

한가람도서관 프로젝트



LOGO

프로젝트명 : 한가람 독서광

개요

- 목적: 책 정보의 저장 및 시각적 표현
- 도구: 안드로이드 스튜디오, VS CODE, 오라클 DB

주요 기능

- 사용자 로그인 확인
- 제목, 내용, 이미지 저장
- 책 이미지 갤러리 뷰

핵심 가치

- 사용자 친화적인 인터페이스
- 효율적인 데이터 관리
- 시각적으로 매력적인 책 표시
- 효과적인 정보 관리
- 직관적인 사용자 경험
- 데이터의 안전한 저장 및 접근



기획 배경

• 문제 인식:

다수의 독자들이 책 정보의 효과적인 관리를 원하나, 기존 애플리케이션들은 사용자의 요구를 충족시키지 못하는 점이 있었음.

• 시장 필요성:

사용자들이 책 정보를 손쉽게 저장하고 시각적으로 향유할 수 있는 애플리케이션에 대한 수요가 존재하나 전자책 또는 상업적 용도위주의 도서 앱시장에 출시된 앱 자체가 많지 않음 (*교보문고, 밀리의서재)

• 차별화 전략:

기존 애플리케이션과의 차별화를 위해 사용자 친화적인 인터페이스와 효율적인 데이터 관리 기능을 제공하는 것을 목표로 설정하였음.

개발 목표

• 개발 목적:

책 정보를 효율적으로 관리하며 사용자가 쉽게 접근할 수 있는 애플리케이션 개발에 주안점을 두었음.

• 기술 선택:

Android Studio를 통한 유연한 앱 개발 환경과 Oracle DB를 이용한 데이터의 안전성 및 안정성 확보에 초점을 맞췄음.

• 수업에서 배운 기술의 적극적 활용:

개발 과정에서 안드로이드 애플리케이션 개발, 데이터베이스 관리, UI/UX 디자인 등 수업에서 습득한 기술을 적극적으로 활용하여 앱 구현에 반영하였음.

* View, Layout 구현, Back-end (Spring Boot) API 구현

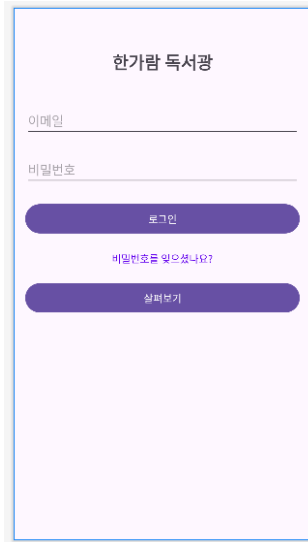
Layout

①



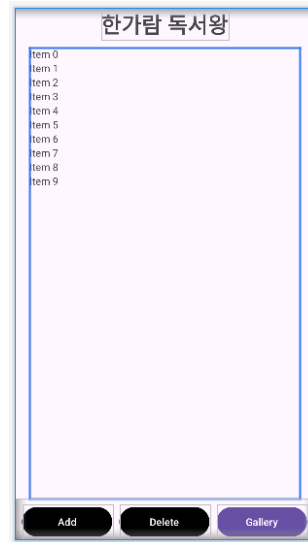
activity_splash.xml

②



activity_login.xml

③



activity_main.xml

④



activity_add.xml

⑥



listitem_memo.xml

Story Board



① 시작 화면 (SplashActivity):

앱을 시작할 때 나타나는 첫 화면입니다. 이 화면은 앱 로고나 브랜드를 짧게 표시한 후에 자동으로 로그인 화면으로 넘어갑니다.

② 로그인 화면 (LoginActivity):

사용자가 자신의 계정 정보를 입력하여 앱에 로그인하는 화면입니다. 로그인이 성공하면 메인 화면으로 이동하고, 실패하거나 로그아웃을 선택하면 다시 시작 화면으로 돌아갑니다.

③ 메인 화면 (MainActivity):

로그인에 성공한 후 사용자에게 표시되는 중심 화면입니다. 여기에서 사용자는 메모를 추가하거나 갤러리를 볼 수 있는 옵션을 선택할 수 있습니다.

④ 메모 추가 화면 (AddActivity):

사용자가 새로운 메모나 이미지를 추가할 수 있는 화면입니다. 메모나 이미지 추가 작업이 끝나면 메인 화면으로 돌아갑니다.

