

약속해

A stylized icon of a mortar and pestle. The mortar is white with a red cross on its front. The pestle is green and is shown in the process of grinding a white pill inside the mortar. There are two more white pills floating above the mortar.

자주 챙겨먹는 영양제와 복용하는 약
사진으로 등록하고, 관리하세요!

20143044 김의연
20163138 이산하
20161671 이채현
20163159 조성래

Contents

2022 국민대학교
캡스톤 디자인 47조

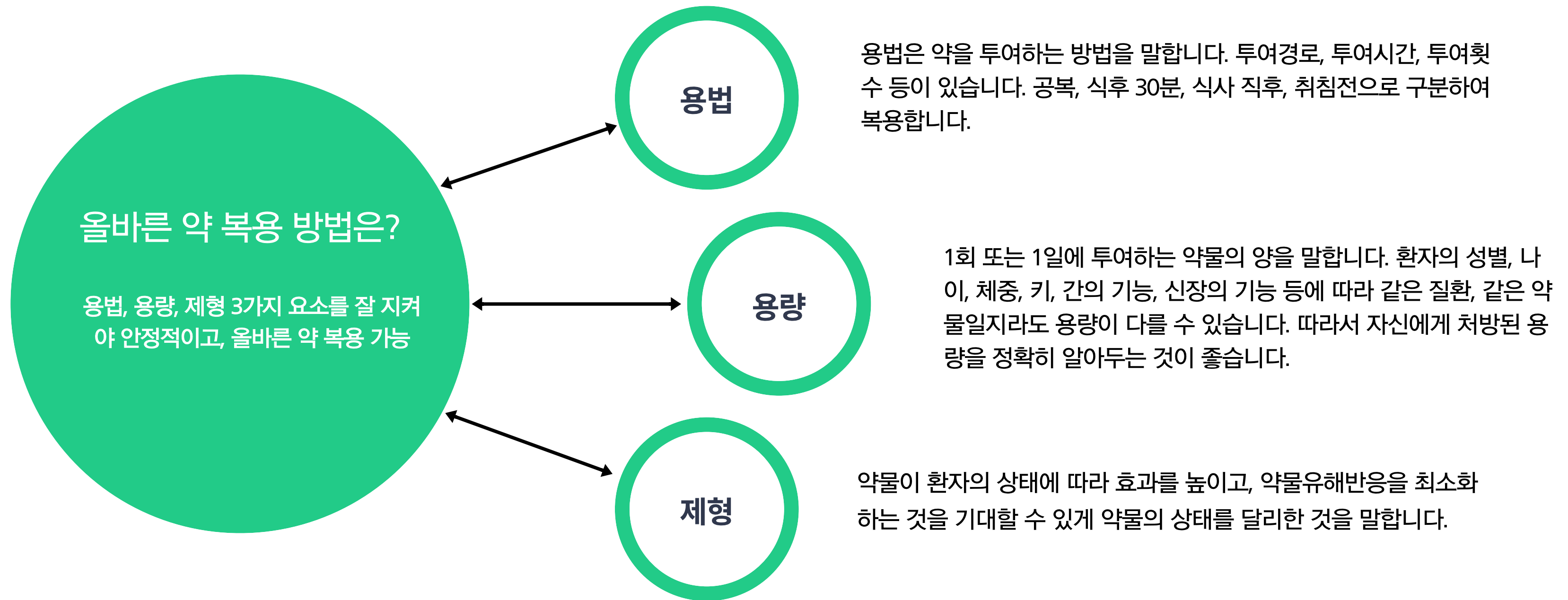


- 프로젝트 필요성 및 목표
- 개발내용
- 기대효과

20143044 김의연
20163138 이산하
20161671 이채현
20163159 조성래

까다로운 약 복용방법

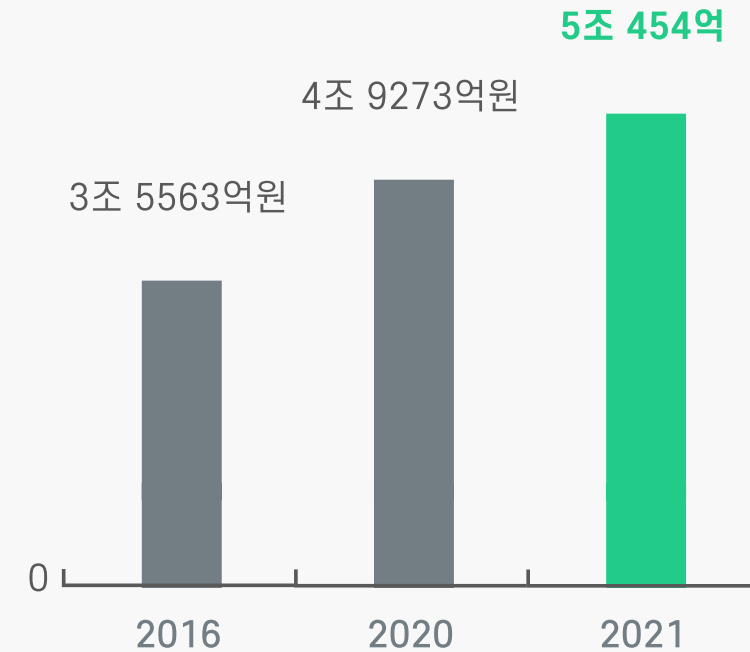
약은 사용에 따라 효과적인 증상의 완화를 불러 올수도, 질병의 악화를 불러 올수도 있기에 약의 올바른 복용은 그 무엇보다 중요합니다.



식약처 건강기능 식품 소비자 인식도 조사, 한국건강기능식품협회 조사

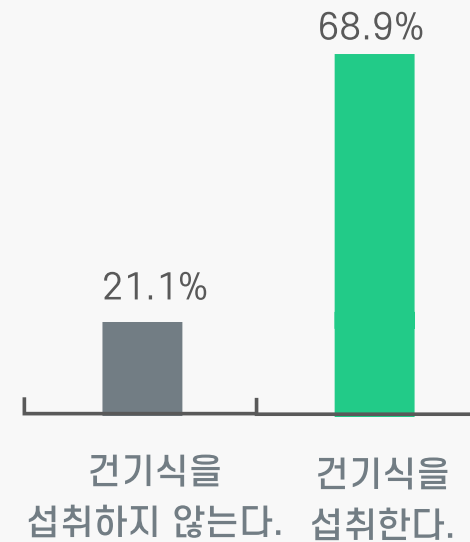
건강기능식품 섭취 증가와 시장의 확대

매년 성장하는
건강기능식품 시장



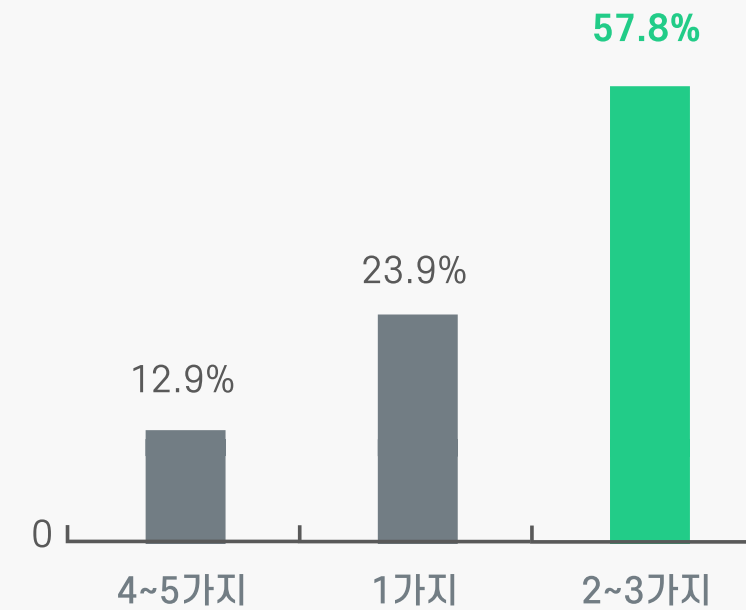
건강기능식품
시장

성인 10명중 7명 건
강기능 식품 섭취



1인당 건강기능식품
섭취 개수

다양한 건강기능식품
섭취



1인당 건강기능식품
섭취 개수

약속해는?

