**ВВЕДЕНИЕ**

**ИнфоТел** — многофункциональное Flutter-приложение для диагностики Android-устройств. Оно предоставляет подробную информацию о состоянии системы, отслеживает работу ключевых компонентов и сохраняет результаты тестирования.

Приложение отличается интуитивно понятным интерфейсом в тёмной цветовой гамме с акцентами на синий цвет, поддержкой русского и английского языков, а также надёжной системой авторизации с продуманной обработкой ошибок.

**Главные особенности:**

* *Глубокая диагностика устройства* — анализ системных параметров, памяти и критически важных компонентов смартфона.
* *Удобная навигация –* продуманная и логичная структура разделов для быстрого доступа к данным.
* *Интерактивные уведомления* — всплывающие сообщения (SnackBar), которые информируют о текущем состоянии устройства и результатах операций.
* *Система авторизации* - безопасная форма регистрации с локальной проверкой данных и обработкой ошибок.

ИнфоТел подойдёт как для опытных пользователей, так и для новичков, желающих быстро и удобно оценить работоспособность своего устройства.

**ОСНОВНЫЕ ФУНКЦИИ ПРИЛОЖЕНИЯ**

1. **Файл main.dart**

В нем импортируются нужные компоненты. Функция main() запускает приложение. В main() вызывается runApp, чтобы запустить приложение, и в начале оно показывает экран HomePage.

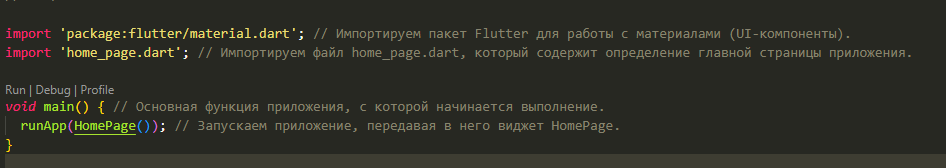
****

Рисунок 1. Main.dart

1. **Файл home\_page.dart**

Это главный экран Flutter-приложения. который:

* Устанавливает локализацию по умолчанию (русский). За это отвечает Locale \_locale = Locale('ru', 'RU');
* Поддерживает смену языка. Метод \_changeLanguage()
* Оформляет всё приложение в тёмном стиле с элементами синего. Через ThemeData.
* Загружает в качестве главного экрана страницу входа (LoginPage), передавая ей возможность смены языка. За это отвечает home: LoginPage(onLocaleChange: \_changeLanguage)
* Отображает список установленных приложений, основные сведения об устройстве, быстрый доступ к разделам с помощью кнопок

**Функции:**

* \_changeLanguage(Locale locale) — изменяет язык приложения и обновляет интерфейс.
* build(BuildContext context) — строит интерфейс:
  + Настраивает локализацию (русский и английский).
  + Устанавливает тему: тёмный фон, фирменные цвета, стили текста и кнопок.
  + Определяет внешний вид полей ввода (скругления, цвета и стили).
  + Загружает экран LoginPage как главный.

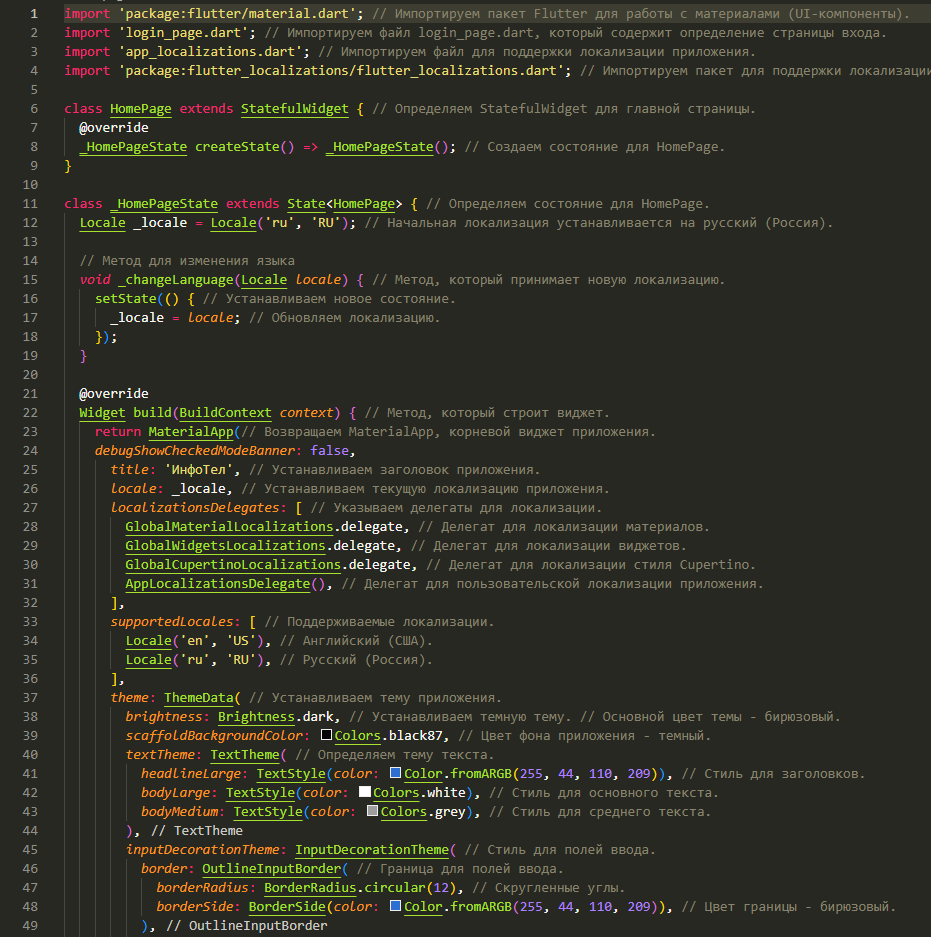
****

Рисунок 2. home\_page.dart

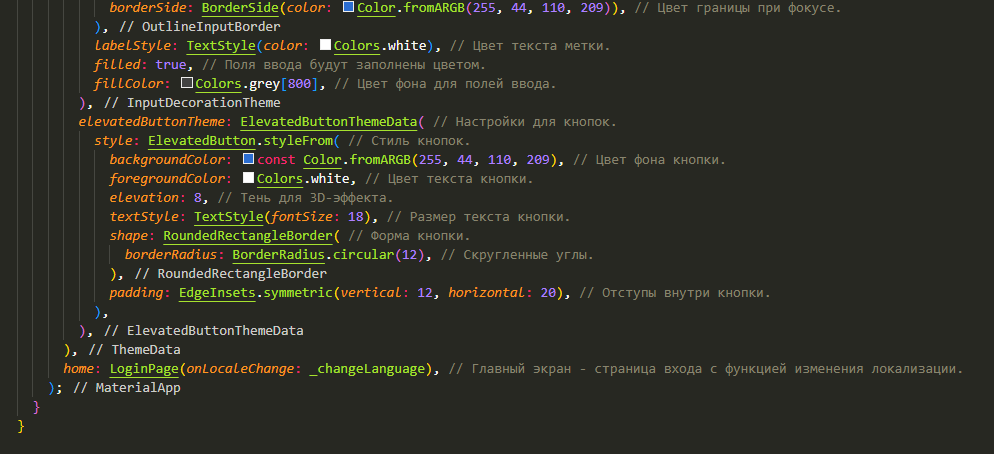


Рисунок 3. home\_page.dart

1. **Файл login\_page.dart**

Это экран входа и регистрации, который:

* Позволяет пользователю ввести имя пользователя и пароль.
* Отображает текущее время. Метод \_startClock().
* Позволяет переключать язык. onLocaleChange – функция для переключения языка (ru/en)
* Проверяет введённые данные через DatabaseHelper и переходит на главный экран (HomeScreen) при успешном входе.
* Позволяет перейти на экран регистрации (RegisterPage).

