

□ ПРОЕКТ: Flutter-приложение «SmartSchedule»

Тип: Мобильное и веб-приложение для расписания с офлайн-режимом и облачной синхронизацией

Технологии: Flutter, Firebase, SQLite, FCM

Роль: Полный цикл — от архитектуры до реализации

Цель проекта: Создание персонального помощника по времени, который работает даже без интернета

1. Общая концепция и цели проекта

Приложение "SmartSchedule" разработано как лёгкий и кроссплатформенный планировщик с возможностью **работы офлайн**, облачной синхронизации, удобного визуального представления расписания, напоминаний и настройки поведения по дням недели.

Цель — не просто дать пользователю таблицу с уроками или задачами, а создать **интуитивный цифровой ежедневник**, который подстраивается под образ жизни и устройства (Android, Web, iOS).

Ключевые задачи:

- Упрощение ручного ввода расписания и событий
- Возможность синхронизации между устройствами
- Push-напоминания о важных событиях
- Работа без интернета (автосинхронизация при подключении)

2. Функциональность и пользовательские сценарии

□ **Блок расписания:**

- Просмотр по дням недели / календарю
- Временные слоты: 5 / 15 / 30 / 60 минут
- Цветовая маркировка событий (по категориям: учёба, работа, личное)
- Поддержка повторяющихся событий (ежедневно, по будням, выборочно)

□ **Добавление / редактирование:**

- Форма добавления события с:
 - временем начала / конца
 - цветом
 - описанием
 - флагом "повторять" и "напомнить за X минут"
- Поддержка "быстрого создания": нажал на слот — заполнил за 3 клика

□ **Синхронизация и авторизация:**

- Авторизация: через Google или email+пароль (Firebase Auth)
- Синхронизация: Firestore (каждое событие = документ)
- Обновления в реальном времени (если есть интернет)
- Автоматическое сохранение изменений в SQLite (если офлайн)

□ **Уведомления:**

- Firebase Cloud Messaging: пуши за 5/10/30 минут до события
- Встроенные напоминания (локальные) на Android через flutter_local_notifications

□ **Настройки:**

- Светлая/тёмная тема
- Формат времени (12/24)
- Старт недели: понедельник / воскресенье
- Язык: RU / EN

3. Архитектура проекта

Платформа:

- Flutter 3.10+
- Null safety, Material 3 design
- Адаптация под Web и Android

Состояние:

- Riverpod — управление состоянием событий, авторизации и локального кэша

- StateNotifierProvider + хранилище (SharedPreferences)

Данные:

- Firestore: коллекции пользователей, под-коллекции events
- SQLite (через Drift): таблица локальных событий, флаги синхронизации
- Слияние данных при включении интернета — приоритет облака

Сервисы:

- Firebase Auth (Google, Email)
- Firebase Cloud Messaging
- Drift (SQL генерация)
- Flutter packages:
 - table_calendar
 - flutter_colorpicker
 - flutter_local_notifications
 - cloud_firestore
 - firebase_auth

CI/CD:

- Flutter build для Web и Android
- Firebase Hosting (для веб-версии)
- Приватный APK под Android, сборка вручную или через Codemagic

4. UI / UX и визуальные решения

Главный экран:

- Панель навигации внизу (Расписание / Добавить / Профиль)
- Отображение недели, текущего дня и всех задач
- Цветные блоки по времени
- Свайп вправо — редактировать, свайп влево — удалить

Экран добавления события:

- Всплывающее модальное окно
- Форма из 5 шагов (время, повторение, цвет, напоминание, описание)
- Быстрые шаблоны (например: "Занятие", "Работа", "Спорт")

Профиль пользователя:

- Имя, аватар (из Google), кнопка «Выйти»
- Количество запланированных событий
- Последняя синхронизация

Дополнительно:

- Иконки `lucide`, `material symbols`
- Визуальные анимации при добавлении и удалении событий
- Плавные переходы между неделями и днями

5. Будущее расширение и потенциал проекта

- Импорт расписания из Excel / PDF / Google Calendar
- Режим "группы": общее расписание для нескольких пользователей
- Виджеты под Android / iOS
- AI-генератор расписания по шаблонам: “Заполни неделю по режиму”
- Web-доступ для админов (например, преподавателей)
- Premium-функции: облачное хранилище, темы, экспорт

□ Результат и статус

Приложение собрано под Android и Web, протестировано на 3 устройствах.
Работает в офлайн-режиме, синхронизируется при подключении.
В базе более 300 событий, сохранённых пользователями.