

Касаемо интерфейса:

Jetpack Compose — это современный инструмент от Google для создания пользовательских интерфейсов (UI) на платформе Android. Это декларативный фреймворк, который позволяет разработчикам описывать интерфейсы с помощью кода, а не с использованием XML-разметки, как это было в традиционном подходе.

У нас в приложении:

Все экраны созданы с использованием этого инструмента. Всего три экрана:

1. Главный экран со всеми заметками. Есть возможность удалить заметку или изменить при нажатии на неё. Также можно добавить новую заметку при нажатии на кнопку справа внизу.

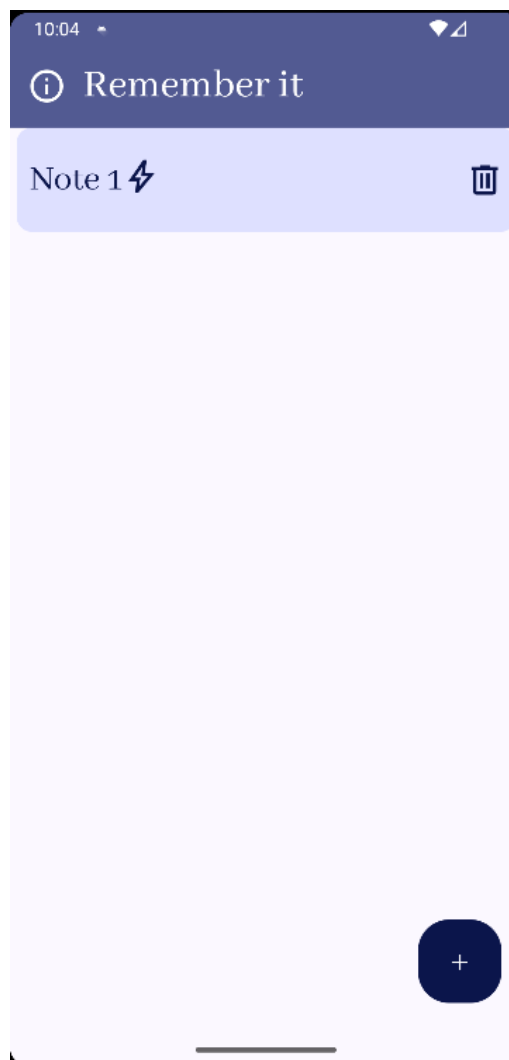


Рисунок 1 - главный экран со всеми заметками

2. Экран с информацией об алгоритме работы приложения.

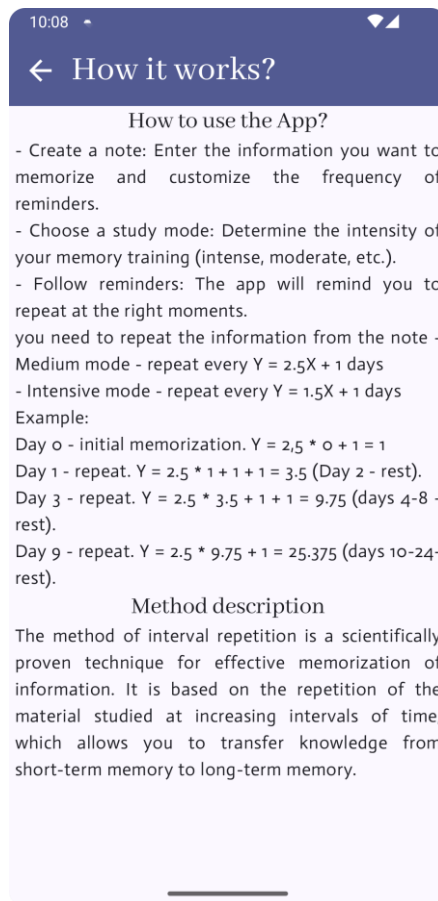


Рисунок 2 - экран с информацией

3. Экран создания/редактирования заметки. Есть возможность задать заголовок и ввести текст заметки. Также полностью очистить поле ввода и сохранить заметку, выбрав алгоритм напоминания.

