

Сопроводительное письмо

к тестовому заданию
«Календарь тренировок»

Исполнитель:

Андрей Ефимов, Москва
+7(916)510-09-34
andrey.efimov.dev@gmail.com

20.12.2025

Содержание

1. Понимание задачи.
2. Реализованный функционал.
3. Результаты тестирования.
4. Временные ресурсы.
5. Ссылка на исходный код в репозиторий GitHub.
6. Выводы и итоги.

Понимание задачи

На основании полученного ТЗ мое понимание задачи включает следующие ключевые требования:

1. Разработать **приложение-календарь** на Swift, фреймворк **SwiftUI**, с отображением разных видов тренировок в виде событий в календаре со следующими требованиями:
 - Отобразить календарь с отметками дней, в которые были тренировки.
 - При клике на день — показать список тренировок.
 - При клике на тренировку — показать детали (дистанция, длительность, тип).
 - Бонус: построить график пульса из данных `diagram_data`.
2. Максимально использовать **ИИ** для написания кода.
3. Использовать архитектуру **MVVM + Coordinator**.
4. Использовать данные для приложения, предоставленные в **JSON** формате.
5. Адаптировать приложения под под **светлую/тёмную тему**.
6. Обеспечить корректную работу на разных устройствах (iPhone SE / iPhone 15 Pro Max).
7. Создать **Unit Tests** тесты для: (1) фильтрации событий по дню, (2) переключение месяцев, (3) форматирования дат и покрытие основных `ViewModel`.
8. Создать приложения без использования сторонних не нативных UI-фреймворков.
9. Загрузить готовый код в репозиторий и предоставить ссылку.

Реализованный функционал

Календарь тренировок

- Месячный просмотр с выделением дней тренировок
- Навигация по месяцам (предыдущий/следующий)
- Визуальные маркеры для дней с активностями

Типы тренировок (5 категорий)

- **Ходьба/Бег** (Walking/Running) - иконка `figure.walk`, цвет желтый
- **Йога** (Yoga) - иконка `figure.mind.and.body`, цвет фиолетовый
- **Водные процедуры** (Water) - иконка `drop.fill`, цвет синий
- **Велоспорт** (Cycling) - иконка `bicycle`, цвет зеленый
- **Силовая** (Strength) - иконка `dumbbell.fill`, цвет оранжевый

Детали тренировки

- **Метаданные**: дистанция, длительность, температура, влажность

- **График ЧСС** (21 точка данных)
- **График активности**
- **Комментарии пользователя**

Техническая реализация

- **SwiftUI** - современный декларативный UI
- **MVVM** - чистая архитектура
- **async/await** - асинхронные операции
- **@Published свойства** - реактивное обновление UI
- **Покрытие тестами** - 8 рабочих unit-тестов

Пользовательский интерфейс

- **WorkoutCard**: Карточка тренировки с иконкой, типом и временем
- **HeartRateChartView**: График сердечного ритма
- **CalendarContainerView**: Основной календарный интерфейс
- **WorkoutDetailView**: Детальный просмотр тренировки
- **DayEventsView**: Список тренировок за день

Технические особенности

1. **Поддержка русского языка** - локализованные названия тренировок
2. **Адаптивный дизайн** - работа на разных размерах экрана
3. **Плавная навигация** - через AppCoordinator
4. **Обработка ошибок** - устойчивость к сетевым проблемам

Статистика

- **9 тестовых тренировок** в демо-данных
- **21 точка данных** на графике **Частота Сердечных Сокращений (ЧСС)**
- **5 категорий** тренировок
- **8 рабочих unit-тестов**
- **3 основных экрана** (календарь, день, детали)

Светлая/Темная тема

- Программная настройка **световой темы** приложения – светлая, темная, системная.
- Реализация **адаптивных цветов** для светлой и темной световых тем.

Стартовая загрузка

- Настройка **LaunchScreen.storyboard**.
- Кастомный интерфейс запуска приложения **LaunchView** с анимированными элементами.

Результаты тестирования

CalendarViewModelTests

Тест	Что проверяет	Статус
testWorkoutsFilteringByDate	Фильтрация тренировок по дате	
testHasWorkoutsOnDay	Проверка наличия тренировок	

Тест	Что проверяет	Статус
testCurrentMonthYearFormat	Форматирование дат	✓
testFetchWorkoutsForSpecificDate	Запрос тренировок по дате	✓
testWorkoutDetailErrorHandler	Обработка ошибок	✓

HeartRateChartIntegrationTest ✓

Тест	Что проверяет	Статус
testHeartRateChartInWorkoutDetail	Интеграция с сервисом	✓
testHeartRateChartViewCreation	Создание SwiftUI view	✓
testHeartRateChartPerformance	Производительность (0.003 сек)	✓

Временные ресурсы

1. Создание кода с помощью ИИ, загрузка в репозиторий GitHub, привязка проекта к GitKraken – **2,5 часа** (12.12.2025)
2. Отладка кода, устранение багов, доработка UI/UX, адаптация светлой/темной темы, реализация UI стартовой загрузки приложения, прогон тестов, подготовка и загрузка **README** файла – **8,0 часов** (15.12.2025).
3. Подготовка отчета (сопроводительное письмо) по выполненному заданию – **1,0 час** (15.12.2025).

ИТОГО: 11,5 часов.

Ссылка на исходный код в репозитории GitHub

<https://github.com/AndreyEfimovDev/CalendarTestTask>

Выводы и итоги

- Все ключевые требования ТЗ выполнены.
- Дополнительно реализован график ЧСС (бонус-задача) и кастомный экран запуска.
- Приложение полностью протестировано, пользовательский интерфейс адаптирован под разные устройства (проверка iPhone SE и iPhone 15 Pro Max) и световые темы (светлая/тёмная).
- Код доступен по ссылке для проверки.