**Техническое задание (ТЗ) на мобильное приложение «MiniHabit»**

**1. Общая информация**

**Название проекта:** MiniHabit (условное)  
**Назначение:** Помощь пользователю в планировании и отслеживании действий/привычек с возможностью выставления напоминаний в заданное время, а также просмотра задач на сегодня.  
**Целевая платформа:** Мобильное приложение (iOS и Android).  
**Целевая аудитория:** Широкий круг пользователей, желающих организовать свои дела/привычки и получать своевременные напоминания.

**2. Цели и задачи**

1. **Автоматизация планирования**: Пользователь задаёт планы (одноразовые или ежедневные), указывает название, дату и/или время, а приложение напоминает ему о наступлении времени.
2. **Отслеживание актуального на сегодня**: Отображение списка задач, предназначенных на текущую дату или ежедневных задач, ещё не выполненных сегодня.
3. **Простота и лаконичность интерфейса**: Минималистичный, современный дизайн в тёмной теме, удобная навигация, чёткая структура с минимумом «шума».
4. **Надёжность и автономность**: Приложение должно корректно работать офлайн и сохранять данные локально; уведомления планируются локально.
5. **Высокая юзабилити**: Большие кнопки, удобные тапы, понятные формы, быстрое добавление/удаление/отметка задач.
6. **Единый визуальный стиль**: Консистентный дизайн во всех экранах; продуманный логотип и иконка, отражающие суть: «календарь + отметка/чек» или «выполненная задача», без избыточных деталей.
7. **Масштабируемость**: В будущем возможное расширение функционала (редактирование задач, категории, статистика, синхронизация и т.п.), поэтому структура проектирования UI/UX должна быть гибкой.

**3. Ключевые сценарии использования**

1. **Добавление нового плана**
   * Пользователь переходит в раздел «Добавить» (через нижнюю навигацию).
   * Вводит название задачи/привычки.
   * Выбирает время (обязательно).
   * Опционально задаёт дату (если задача одноразовая) или переключает «Повторять каждый день» (если ежедневная).
   * Подтверждает добавление.
   * После добавления сразу обновляется список «Планы» и «Сегодня» (если дата совпадает или ежедневная). Запускается (или пересоздаётся) локальное напоминание.
2. **Просмотр всех планов**
   * Пользователь находит раздел «Планы» (нижняя вкладка).
   * Видит список всех созданных задач, отсортированный от ближайшей по дате/времени к более отдалённым.
   * Каждая карточка показывает: название, краткую информацию о расписании («Каждый день в HH:MM» или «DD.MM.YYYY в HH:MM»), а также, при желании, строку «Следующее: Сегодня в HH:MM» или «DD.MM.YYYY в HH:MM».
   * Между карточками — отступ примерно 10 dp, чтобы не сливались.
   * Кнопка удаления (иконка «✕») справа: размер 40×40 dp, круглой формы. При нажатии удаляется задача и отменяется связанное напоминание.
3. **Просмотр задач на сегодня**
   * Пользователь открывает раздел «Сегодня».
   * Видит список задач, у которых дата совпадает с сегодняшней или ежедневные задачи, ещё не отмеченные сегодня.
   * Сортировка в пределах сегодняшних задач — по времени (от ближайшего).
   * Каждая карточка: название, время (HH:MM) и кнопка «✓» справа размером 40×40 dp. При нажатии:
     + Если задача ежедневная: отмечается как выполненная (устанавливается флаг «выполнено сегодня»), напоминание на сегодня отменяется; приложение пересоздаёт новое напоминание на следующий день в то же время.
     + Если одноразовая: удаляется задача и отменяется напоминание.
   * После отметки список «Сегодня» обновляется мгновенно: удалённая/отметленная задача исчезает.
4. **Уведомления (напоминания)**
   * Приложение запрашивает у пользователя разрешение на локальные уведомления при первом создании задачи или при старте (в зависимости от платформы).
   * При добавлении и при каждом изменении списка задач (добавление/удаление/отметка) пересоздаются локальные уведомления:
     + Для одноразовых задач: если дата+время ещё в будущем, запланировать уведомление. Если время/дата уже прошли, уведомление не планируется.
     + Для ежедневных: если задача ещё не выполнена сегодня и время ещё не прошло — запланировать уведомление на сегодня; если время уже прошло или задача уже помечена выполненной сегодня — запланировать уведомление на следующий день в указанное время.
   * При удалении задачи или отметке ежедневной на сегодня — отменить текущее запланированное уведомление, после этого пересоздать для следующего дня (для ежедневных) или вовсе не планировать (для одноразовых).
5. **Обновление данных в реальном времени**
   * После добавления новой задачи через раздел «Добавить»: списки «Планы» и «Сегодня» обновляются без перезапуска приложения.
   * Раздел «Сегодня» должен мгновенно отражать новые задачи, если они актуальны для сегодняшней даты. Это достигается вызовом методов обновления (load + фильтрация) у соответствующего экрана сразу после добавления.