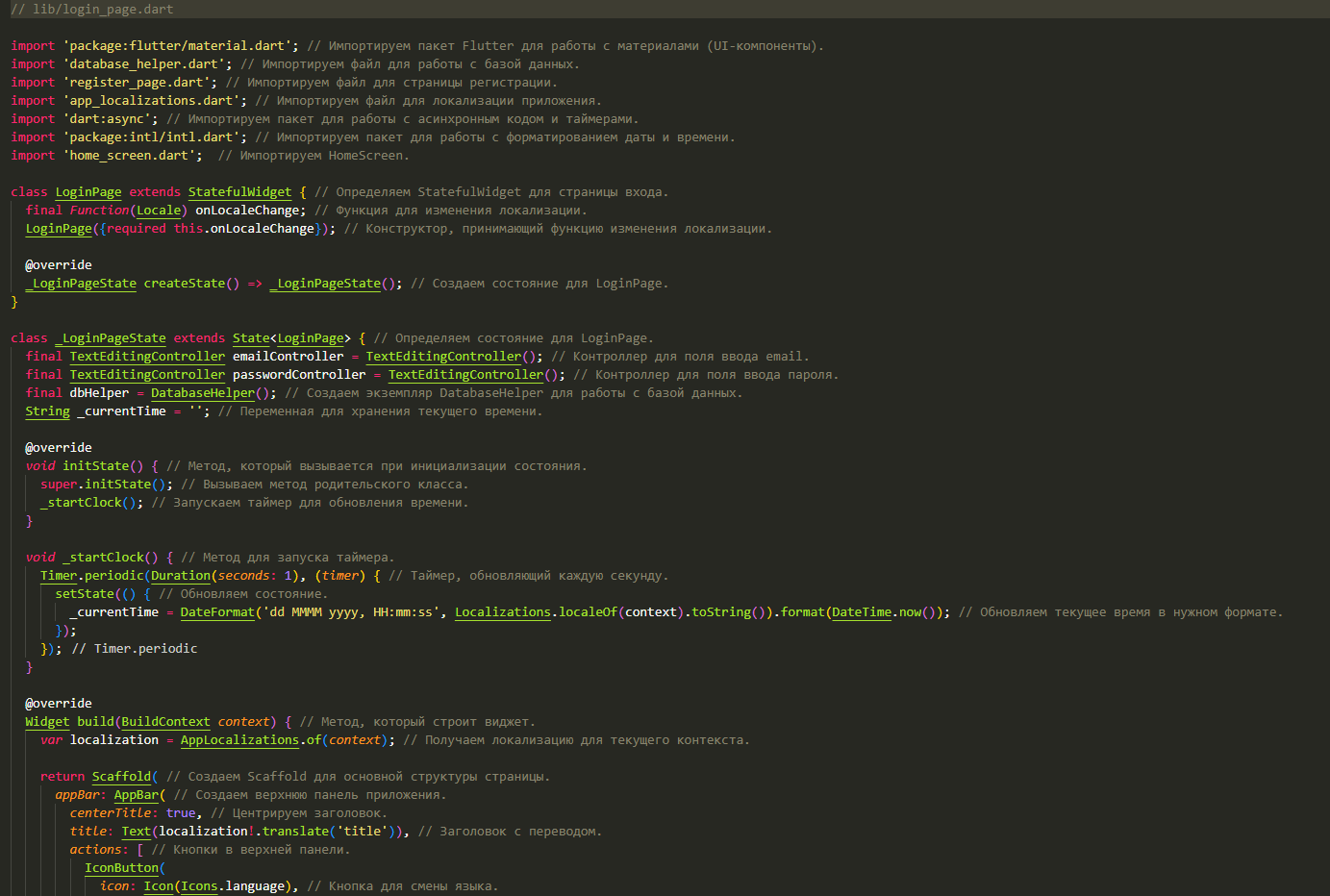
****

Рисунок 4. login\_page.dart

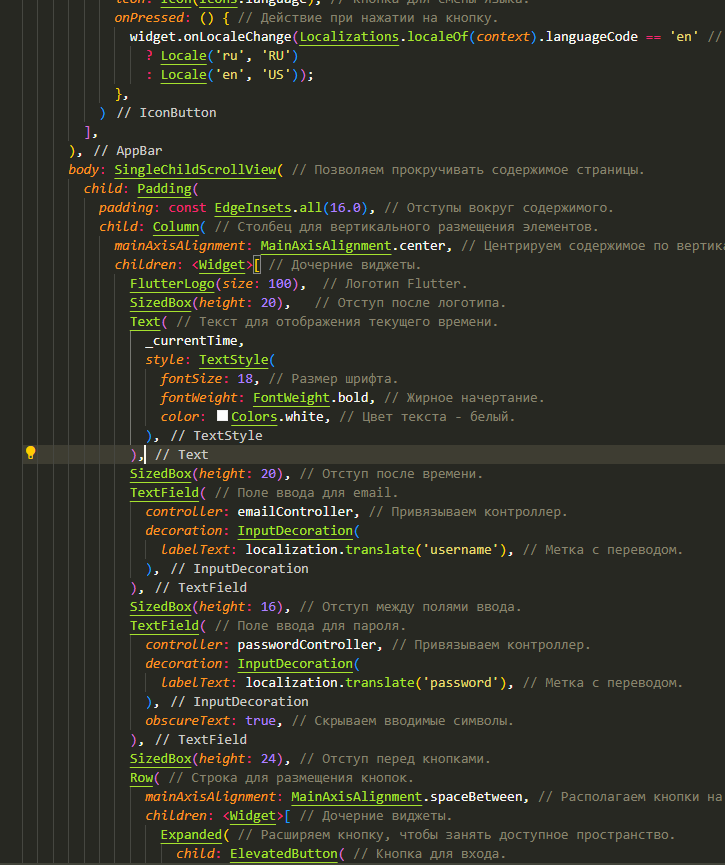


Рисунок 5. lodin\_page.dart

Body (Основное содержимое): Поля ввода – email и password, Кнопка "Войти" – проверяет аутентификацию через dbHelper, Кнопка "Регистрация" – переход на RegisterPage

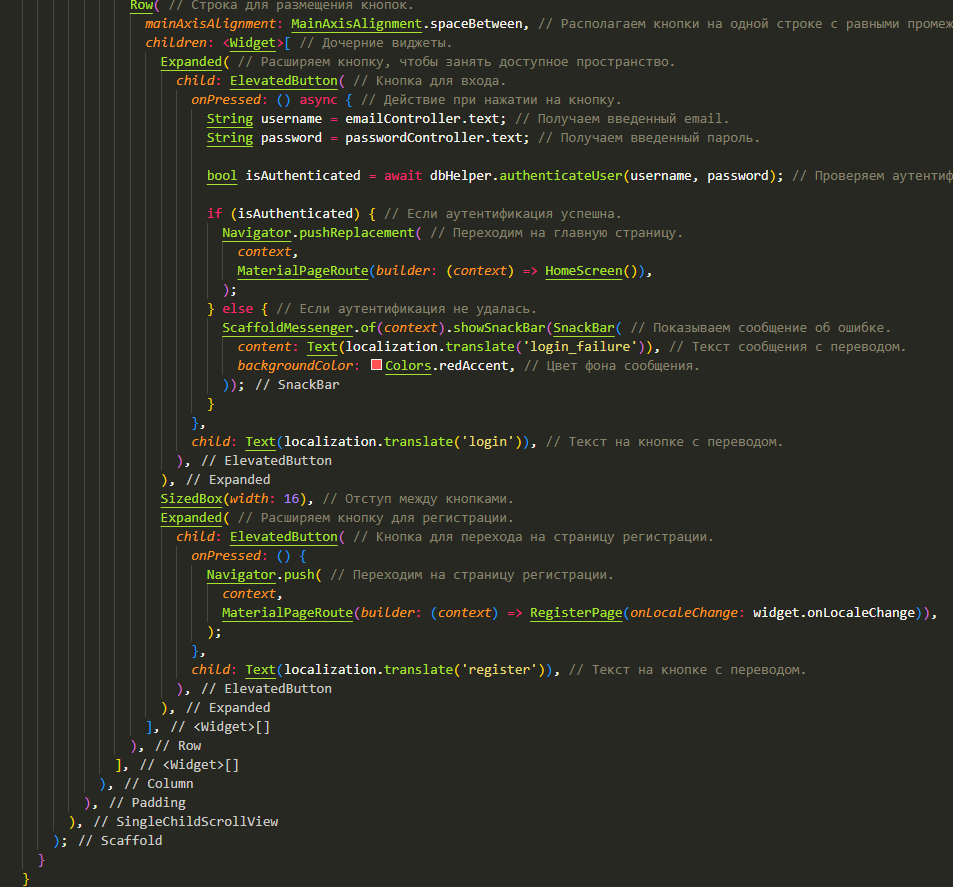


Рисунок 6. login\_page.dart

1. **Файл register\_page.dart.**

Этот файл определяет экран регистрации пользователя (RegisterPage), который отображает:

* Поле ввода email (имя пользователя).
* Поле ввода пароля.
* Кнопку регистрации.
* Сообщения об ошибке или успехе регистрации.

