보에 해당하는 위도, 경도가 정확히 지도에 표시되는가?
- 마커 클릭시 정확한 위치 정보를 가져오는가?
- 프론트 연동 시 지도/관측지/예보 호출이 정상 동작하는가?

#### 예외 처리 확인

- 한국 외 좌표 입력 시 418 규칙이 일관되게 적용되는가?

## 4. 2차 빌드 개발 대상

### 4.1 선정 기능명

선정 기능명은 다음과 같다.

**UIR-300: 별지도 화면**

**FR-300: 별지도 기능**

#### 4.1.1 선정 기능 요구사항

##### **UIR-300: 별지도 화면**

UIR-301: 별지도 화면 구성 - 별자리와 주요 천체가 시각적으로 구분되어 표시되어야 한다.

UIR-302: 가이드 연동 시야 - 교육 가이드의 진행 방향에 따라 자동으로 시야가 이동하며 해당 방향의 별지도가 렌더링되어야 한다.

UIR-303: 천체 선택 정보 표시 - 사용자가 천체를 터치하면 이름, 밝기, 거리 등의 기본 정보가 팝업 형태로 표시되어야 한다.

UIR-304: 저휘도 - 사용자는 애플리케이션을 어두운곳에서 사용하므로, 어두운 테마 모습을 가져야한다.

UIR-305: 글꼴 크기 고정 - 별지도 내의 모든 텍스트는 줌 인/아웃과 관계없이 고정된 크기(예: 14sp)로 유지되어야 한다.

UIR-306: 적경·적위 격자 표시 - 적경(RA)과 적위(Dec) 격자를 켜거나 끌 수 있는 토글 버튼이 제공되어야 한다.

UIR-307: 방위각·고도 격자 표시 - 방위각(Azimuth)과 고도(Altitude) 격자를 표시하거나 숨길 수 있는 UI 옵션을 제공해야 한다.

UIR-308: 동서남북 방향 표시 - 별지도 상단 또는 주변에 N, E, S, W 방향이 시각적으로 표시되어야 한다.

UIR-309: 별자리 선 토글 - 별자리 연결선을 켜거나 끌 수 있는 토글 버튼을 제공해야 한다.

UIR-310: 별자리 그림 토글 - 별자리 그림을 켜거나 끌 수 있는 토글 버튼을 제공해야 한다.

UIR-311: 대기효과 토글 - 대기 산란 효과(Atmosphere Rendering)를 On/Off할 수 있는 설정을 제공해야 한다.

UIR-312: 심천체 표시 옵션 - 은하, 성운, 성단 등 심천체(Deep Sky Object)의 표시 여부를 설정할 수 있어야 한다.

UIR-313: 관측 위치 조정 - 사용자는 지도나 위도·경도 입력을 통해 관측 위치를 직접 수정할 수 있어야 한다.

UIR-314: 시선 트래킹 토글 - 시선 트래킹 기능을 켜면 기기의 방향(자이로 기준)에 따라 시야가 실시간으로 움직여야 한다.

UIR-315: 시간 조정 인터페이스 - 사용자가 원하는 날짜·시간을 선택하여 그 시점의 하늘을 시각적으로 확인할 수 있는 UI가 제공되어야 한다.

UIR-316: 천체 상세보기 인터페이스 - 사용자가 상세정보보기를 클릭하면, 해당하는 천체의 상세정보(출몰시각, 적경, 적위, 기타 등)와 연동된 관측후기 및 교육콘텐츠 내용 출력

### **FR-300: 별지도 기능**

FR-301: 3D 별지도 렌더링 - 구면좌표계를 기반으로 별자리, 천체, 격자, 심천체를 3D 환경에서 렌더링해야 한다.

FR-302: 센서 기반 정합 - GPS, 자이로, 자기장 센서 데이터를 이용하여 사용자의 실제 방향과 별지도의 방향을 실시간으로 일치시켜야 한다.

FR-303: 천체 선택 처리 - 터치 좌표로 선택된 천체를 식별하고 해당 정보를 UI에 전달하여 표시해야 한다.

FR-304: 정합성 검증 - 렌더링된 별자리 위치는 실제 하늘과  $\pm 1^\circ$  이내의 오차 범위에서 정합되어야 한다.

FR-305: 관측 위치 재설정 - 사용자가 지정한 위도·경도를 기준으로 천구 중심 좌표를 재계산하고 반영해야 한다.

FR-306: 천구 경사 보정 - 위도에 따른 천구의 기울기를 계산하여 렌더링에 반영해야 한다.

FR-307: 천체 검색 기능 - 천체명을 입력받아 해당 천체의 적경·적위 좌표를 계산하고 방향과 각거리를 시각적으로 표시해야 한다.

FR-308: 핀치 줌 제스처 - 두 손가락 제스처를 감지하여 별지도를 부드럽게 확대 및 축소해야 한다.

FR-309: 시간 변경 처리 - 사용자가 선택한 날짜·시간을 기준으로 천체의 위치를 계산하고 렌더링 결과를 갱신해야 한다.

FR-310: 별 등급 필터링 - 설정된 등급 한계(예: 6등성 이하)에 따라 렌더링할 별의 수를 제한해야 한다.

FR-311: 대기효과 렌더링 - 대기 산란과 굴절 효과를 세이더 단계에서 계산하며, 설정에 따라 효과를 켜거나 꺼야 한다.

FR-312: 심천체 렌더링 - DSO(은하, 성단, 성운 등) 데이터를 기반으로 심천체를 별지도에 표시해야 한다.