**4. Функциональные требования**

**4.1. Структура экранов и навигация**

* **Нижняя навигация (Tab Bar)** с тремя вкладками:
  1. **«Планы»** — просмотр всех задач.
  2. **«Сегодня»** — просмотр задач на текущий день.
  3. **«Добавить»** — экран для создания новой задачи.
* На каждом экране отсутствует стандартный отступ под системный ActionBar: используется свой заголовок или вообще заголовок может быть минимален, в зависимости от дизайна.
* Экран «Добавить» не входит в стек возврата (чтобы при нажатии кнопки «Добавить» из этого экрана не было привычного «назад» в стеке): после сохранения приложение переключается на «Планы» или «Сегодня» по логике, без лишнего экрана-на-зад.

**4.2. Экран «Планы»**

* Заголовок: «Планы».
* Список карточек задач:
  + Сортировка: ближайшая по дате/времени сверху.
  + Всё содержимое: карточки выстраиваются одна под другой, с отступом между ними ~10 dp.
  + Карточка: фон тёмного оттенка (#1E1E1E или аналог), скруглённые углы, паддинги внутри (например, 16 dp).
  + Внутри карточки:
    - Слева:
      * Название задачи крупным шрифтом (например, 18sp), белый/светлый цвет.
      * Под названием: строка описания расписания («Каждый день в HH:MM» или «DD.MM.YYYY в HH:MM»), цвет чуть светлее или приглушённый (например, #BBBBBB).
      * (Опционально) Ещё строка «Следующее: Сегодня в HH:MM» или «DD.MM.YYYY в HH:MM», шрифт мелкий (#888888, 12sp).
    - Справа: кнопка удаления:
      * Размер 40×40 dp, круглая (border-radius 20 dp).
      * Фон: красный оттенок (#EF4444). Иконка «✕» или графический крестик.
      * Иконка или текст «✕» белого цвета, выровнен по центру.
  + При тапе по пустому месту карточки можно не реализовывать действие (или позже добавить редактирование). Сейчас tap не вызывает ошибку, но placeholder-метод noop.
* Если список пуст: показывается информационная зона (Label) с текстом «Нет планов. Перейдите на вкладку «Добавить», чтобы создать план.», центрировано горизонтально, цвет приглушённый (#888).

**4.3. Экран «Сегодня»**

* Заголовок: «Сегодня».
* Список карточек задач на текущий день:
  + Выводятся только задачи, удовлетворяющие:
    - Ежедневные задачи, у которых флаг lastDone ≠ сегодняшняя дата.
    - Одноразовые задачи, у которых дата = сегодняшняя дата.
  + Сортировка по времени (HH:MM) возрастание.
* Карточка:
  + Оформление аналогичное «Планы» (фон, скругления, паддинг).
  + Слева: название крупным и время под ним мелким.
  + Справа: кнопка «✓»:
    - Размер 40×40 dp, круглая (border-radius 20 dp).
    - Фон зелёный (#4ADE80), иконка «✓» чёрная или тёмная (#000) для контраста.
  + При нажатии:
    - Для ежедневной: установить lastDone = сегодня, сохранить, пересоздать уведомления; карточка исчезает из списка «Сегодня», останется в «Планы» для дальнейших дней.
    - Для одноразовой: удалить задачу из списка "plans" и сохранить, уведомление отменяется; карточка исчезает.
* Пустой список: Label «Нет планов на сегодня. Перейдите на вкладку «Добавить».»