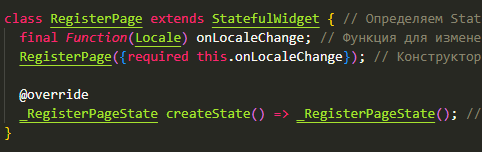
****

Рисунок 7. register\_page.dart

Этот код отвечает за определение страницы.

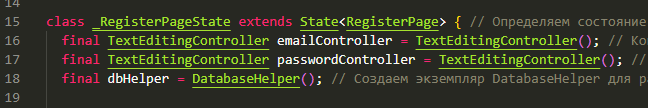


Рисунок 8. register\_page.dart

Состояние страницы. emailController — контролирует ввод в поле email, passwordController — контролирует ввод пароля, dbHelper — доступ к базе данных.

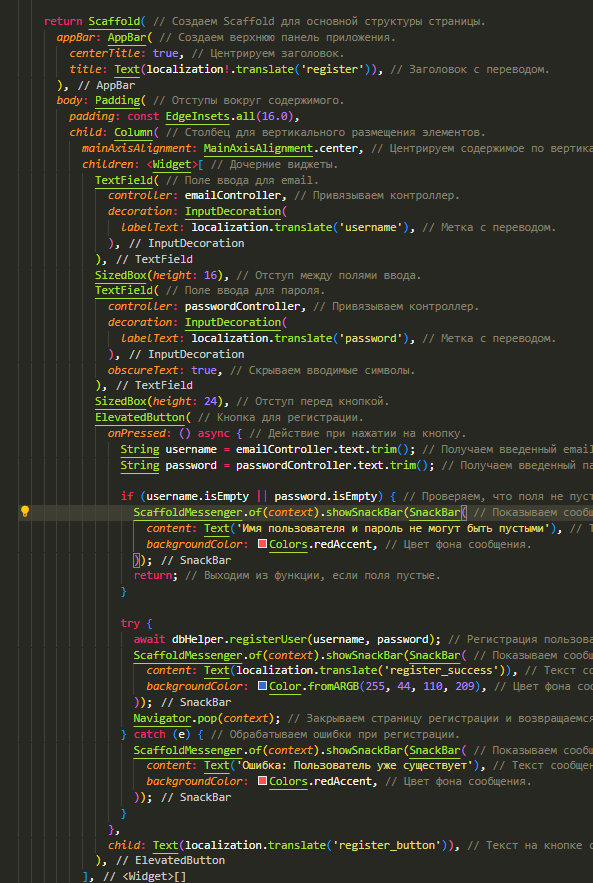


Рисунок 9. register\_page.dart

Содержимое страницы.

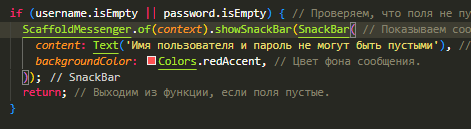


Рисунок 10. register\_page.dart

Проверка полей.

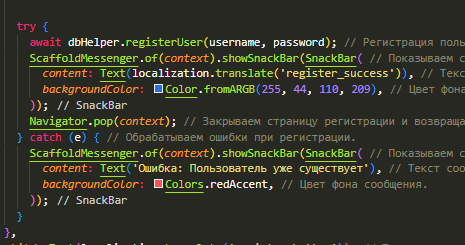
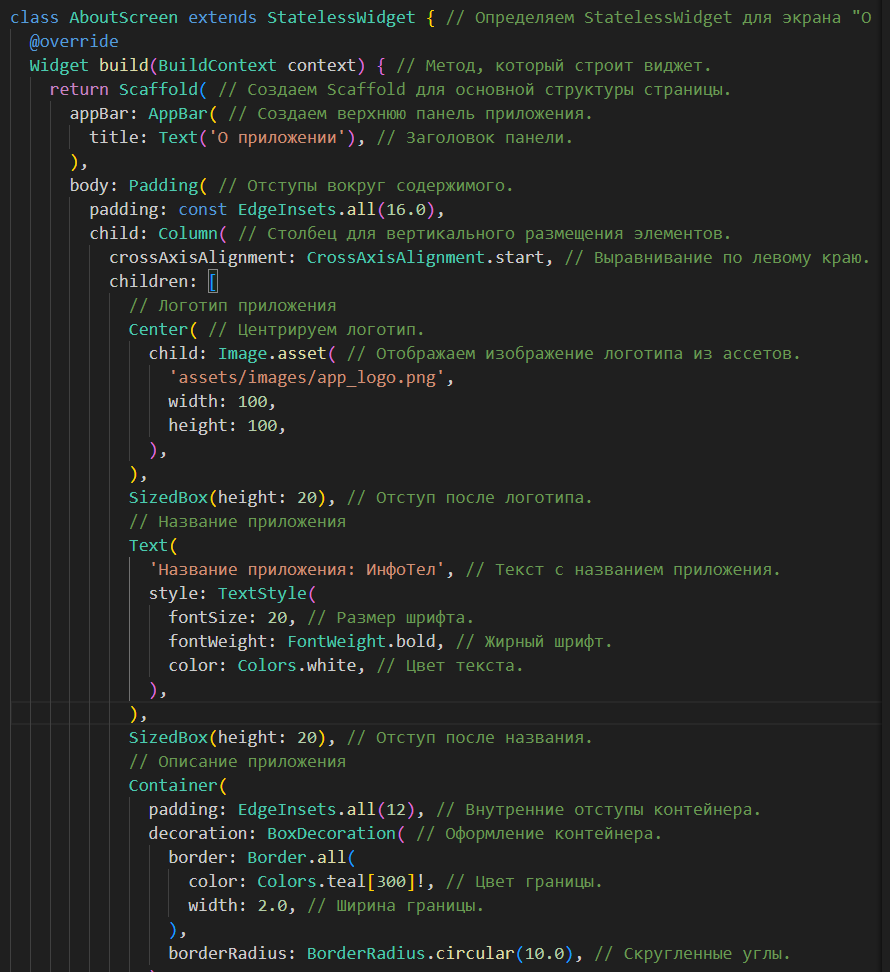


