

프로젝트

DX 이주빈

I. 프로젝트 개요 : 제목과 간략 설명

제목 : (Project) Oasis

메모, 뉴스, 알람, 게임, 메뉴 추천등 의 기능

바쁜 현대인의 삶에서 잠시 쉬어갈수 있는 앱이라는 의미

II. 시스템 설계

기술 스택: Android Studio, SQLite, NFC, RSS(XML), Threads, AsyncTask, Custom Widgets

전체 구조



구조도



1) 주요 기능 설계

메인 페이지

- 구성 요소:
 - 툴바:
 - 좌측: 햄버거 버튼 → 사이드 메뉴 오픈
 - 우측: 홈 버튼 → 메인 페이지 이동
 - 버튼: To-Do, 뉴스, 알람, 게임, 저녁 메뉴로 이동

공통 로직

- **Android Activity 및 Lifecycle:**

- `AppCompatActivity` 를 상속받아 액티비티를 정의하고, 생명 주기 메서드(`onCreate` , `onResume` , `onPause` , `onDestroy`)를 사용.

- **View 및 Layout:**

- `FrameLayout` , `Button` , `ImageView` , `TextView` 와 같은 Android 뷰를 활용하여 UI를 구성.
- `Toolbar` 와 `DrawerLayout` 을 사용해 상단 툴바 및 사이드바 메뉴를 구현.
- `NavigationView` 로 네비게이션 메뉴 구성.

- **Event Handling:**

- 버튼 클릭 이벤트(`setOnClickListener`)와 `OnClickListener` 를 통해 사용자 상호작용 처리.
- 뷰 클릭 이벤트를 사용하여 이미지 클릭 처리.

- **Multi-threading 및 AsyncTask:**

- `AsyncTask` 를 활용해 백그라운드에서 작업 수행(`MouseTask`).
- `Handler` 와 `Looper` 를 사용해 UI 업데이트를 일정 간격으로 수행.

- **Resource Management:**

- `R.drawable` 와 `R.raw` 에서 이미지 및 오디오 리소스를 로드.
- `findViewById` 로 XML 레이아웃에서 뷰를 참조.

- **Intent 및 Activity Navigation:**

- `Intent` 를 사용해 다른 액티비티로 이동 (`MainActivity` , `MemoMainActivity` 등).

- **Logging:**

- `Log.d` 를 사용해 디버깅 메시지 출력.

- **UI 업데이트:**

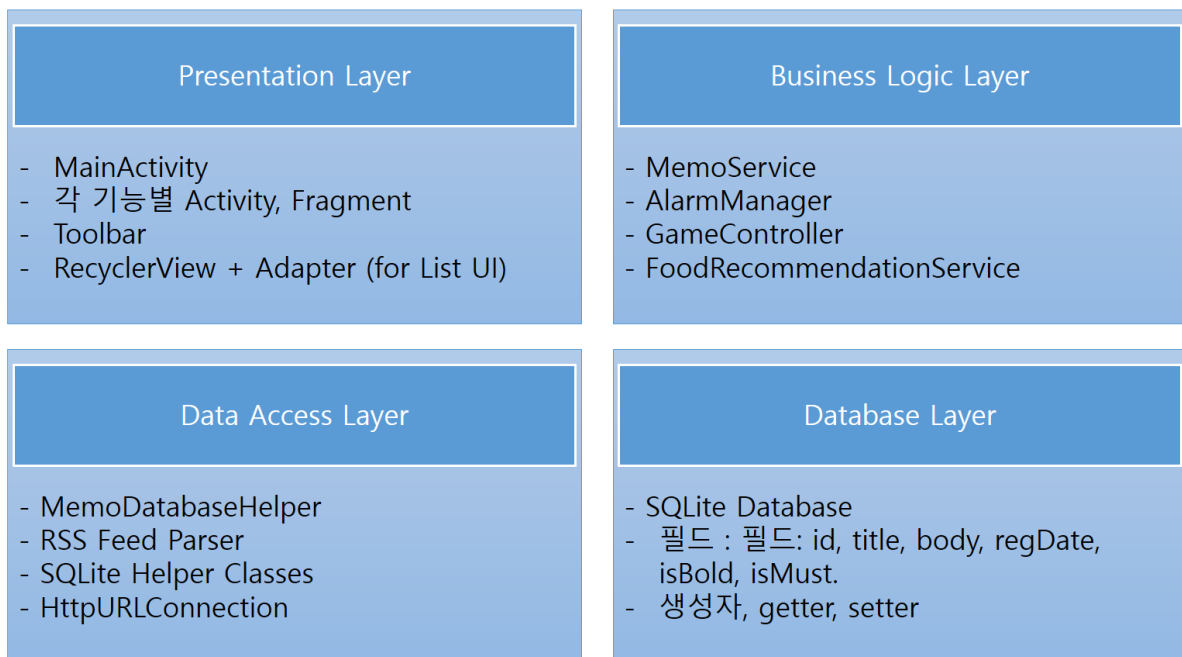
- `TextView` 에 상태(레벨, 점수, 시간 등)를 실시간으로 업데이트.