⑤ 갤러리 화면 (Gallery):

저장된 이미지들을 갤러리 형태로 볼 수 있는 화면입니다. 사용자는 여기에서 자신의 책 이미지 컬렉션을 볼 수 있습니다.

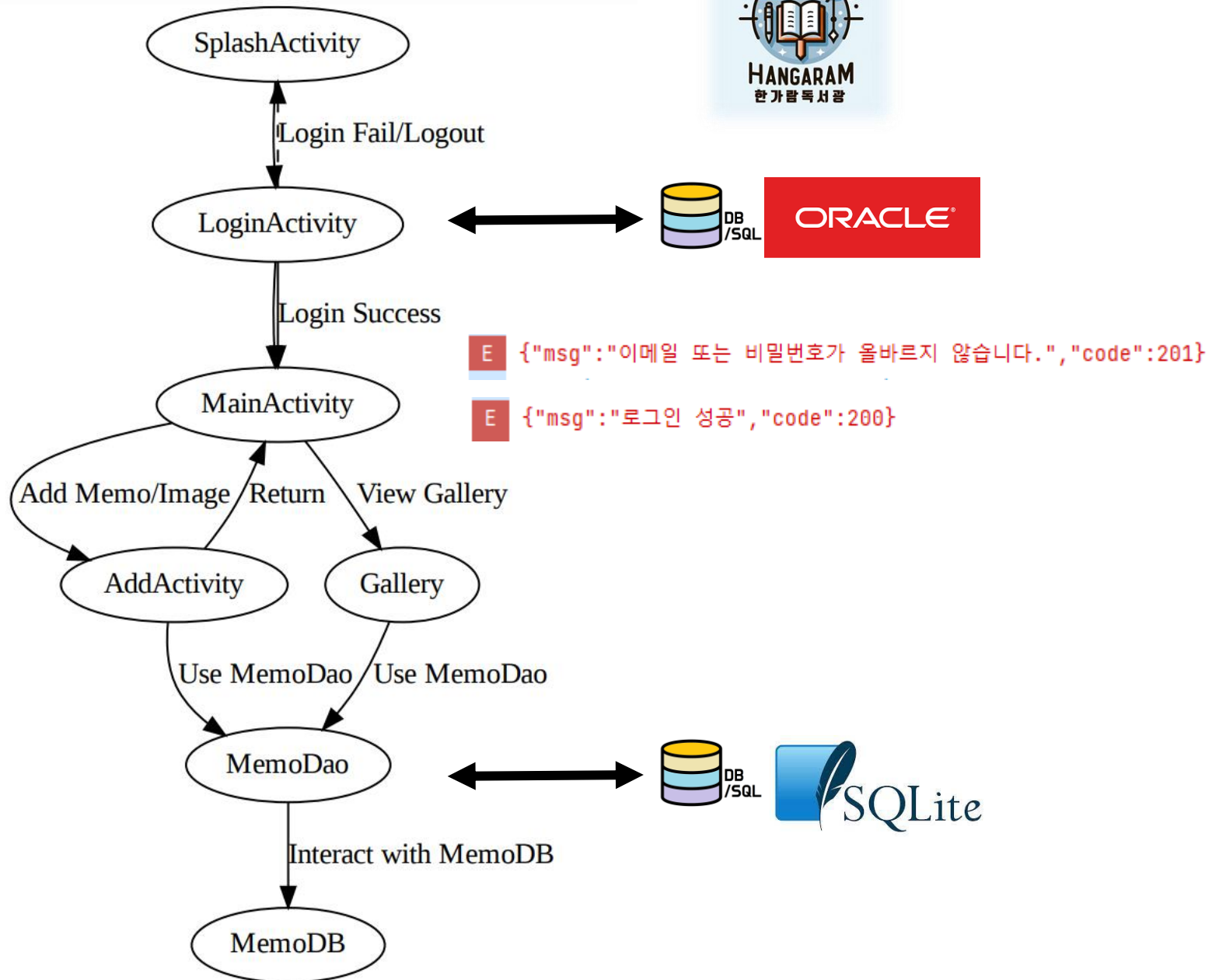
⑥ 메모 DAO (MemoDao):

데이터베이스에 접근하여 메모 데이터를 관리하는 데 사용되는 데이터 접근 객체(Data Access Object)입니다. 메모 추가 화면과 갤러리 화면에서 이 DAO를 통해 데이터 작업을 수행합니다.