Рисунок 11. register\_page.dart

Регистрация в БД

1. Файл

****

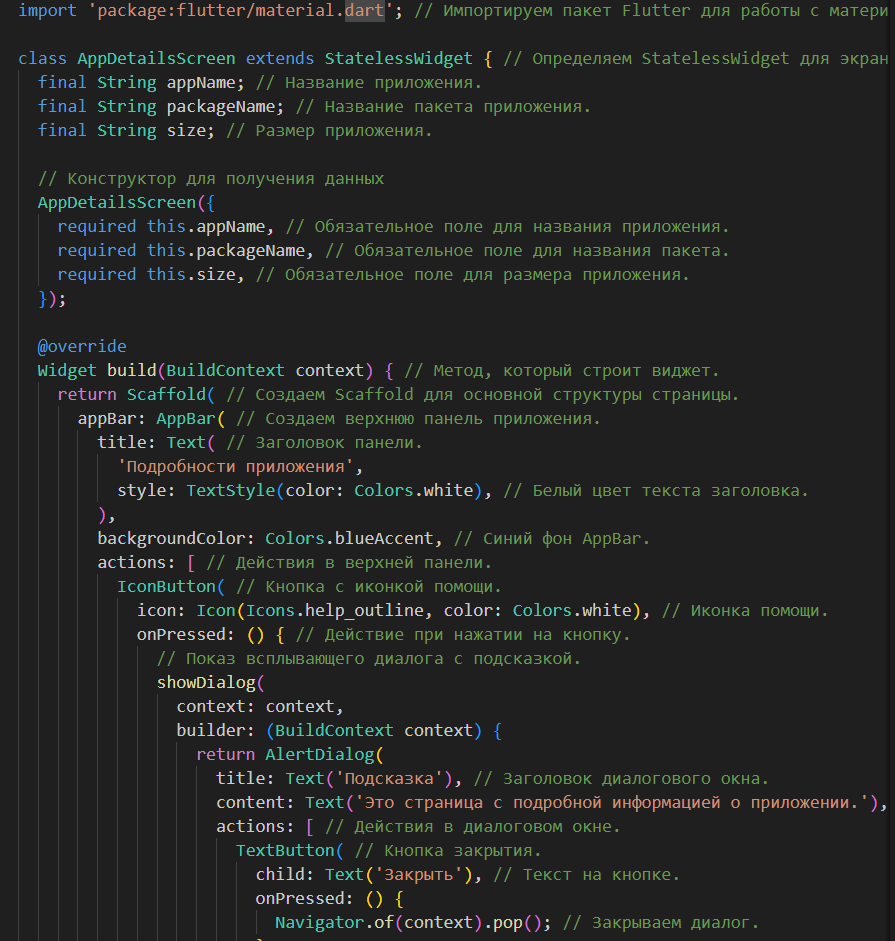
**Рис-7. About\_screen.dart**

**Информационный экран с полным набором контактных данных и возможностью быстрой связи, основные элементы:**

1. **Информационный блок:**
2. **Интерактивные элементы:**
3. **WebView экран:**

**Функции:**

1. **Инициирование телефонного звонка (\_makePhoneCall())**
2. **Открытие почтового клиента (\_sendEmail())**

****

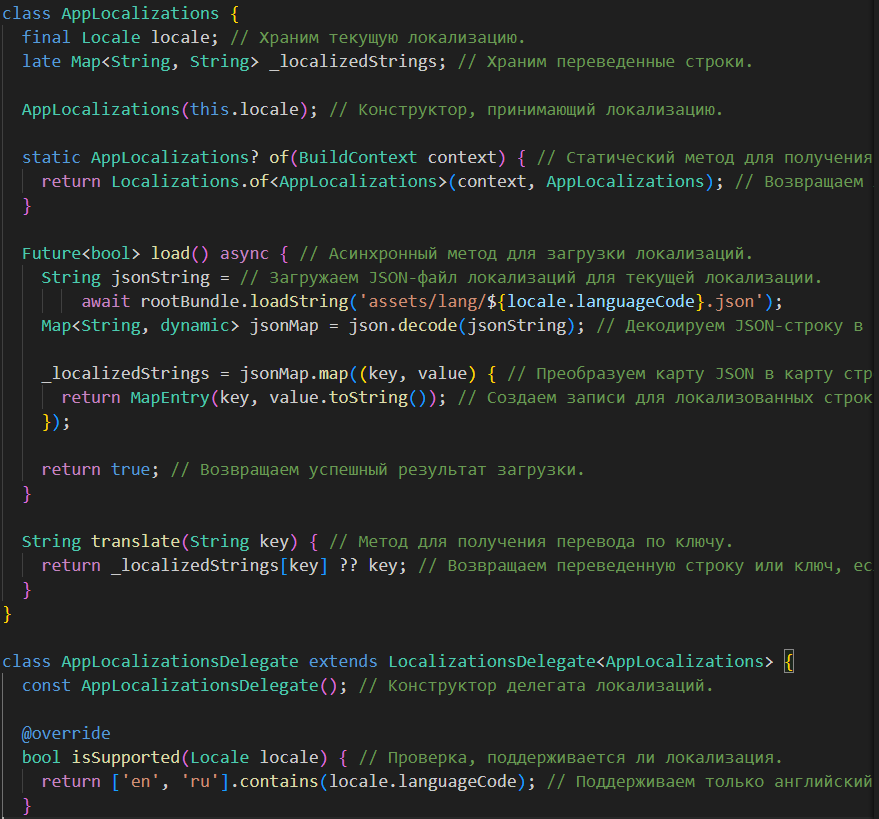
**Рис-8. App\_detail\_screen.dart**

**Детализированный экран отображения информации о приложении с четкой структурой и удобной навигацией, основные элементы:**

1. **Заголовок экрана с кнопкой помощи**
2. **Информационные блоки: название приложения, имя пакета, размер приложения**
3. **Иконки для каждого параметра**

**Функции:**

1. **Название приложения (appName)**
2. **Идентификатор пакета (packageName)**
3. **размер приложения (size)**

****

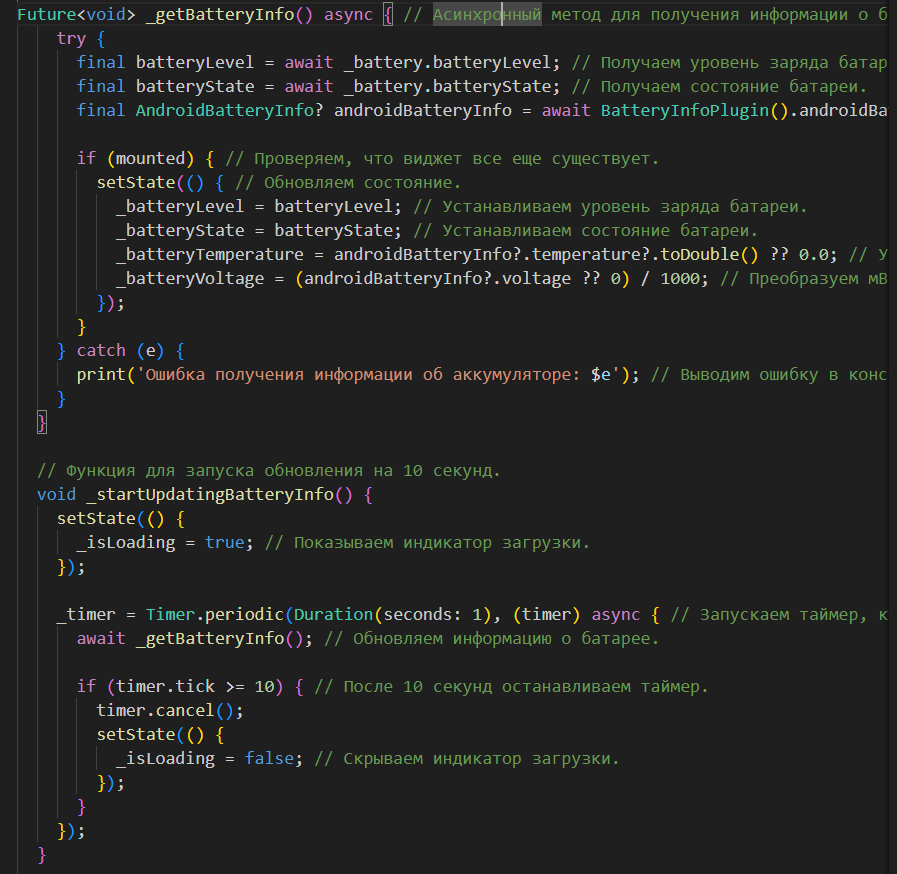
**Рис-9. App\_localizations.dart**

**Полнофункциональная система локализации с поддержкой JSON-файлов, основные элементы:**

1. **Хранение локализованных строк, загрузка переводов из JSON-файлов (AppLocalizations)**
2. **Проверка поддерживаемых языков,** **Загрузка данных локализации (AppLocalizationsDelegate)**

**Функции:**

1. **Получение экземпляра локализаций из контекста (AppLocalizations.of())\**
2. **Загрузка переводов из JSON (load())**
3. **Получение перевода по ключу (translate())**

****

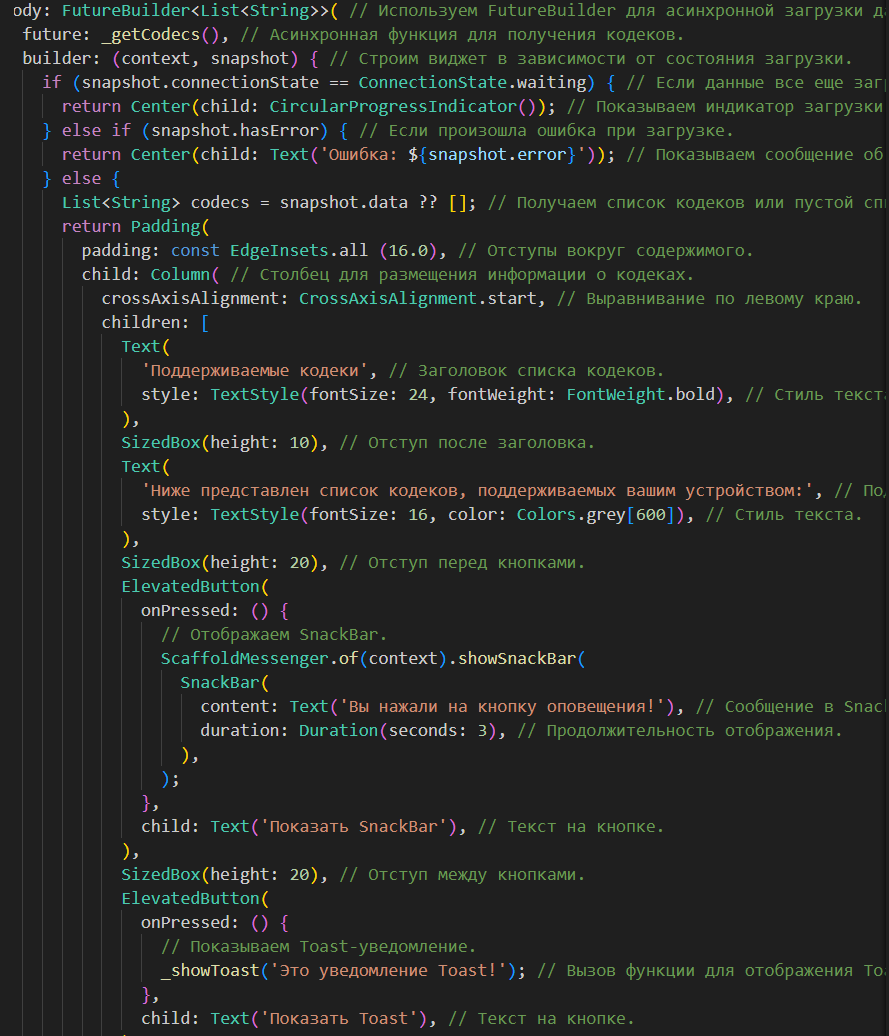
**Рис-10 Battery\_info\_screen.dart**

**Комплексный экран мониторинга состояния батареи с автоматическим обновлением данных, основные компоненты:**

1. **Визуальные индикаторы**
2. **Данные о батарее (состояние, температуру, уровень заряда и тд.)**
3. **Функциональные элементы**

**Функции:**

1. **Получение текущих данных о батарее (\_getBatteryInfo())**
2. **10-секундное обновление состояния (\_startUpdatingBatteryInfo())**
3. **Универсальный компонент для отображения (температуры , напряжения, состояния (\_buildInfoCard())**

****

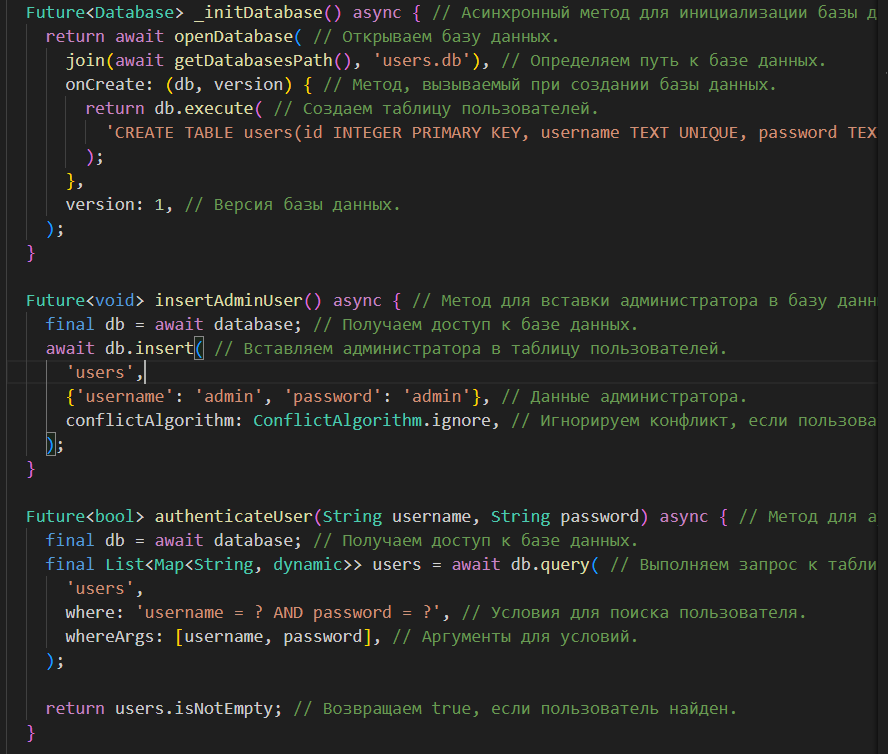
**Рис-11 Codecs\_screen.dart**

**Информативный экран для отображения поддерживаемых медиа-кодеков устройства с дополнительными возможностями демонстрации различных типов уведомлений, основные элементы:**

1. **Toast-уведомления**
2. **SnackBar сообщения**
3. **FutureBuilder для асинхронной загрузки**
4. **Проверка системных файлов media\_codecs.xml**
5. **Парсинг XML содержимого**

**Функции:**

1. **Получение списка кодеков устройства (\_getCodecs())**
2. **AlertDialog с заголовком и текстом (\_showDialog())**
3. **Использует RegExp для поиска тегов MediaCodec ( \_parseCodecs())**

****

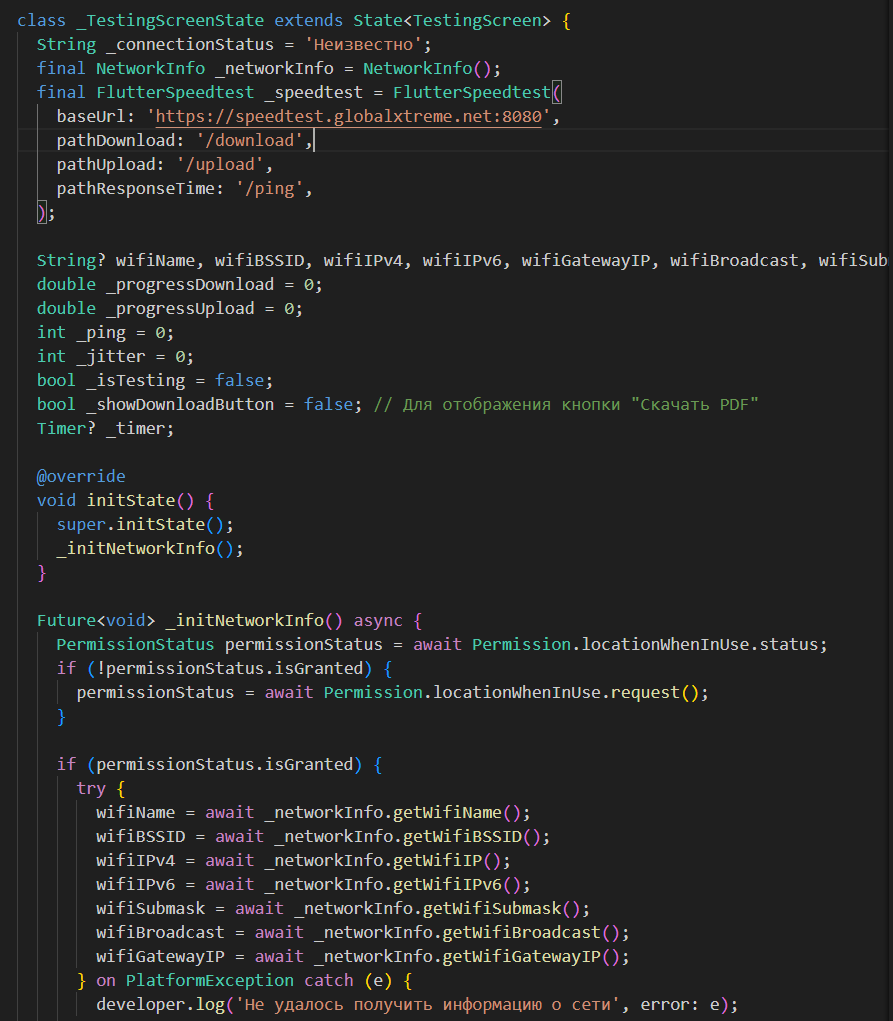
**Рис-12 Database\_helper.dart**

**Легковесный SQLite-менеджер для аутентификации пользователей с базовыми функциями регистрации и проверки учетных данных, основные элементы:**

1. **Приватный конструктор \_internal()**
2. **Статический экземпляр \_instance**
3. **Фабричный метод DatabaseHelper()**

**Функция:**

1. **Инициализация БД при первом обращении (\_initDatabase())**
2. **Отказ при существующем пользователе, генерация исключения при ошибке (registerUser())**
3. **Игнорирование при существующей записи (insertAdminUser())**
4. **Возвращает: true при успешной аутентификаци (authenticateUser())**

****

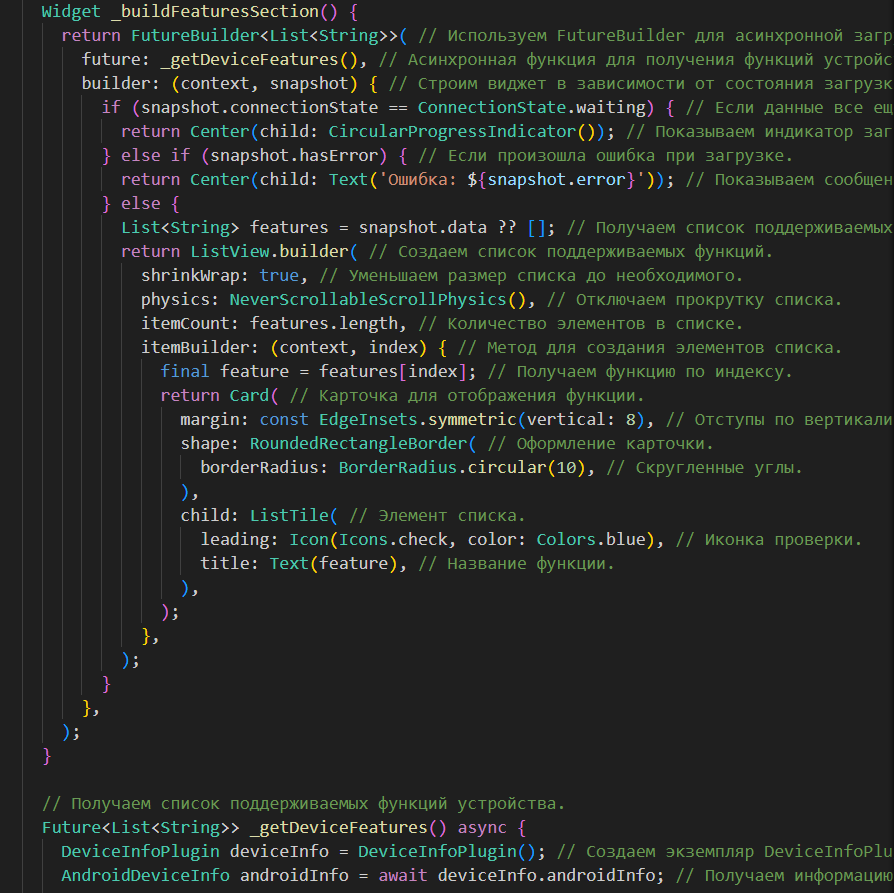
**Рис-13 Testing\_screen.dart**

**Комплексный инструмент для диагностики сети с возможностью сохранения результатов, основные элементы:**

1. **Информационная панель сети (название сети, BSSID, IPv4/IPv6 адреса, шлюз и маску подсети)**
2. **Результаты теста скорости (скорость загрузки, скорость отдачи, пинг и джиттер)**

**Функции:**

1. **Получает данные о Wi-Fi подключении (\_initNetworkInfo())**
2. **Формирует PDF с помощью pdf пакета, включает все сетевые параметры (\_generateAndSavePDF())**
3. **40-секундное ограничение (проверка mounted перед setState**

****

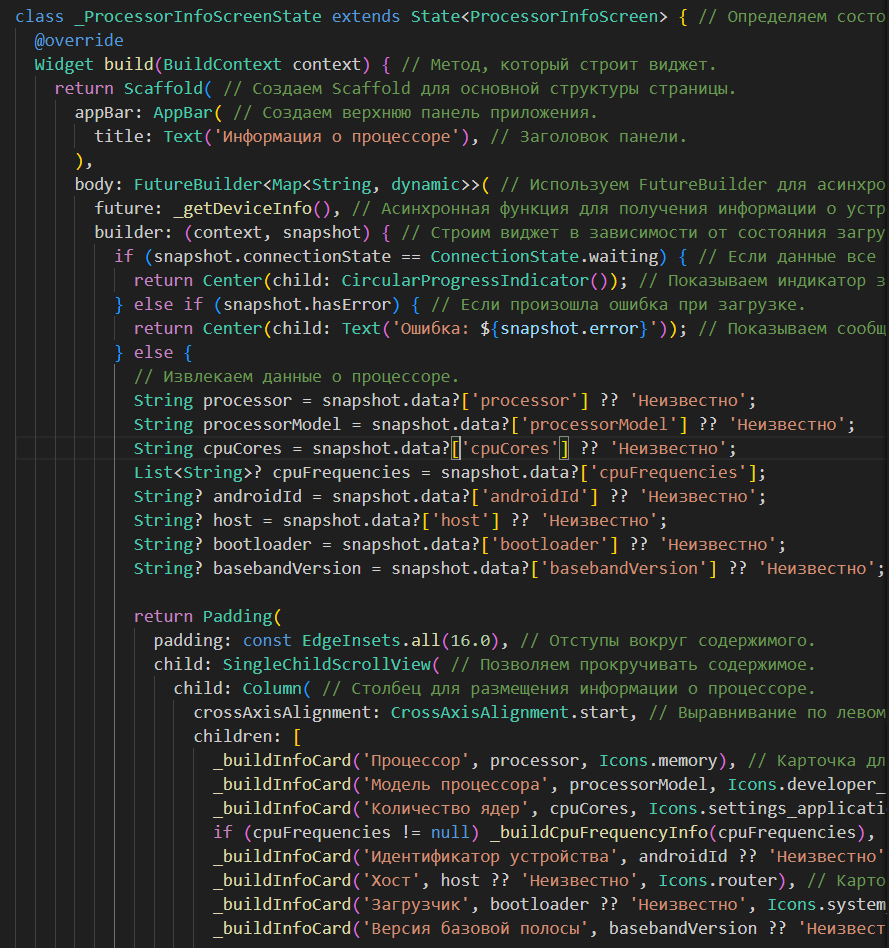
**Рис-14 Device\_features\_screen.dart**

**Этот экран отображает список поддерживаемых функций устройства, основные элементы:**

1. **Базовая структура экрана с AppBar и телом (body).**
2. **Асинхронно загружает данные о функциях устройства.**

**Функции:**

1. **Создает заголовок секции с иконкой (\_buildSectionTitle(String title, IconData icon))**
2. **Отображает список функций устройства в виде карточек (\_buildFeaturesSection())**
3. **Запрашивает системные функции устройства через DeviceInfoPlugin.**

****

**Рис-15 Processor\_info\_screen.dart**

**Экран отображает подробную информацию о процессоре и системе устройства, основные элементы:**

1. **Создает унифицированную карточку для отображения информации.**
2. **Переход на экраны SystemFoldersScreen и CodecsScreen.**
3. **Асинхронно загружает данные о процессоре и системе.**

**Функции:**

1. **Собирает комплексные данные о процессоре и системе (\_getDeviceInfo())**
2. **Читает файлы ядер процессора (\_getCpuFrequencies())**
3. **Запрашивает разрешение manageExternalStorage для доступа к системным файлам (\_requestPermissions())**
4. **Форматирует список частот процессора в виджет Card (\_buildCpuFrequencyInfo())**

****

**Рис-16 Reports\_screen.dart**

**Экран предназначен для генерации PDF-отчетов с технической информацией о процессоре и системе устройства, основные элементы:**

1. **Управление разрешениями на доступ к хранилищу**
2. **Кнопки генерации и скачивания отчета**
3. **Ядро для создания PDF-документов**

**Функции:**

1. **Сбор системной информации (\_getProcessorInfo())**
2. **Читает максимальные частоты каждого ядра (\_getCpuFrequencies())**
3. **Загружает кастомный шрифт (Roboto) из assets для использования в PDF. (\_loadFont())**

****

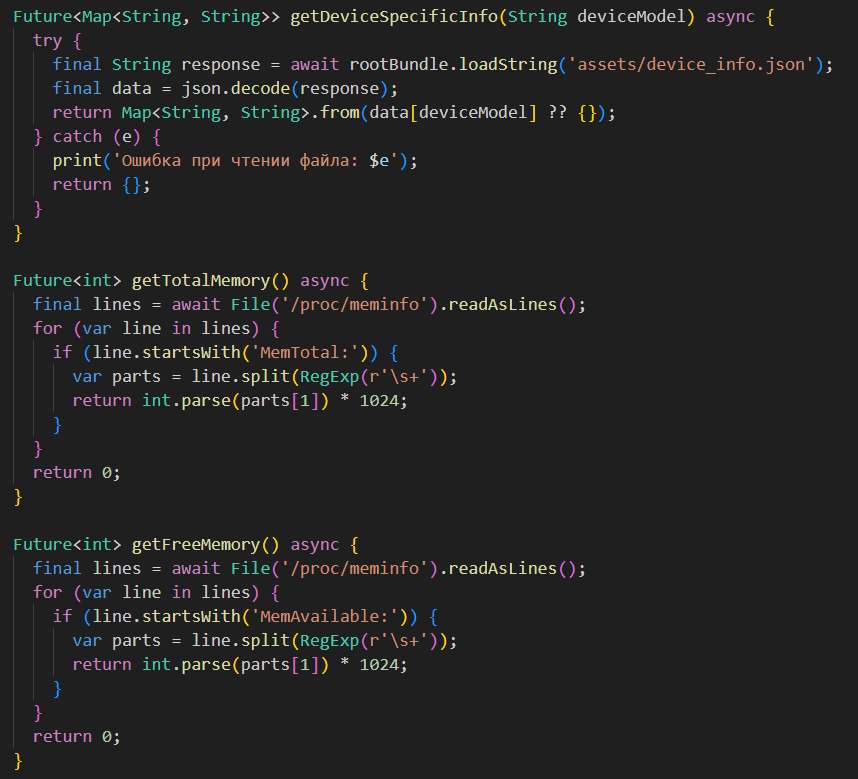
**Рис-17 System\_folders\_screen.dart**

**Экран предоставляет полную информацию о файловой системе Android, о сновные элементы:**

1. **Асинхронная загрузка данных о папках**
2. **Унифицированный виджет для отображения информации**

**Функции:**

1. **Проверяет существование стандартных системных папок (\_getSystemFolders())**
2. **Обнаруживает все доступные хранилища (\_getExternalStorage())**
3. **Парсит /proc/mounts для получения точек монтирования ( \_getMountPoints())**

****

**Рис-17 System\_info\_helper.dart**

**Скрипт предназначен для сбора детальной информации об Android-устройстве, основные элементы:**

1. **Основной инструмент получения системной информации**
2. **Чтение /proc/meminfo и вывод команды df**

**Функции:**

1. **Рассчитывает общий и свободный объем памяти устройства (getMemoryInfo()).**
2. **Собирает базовые данные об устройстве (getSystemInfo())**
3. **Выполняет команду df для получения данных о хранилище (getStorageInfo()**

****

**Рис-18 System\_info\_screen.dart**

Экран отображает **полную техническую информацию об устройстве** и предоставляет инструменты для работы с данными, основные элементы:

1. Сворачиваемые блоки для памяти и системной информации
2. **Кнопки действий (к**опирование данных, генерация PDF-отчета)

Функции:

1. Копирует выбранные данные в буфер обмена с уведомлением пользователя (\_copyToClipboard())
2. Запрашивает разрешение на запись PDF-документ (\_saveAsPdf())
3. Объединяет данные от getMemoryInfo() и getSystemInfo()

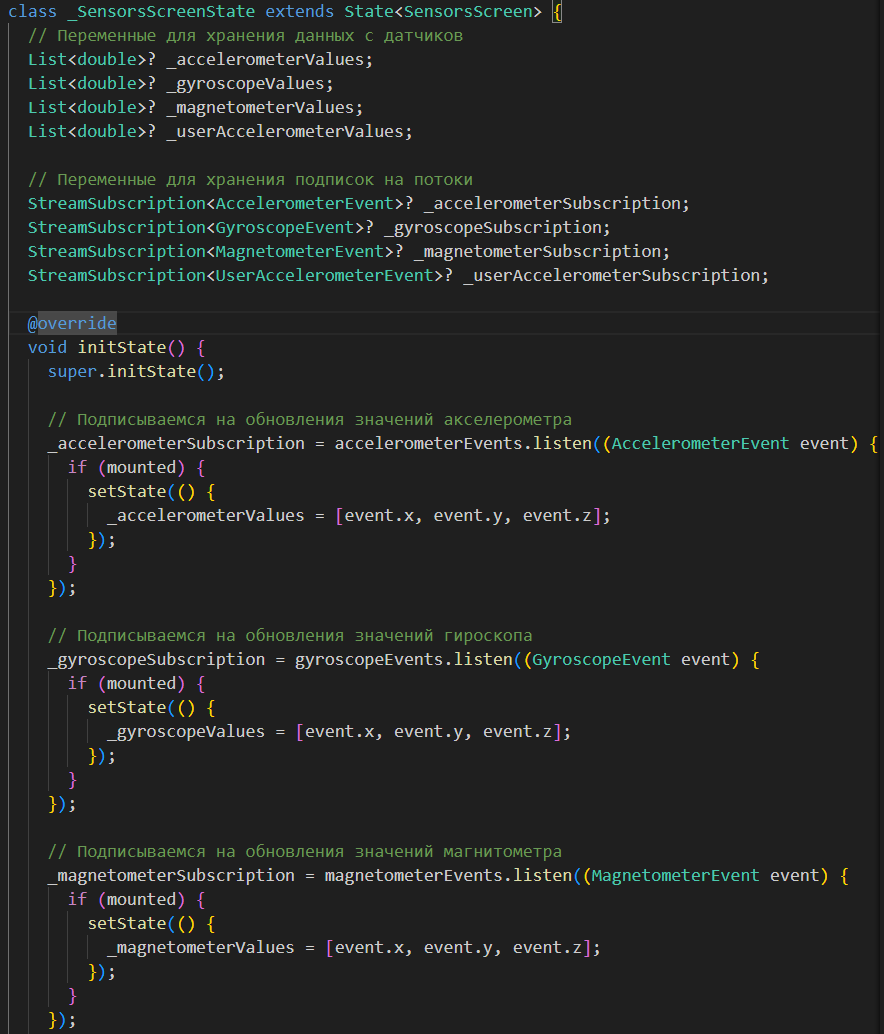


Рис-19 Sensors\_screen.dart

Экран отображает **реальные показания датчиков устройства** в реальном времени, основные элементы:

1. Подписки на потоки данных датчиков (акселерометр, гироскоп и др.).
2. Унифицированные виджеты для отображения данных.