- **Animation 및 Delay:**

- 이미지 교체와 지연 처리 (`postDelayed`).

2) 레이어

- **UI 레이어:**
 - 메인 페이지와 각 기능 화면 (툴바 포함)
- **비즈니스 로직:**
 - 기능별 클래스와 데이터 처리 로직 (ToDo, 뉴스, 알람 등)
- **데이터 관리:**
 - SQLite 데이터베이스와 외부 RSS 피드 활용



3) 데이터

Database Class (UML class Diagram)

1. MemoDto 클래스:

- 메모의 데이터 구조를 정의합니다.

- 필드: `id`, `title`, `body`, `regDate`, `isBold`, `isMust`.
- 생성자 및 getter/setter 메서드가 포함되어 있습니다.

2. MemoDatabaseHelper 클래스:

- SQLite 데이터베이스를 관리합니다.
- 메서드:
 - `onCreate` 와 `onUpgrade`: 테이블 생성 및 업그레이드를 처리.
 - `getAllMemos`: 모든 메모를 가져옵니다.
 - `saveMemo`: 새 메모를 저장.
 - `updateMemo`: 메모 업데이트.
 - `getMemoById`: 특정 ID로 메모 검색.
 - `deleteMemo`: 메모 삭제.

데이터 흐름

1. **사용자 입력:** 사용자가 UI를 통해 데이터를 입력/조작.
2. **비즈니스 로직:** 입력된 데이터를 서비스 클래스에서 처리.
3. **데이터 액세스:** 데이터베이스 헬퍼를 통해 데이터를 저장하거나 로드.
4. **UI 업데이트:** 처리 결과를 UI로 다시 전달해 화면 업데이트.

4) 세부 로직

To-Do 리스트

- 데이터 흐름:
 - SQLite Helper → DTO → Adapter → RecyclerView
- 기능:

- 메모 추가, 수정, 삭제
- Long-press: 삭제/강조 표시
- NFC 태그 → 특정 메모로 이동
- FAB 버튼 → 새로운 메모 작성

이하 하단 III)주요 기능에서 설명

III. 주요 기능 : 주요 구현 코드와 설명

i. 필수 구현 내용 및 ii. 추가 구현 내용

0. 메인화면 : 각 버튼을 클릭하여 개별 기능 시행

1. ToDo(할일) 관리

■사용 기술

- BroadcastReceiver, LocalBroadcastManager
- ListView
- FloatingActionButton (FAB)
- AlertDialog
- DatabaseHelper
- DTO (Data Transfer Object)
- NFC (Near Field Communication)
- lastProcessedTag
- onActivityResult
- notifyDataSetChanged
- bytesToHex

■필수 구현

- SelfTest 및 수업 내용을 기반으로 할일 관리기능 구현
- 등록, 수정 및 조회, 삭제 기능(DB헬퍼와 DTO활용)

■추가 구현

- 할일(TODO) 상태와 한일(DONE) 상태 구분 - 회색으로 변환
- 상태별 조회 - 중요표기 (색 변환 및 제목에 별모양 추가)
- NFC기능 추가 : idx:num형태의 nfc수신하여 num번째 문서로 이동 및 없는 번호일경우 생성으로 이동 및 lastProcessedTag을 활용하여 중복태깅 방지
- bytesToHex- 헬퍼 메서드로 태그 ID를 처리.
- 연결된 콘텍스트 메뉴 - 목록의 글을 누르면 위의 상태를 지정가능, 삭제가능
- 메모 리스트에 작성된 메모의 개수에 따라 변동되는 배경라인