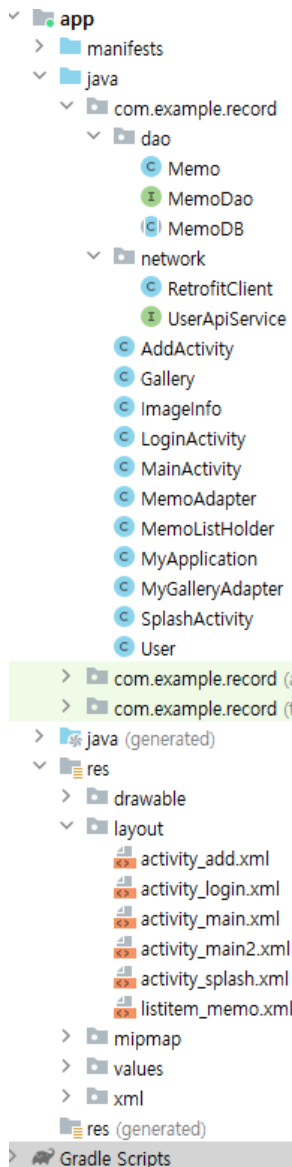
⑦ 메모 DB (MemoDB):

실제 메모 데이터가 저장되는 데이터베이스입니다. 메모 DAO를 통해 이 데이터베이스와 상호작용하여 데이터를 저장하고 검색합니다.

Flow Chart



PROJECT STRUCTURE



애플리케이션 구조

- 애플리케이션은 MVC(Model-View-Controller) 패턴을 따르며, 다음과 같은 주요 패키지와 컴포넌트로 구성

패키지 구조

- DAO (Data Access Object):** 데이터 접근과 관련된 클래스를 포함합니다.
- Memo:** 메모 데이터의 모델 클래스입니다.
- MemoDao:** 데이터베이스 작업을 위한 인터페이스입니다.
- MemoDB:** 애플리케이션의 데이터베이스를 나타냅니다.
- network:** 네트워크 통신을 위한 클래스를 포함합니다.
- RetrofitClient:** HTTP 클라이언트 구성을 위한 클래스입니다.
- UserApiService:** 사용자 관련 API 호출을 관리하는 인터페이스입니다.
- Activities:** 사용자 인터페이스와 상호작용을 담당하는 액티비티들을 포함합니다.
- SplashActivity:** 앱 시작 시 초기 화면을 표시합니다.
- LoginActivity:** 사용자 로그인을 처리합니다.
- MainActivity:** 메모와 책 이미지의 메인 갤러리 뷰를 보여줍니다.
- AddActivity:** 새로운 메모나 책 이미지를 추가하는 화면입니다.
- Gallery:** 저장된 책 이미지들을 갤러리 형태로 보여줍니다.

리소스 관리

- res/layout:** 각종 레이아웃 XML 파일들을 포함합니다.
- activity_splash.xml:** 스플래시 화면의 레이아웃을 정의합니다.
- activity_login.xml:** 로그인 화면의 레이아웃을 정의합니다.
- activity_main.xml:** 메인 갤러리 뷰의 레이아웃을 정의합니다.
- activity_add.xml:** 메모 추가 화면의 레이아웃을 정의합니다.
- listitem_memo.xml:** 메모 리스트 아이템의 레이아웃을 정의합니다.
- mipmap:** 앱 아이콘과 같은 이미지 리소스들을 포함합니다.
- values:** 문자열, 색상, 스타일 등의 리소스 값을 정의합니다.

Gradle Scripts

- build.gradle (Module: app):** 앱의 빌드 설정을 담당합니다. 의존성 관리, 플러그인 설정, 빌드 버전 관리 등이 포함
- build.gradle (Project: YourProjectName):** 전체 프로젝트의 빌드 설정을 관리합니다.

주요 기능

사용자 인증: 사용자는 LoginActivity를 통해 인증을 받아 애플리케이션에 접근할 수 있습니다.

데이터 관리: 사용자는 MainActivity와 AddActivity를 통해 메모와 책 이미지를 추가하고 관리할 수 있습니다.

갤러리 뷰: Gallery 액티비티에서는 사용자가 저장한 책 이미지를 갤러리 형태로 볼 수 있습니다.

기술 스택

프로그래밍 언어: Java

데이터베이스: Room Persistence Library

네트워크 통신: Retrofit

보안

네트워크 보안: HTTPS 프로토콜을 사용하여 암호화된 통신을 확보합니다.

데이터 보안: 로컬 데이터베이스에 저장되는 정보는 암호화하여 보호합니다.



※ TextWatcher

TextWatcher는 Android에서 텍스트 입력 필드(EditText)에 입력되는 텍스트의 변경 사항을 감지하고 대응하기 위한 인터페이스입니다. 사용자가 텍스트를 입력, 수정, 삭제하는 것을 실시간으로 감지하고, 이에 따라 특정한 동작을 수행할 수 있게 해줍니다.

※ Retrofit

Retrofit은 Android 및 Java 애플리케이션에서 REST API를 손쉽게 사용할 수 있도록 도와주는 유형 안전한 HTTP 클라이언트 라이브러리입니다. Square에 의해 개발되었으며, 네트워크 요청을 간편하게 만들고, 응답을 처리하는 과정을 단순화합니다.

세부 기술 내역

Room Persistence Library:

Room은 SQLite 데이터베이스 작업을 추상화하고, 보다 유연한 데이터베이스 액세스를 제공하는 라이브러리입니다. Memo, MemoDao, MemoDB는 Room 데이터베이스의 일부로서, 메모 데이터를 저장하고 접근하기 위해 사용됩니다.

Retrofit:

Retrofit은 REST API 통신을 위한 타입-안전한 HTTP 클라이언트입니다. RetrofitClient는 네트워크 요청을 위한 기본 설정을 담당하며, UserApiService는 사용자 관련 네트워크 호출의 인터페이스를 정의합니다.

Android Activity & Intent:

SplashActivity, LoginActivity, MainActivity, AddActivity, Gallery는 Android 애플리케이션의 화면을 나타내는 Activity입니다. Activity 간 전환은 Intent 시스템을 사용하여 수행됩니다. 예를 들어, SplashActivity에서 LoginActivity로 전환할 때 Intent를 사용합니다.

Android Adapter & ViewHolder 패턴:

MemoAdapter, MemoListHolder, MyGalleryAdapter는 RecyclerView의 Adapter와 ViewHolder 패턴을 구현하여 데이터를 리스트 형태로 표시합니다. 이 패턴은 높은 성능의 스크롤 가능한 리스트 또는 그리드를 생성하는 데 도움을 줍니다.

XML Layout:

res/layout 디렉토리에 있는 XML 파일들은 애플리케이션의 각 Activity의 사용자 인터페이스를 정의합니다. activity_splash.xml, activity_login.xml, activity_main.xml 등은 각각의 화면 레이아웃을 구성합니다.

Drawable Resources:

mipmap 폴더는 해상도에 따라 다른 이미지 리소스를 제공하여 앱 아이콘을 관리합니다.

Gradle Build System:

Gradle Scripts는 애플리케이션의 빌드 및 의존성 관리를 담당합니다. build.gradle 파일은 라이브러리 의존성, 애플리케이션의 빌드 설정, 버전 관리 등을 정의합니다.

MVC Architecture:

Model-View-Controller (MVC) 아키텍처 패턴이 암시적으로 사용됩니다. 여기서 Model은 dao와 network 패키지, View는 layout XML과 Activity 클래스, Controller는 Activity 클래스에서의 사용자 인터랙션 처리를 의미합니다.

■ Retrofit:

Retrofit은 REST API 통신을 위한 유형 안전한 HTTP 클라이언트 라이브러리입니다. 네트워크 요청을 쉽게 구성하고 응답을 효율적으로 처리할 수 있도록 도와줍니다. JSON, XML 등의 데이터 형식을 자바 객체로 쉽게 변환할 수 있는 기능을 제공합니다.

■ RecyclerView:

RecyclerView는 데이터 세트의 변경에 대응하여 유연하게 뷰를 제공하는 Android 라이브러리입니다.

리스트 또는 그리드 형태로 데이터를 표시하는 데 사용됩니다.

성능 최적화를 위해 뷰 홀더 패턴을 사용합니다.

■ TextWatcher:

TextWatcher 인터페이스는 텍스트 입력 필드(EditText)의 텍스트 변경을 감지합니다.