**4.4. Экран «Добавить»**

* Заголовок: «Новый план» или «Добавление задачи».
* Форма ввода:
  + **TextField** для названия: placeholder «Введите название задачи».
  + **TimePicker** для выбора времени (обязательно).
  + **DatePicker** для выбора даты (если одноразовая).
  + **Switch** или чекбокс «Повторять каждый день»: если включён, при сохранении игнорировать дату (задача воспринимается как ежедневная).
* Кнопка «Добавить» или «Сохранить»:
  + Широкая кнопка с отступами, скруглениями (например, padding 10×16, border-radius 8).
  + Цвет: акцентный (синий #3B82F6 или аналог), текст белый.
* При переходе на экран поля сбрасываются (при каждом заходе новые значения по умолчанию: текущая дата/время, пустое название, switch «выкл»).
* После нажатия «Добавить»:
  + Если название пустое — не сохранять (можно кратко подсветить ошибку или не позволить сохранить).
  + Формирование объекта:
  + {
  + id: уникальный идентификатор (например, timestamp),
  + title: string,
  + date: "YYYY-MM-DD" или null,
  + time: "HH:mm",
  + recurring: Boolean,
  + lastDone: null
  + }
  + Сохранение в локальное хранилище (JSON-массив).
  + Планирование/перезапланирование уведомлений.
  + Обновление экранов «Планы» и «Сегодня» (через вызов методов обновления у соответствующих компонентов).
  + Переход/переключение на экран «Планы» (или другой логически предпочтительный).

**4.5. Логика уведомлений**

* **Запрос разрешений**: при первом создании напоминания или при старте приложения необходимо запросить локальные уведомления. Если отказано, уведомления не будут работать, но остальные функции остаются.
* **Хранение данных уведомлений**: каждый план содержит id, что позволяет отменять или перезапланировать конкретное уведомление.
* **Отмена уведомлений**:
  + При удалении задачи: отменить уведомление по id.
  + При отметке ежедневного: отменить текущее уведомление на сегодня и запланировать новое на следующий день.
* **Перезапуск уведомлений**: при любом изменении списка задач (добавление/удаление/отметка) выполнять логику «отменить старые» + «запланировать новые» для всего массива. Либо отменять только точечно (по id), но проще — очистить все запланированные и запланировать актуальные.
* **Поведение при старте приложения**: при запуске загружать все сохранённые задачи и запускать логику планирования уведомлений, чтобы учесть те, что были созданы ранее.