FR-313: 시선 트래킹 처리 - 자이로 데이터를 실시간으로 수집해 View Matrix를 갱신하여 시야를 조정해야 한다.

FR-314: 천체 고정 기능 - 선택된 천체를 화면 중앙에 고정(Pin)하여 시점 이동 시에도 중앙에 유지되어야 한다.

FR-315: 천체 출몰 시각 계산 - 관측 위치와 날짜를 기준으로 천체의 출현(Rise)-소멸(Set) 시각을 계산해야 한다.

FR-316: 렌더링 주기 동기화 - 센서 및 시간 데이터를 일정 주기(예: 30Hz)로 동기화하여 부드러운 렌더링을 유지해야 한다.

## 4.2 태스크 목록

태스크 번호	태스크 명	구현 기술	담당자	소요기 간	비고
T-201	오픈소스 분석 및 기술 검토	Staroid(Java), Stellarium(C++), OpenGL ES	서범수	14d	-
T-202	별지도 오픈소스 선정 및 연결 (또는 자체 구현)	-	서범수	14d	T-401
T-203	천체 렌더링 구현(별·행성·위성)	-	서범수	14d	T-402
T-204	별자리 렌더링 구현(선·그림)	-	서범수	7d	T-403
T-205	심천체 렌더링 구현(은하·성운·성단)	-	서범수	3d	-
T-206	좌표계 구현 (적도, 지평좌표계)	수학 모델링(Matrix Transform)	서범수	3d	-
T-207	대기효과 구현	-	서범수	7d	T-403
T-208	천체 검색 기능 구현	-	서범수	7d	T-403, T-404
T-209	관측 위치 조정 기능 구현	Kotlin, GPS API, Coordinate Transform	서범수	7d	-
T-210	시간 조정 기능 구현	Kotlin, Time API, Julian Date 계산	서범수	7d	-
T-211	시선 트래킹 기능 구현	Android SensorManager, Gyroscope, Coroutine	서범수	3d	-
T-212	SkyMapController 모듈화	Kotlin Singleton, JS Wrapper, Hilt DI	서범수	3d	이전 전체
T-213	천체 상세정보 구현	Koltin, Hilt DI	서범수	3d	T-408
T-213	천체명 번역 모듈	Kotlin, JSON 매팅, 데이터 로컬라이제이션	서범수	1d	-
T-214	문서 작성	HancomDocs	서범수	1d	-

## 4.3 종료일

2025년 10월 29일(수)

- 해당 빌드는 수업 일정과 팀의 프로젝트 개발 일정을 고려하여 9월 23일을 종료일로 설정한다.

## 4.4 중간 점검일

2025년 10월 15일(수)

- 중간 점검에서는 별지도 모듈의 구현이 설계서 및 요구사항과 부합하는지를 중심으로 검증하며, 기능별 연동 및 UI 구조의 완성도를 점검한다.

주요 점검 항목은 다음과 같다.

### 요구사항 추적성 확인

- UIR-300~315 및 FR-300~315에 정의된 세부 요구사항이 모두 반영되었는가?
- 기능 명세서 상의 설계 흐름과 실제 구현 흐름(렌더링, 센서, UI)이 일치하는가?

### 문서와 기능 연결 확인

- SkyMap 구조 설계서 내의 데이터 흐름도와 실제 WebView–AppBridge 연동 구조가 일치하는가?
- StellariumController의 명세서 내 함수 구조가 실제 Kotlin 코드와 일치하는가?
- AppBridge 통신 시 JS ↔ Kotlin 간 메시지 규격(JSON 포맷, 함수명, 응답 구조)이 설계와 동일한가?

### 기능 동작 확인

- Stellarium Web Engine이 WebView 환경에서 정상적으로 구동되는가?
- 자이로센서 기반 시선 트래킹이 실시간으로 반응하며, 오차가  $\pm 1^\circ$  이내로 유지되는가?
- 시간 변경(시각/날짜) 시 천체 위치 계산이 정확히 반영되는가?
- 천체 선택 시 NAME·적경(RA)·적위(Dec)·밝기·거리 등의 정보가 정상적으로 출력되는가?
- 핀치 제스처(FOV 제어)가 렌더링 해상도 저하 없이 부드럽게 작동하는가?
- 격자(적경/적위, 방위각/고도) 및 별자리 선·그림 툴이 정상적으로 반응하는가?
- 저휘도 모드 및 심천체 표시 옵션이 UI 툴 상태에 따라 즉시 적용되는가?
- 천체 검색 기능에서 입력한 이름이 번역 모듈을 통해 정확히 탐색되는가?
- StellariumController를 통한 시야 이동(적경·적위 기반)이 정상 작동하는가?
- Education Mode 내에서 Live2D 캐릭터와 별지도 간 동기화(시야, 시간, 위치)가 유지되는가?

## 5. 3차 빌드 개발 대상

### 5.1 선정 기능명

선정 기능명은 다음과 같다.

**FR-600: 마이페이지**

**UIR-6300: 로그인/회원가입 페이지**

#### 5.1.1 선정 기능 요구사항

**FR-600: 마이페이지**

FR-601: 회원가입 - 사용자 개개인은 개인정보를 입력하여 회원가입을 할 수 있고, 이메일이 중복되지 않을 경우 여러 계정을 생성할 수 있다.

FR-602: 로그인 - 생성한 계정으로 로그인 할 수 있어야 한다.

FR-603: 이메일 찾기 - 사용자가 비밀번호를 잊어버렸을 경우 본인 인증을 통해 가입한 계정을 확인할 수 있다.

FR-604: 비밀번호 찾기 - 사용자가 비밀번호를 잊어버렸을 경우 본인 인증을 통해 이메일로 임시 비밀번호를 발급받을 수 있어야 한다.

FR-605: 자동 로그인 - 리프레시 토큰이 유효한 경우, 앱을 실행해도 로그인이 유지되어야 한다.

**UIR-6300: 로그인/회원가입 페이지**

UIR-6301: 로그인 화면 - 이메일, 비밀번호 입력 필드와 로그인 버튼, 하단에는 '아이디 찾기', '비밀번호 찾기', '회원가입' 버튼이 배치된다. 로그인 실패 시 에러 메시지를 표시해야 한다.

UIR-6302: 비밀번호 찾기 화면 - 본인 인증 절차와 임시 비밀번호 발급 절차를 포함한다. 인증 성공 시 이메일로 임시 비밀번호를 전송했다는 안내 메시지를 표시한다.

UIR-6303: 이메일 찾기 화면 - 이름 및 본인 인증 정보를 입력하여 가입된 이메일을 확인할 수 있는 UI를 제공한다. 찾은 이메일은 마스킹 처리하여 보여주어야 한다.

UIR-6304: 회원가입 화면 - 이메일, 비밀번호, 이름 등 필수 입력 필드와 약관 동의 체크박스를 포함한다. 이메일 중복 체크 기능 및 인증번호 발송/확인 절차를 포함한다. 회원가입 완료 후 자동으로 로그인 화면 또는 메인 화면으로 이동한다.

UIR-6305: 약관 동의 화면 - 필수 약관 및 선택 약관을 구분하여 표시하며, 사용자는 약관을 확인 후 체크박스로 동의할 수 있어야 한다. 모든 필수 약관에 동의해야 다음 단계로 진행 가능하다.

UIR-6306: 이메일 인증 화면 - 입력한 이메일 주소로 인증 코드를 전송하고, 해당 코드를 입력받는 UI를 제공한다. 인증 성공 시 회원가입이 완료된다.

UIR-6307: 뒤로가기 - 로그인, 회원가입, 비밀번호 찾기 화면 등에서 뒤로가기를 눌렀을 때 이전 단계로 이동할 수 있어야 한다.

## 5.2 태스크 목록

태스크 번호	태스크 명	구현 기술	담당자	소요기간	비고
T-301	회원가입	Kotlin, Validator, BCryptPasswordEncoder, 예외/응답 규격화	윤태영	2d	-
T-302	로그인	Spring Security, JWT(Access/Refresh), 예외 처리, 토큰 재발급	윤태영	3d	-
T-303	이메일 인증	Spring Mail(Gmail SMTP), Thymeleaf	윤태영	2d	-
T-304	아이디 찾기	Spring Mail(Gmail SMTP), Thymeleaf	윤태영	2d	-
T-305	비밀번호 찾기	Spring Mail(Gmail SMTP), Thymeleaf	윤태영	2d	-
T-306	로그인 UI	Jetpack Compose, 폼 검증	서범수	2d	-
T-307	회원가입 UI	Jetpack Compose, 입력 검증, 네트워크 연동	서범수	3d	-
T-308	약관 동의 UI	체크박스 UI, 상태 관리	서범수	2d	-
T-309	이메일 인증 UI	Jetpack Compose	서범수	1d	-
T-310	이메일 찾기 UI	Jetpack Compose	서범수	1d	-
T-311	비밀번호 찾기 UI	Jetpack Compose	서범수	1d	-
T-312	TokenDataStore	Android DataStore, JWT	서범수	3d	-
T-313	Auth Controller 연동	Retrofit, Interceptor, Token 주입, 401 처리	서범수	5d	선행 T-300~ T-313
T-314	Auth 문서 작성	문서화(API 명세서), 빌드명세서	서범수, 윤태영	1d	-

## 5.3 종료일

2025년 09월 23일(월)

- 해당 빌드는 수업 일정과 팀의 프로젝트 개발 일정을 고려하여 9월 23일을 종료일로 설정한다.

## 5.4 중간 점검일

2025년 9월 10일(수)

중간 점검에서는 아래의 항목들을 기준으로 요구사항 대비 구현의 적합성, UI-API 연동의 정상 동작 여부, 보안 및 예외 처리 안정성 등을 집중적으로 점검한다.

### 회원가입 및 이메일 인증 기능 점검 (T-301, T-303, T-309)

- 회원가입 후 이메일 인증이 정상적으로 발송되는지 확인한다.
- 인증 링크 클릭 또는 인증코드 입력 시 정상적으로 계정 활성화가 되는지 검증한다.
- 인증 실패 및 만료 케이스에 대한 예외 처리도 확인한다.

### 비밀번호 찾기 및 임시 비밀번호 발급 기능 점검 (T-305, T-311)

- 비밀번호 찾기 요청 시 인증 절차가 정상적으로 진행되는지 점검한다.
- 임시 비밀번호가 이메일로 정상 발송되는지, UI 상에 안내가 적절하게 표시되는지 확인한다.
- 임시 비밀번호 사용 후 정상 로그인이 가능한지 검증한다.

### 환경 변수 및 보안 설정 점검 (T-313 관련)

- API Key, SMTP 설정, JWT Secret Key 등의 정보가 환경 변수로 안전하게 관리되고 있는지 점검한다.
- 빌드 환경에서 노출되는 정보가 없는지 보안 점검을 수행한다.

### 예외 및 에러 처리 점검 (T-301~T-314 공통)

- 회원가입, 로그인, 인증, 비밀번호 찾기 과정에서 발생할 수 있는 예외 상황에 대해 적절한 메시지와 상태 코드가 반환되는지 점검한다.
- 클라이언트(UI)에서도 에러 상태를 직관적으로 표시하고 있는지 확인한다.

### 토큰 자동 회전(갱신) 로직 점검 (T-302, T-312, T-313)

- 401 Unauthorized 발생 시 Refresh Token을 통해 Access Token이 자동 재발급되는지 확인한다.
- 토큰 만료 시점 처리와 사용자 경험(UX)이 끊기지 않는지 점검한다.

### UI 완성도 및 사용성 점검 (T-306~T-311)

- 화면 간 전환이 자연스럽고 응답 속도가 적절한지 확인한다.
- 잘못된 입력에 대해 실시간 검증이 이루어지고 있는지 점검한다.
- 불필요한 공백, 레이아웃 깨짐, 시각적 오류가 없는지 점검한다.

## 6. 4차 빌드 개발 대상

### 6.1 선정 기능명

선정 기능명은 다음과 같다.

**FR-600: 마이 페이지**

**UIR-600: 마이페이지 화면**

**UIR-6100: 캘린더 화면**

**UIR-6200: 사용자 정보 표시 화면**

#### 6.1.1 선정 기능 요구사항

##### **FR-600: 마이 페이지**

FR-606: 계정 인증 – 생성한 계정으로 로그인 할 수 있어야 하며, 계정을 잊어버렸을 경우 본인 인증을 통해 가입한 계정을 확인할 수 있고, 이메일로 임시 비밀번호를 발급 받을 수 있어야 한다.

FR-607: 계정 관리 – 마이 페이지에서 본인의 정보를 조회 및 닉네임, 전화번호, 비밀번호 등의 정보를 수정할 수 있어야 하며, 사용자가 원할 때 로그아웃과 회원 탈퇴를 할 수 있어야 한다.

FR-608: 프로필 사진 - 마이페이지의 프로필에서 프로필 사진을 업로드하여 생성하거나 수정할 수 있어야 한다.

FR-609: 개인 캘린더 – 관측 일정을 캘린더에 계획하거나, 과거에 했던 관측을 기록할 수 있어야 하고, 작성한 계획이나 기록을 열람할 수 있어야 한다.

FR-610: 세부 계획 – 관측 일정을 계획하거나 수정할 때, 별지도와 관측지 UI나 지도를 통하여 관측 대상, 장소, 날짜 및 시간을 정할 수 있어야 한다.

##### **UIR-600: 마이페이지 화면**

UIR-601: 프로필 - 사용자 프로필에는 사용자의 닉네임, 프로필 사진, 관측 횟수가 표시된다.

UIR-602: 캘린더 뷰 - 월간·주간·일간 뷰를 전환할 수 있는 캘린더 컴포넌트를 제공하고, 관측 일정, 기록은 날짜 아래 색깔 별로 점으로 표시한다.

UIR-603: 하단 메뉴 바 - 하단에 메뉴 바를 두어 사용자가 작성한 관측 계획 및 게시글을 조회와 고객센터 페이지로 넘어갈 수 있어야 한다.

UIR-604: 로그아웃 - 로그아웃 버튼 클릭 시 로그아웃 하겠냐는 질문 팝업이 표시되며, 확인을 누르면 로그인 화면으로 이동한다.

UIR-605: 회원탈퇴 - 회원탈퇴 버튼 클릭 시 비밀번호 재입력을 한 후 탈퇴 하겠냐는 질문 팝업이 표시되며, 확인을 누르면 사용자의 계정이 삭제되고 로그인 화면이 표시된다.

## **UIR-6100: 캘린더 화면**

UIR-6101: 일정 상세 보기 - 캘린더의 일정 점을 터치하면 팝업 또는 상세 화면으로 이동하여, 날짜·시간·장소·대상·메모·알림 설정을 확인·수정할 수 있어야 한다.

UIR-6102: 일정 추가 - 화면 상단의 '+' 플로팅 액션 버튼(FAB)을 눌러 관측 계획 입력 화면으로 이동한다.

UIR-6103: 관측 일정 작성 - 관측 일정을 계획할 때 관측 일자, 관측지, 관측 대상, 메모를 입력할 수 있어야 한다.

## **UIR-6200: 사용자 정보 화면**

UIR-6201: 사용자 정보 표시 - 사용자 정보 화면에는 사용자의 닉네임, 생년월일, 이메일(수정 불가), 연동 계정(수정 불가), 프로필 사진이 표시된다.

UIR-6202: 정보 수정 - 닉네임, 프로필 사진 옆에 수정 아이콘을 누르면 생년월일, 닉네임, 핸드폰 번호의 정보를 수정할 수 있어야 한다.

UIR-6203: 비밀번호 변경 - 비밀번호 변경 버튼을 누르면 정해진 절차에 따라서 비밀번호를 변경할 수 있어야 한다.

## 6.2 태스크 목록

태스크 번호	태스크 명	구현 기술	담당자	소요기간	비고
T-401	계정 관리	Spring Security, JPA(Hibernate), MySQL	윤태영	3d	
T-402	비밀번호 변경	Spring Security, BCryptPasswordEncoder	윤태영	3d	
T-403	로그아웃 및 회원탈퇴	JWT 리프레시 토큰 무효화	윤태영	3d	
T-404	관측 일정 계획	Spring MVC(REST), JPA(Hibernate)+MySQL	윤태영	4d	
T-405	관측 기록	Spring MVC(REST), JPA+MySQL, Multipart 파일 업로드(이미지)	윤태영	4d	
T-406	캘린더 조회	Pageable+정렬	윤태영	3d	
T-407	사용자 정보 표시 UI	Jetpack Compose, ViewModel + Flow, Retrofit/OkHttp	김채영	3d	
T-408	회원 탈퇴 팝업	이중 확인/체크박스, 탈퇴 API 호출 후 자동 로그아웃/세션 삭제	김채영	3d	
T-409	캘린더 UI	Jetpack Compose, ViewModel	김채영	4d	
T-410	일정 상세조회 화면	Jetpack Compose, ViewModel	김채영	2d	
T-411	일정 계획 작성 화면	폼 검증(즉시 검증 + 안내)	김채영	3d	
T-412	마이페이지 I.A 작성	정보구조(I.A), 화면흐름/시나리오, draw.io	김채영	7d	
T-413	User API 명세서 작성	springdoc-openapi/Swagger 스키마, 요청/응답 예시 JSON	윤태영	5d	
T-414	User Controller 연동	Retrofit2 + OkHttp Interceptor, 코루틴/Flow, 에러 핸들링	김채영	5d	

## 6.3 종료일

2025년 10월 20일(월)

- 해당 빌드는 수업 일정과 팀의 프로젝트 개발 일정을 고려하여 10월 20일을 종료일로 설정한다.

## 6.4 중간 점검일

2025년 9월 20일(토)

중간 점검에서는 아래와 같은 점검 항목들에 대해 잘 부합하는지에 대해 점검한다.

### 요구사항 추적성 확인

- 설계 단계에서 정의한 요구사항과의 추적성이 일치하는가?

### 기능 동작 확인

- 비밀번호 변경 시 현재 비밀번호 검증 정책이 잘 적용되는가?
- 로그아웃 및 회원 탈퇴 시 사용하던 토큰이 만료되는가?
- 프론트와 연동시 API가 정상적으로 작동하는가?

### 예외 처리 확인

- 예외/에러 처리는 잘 이루어지고 있는가?

### UI 인터페이스 동작 확인

- 부자연스러운 UI는 없는가?

### 문서와 기능 연결 확인

- 유즈케이스 시나리오대로 흐름이 동작하는가?
- IA 흐름과 UI의 흐름이 일치하는가?

## 7. 5차 빌드 개발 대상

### 7.1 선정 기능명

선정 기능명은 다음과 같다.

**FR-500: 관측 커뮤니티**

**UIR-500: 커뮤니티 화면**

**UIR-5100: 게시글 작성 화면**

**UIR-5200: 게시글 상세 조회 화면**

#### 7.1.1 선정 기능 요구사항

##### **FR-500: 관측 커뮤니티**

FR-501: 게시물 작성 및 공유 - 사용자는 관측 후기 또는 교육 콘텐츠를 작성하여 커뮤니티에 새 게시물을 업로드할 수 있어야 한다.

FR-502: 게시물 조회 - 사용자는 커뮤니티에서 다른 사용자가 올린 게시물을 최신순, 인기순, 조회순으로 스크롤 하여 볼 수 있어야 한다.

FR-503: 댓글 작성 및 관리 - 사용자는 각 게시물에 댓글을 작성, 수정, 삭제할 수 있어야 하며, 다른 사용자와 의견을 주고받을 수 있어야 한다.

FR-504: 좋아요/평점 - 사용자는 각 게시물 및 댓글에 좋아요를 줄 수 있고, 교육 프로그램에 평점, 좋았던 점, 개선해야 할 사항의 피드백을 남길 수 있어야 한다.

##### **UIR-500: 커뮤니티 화면**

UIR-501: 카테고리 탐색 - 상단에 "관측 후기", "자유 게시판", "교육 콘텐츠" 등 탭 형태의 카테고리 바를 배치하여, 터치 한 번에 해당 게시판으로 이동할 수 있어야 한다.

UIR-502: 검색 필터 - 키워드 검색 입력창과 함께 "태그", "작성자", "날짜순/인기순" 필터 드롭다운을 제공해야 한다.

UIR-503: 게시판 목록 - 선택한 카테고리의 게시물이 썸네일·제목·작성자·작성일·평점과 함께 리스트 또는 카드로 표시되고, 무한 스크롤을 지원해야 한다.

UIR-504: 커뮤니티 홈 - 커뮤니티 홈 화면에서는 최근 게시된 '관측 후기글', '교육 프로그램글' 그리고 좋아요 수가 높은 '인기 자유 게시글'을 조회할 수 있다.

##### **UIR-5100: 게시글 작성 화면**

UIR-5101: 입력 항목 - 게시글 작성 화면은 제목, 내용, 이미지 업로드 등의 항목으로 구성되며, 관측 후기 게시글은 관측 일시, 관측 장소, 관측 대상, 관측 장비의 추가 항목이 있고, 교육 프로그램 게시글은 난이도, 교육 대상의 추가 항목을 작성해야 한다.

UIR-5102: 저장 및 게시 - 작성하고 있는 내용을 중간에 저장 할 수 있도록 임시 저장을 할 수 있어야 하며, 커뮤니티에 업로드 할 것인지 선택할 수 있어야 한다.