2. 뉴스 보기

- 기술
 - HttpURLConnection
 - XML Parser
 - AsyncTask
- 필수 구현
 - 특정 신문사의 RSS 뉴스기사 목록 출력
 - 뉴스 목록 선택시 해당 뉴스 화면으로 이동
- 추가 구현
 - 탭 형식으로 여러 신문사 구독
 - 해당 기사의 작성 일 및 신문사 지정 카테고리가 존재할경우 표기

3. 알람관리

- 사용 기술
 - AlarmManager (알람 관리)
 - Calendar (날짜 및 시간 관리)
 - PendingIntent (알람 설정 및 취소)
 - CountdownTimer (타이머 및 UI 업데이트)
 - Dynamic View Management (뷰 동적 업데이트)
 - DatePickerDialog & TimePickerDialog (날짜 및 시간 선택 다이얼로그)

- SystemClock (경과 시간 계산)
- Settings.ACTION_MANAGE_OVERLAY_PERMISSION (다른 앱 위에 그리기 권한 처리)

■기본 구현

- 설정한 시간(sec)후 작동하는 타이머
- 날짜와 시간을 설정후 작동하는 알람
- 알람등록과 등록취소
- 설정한 알람까지 남은 시간을 표기해주는 아이콘과 텍스트
- 별개의 페이지로 구현되는 알람
- 사전 허가를 통해 다른 화면 혹은 앱을 사용하고 있더라도 그 위에 알람을 표기
- 애니메이션을 활용하여 빠르게 변화하는 화면 색을 통해 기상에 도움을 줌

4. (조개) 잡기 게임

■사용 기술

- (Multi) Thread
- AsyncTask
- Custom Widget
- Handler & Looper - UI update
- Dynamic View Management

■기본 구현

- 캐릭터가 랜덤한 위치에 나타남
- 캐릭터를 선택하여 득점
- 캐릭터와 배경, BGM 등을 기획, 사용

■추가 구현

- 게임시작 버튼을 통해 게임을 시작할수 있고, 상단에 플레이 방식을 표기하는 ui
- 페이지 상단에 현재 레벨, 잡은 개수, 지난 시간을 표시하는 UI
- 클리어시 시간을 표기하며 이를 기반으로 플레이 랭크(S,A,B,C등) 표기
- 다음 진행 여부 확인 및, 아니오 클릭시 되묻기, 선택지 무시후 배경클릭으로 진행했으면 다시 여부를 확인하도록 강제
- 조개를 누르는데 성공시 해당 위치에 열린 진주조개 표기
- Level Up에 따른 속도 증가
- 바다사자 캐릭터 표시 후 선택시 감점 기능

추가요소

5. 저녁메뉴 추천

■사용 기술

- ArrayAdapter & Spinner
- HashMap
- ArrayList & Arrays:(Dynamic)

■구현

- 스피너를 통해 메뉴 카테고리 와 타입 선택
- 탐색 버튼을 통해 이동한 화면에서, 토너먼트형식으로 보다 선호하는것을 선택해나가면 최종적으로 한 요소가 선택되며, 해당요소로 스피너가 자동 변환
- 추천 버튼 클릭시 두 타입을 기준으로, 미리 저장된 데이터 기반 추천 음식 목록을 하단에 생성

6. 툴바 및 사이드바

FrameLayout, Button, ImageView, TextView와 같은 Android 뷰를 활용하여 UI를 구성.

Toolbar와 DrawerLayout을 사용해 상단 툴바 및 사이드바 메뉴를 구현.

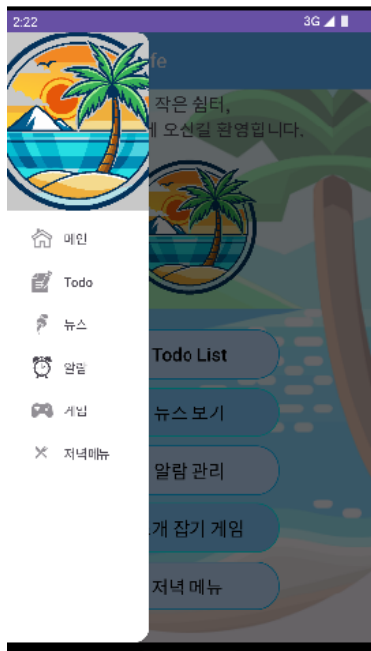
NavigationView로 네비게이션 메뉴 구성.

각자에서 사용되는 사진은 res밑의 drawable에서,

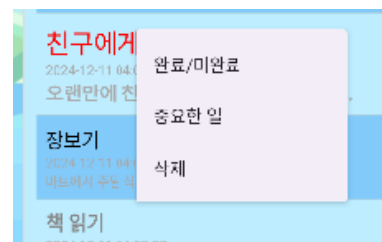
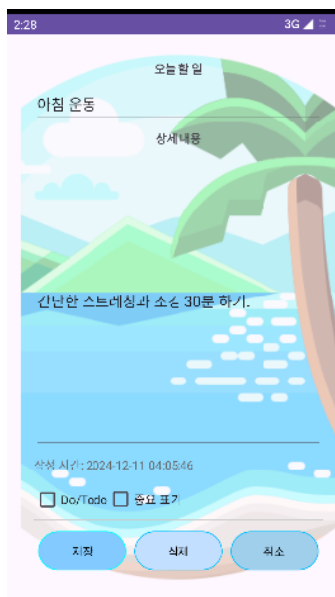
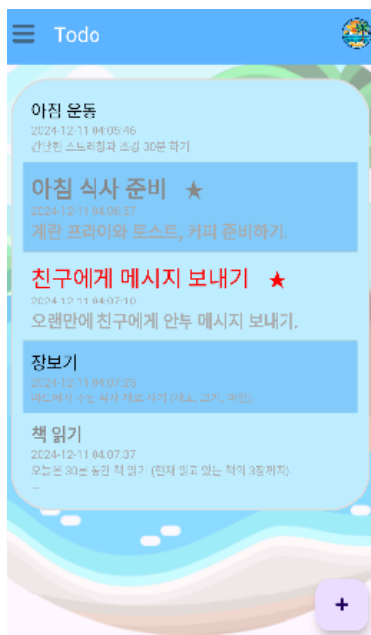
툴바, 사이드바 등에 사용되는 아이콘은 values/themes에서vector asset으로 재구성한 이미지를 사용

IV. 실행화면 : 주요 실행화면 캡처와 설명

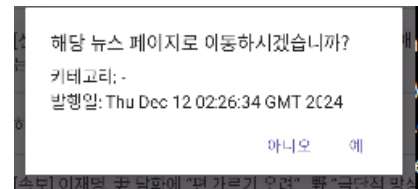
1. 메인 페이지 / 툴바 / 사이드 바



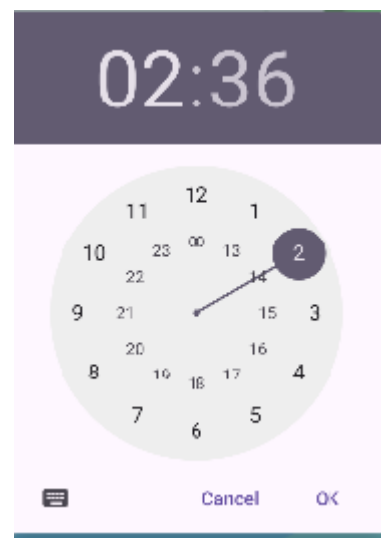
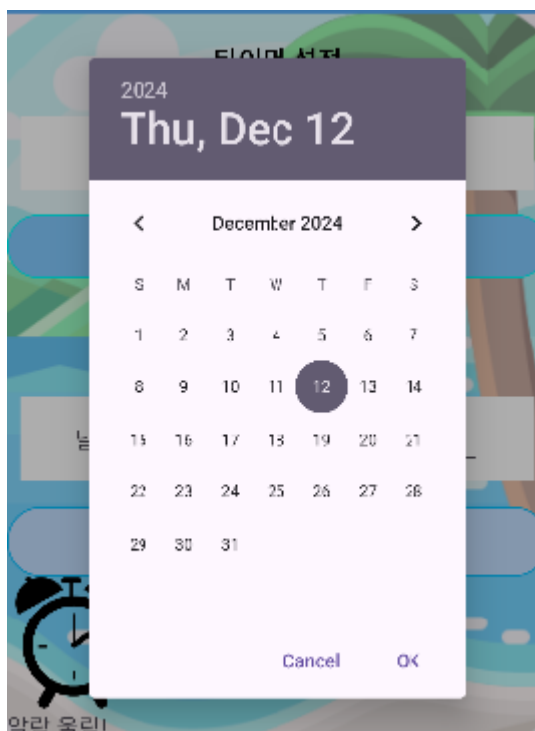
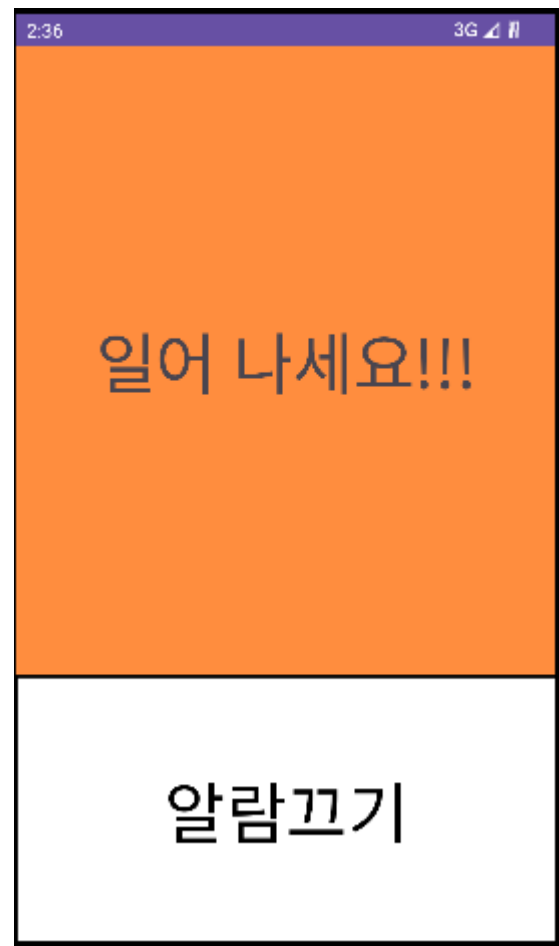
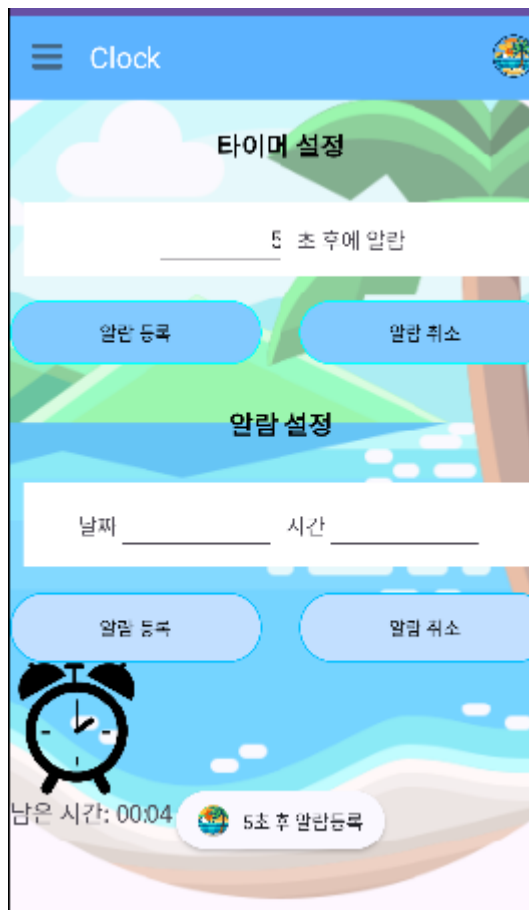
2. Todo List



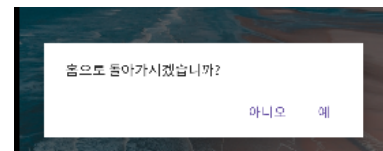
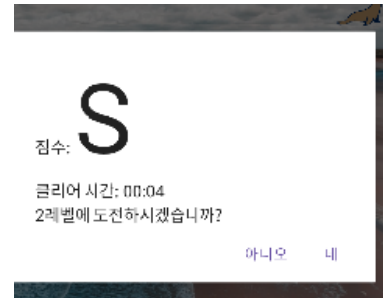
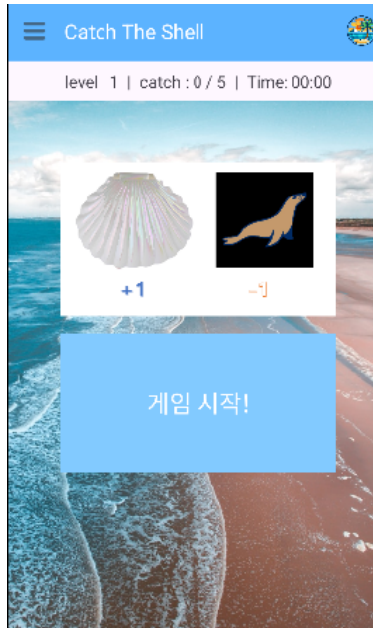
3. 뉴스



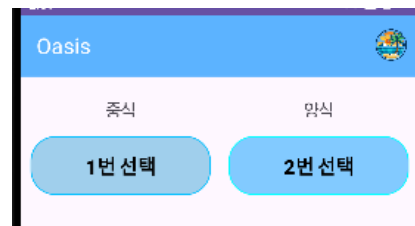
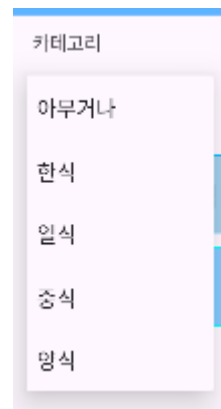
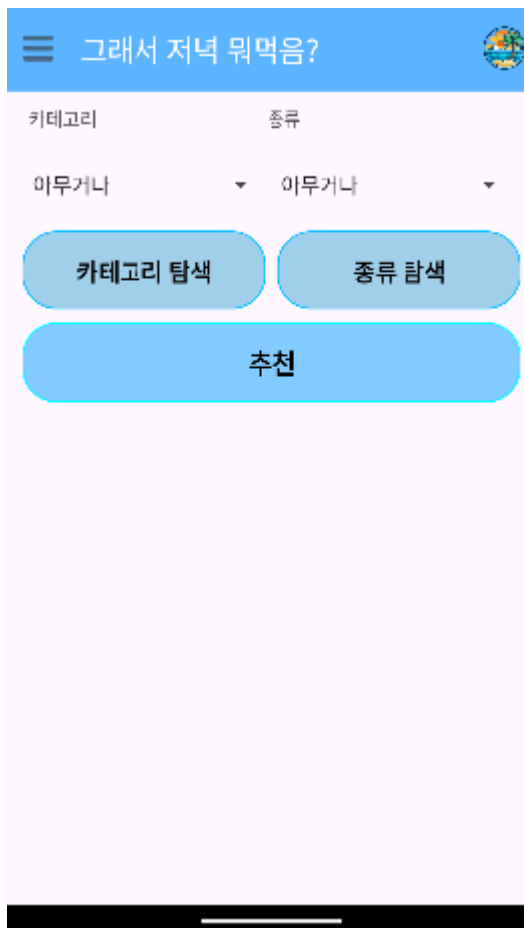
4. 알람



5. 조개 잡기 게임



6. 저녁메뉴 탐색





E. 향후 개선점 : 미흡한 점이나 추가 구현해보고 싶은 내용

알람 기능을 현재 새로 생성시 이전 알람을 갱신하도록 제작되었으나,
리스트 형태로 수정하여 다수의 알람을 동시 설정 가능하게 개선하고 싶습니다.