**5. Дизайн и UI/UX требования**

**5.1. Общая стилистика**

* **Тёмная тема**: фон большинства экранов — очень тёмный (близкий к чёрному или тёмно-серому, например, #121212 или #1E1E1E). Текст и иконки — светлые оттенки (#FFFFFF, #BBBBBB, #888888). Акценты (кнопки) — яркие цвета (синий, зелёный, красный).
* **Минимализм**: минимум лишних визуальных элементов, чёткие формы, достаточно пустого пространства для восприятия.
* **Скруглённые карточки**: радиус скругления ~12 dp.
* **Отступы и интервалы**:
  + Между карточками вертикальный отступ ~10 dp (например, margin-top и margin-bottom по 5 dp или margin-bottom 10 dp).
  + Внутри карточки паддинг ~16 dp.
  + Отступы по сторонам от границ экрана (левый/правый) ~16 dp.
  + Между элементами формы (TextField, метки, переключатель) — ~16 dp.
* **Кнопки действий**:
  + Размер иконок/кнопок подтверждения/удаления: 40×40 dp, круглые (border-radius 20 dp).
  + Основные кнопки (Добавить, Сохранить): горизонтальные по ширине с адекватными паддингами (10–16 dp), скруглённые углы (8–12 dp).
* **Шрифты**:
  + Заголовки: 20 sp (или 18–20), полужирные.
  + Текст задачи в карточке: 18 sp, белый.
  + Мелкий текст (описание расписания): 14 sp (#BBBBBB) или 12 sp (#888888) для второстепенной информации.
* **Навигация**:
  + Нижнее меню (Tab Bar) с тремя вкладками. Иконки вкладок опционально: простые контурные символы «Планы» (лист/список), «Сегодня» (иконка календаря или солнца), «Добавить» (плюс).
  + Поддержка индикации выбранной вкладки: иконка/текст другого цвета (акцентный) для активной вкладки.
* **Заголовок экранов**:
  + Убираем стандартный ActionBar (если он создаёт нежелательный отступ сверху).
  + Используем собственный header: невысокий блок (height ~56 dp) с фоном тёмно-серым (#1F1F1F) и выровненным по центру текстом заголовка.
* **Форма добавления**:
  + Элементы формы («Название», «Время», «Дата», «Повторять каждый день») располагаются вертикально, с равными отступами ~16 dp между ними.
  + Поле ввода (TextField): фон чуть светлее фона экрана (#1E1E1E), белый текст, внутренняя вставка ~10 dp, скруглённые углы (~8 dp).
  + TimePicker и DatePicker стандартизированы под тему (тёмный фон, светлый текст), если платформа поддерживает стилизацию; иначе помещены в контейнер с нужным фоном.
* **Placeholder состояния (пустой список)**:
  + Центрированная подсказка, приглушённый цвет, шрифт ~16 sp.
* **Анимации и отклики**:
  + Лёгкий отклик на нажатие: изменение прозрачности или фонового оттенка нажатой карточки/кнопки.
  + Анимация добавления/удаления карточки: можно добавить простое плавное исчезновение или сдвиг.
* **Иконка приложения (логотип)**:
  + Простой минималистичный стиль: символ календаря (или стилизованного круга с «шапкой» календаря) и внутри или поверх — знак «галочка» (чек) или символ подтверждения.
  + Монохромный или с одним акцентным цветом (например, зелёная галочка на тёмном фоне).
  + Логотип не перегружен деталями: простые геометрические формы, чистые линии, легко узнаваем в маленьком размере (иконка приложения).
  + Рекомендации:
    - Форма: квадрат/круг, без сложных теней, возможно плоский материал-дизайн.
    - Цветовая палитра: тёмный фон (или градиент тёмных оттенков) + яркий акцент (галочка).
    - Внутри иконки оставлять достаточно «пустого» пространства (padding), чтобы символы не сливались с краем.

**5.2. Цветовая палитра (пример)**

* Фон экрана: #121212 или #1E1E1E.
* Карточки: #1E1E1E или чуть отличающийся оттенок от фона.
* Текст основной: #FFFFFF.
* Текст второстепенный: #BBBBBB.
* Текст вспомогательный: #888888.
* Акцентные кнопки:
  + Добавить/Сохранить: #3B82F6 (синий) или другой яркий, контрастный на фоне.
  + Удалить: #EF4444 (красный).
  + Подтвердить (✓): #4ADE80 (зелёный).
* Заголовок header: #1F1F1F фон, текст #FFFFFF.
* Нижняя навигация: фон #1F1F1F, активный пункт акцентный цвет, неактивные — #BBBBBB.

**5.3. Размеры, отступы, спейсинг**

* **Header**: высота ~56 dp, без лишних внутренних отступов сверху.
* **Сторонние отступы**: слева/справа у контента ~16 dp.
* **Между элементами**: ~10–16 dp в зависимости от контекста.
* **Карточки**: margin-top и margin-bottom по ~5 dp (итого между списком 10 dp), margin-слева/справа ~16 dp. Паддинг внутри карточки ~16 dp.
* **Кнопки подтверждения/удаления**: размер 40×40 dp, радиус скругления 20 dp; иконка внутри центрирована.
* **TextField**: высота авто, внутренняя вставка ~10 dp, скруглённые углы ~8 dp.
* **Кнопка «Добавить»**: ширина — по контенту или растягивается по ширине контейнера с отступами; высота ~40–48 dp; скругление ~8–12 dp.

**5.4. