

TO DO APP

[HTTPS://GITHUB.COM/ORIGINALUNIT337/TO DO APP/
TREE/DEVELOP](https://github.com/OriginalUnit337/to-do-app/tree/develop)

DEVELOPED BY FESHCHANKA VITALI



АРХИТЕКТУРА

- ИСПОЛЬЗУЕТСЯ ЧИСТАЯ АРХИТЕКТУРА. ПРИЛОЖЕНИЕ РАЗДЕЛЕНО НА 3 СЛОЯ:
- PRESENTATION
- DOMAIN
- DATA

СЛОЙ DATA

- DATA СЛОЙ ОТВЕЧАЕТ ЗА РАБОТУ С БД. ПОЛУЧЕНИЕ И ОТПРАВКА ДАННЫХ В БД. СОСТОИТ ИЗ:
- МОДЕЛЕЙ (КОЛЛЕКЦИИ, ДОКУМЕНТА И ЗАМЕТКИ).
- СЕРВИСА, КОТОРЫЙ ДЕЛАЕТ ЗАПРОСЫ НА АПИ.
- РЕАЛИЗАЦИЯ РЕПОЗИТОРИЯ, КОТОРЫЙ ОБРАБАТЫВАЕТ РЕЗУЛЬТАТЫ ЗАПРОСОВ С СЕРВИСА.

СЛОЙ DOMAIN

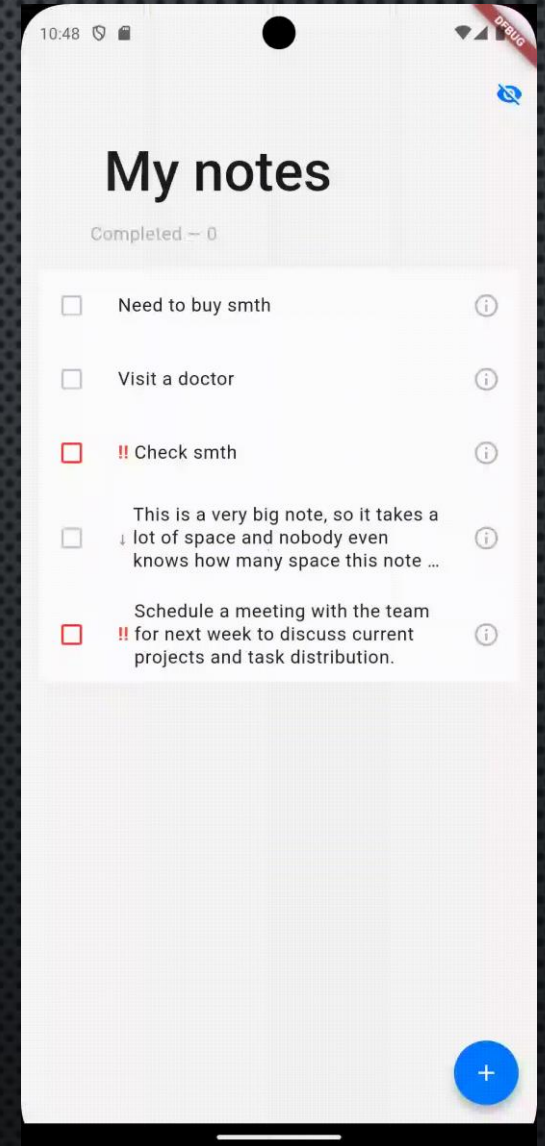
- СЛОЙ DOMAIN является связующим звеном между DATA и PRESENTATION. Состоит из:
- ENTITY
- АБСТРАКТНОГО РЕПОЗИТОРИЯ
- USECASES КОТОРЫЕ ВЫЗЫВАЮТ АБСТРАКТНЫЕ МЕТОДЫ РЕПОЗИТОРИЯ.

СЛОЙ UI

- СЛОЙ UI ОТВЕЧАЕТ ЗА ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС. НА ЭТОМ СЛОЕ РЕАЛИЗОВАНА ЛОКАЛИЗАЦИЯ И ТЕМА (СВЕТЛАЯ, ТЕМНАЯ). СОСТОИТ ИЗ:
- 2 ЭКРАНА, ГДЕ У КАЖДОГО СВОЙ BLOC
- ФАЙЛ ТЕМЫ
- ФАЙЛ ПАЛИТРЫ, ГДЕ ЗАДАНЫ ВСЕ НЕОБХОДИМЫЕ ЦВЕТА
- ФАЙЛ С НАСТРОЙКОЙ ШРИФТОВ.

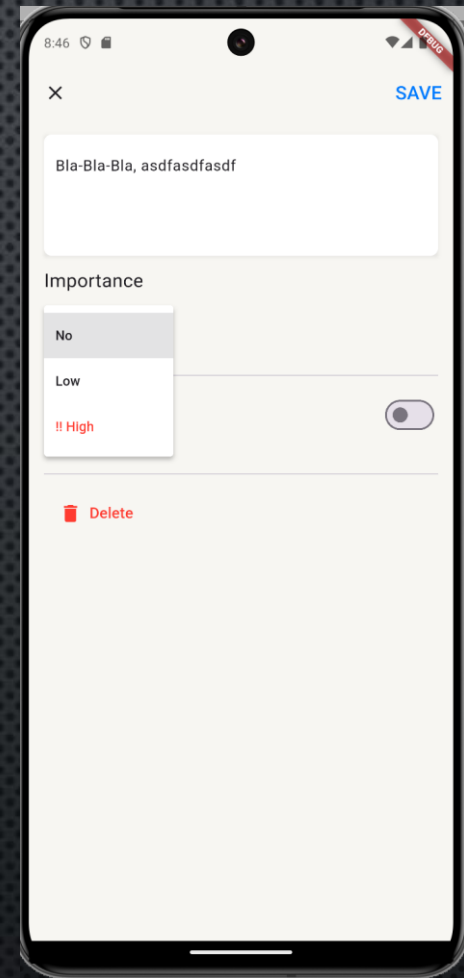
РАЗРАБОТКА UI

- РАЗРАБОТКА НАЧАЛАСЬ С UI СЛОЯ. БЫЛИ СОЗДАНЫ ФАЙЛ С ПАЛИТРОЙ ЦВЕТОВ КАК ПО ДИЗАЙНУ. ФАЙЛ С ШРИФТАМИ. 2 ЭКРАНА – ГЛАВНЫЙ ЭКРАН СО СПИСКОМ ВСЕХ ЗАМЕТОК, ГДЕ ВИДНА ЧАСТЬ ЗАМЕТКИ (ТЕКСТ ЗАМЕТКИ ОБРЕЗАЕТСЯ ЕСЛИ ОН ПРЕВЫШАЕТ 3 СТРОКИ ТЕКСТА), ВАЖНОСТЬ ЗАМЕТКИ ТАКЖЕ ВЫДЕЛЕНА СПЕЦИАЛЬНЫМИ ИКОНКАМИ И ЦВЕТОМ.
- КАЖДУЮ ЗАМЕТКУ МОЖНО СВАЙПАТЬ **ВПРАВО** ДЛЯ **ОТМЕТКИ**, ЧТО ЗА МЕТКА **ВЫПОЛНЕНА** (НО ЭТО МОЖНО ДЕЛАТЬ ПРИ УСЛОВИИ ЧТО «ГЛАЗИК» В ВЫКЛЮЧЕННОМ СОСТОЯНИИ, ТО ЕСТЬ СПРЯТАНЫ ЗАМЕТКИ, КОТОРЫЕ ВЫПОЛНЕННЫ), СВАЙП **ВЛЕВО** ДЛЯ УДАЛЕНИЯ ЗАМЕТКИ.
- ЧЕКБОКС ВЫПОЛНЯЕТ АНАЛОГИЧНУЮ ФУНКЦИЮ СВАЙПА **ВПРАВО**
- **«ГЛАЗИК»** : ИСПОЛЬЗУЕТСЯ ДЛЯ ТОГО, ЧТОБЫ СПРЯТАТЬ ВЫПОЛНЕННЫЕ ЗАМЕТКИ
- ПОД ЗАГОЛОВКОМ МОЖНО УВИДЕТЬ КОЛИЧЕСТВО ВЫПОЛНЕННЫХ ЗАМЕТОК



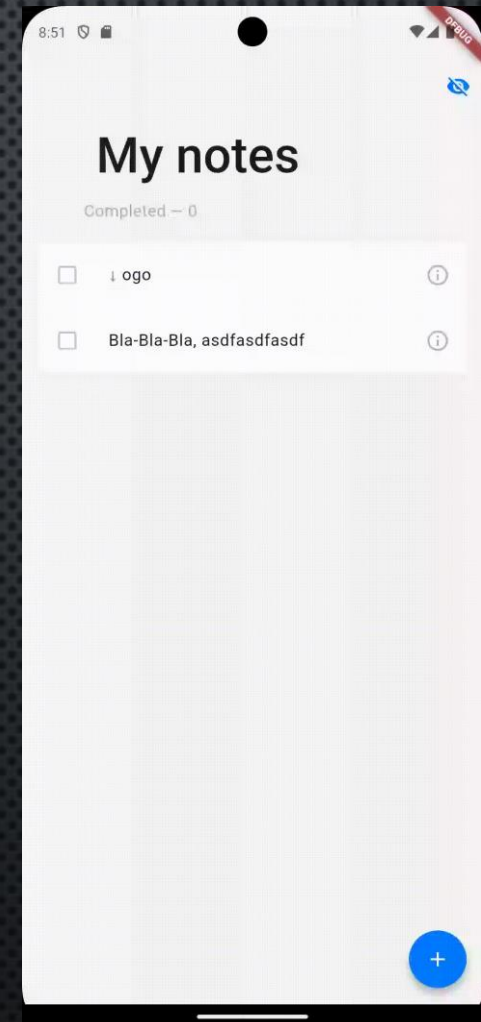
РАЗРАБОТКА UI

- ВТОРОЙ ЭКРАН С ИНФОРМАЦИЕЙ О ЗАМЕТКИ ИСПОЛЬЗУЕТСЯ КАК ДЛЯ СОЗДАНИЯ НОВОЙ ЗАМЕТКИ, ТАК И ДЛЯ РЕДАКТИРОВАНИЯ ТЕКУЩЕЙ.