UIR-5103: 미디어 첨부 - 사용자는 갤러리에서 관측 때 촬영한 사진을 선택하거나 즉시 촬영하여 이미지를 첨부할 수 있어야 한다.

UIR-5104: 유효성 검증 - 필수 입력 항목이 비어있거나, 오류가 있을 경우 안내 메시지를 통해 알려주고 저장을 제한한다.

## **UIR-5200: 게시글 상세 조회 화면**

UIR-5201: 수정 및 삭제: 본인이 작성한 게시글을 상세 조회하면 수정 및 삭제 버튼을 클릭하여 수정/삭제가 가능해야 한다.

UIR-5202: 게시글 상세 보기 – 리스트에서 게시글을 선택하면 풀스크린 모달 또는 새 화면으로 이동하여, 사진 확대, 본문 전체, 평점·댓글·공유 버튼을 확인할 수 있어야 한다.

UIR-5203: 댓글·좋아요 - 게시글 상세 화면 하단에 댓글 입력창과 좋아요 버튼을 배치하고, 댓글은 스레드 형태로 표시해야 한다.

## 7.2 태스크 목록

태스크 번호	태스크 명	구현 기술	담당자	소요기간	비고
T-501	게시물 작성/수정/삭제	Spring Security(AuthenticationPrincipal), 소프트삭제(status/삭제표시)	윤태영	5d	
T-502	게시글 조회 및 평점, 좋아요 토클, 평점 평균/카운트 원자적 갱신	Spring Data JPA(Pageable/Sort), 좋아요 토클, 평점 평균/카운트 원자적 갱신	윤태영	3d	선행: FR-400.01
T-503	댓글 작성/수정/삭제	N+1 방지(fetch join), 예외 처리	윤태영	4d	선행: FR-400.01
T-504	이미지 업로드	Spring MVC Multipart, 파일 검증, 오브젝트 스토리지(MinIO 로컬/S3 운영)	윤태영	5d	
T-505	커뮤니티 홈 UI	Compose HorizontalPager, LazyVerticalGrid, Post API 홈 화면 게시글 불러오기 연동	김채영	5d	
T-506	게시글 목록 UI	Compose LazyColumn, LazyVerticalGrid, Post API 타입별 목록 조회 연동, 작성자, 제목 검색 필터링	김채영	5d	
T-507	게시글 작성 UI	폼 검증, 이미지 선택, DatePicker, 필수 작성 목록 검증, Post API 게시글 생성 연동	김채영	4d	
T-508	게시글 상세 조회 UI	상세 본문/이미지 캐러셀, 좋아요/댓글수 로드, Post API 상세조회 연동, 교육 게시글 평점 매기기 연동	김채영	3d	선행: UIR-500.03
T-509	게시글 댓글 작성 UI	댓글 관련 API 댓글/대댓글 작성 연동	김채영	3d	
T-510	게시글 댓글 목록 UI	댓글 관련 API 댓글 목록 연동, 댓글 좋아요 토클 연동, 댓글 삭제 연동	김채영	3d	
T-511	커뮤니티 페이지 화면 I.A 작성	IA(화면 흐름/상태 디아이어그램), Figma 산출물	김채영	5d	
T-512	Community API 명세서 작성	springdoc-openapi, 요청/응답 예시 JSON, 에러코드/메시지 표준화	윤태영	5d	선행: FR-400.01~04
T-513	Community Controller 연동	Retrofit2, OkHttp Interceptor, 에러 핸들링	김채영	5d	선행: API-001

## 7.3 종료일

2025년 10월 13일(월)

- 해당 빌드는 수업 일정과 팀의 프로젝트 개발 일정을 고려하여 10월 13일을 종료일로 설정한다.

## 7.4 중간 점검일

2025년 10월 1일

중간 점검에서는 아래와 같은 점검 항목들에 대해 잘 부합하는지에 대해 점검한다.

### 요구사항 추적성 확인

- 설계 단계에서 정의한 요구사항과의 추적성이 일치하는가?

### 문서와 기능 연결 확인

- IA 흐름과 커뮤니티/게시글/댓글 UI 플로우가 일치하는가?
- Community API 명세서의 요청/응답/상태코드가 구현과 일치하는가?

### 기능 동작 확인

- 게시물 작성/수정/삭제 권한이 소유자 기준으로 정확히 검증되는가?
- 댓글 작성/수정/삭제 권한이 소유자 기준으로 정확히 검증되는가?
- 좋아요(토글) 및 평점(사용자당 1회) 제약이 잘 보장되고 있는가?
- 게시글/댓글 수, 좋아요 수, 평점 수/평균 갱신이 원자적으로 처리되는가?
- 이미지 저장소(S3/MinIO) 경로, 공개 URL, 권한이 설정대로 동작하는가?

### 예외 처리 확인

- 커뮤니티 홈/목록 UI에서 로딩/빈상태/에러 상태 표시가 일관되게 구현되었는가?

## 8. 6차 빌드 개발 대상

### 8.1 선정 기능명

선정 기능명은 다음과 같다.

**FR-200: 마스코트 캐릭터**

**UIR-200: 마스코트 캐릭터 인터페이스**

#### 8.1.1 선정 기능 요구사항

##### **FR-200: 마스코트 캐릭터**

FR-201: 친근한 디자인 – 마스코트 캐릭터는 친근한 디자인으로 사용자가 흥미와 친숙함을 느낄 수 있어야 한다.

FR-202: Live 2D - 마스코트 캐릭터는 사용자의 집중을 유지하기 위해 Live2D 기반으로 동작 및 감정을 표현해야 한다.

FR-203: 동작 표현 – 마스코트 캐릭터는 대기, 화남, 슬픔, 행복, 당황, 등장, 퇴장 동작을 표현해야 한다.

FR-204: 말풍선 UI – 마스코트 캐릭터는 말풍선 UI를 통해 대화하는 형식을 사용해야 한다.

FR-205: 모듈화 - 마스코트 캐릭터는 메인화면, 별지도, 교육 콘텐츠 등 다양한 화면에서 동일하게 조작 할 수 있도록 모듈화되어야 한다.

##### **UIR-200: 마스코트 캐릭터 인터페이스**

UIR-201: 위치 조정 – 캐릭터의 위치는 상황에 따라 유동적으로 조정할 수 있어야 한다.

UIR-202: 크기 조정 – 마스코트 캐릭터의 크기는 환경에 맞게 조정할 수 있어야 한다.

UIR-203: 상황 반응 - 마스코트 캐릭터는 상황에 따라 감정을 표현해야 하며, 이는 시각적으로 명확해야 한다.

UIR-204: 텍스트 가독성 – 캐릭터의 설명 텍스트는 말풍선에 명확한 글씨로 표시되어야 한다.

UIR-205: 터치 간섭 금지 – 캐릭터 UI는 다른 UI 요소의 터치 이벤트를 방해하지 않아야 한다.

## 8.2 태스크 목록

태스크 번호	태스크 명	구현 기술	담당자	소요기간	비고
T-601	마스코트 캐릭터 디자인	일러스트 제작 툴	서범수, 디자이너	14d	-
T-602	Live2D App 이식	Live2D Cubism SDK for Java, OpenGL ES	서범수	14d	-
T-603	마스코트 캐릭터 디자인 Live2D화	Live2D Cubism Editor	서범수	3d	선행 T-600
T-604	Live2D 동작 구현	Cubism Motion API, Gesture Handling	서범수	14d	선행 T-600~ 602
T-605	Live2D 동작 연동	Live2D Cubism SDK for Java, OpenGL ES	서범수	2d	선행 T-600~ 603
T-606	마스코트 캐릭터 말풍선	Compose Canvas / Modifier	서범수	2d	-
T-607	마스코트 캐릭터 UI	Jetpack Compose, AndroidView Wrapper	서범수	3d	선행 T-601
T-608	Live2D 모듈화	DI(Hilt), 모듈 분리	서범수	3d	선행 T-600~ T-606
T-609	App Icon 변경	AndroidManifest, Adaptive Icon	서범수	0.5d	선행 T-600
T-609	마스코트 캐릭터 문서 작성	빌드명세서	서범수	1d	-

## 8.3 종료일

2025년 10월 12일(일)

- 해당 빌드는 수업 일정과 팀의 프로젝트 개발 일정을 고려하여 10월 12일을 종료일로 설정한다.

## 8.4 중간 점검일

2025년 10월 2일(목)

중간 점검에서는 아래와 같은 점검 항목들에 대해 잘 부합하는지에 대해 점검한다.

### 요구사항 추적성 확인

- FR/UIR 요구사항과 실제 구현 중인 기능이 일치하는가?
- 태스크 번호(T-600~T-609)가 요구사항(FR-200~205, UIR-200~205)과 정상적으로 매핑되는가?

### Live2D App 이식 및 렌더링 점검

- Live2D 모델이 Android Compose 환경에서 정상적으로 출력되는가? (T-601)
- 화면 해상도 및 비율에 따라 캐릭터가 올바르게 스케일링되는가? (UIR-202)

### Live2D 동작 구현 및 연동 점검

- 감정 동작(대기, 화남, 슬픔, 행복, 당황, 등장, 퇴장)이 정상적으로 표현되는가? (FR-203 / T-203)
- Controller를 통한 상태 전환이 문제 없이 동작하는가? (T-604)

### UI 인터페이스 동작 확인

- 캐릭터가 우측 하단 또는 지정된 위치에 정상적으로 고정되는가? (UIR-201)
- 주변 UI와 터치 이벤트 충돌이 없는가? (UIR-205)
- 말풍선 UI 및 텍스트 가독성이 명확한가? (FR-204 / UIR-204 / T-605)

### 모듈화 구조 검증

- Controller 및 View가 독립적으로 동작하며, 메인화면·별지도·교육화면 간 전환 시에도 정상 작동하는가? (FR-205 / T-607)

### 리소스 및 아이콘 점검

- App Icon이 정상적으로 적용되고, 런처에서 시각적으로 표시되는가? (T-608)

### 문서화 상태 점검

- 현재까지의 구현 내용이 빌드 명세서 및 요구사항 정의서와 일치하는가? (T-609)
- 요구사항 ID와 태스크 번호 간 추적 표가 정리되어 있는가?

# 9. 7차 빌드 개발 대상

## 9.1 선정 기능명

선정 기능명은 다음과 같다.

**UIR-100: 천체관측 교육 프로그램 화면**

**FR-100: 천체관측 교육 프로그램**

### 9.1.1 선정 기능 요구사항

**UIR-100: 천체관측 교육 프로그램 화면**

UIR-101: 교육프로그램 화면 구성 - 교육프로그램 전용 화면 (Live2D+별지도 통합 구성)

UIR-102: 대화식 정보 제공 - 캐릭터가 이야기하듯 단계별로 설명을 진행해야 한다.