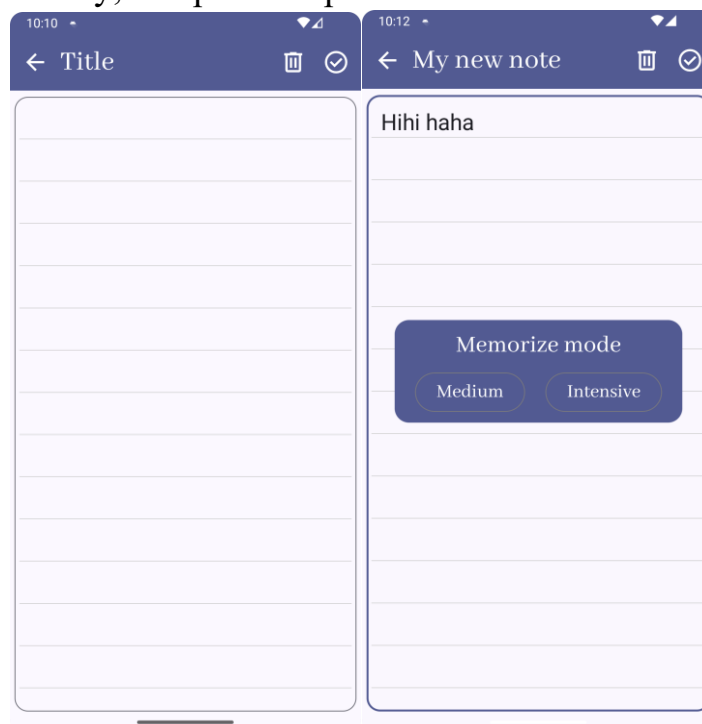


Рисунок 3 - экран создания/редактирования заметки

Material Theme — это система дизайна, разработанная Google, которая предоставляет стандарты для создания привлекательного, доступного и интуитивно понятного пользовательского интерфейса. Она базируется на принципах **Material Design**, а также предлагает разработчикам инструменты для настройки внешнего вида и взаимодействий их приложений.

У нас в приложении:

Приложение использует Material Theme и все Composable функции используют настройки, заданные в ней. Также есть поддержка тёмной темы.

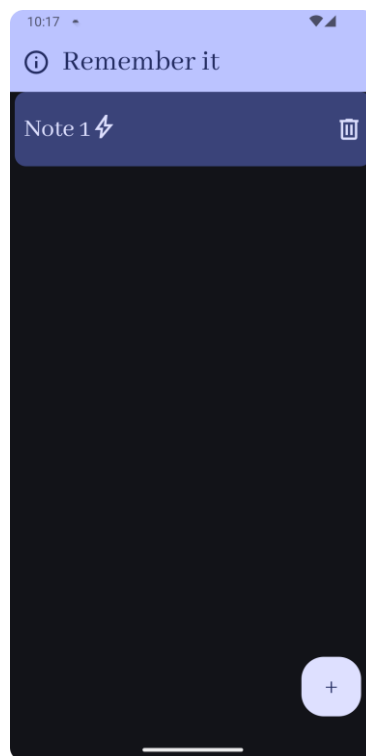


Рисунок 4 - поддержка тёмной темы приложением

NavController — это компонент библиотеки **Jetpack Navigation**, который управляет навигацией в приложении Android. Он отвечает за перемещение между экранами (фрагментами, активностями) и обработку стека навигации, что упрощает реализацию сложных навигационных сценариев.

У нас в приложении:

В приложении есть специальная функция реализующая NavHost и перемещение по экранам приложения.

Локализация — приложение поддерживает русский язык. Он будет использоваться, в случае если системный язык устройства - русский.

Касается логики:

ViewModel — это компонент архитектуры Android. Основная цель ViewModel — обеспечить сохранение данных и состояния UI при изменении конфигурации, таких как поворот экрана, а также разделение логики приложения и его пользовательского интерфейса.

У нас в приложении:

Реализована одна ViewModel для обработки хранения состояния заметок и для взаимодействия с локальной базой данных, хранящей эти заметки. Она принимает в себя интерфейс dao, служащий для выполнения операций с базой данных. Эта ViewModel обрабатывает NoteState, дата класс, используемый для хранения текущего состояния приложения.

Room — это библиотека Jetpack, которая предоставляет абстракцию над SQLite и упрощает работу с базами данных в Android. Она помогает разработчикам сохранять, изменять и запрашивать данные, обеспечивая при этом удобный API и соблюдение принципов архитектуры.

У нас используется всего одна таблица для хранения информации о заметке, а именно: заголовок, текст, режим напоминания и id.

Для взаимодействия с ней реализован интерфейс dao, в котором реализовано 3 запроса:

1. Вставить (или заменить заметку с таким же id)
2. Удалить заметку
3. Получить все заметки из базы данных

```
1 package com.example.rememberit.model
2
3 import androidx.room.Dao
4 import androidx.room.Delete
5 import androidx.room.Query
6 import androidx.room.Upsert
7
8 @Dao
9 interface NoteDao {
10
11     @Upsert
12     suspend fun upsertNote(note: Note)
13
14     @Delete
15     suspend fun deleteNote(note: Note)
16
17     @Query("SELECT * FROM note")
18     suspend fun getNotes(): List<Note>
19 }
```

Рисунок 5 - dao интерфейс

title	text	mode	id
Note 1	Hihih	Intensive	18

Рисунок 6 - пример записи в БД

WorkManager — это библиотека Jetpack для упрощения выполнения фоновых задач, которые должны быть надёжно выполнены даже при изменении конфигурации устройства (например, перезапуске устройства).

У нас в приложении `workManager` служит для отправки уведомлений с напоминанием текст заметки. Создаётся `NotificationChannel` в который и отправляется уведомление. Частота отправления заметки зависит от выбранного при её сохранении режима.