사진 촬영 기능과 약에 대한 이미지를 저장소에서 불러올 수 있습니다.



복용 기록을 남겨 약의 중복 복용을 방지하고, 문제가 생길 경우 복용 기록을 바탕으로 원인을 찾을 수 있습니다.



효율적인 약물관리



복용해야 할 약과 건강기능식품을 시간에 맞추어 알림을 설정, 원활한 관리를 할 수 있게 도와줍니다.



근처 약국의 위치를 제공하여 필요할 때 간편하게 위치를 찾을 수 있습니다

개발과정

STEP 1

1차 스케치 초안
github 페이지
구성

STEP 2

flutter, Dart
학습

스케치 초안 바탕으로
diagram 작성

스케치 초안
화면 구성
(홈, 설정, 기록)

주요 기능 구현

최종 테스트

STEP 3

어플리케이션
배포

단순 알람 및 알람 스케줄링
사진 촬영 및 갤러리에서 사진 불러오기
로컬 데이터 저장소 사용
각종 UI, 위젯

develop

<Test Model>

Google

Pixel 2



Flutter

SDK 2.10.4



Dart

v3.36.0

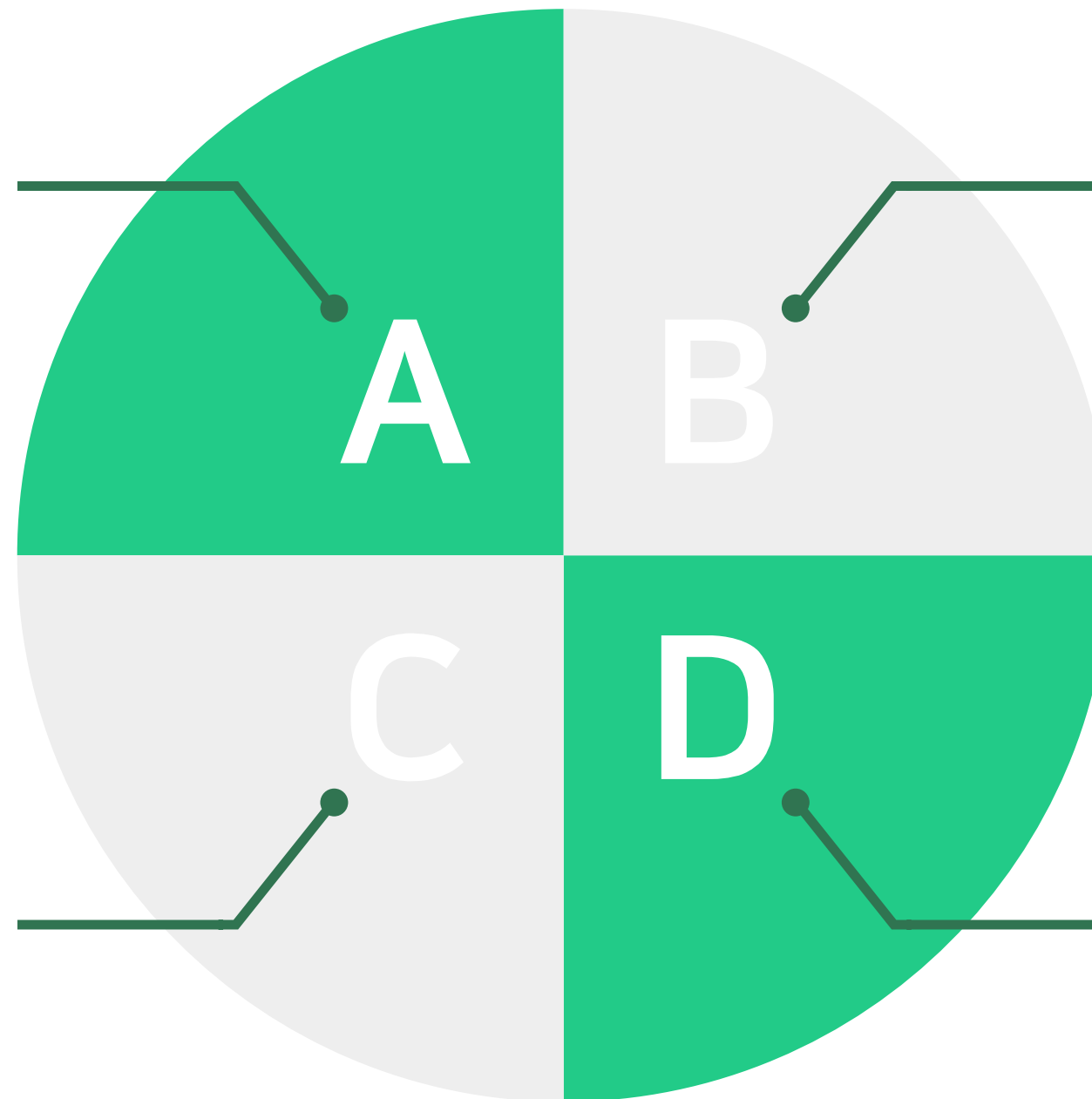


version 1.66



2021.1.1 Patch 2

Android Studio

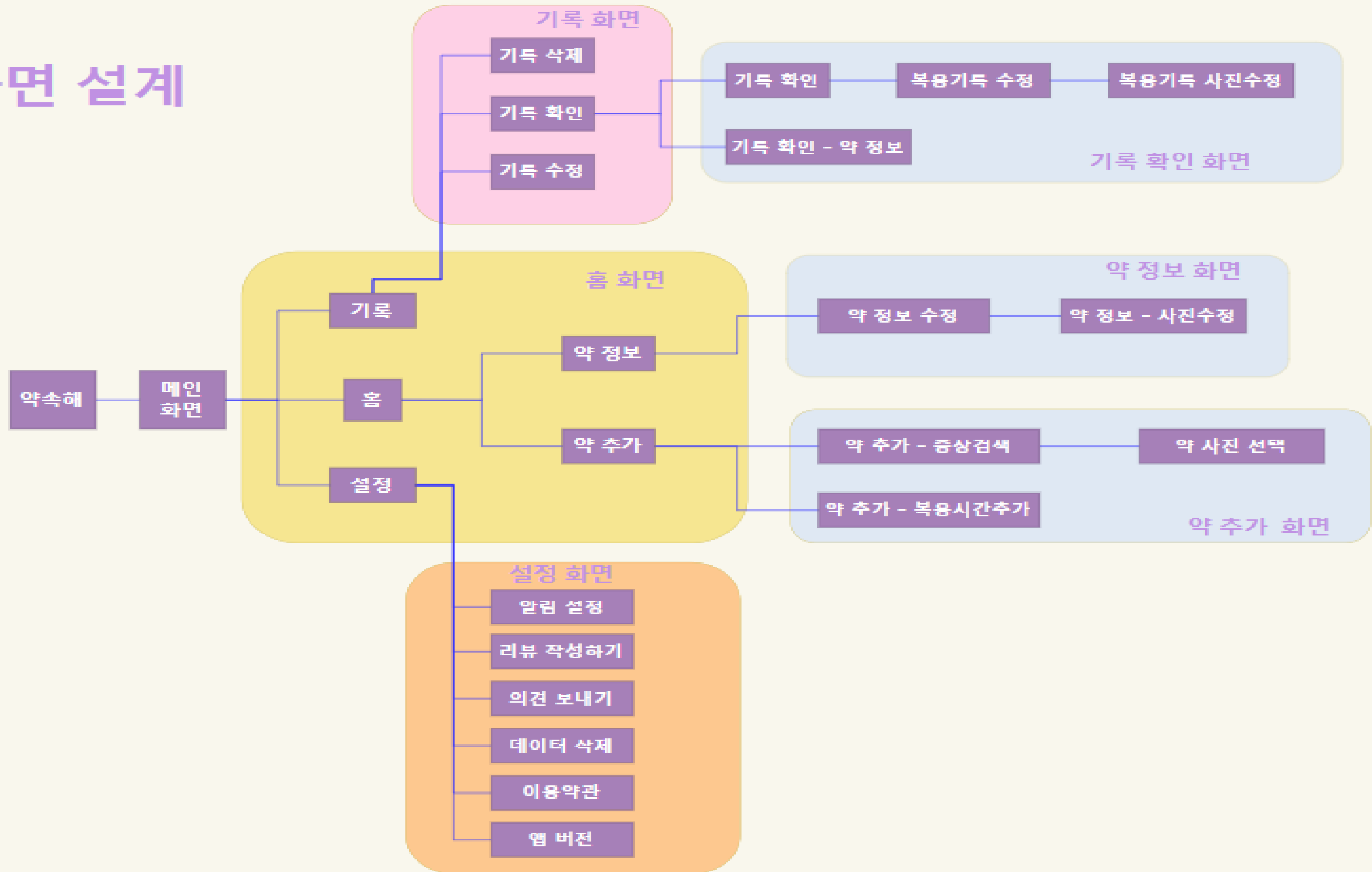


git

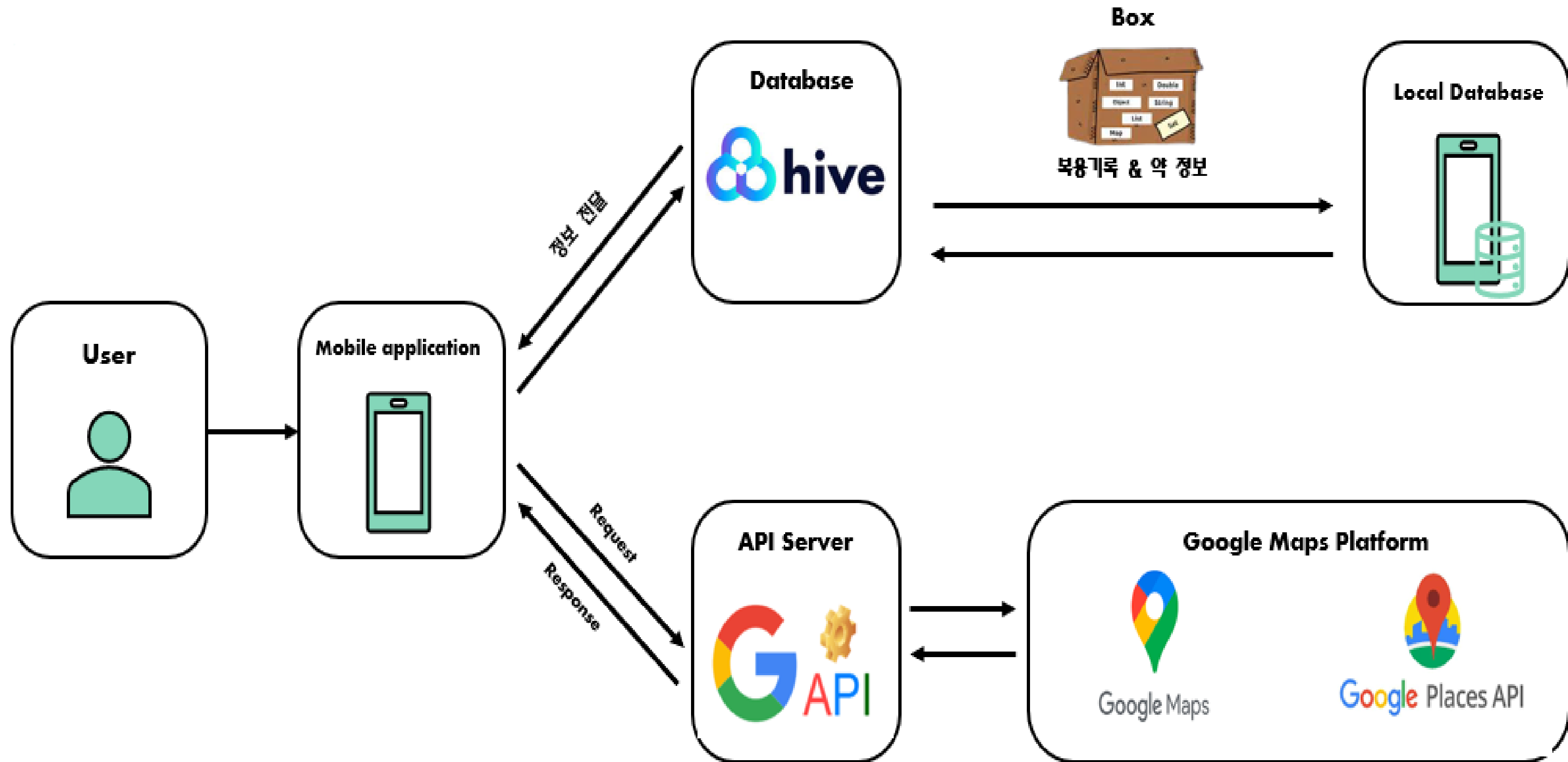
v2.35.1



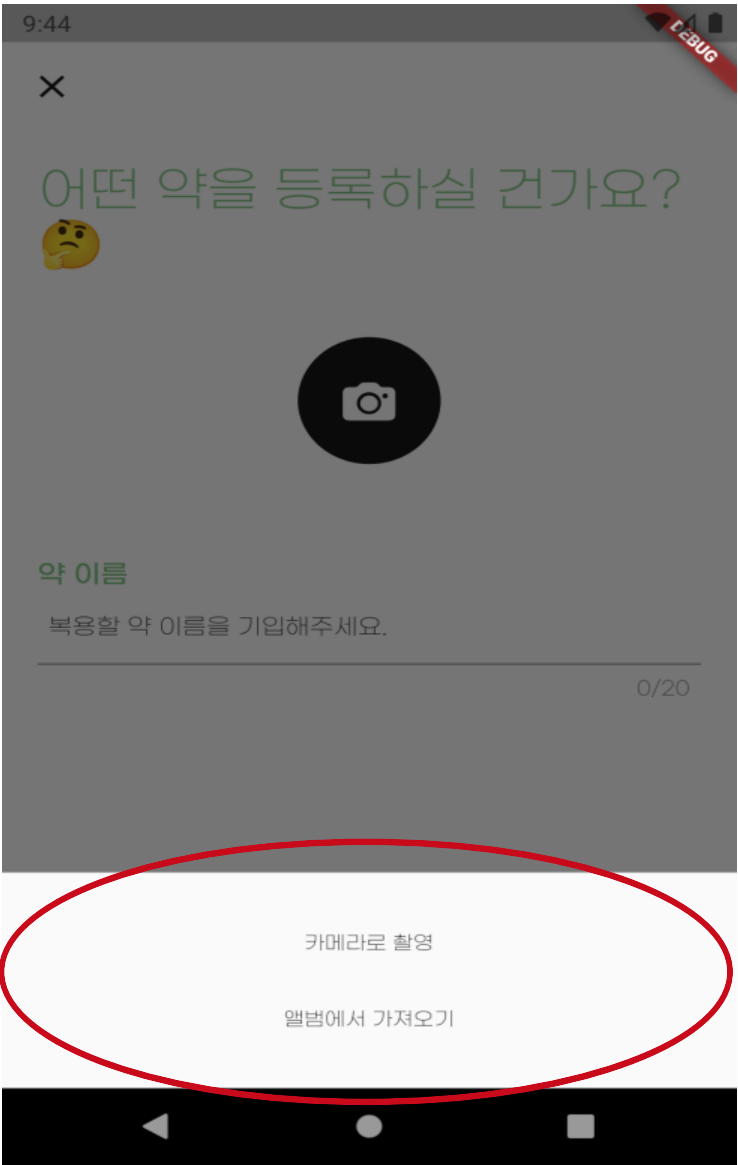
화면 설계



System Structure



주요 기능 & 구현 방법 - 약 정보 등록



ImgaePicker()



flutter_local_nofications



Database



사진 촬영 & 앨범에서 업로드



시간 및 알람 설정

주요 기능 & 구현 방법 - 약 정보 등록(1)