UIR-103: 단계 전환 - 이전, 다음 넘길수있어야함

UIR-104: 관측 교육 콘텐츠 - 관측 교육 콘텐츠의 내용에 따라 화면이 유동적으로 변화하여야 한다.

UIR-105: 피드백 입력 인터페이스 - 학습 종료후 사용자가 별점과 의견을 입력할수있는 피드백 UI가 제공되어야한다.

UIR-106: 학습 자동 진행 제어 - 자동진행(Auto)모드를 on/off 할수있는 토글 UI를 제공해야한다.

**FR-100: 천체관측 교육 프로그램**

FR-101: 단계별 학습 - 사용자가 천체관측 방법을 쉽게 이해할 수 있도록 마스코트 캐릭터를 활용한 단계별 학습 가이드라인을 제공해야 한다.

FR-102: 교육프로그램 선택 - 사용자는 원하는 교육을 선택해서 들을수 있어야한다.

FR-103: Live2D 연동 제어 - 대사, 감정표현 등의 내용들이 정상 작동한다.

FR-104: 별지도 연동 제어 - 시야이동, 천체선택, 토글등이 원활하게 이루어진다.

FR-105: 가이드 건너뛰기 - 가이드라인은 사용자가 원하지 않으면 넘어갈 수 있어야 한다.

FR-106: 시야-대사 동기화 - 캐릭터 대사와 별지도 시야 이동 간 타이밍 일치

FR-107: 자동 진행 제어 기능 - Auto모드가 활성화되면 단계가 일정 시간 간격으로 자동 전환되어야하며, 비활성화시 수동 전환만 가능해야 한다.

FR-108 확장성 확보 - JSON 구조를 유지한 상태로 추후 퀴즈나 영상 삽입 기능을 손쉽게 추가 할 수 있어야 한다.

## 9.2 태스크 목록

태스크 번호	태스크 명	구현 기술	담당자	소요기간	비고
T-701	별지도 Controller 연동	Kotlin / WebView / JS Bridge	서범수	5d	-
T-702	Live2D Controller 연동	Kotlin / Live2D SDK / Coroutine	서범수	5d	-
T-703	EduEngine 파서 구현	Kotlin / JSON Parser / StateFlow	서범수	7d	T-304과 동시진행
T-704	JSON 형식 구현	JSON Schema / Kotlin Data Class	서범수	7d	T-303과 동시진행
T-705	JSON 시나리오 로드 구현	Kotlin / Asset Loader / ViewModel	서범수	3d	T-304
T-706	교육프로그램 Overlay UI 구현	Jetpack Compose / Material3	서범수	3d	-
T-707	자동 진행 모드 구현	Jetpack Compose / Material3	서범수	1d	T-306
T-708	피드백 UI 구현 및 API 연동	Coroutine / Flow / Timer	서범수	3d	-
T-709	교육프로그램 문서 작성	Compose / Retrofit / Hilt	서범수	1d	-

## **9.3 종료일**

2025년 11월 10일(월)

- 해당 빌드는 수업 일정과 팀의 프로젝트 개발 일정을 고려하여 11월 10일을 종료일로 설정한다.
- 해당 일정까지 모든 주요 기능(EduProgram, Skymap, Feedback 모듈 등)의 통합 테스트를 완료하고 시연 가능한 형태로 제출을 목표로 한다.

## **9.4 중간 점검일**

2025년 11월 5일(수)

중간 점검일은 전체 개발 진행 상황을 검토하고,

- 기능별 통합 상태(별지도-Live2D 연동, 자동 진행, 피드백 API 등)를 확인하기 위한 내부 리뷰 및 발표용 점검 시점으로 설정하였다.
- 이 시점까지 EduProgram의 주요 동작 흐름(시나리오 로드 → 자동진행 → 종료 후 피드백 입력)을 정상적으로 확인하는 것을 목표로 한다.

## 10. 개발 환경

번호	도구 / 환경	담당자	완료일	비고
1	Kotlin 1.9.25	윤태영	2025.03.04	Gradle 의존성으로 자동 설치
2	Java 21	윤태영	2025.03.04	Kotlin과 호환
3	Spring Boot 3.4.3	윤태영	2025.03.04	
4	IntelliJ IDEA	윤태영	2025.03.04	Kotlin 및 SpringBoot에 최적화된 IDE
5	Docker / Docker Compose	윤태영	2025.03.11	
6	MySQL 8.0	서범수	2025.03.13	
7	MongoDB 6.0	서범수	2025.03.13	
8	Mongo-express	서범수	2025.03.13	
9	GitHub	서범수	2024.12.02	버전 관리 및 협업
10	Swagger UI	서범수	2025.03.13	API 테스트 및 자동 문서화
11	Notion	김채영	2024.10.29	문서화 공유 도구
12	Android 15 / API 35	김채영	2025.03.04	
13	Live2D Cubism SDK for Java 5.4.1	서범수	2025.09.15	마스코트 캐릭터
14	StellariumWebEngine	서범수	2025.10.11	별지도

## **11. 기타 사항**

본 문서는 여러 차수의 문서가 작성될수 있으므로 다음과 같은 예시에서 문서번호에 .1 .2와같은 수식어를 붙여 차별점을 둔다.

예시) Byeoldori\_Doc\_06.1\_BuildDevPlan\_v1\_2025-03-26