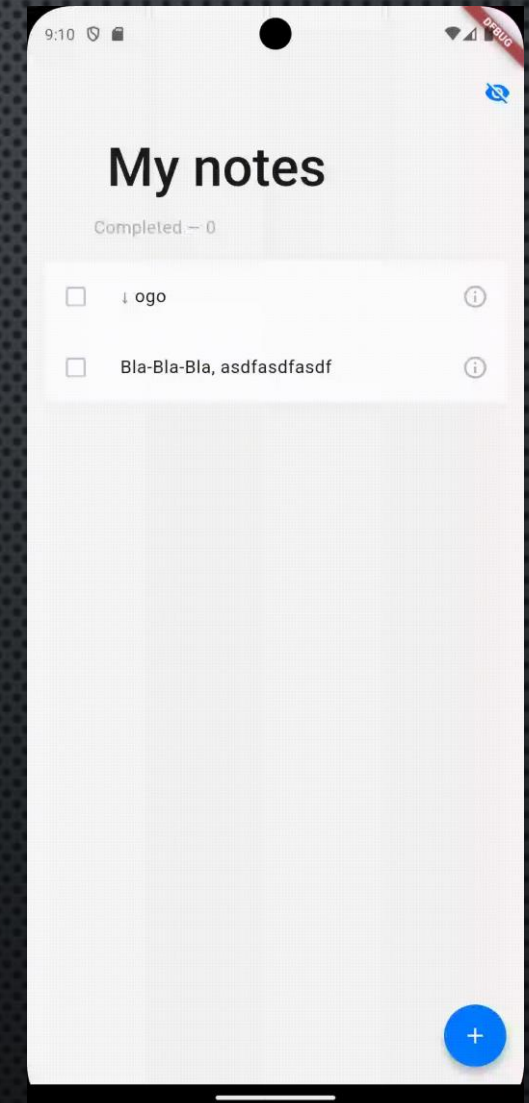
ЛОКАЛИЗАЦИЯ И ТЕМА

- ДАЛЕЕ БЫЛИ РЕАЛИЗОВАНЫ ЛОКАЛИЗАЦИЯ И СМЕНА ТЕМЫ



РАБОТА С API

- ДАЛЕЕ Я ПЕРЕШЕЛ К РАБОТЕ С API. ДЛЯ НАЧАЛА БЫЛИ СОЗДАНЫ МОДЕЛИ ДОКУМЕНТОВ, КОТОРЫЕ СООТВЕТСТВУЮТ МОДЕЛИ, КОТОРУЮ ВОЗВРАЩАЕТ API. МОДЕЛЬ ДОКУМЕНТА ИСПОЛЬЗУЕТ ДЖЕНЕРИК, ЧТО ПОЗВОЛЯЕТ ПОДСТАВЛЯТЬ ЛЮБОЙ НЕОБХОДИМЫЙ МОДЕЛЬ.
- БЫЛ РЕАЛИЗОВАН ПОЛНЫЙ ЦИКЛ ВЫЗОВА. БЛОК ВЫЗЫВАЕТ UseCase, UseCase вызывает метод репозитория, Репозитории вызывает сервис. В БЛОК ВОЗВРАЩАЕТСЯ DataState (либо Success, либо Failed) и ПРОИСХОДИТ СООТВЕТСТВУЮЩАЯ ОБРАБОТКА СОСТОЯНИЯ.



РЕАЛИЗОВАНО НА ДАННЫЙ МОМЕНТ

- НА ДАННЫЙ МОМЕНТ РЕАЛИЗОВАНЫ 2 МЕТОДА С API:
- ПОЛУЧЕНИЕ ВСЕХ ЗАМЕТОК ИЗ БД
- ОБНОВЛЕНИЕ ЗАМЕТКИ

МОЖНО ОТМЕЧАТЬ ВЫПОЛНЕНИЕ ЗАМЕТОК, А ТАКЖЕ ИХ РЕДАКТИРОВАТЬ.

TODO:

- СОЗДАНИЕ ЗАМЕТКИ АПИ
- УДАЛЕНИЕ ЗАМЕТКИ АПИ
- РЕАЛИЗОВАТЬ ТАКОЙ ЖЕ ФУНКЦИОНАЛ КАК В АПИ, ТОЛЬКО В **ЛОКАЛЬНОЙ БД**
- СИНХРОНИЗАЦИЯ МЕЖДУ РЕМОУТ И ЛОКАЛЬНЫМ.