- main.dart

```
Run | Debug | Profile
void main() async {
  WidgetsFlutterBinding.ensureInitialized();

  await notification.initializeTimeZone();
  await notification.initializeNotification();

  await hive.initializeHive();
  await initializeDateFormatting();
  runApp(const MyApp());
}
```

- Hive DB 초기화

```
class YaksokHive{
  Future<void> initializeHive() async{
    await Hive.initFlutter();

    Hive.registerAdapter<Medicine>(MedicineAdapter());
    Hive.registerAdapter<MedicineHistory>(MedicineHistoryAdapter());

    await Hive.openBox<Medicine>(YaksokHiveBox.medicine); //medicine이라는 Box
    await Hive.openBox<MedicineHistory>(YaksokHiveBox.medicineHistory); //medicine_history라는 Box
  }

  class YaksokHiveBox{
    static const String medicine = 'medicine';
    static const String medicineHistory = 'medicine_history';
  }
}
```

- Hive.openBox('medicine')

- medicine Box를 생성 repository에서 관리

주요 기능 & 구현 방법 - 약 정보 등록(2)

```
class MedicineRepository{
  Box<Medicine>? _medicineBox;

  Box<Medicine> get medicineBox{
    _medicineBox ??= Hive.box<Medicine>(YaksokHiveBox.medicine);
    return _medicineBox!;
  }

  void addMedicine(Medicine medicine) async{
    int key = await medicineBox.add(medicine);

    log('[addMedicine] add (key:$key) $medicine');
    log('result ${medicineBox.values.toList()}');
  }

  void deleteMedicine(int key) async{
    await medicineBox.delete(key);

    log('[deleteMedicine] delete (key:$key)');
    log('result ${medicineBox.values.toList()}');
  }

  void updateMedicine({
    required int key,
    required Medicine medicine,
  }) async {
    await medicineBox.put(key, medicine);

    log('[updateMedicine] update (key:$key) $medicine');
    log('result ${medicineBox.vlaues.toList()}');
  }

  int get newId{
    final lastId = medicineBox.values.isEmpty ? 0 : medicineBox.values.last.id;
    return lastId+1;
  }
}
```

```
Widget build(BuildContext context) {
  return CupertinoButton(
    onPressed: () {
      showModalBottomSheet(context: context, builder: (context) {
        onPressedUpdate: () { //약 정보 수정
          Navigator.push(context, FadePageRoute(page: AddMedicine));
        },
        onPressedDeleteMedicine: () { //약 정보 삭제
          //알람 삭제
          //hive 약 데이터 삭제
          notification.deleteMultipleAlarm(alarmIds);
          medicineRepository.deleteMedicine(medicineAlarm.key);
          Navigator.pop(context);
        },
        onPressedDeleteAll: () { //약 정보와 히스토리 모두삭제
          //알람 삭제
          //hive 히스토리 데이터 삭제
          //hive 약 데이터 삭제
          notification.deleteMultipleAlarm(alarmIds);
          historyRepository.deleteAllHistory(keys);
          medicineRepository.deleteMedicine(medicineAlarm.key);
          Navigator.pop(context);
        },
      ), // MoreActionBottomSheet
    ),
  );
}
```

- 약 객체 모델 삭제

- medicine_repository -> 생성된 약 객체 모델 저장

주요 기능 & 구현 방법 - 약국 검색(1)

(1) Google API를 사용하기 위해 pubspec.yaml파일에 패키지를 추가

Google Cloud Platform에서 구글 Open API 키 값을 받아 AndroidManifest.xml에 추가

```
dependencies:  
  flutter:  
    sdk: flutter  
  
  #google_map_api  
  google_maps_flutter: ^2.1.5  
  flutter_google_places: ^0.3.0  
  geolocator: ^8.2.1
```

- pubspec.yaml에 google_map_api를 설치

```
<!--google map-->  
<meta-data android:name="com.google.android.geo.API_KEY"  
  android:value="A-XXXXXXXXXXXXXXXXXXXX" />  
  
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>  
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

- AndroidManifest.xml

API 키

<input type="checkbox"/>	이름
<input type="checkbox"/>	✓ places api (Android)
<input type="checkbox"/>	✓ Maps API key (ios)
<input type="checkbox"/>	✓ Maps API key (android)

-> Google Cloud Platform에서 받은 API

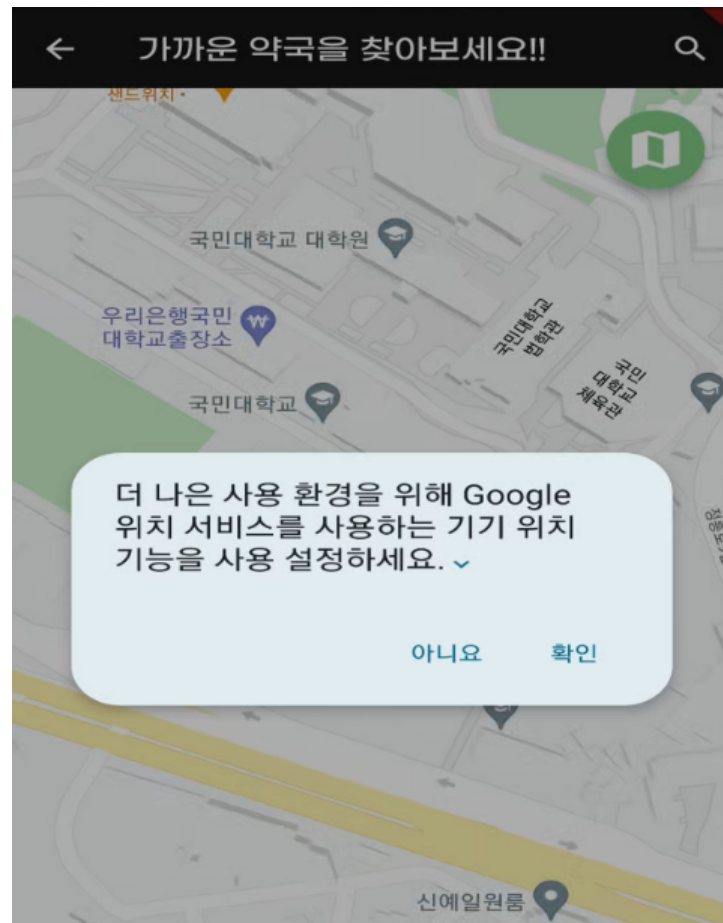
주요 기능 & 구현 방법 - 약국 검색(2)

- setting_map_page.dart -> SettingMapPage는 사용자의 위치정보를 받아오고 API 서버로부터 받아온 Google Map 정보를 표시함

<_getUserLocation()은 initState()에서 호출되어 위치정보 허용을 묻고, 사용자의 위치정보를 받아옴>

```
void _getUserLocation() async{
  LocationPermission permission = await Geolocator.requestPermission(); //지도
  Position position = await Geolocator.getCurrentPosition(desiredAccuracy: LocationAccuracy.high);
  setState(() {
    _initialPosition = LatLng(position.latitude, position.longitude);
  });
}
```

```
GoogleMap(
  myLocationButtonEnabled: true,
  myLocationEnabled: true,
  mapType: _currentMapType,
  //위성지도와 기본지도 변경 함수
  initialCameraPosition: CameraPosition(target: _initialPosition??
  onMapCreated: (GoogleMapController controller) {
    setState(() {
      _controller = controller;
    });
  },
  markers: markers.toSet(),
  onTap: (coordinate) {
    _controller.animateCamera(CameraUpdate.newLatLng(coordinate));
    addMarker(coordinate);
  },
), // GoogleMap
```



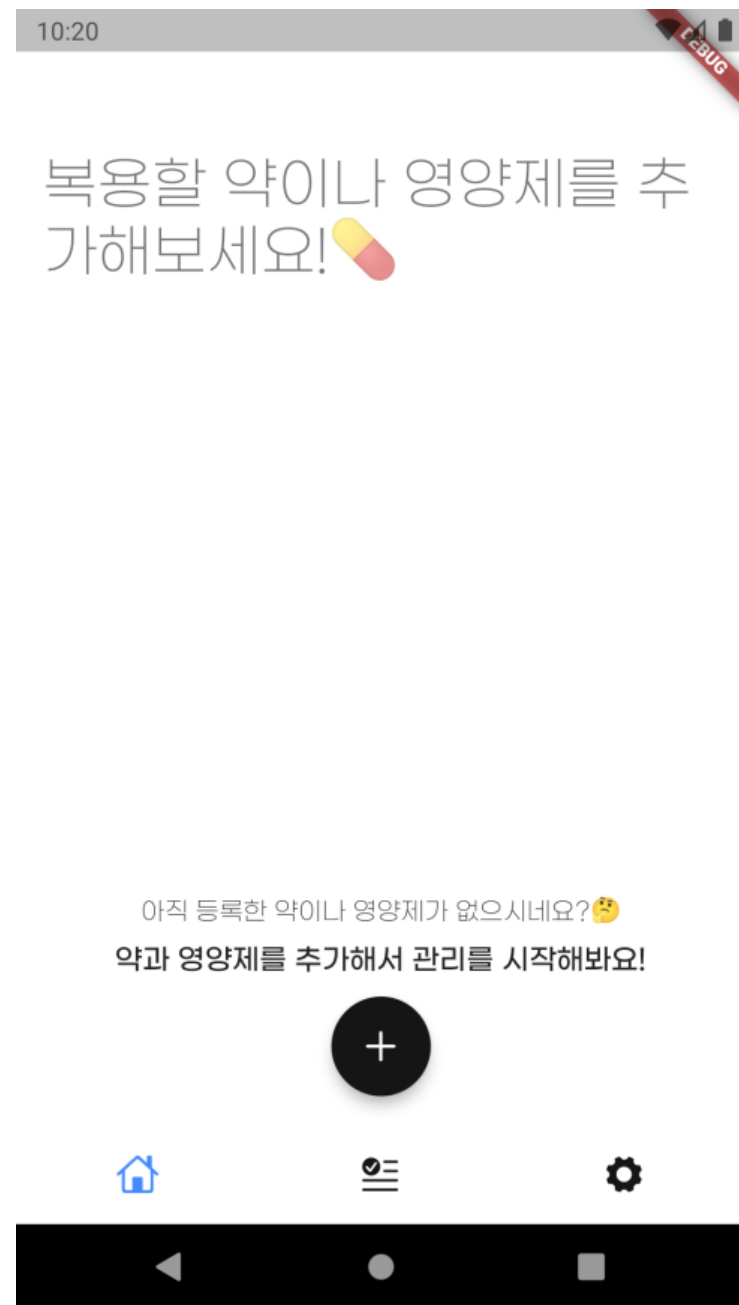
위치 정보 권한 설정



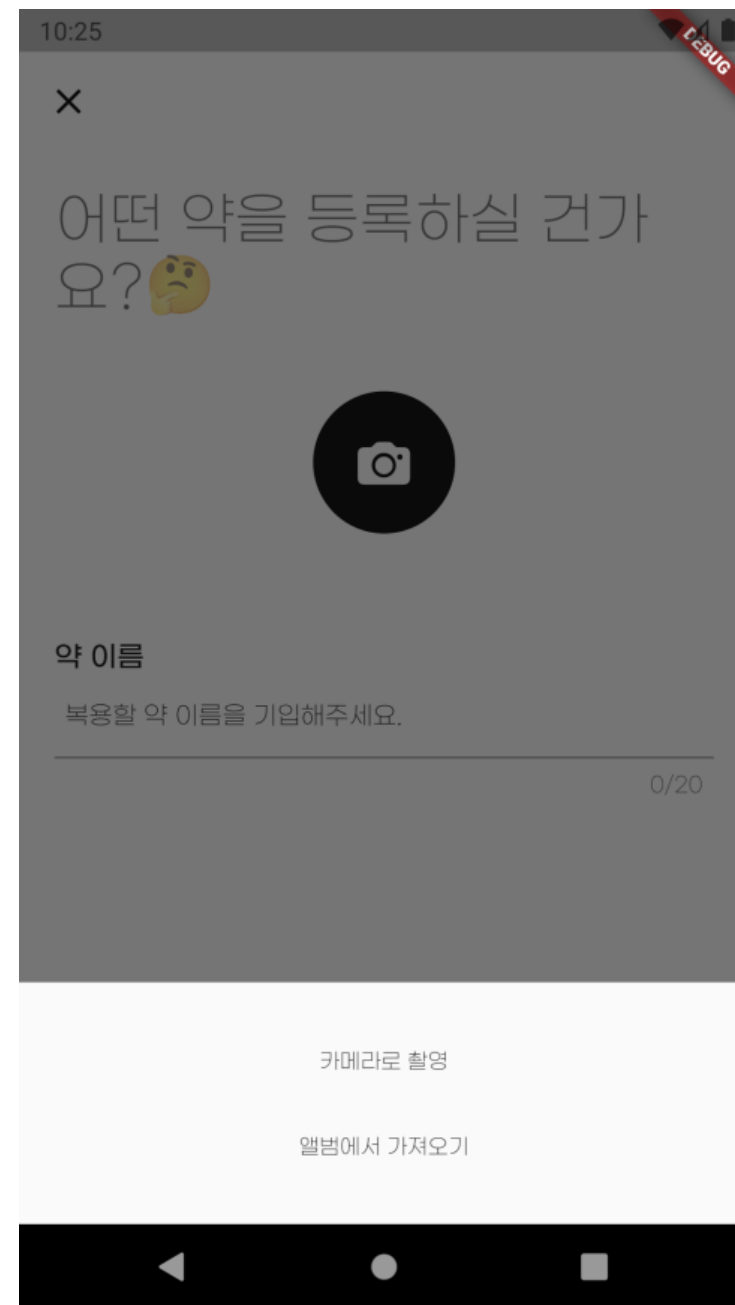
위치 정보 기반 검색

1. 복용할 약 설정

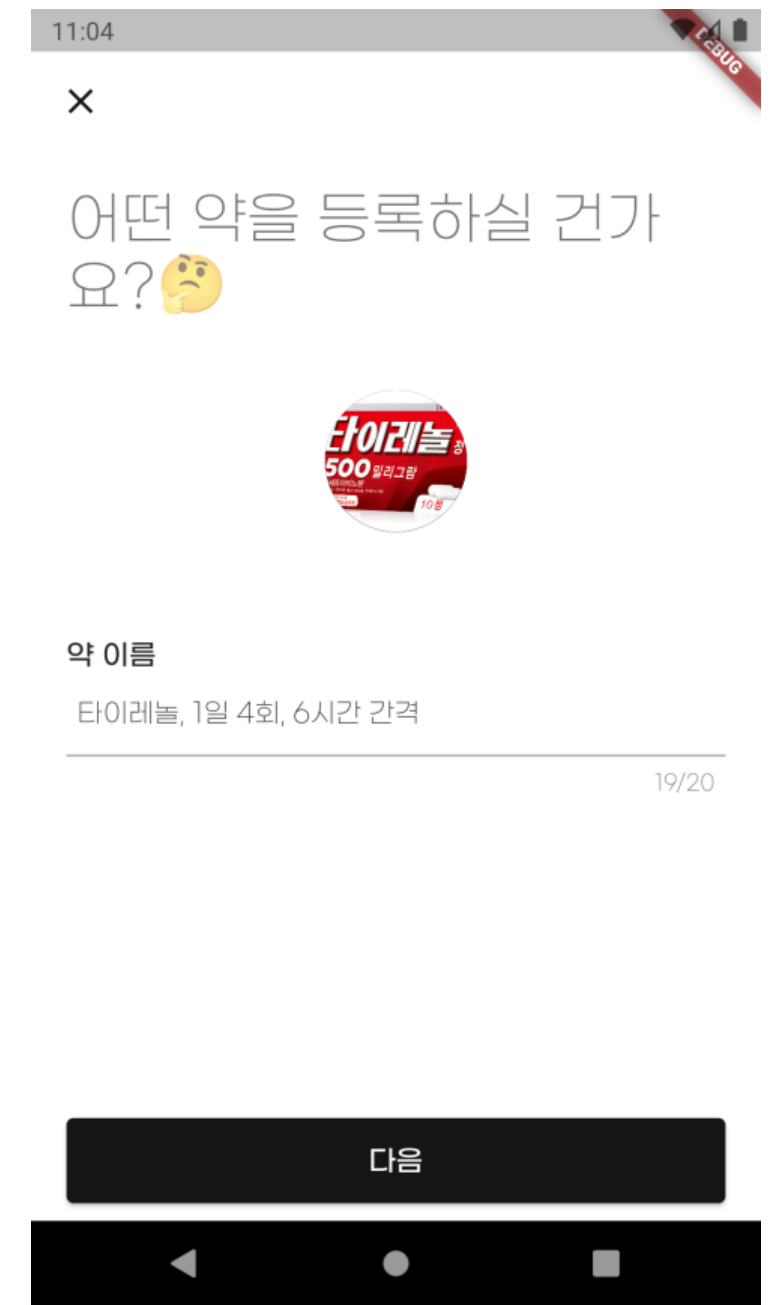
저장소에서 약의 이미지를 불러오거나 사진으로 촬영하여 약에 대한 사진을 등록합니다.
약의 이름 옆에 간단한 정보를 추가적인 정보를 메모합니다 (ex. 복용시간, 간격)



<최초 실행 화면>



<이미지 등록>



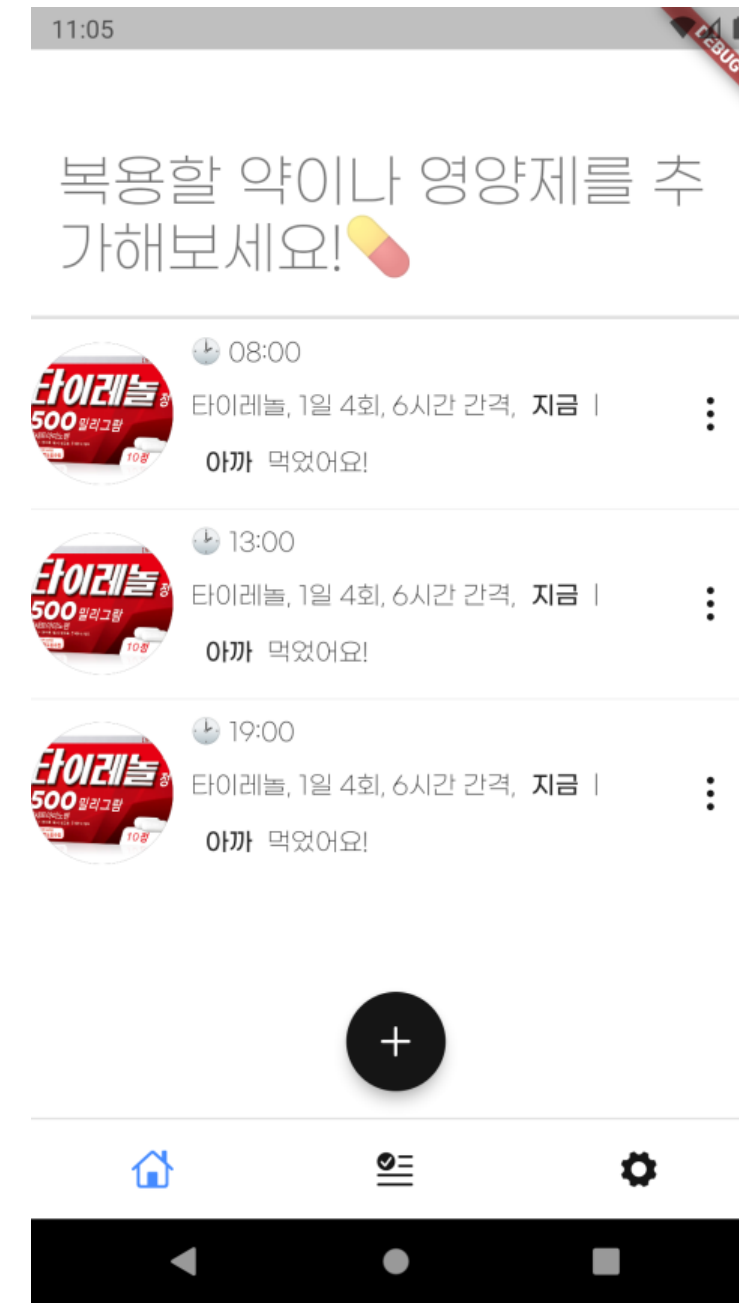
<약 이름 입력>

2. 복용 시간 설정

약 복용 시간을 설정하여 설정한 시간이 오면 알림이 올 수 있도록 설정합니다.



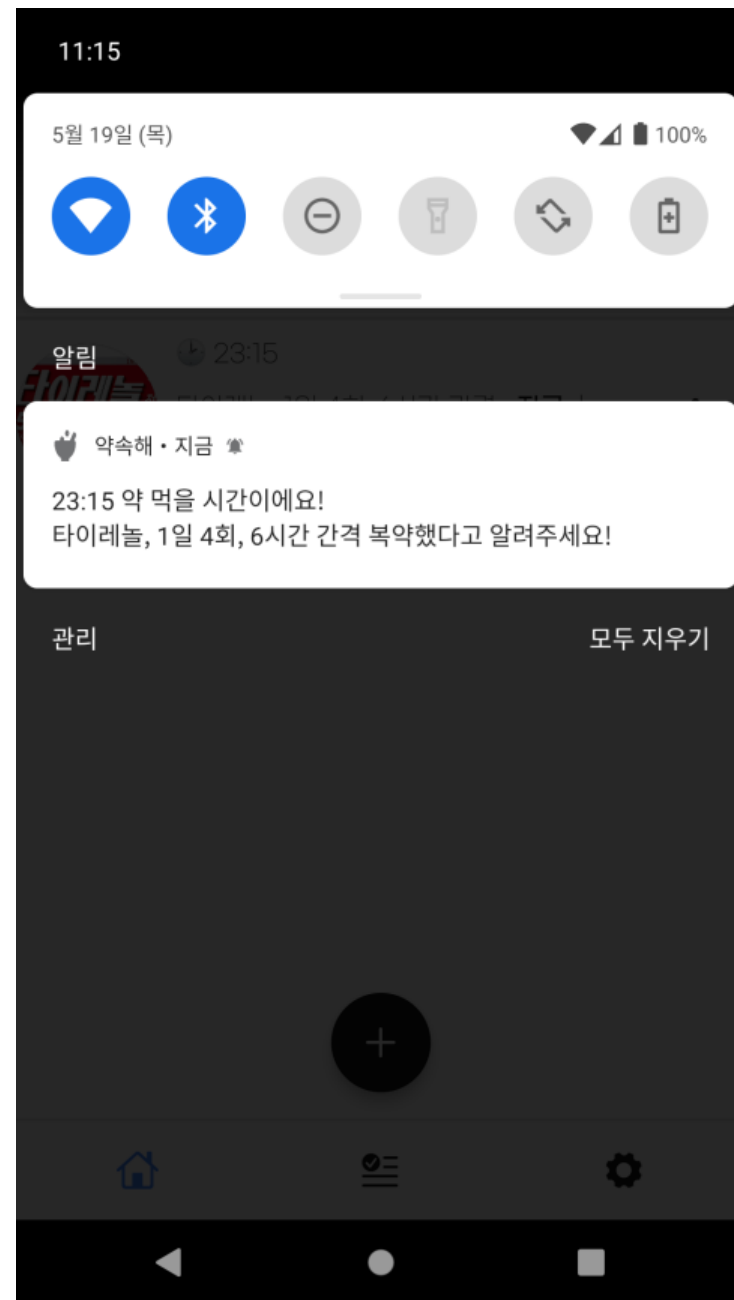
<시간 설정>



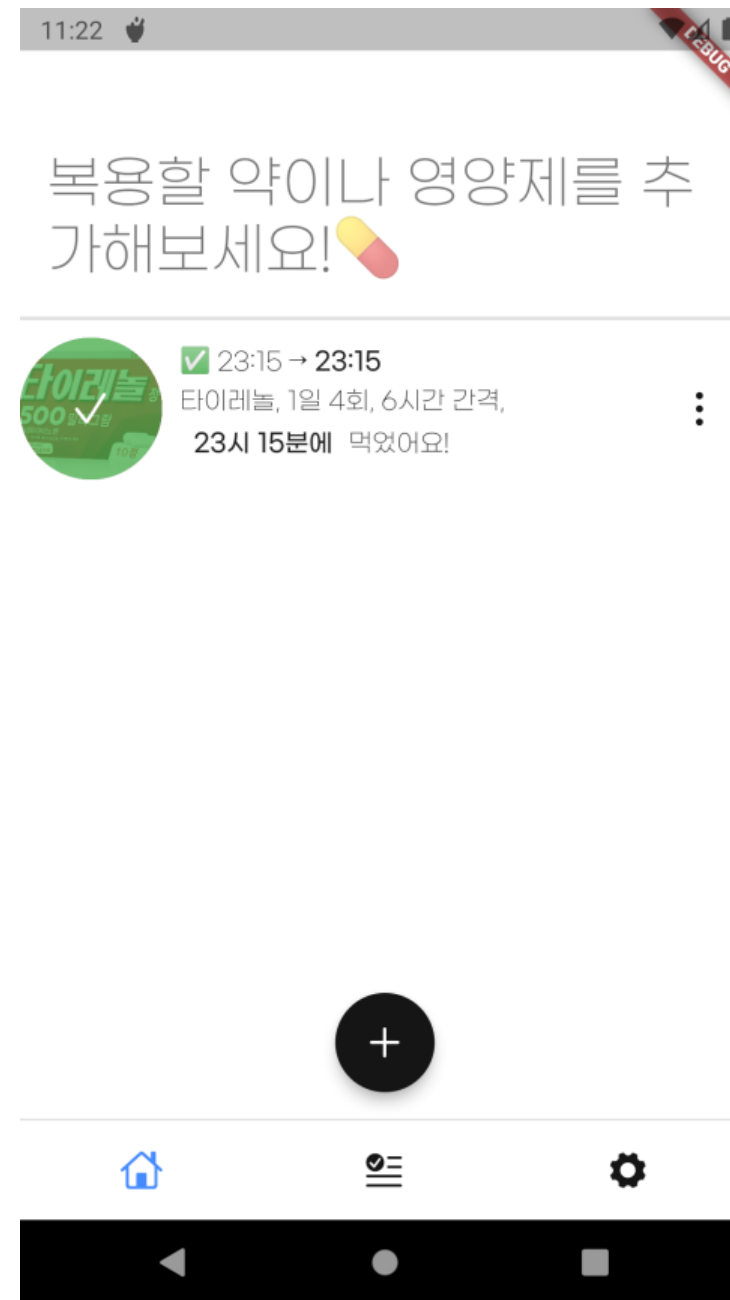
<등록 완료>

3. 알림 및 복용기록 확인

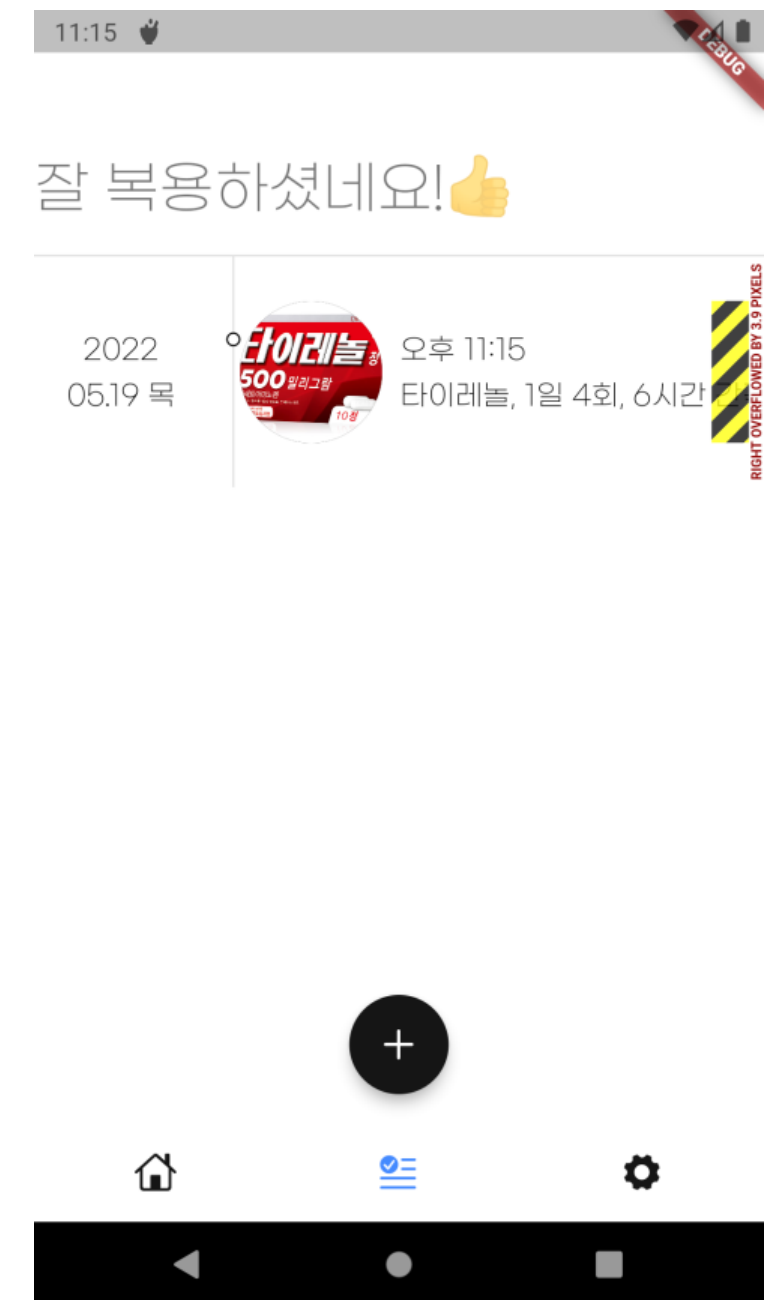
복용할 시간이 되면 알림이 울리고, 체크시 자동으로 복용기록을 남기게 됩니다.



<알림확인>



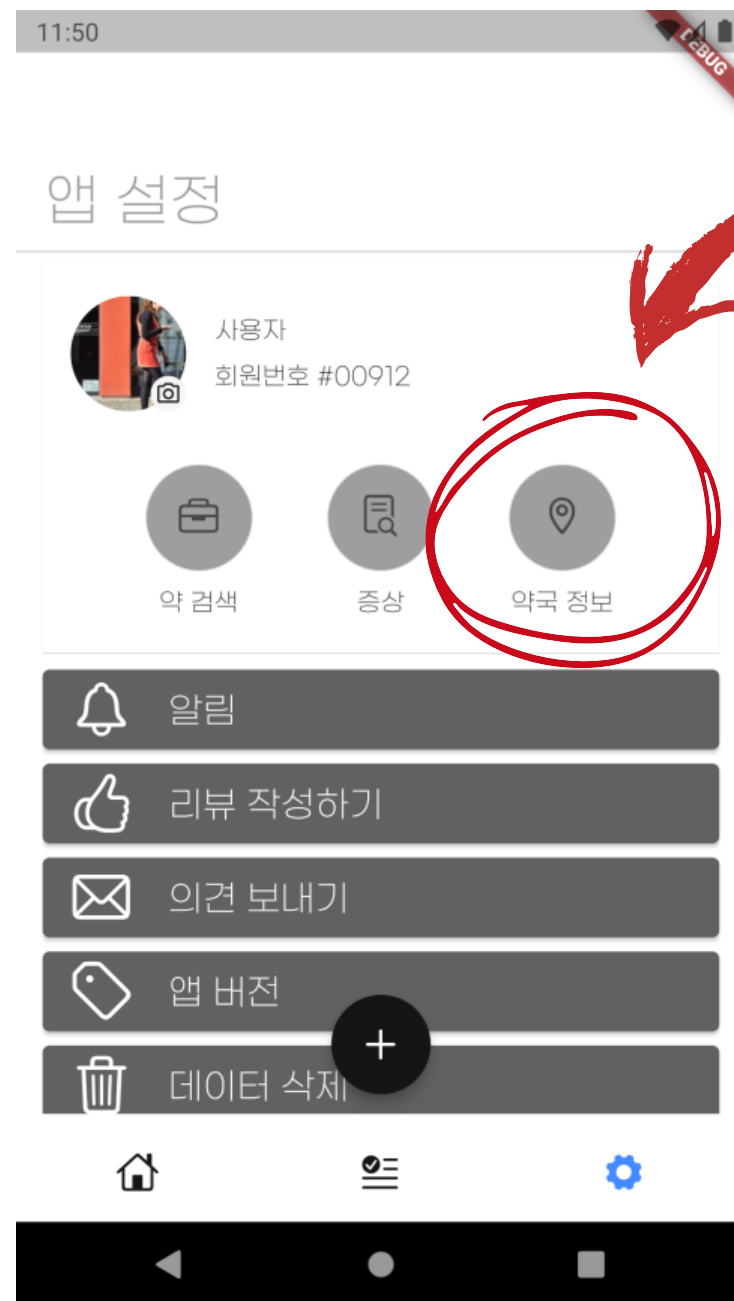
<복용체크>



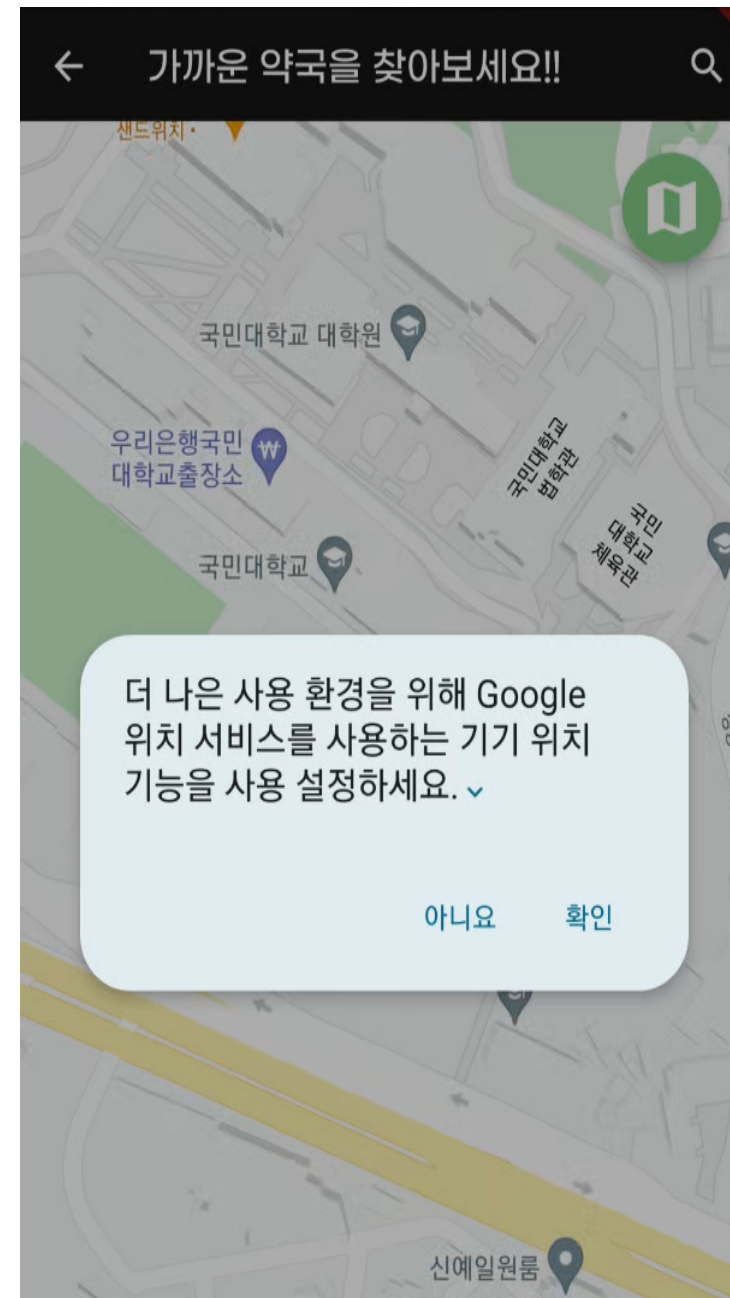
<복용 기록 저장>

지도 & 약국 검색

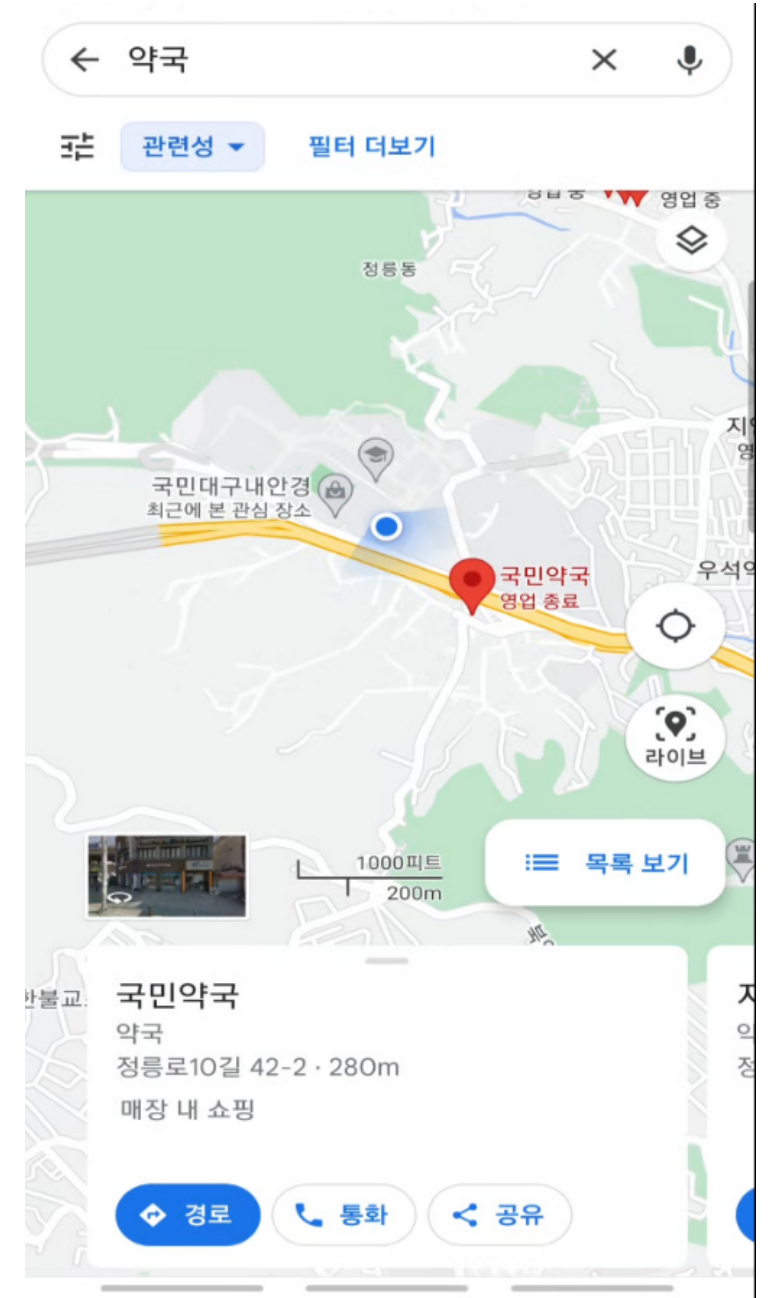
어플리케이션 내 지도를 제공하여 주변 약국을 검색 할 수 있습니다.



<약국 정보 선택>



<위치 정보 권한 설정>



<위치 기반 주변 약국 검색>

기대효과

01

약 복용 시간을 반드시 지켜야 하는 환자

02

한가지 약이 아닌 여러가지 약을 복용하는 환자



03

주기적으로 영양제를 섭취하는 소비자

04

평소 약물 관리에 어려움을 가지고 있는 환자