Функции:

1. Создает заголовок секции с иконкой (\_buildSectionTitle)
2. Отменяет подписки при закрытии экрана для утечки памяти (**Очистка подписок (**dispose**))**
3. Преобразует сырые значения в читаемый формат.

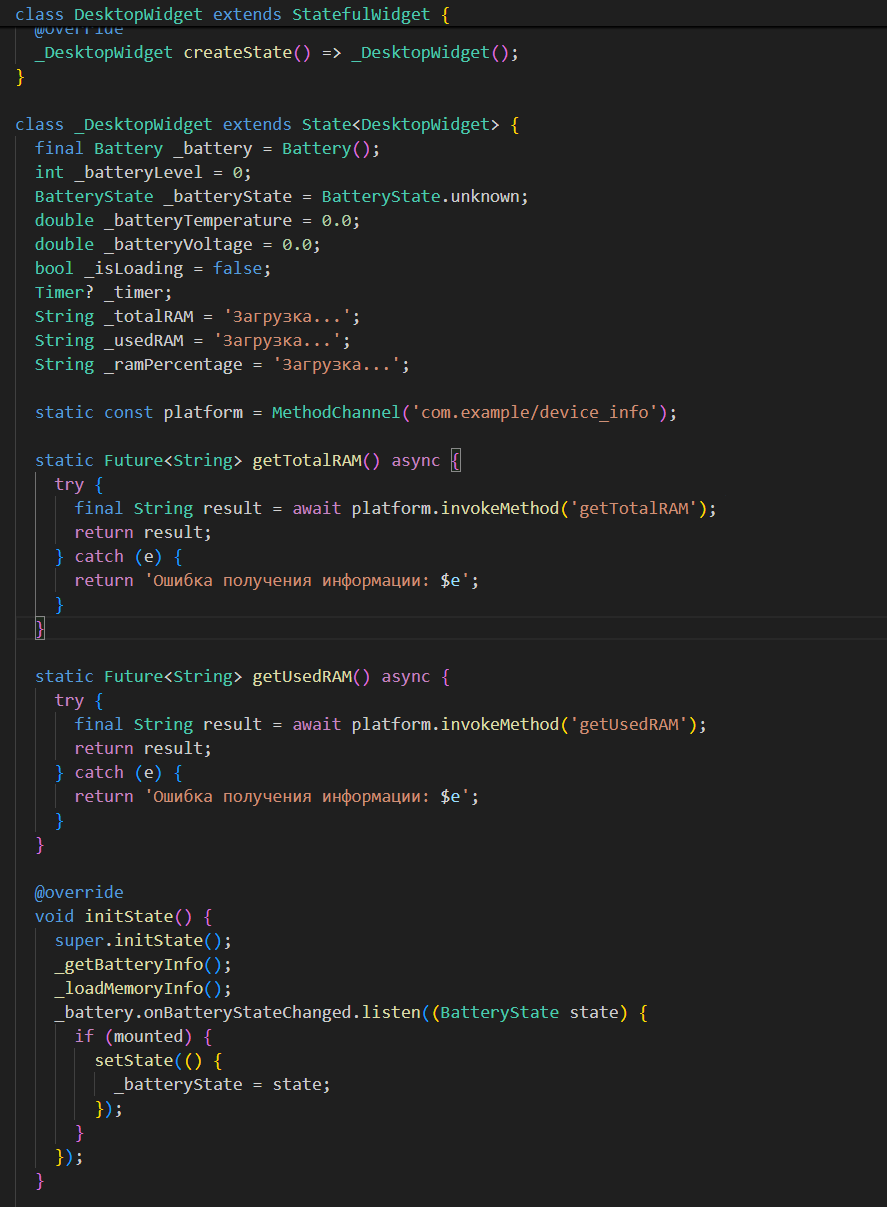


Рис-20 Desktop\_widget.dart

Компактный виджет для мониторинга ключевых параметров системы с автоматическим обновлением и удобной визуализацией данных, **Основные элементы:**

1. **Индикатор заряда батареи**:
   * Круговой индикатор с процентным значением
   * Цветовая индикация (зеленый/оранжевый/красный)
   * Текстовое описание состояния (зарядка/разрядка)
2. **Карточки информации**:
   * Температура батареи (°C)
   * Напряжение (V)
   * Использование RAM (ГБ/%)
   * Общий объем RAM (ГБ)

Функции:

1. Получает текущие параметры батареи (\_getBatteryInfo())
2. Загружает данные об использовании памяти (\_loadMemoryInfo())
3. Запускает 10-секундное обновление данных (\_startUpdatingBatteryInfo())
4. Создает унифицированную карточку данных (\_buildInfoCard())



Рис-21 Test\_display\_screen.dart

Скрипт предназначен для проверки битых пикселей дисплея телефона, основные элементы:

1. Циклический перебор предустановленных цветов
2. Реакция на касание экрана (onTap)

Функции:

1. Циклически переключает индекс цвета в списке (\_changeColor())
2. Перехватывает касания по всему экрану (GestureDetector)

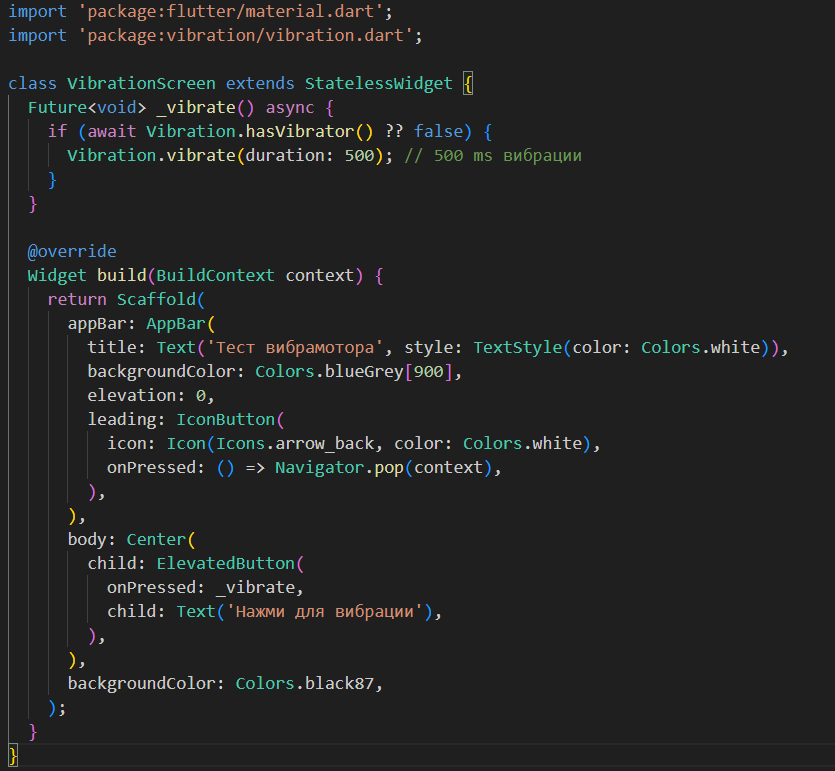


Рис-22 Vibration\_screen.dart

Скрипт предназначен для **тестирования вибромотора устройства, основные устройства:**

1. Кнопка, расположенная по центру экрана для активации теста
2. Проверка наличия вибромотора через Vibration.hasVibrator()

Функции:

1. Проверяет наличие вибромотора, активирует вибрацию на 500 мс (\_vibrate())
2. Работает через пакет vibration