사용자가 텍스트를 입력, 수정, 삭제할 때 이를 실시간으로 감지하고 반응할 수 있습니다.

■ BaseAdapter:

BaseAdapter는 ListView, GridView와 같은 뷰에 데이터를 제공하는 데 사용되는 기본 어댑터 클래스입니다.

데이터를 뷰에 바인딩하는 데 필요한 메소드를 구현합니다.

■ Intent:

Intent는 Android에서 다양한 컴포넌트 간의 통신을 위해 사용됩니다.

액티비티, 서비스 및 브로드캐스트 리시버 간의 메시지 전달에 사용됩니다.

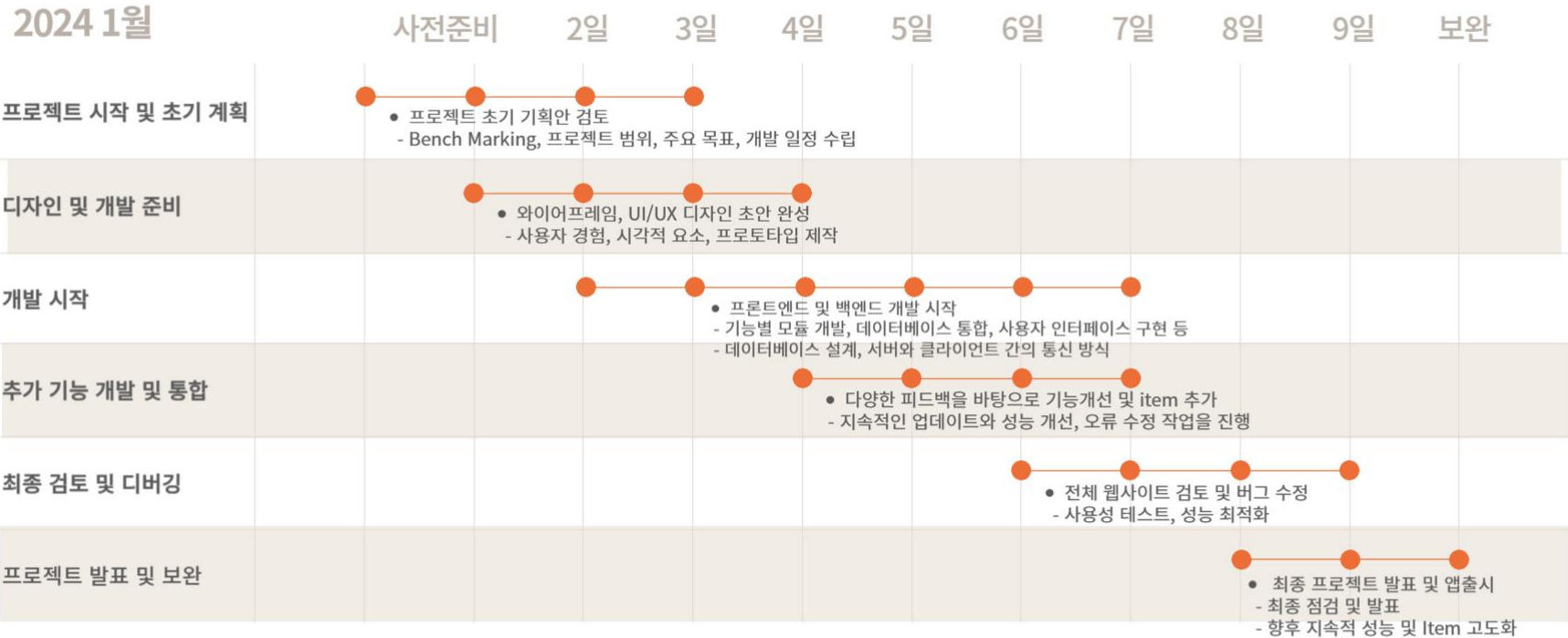
■ Handler와 Looper:

Handler와 Looper는 Android에서 스레드 간 통신을 관리하는 데 사용됩니다.

특히, 백그라운드 스레드에서 메인 스레드(UI 스레드)로 작업을 전달하는 데 사용됩니다.

PROJECT SCHEDULE

2024 1월



추가적인 기능 향상과

영화, 여행, 맛집, 신문기사 등 개인 일상 생활과 정보를 통합적으로 관리 하는

개인 정보 종합관리 앱으로 확대

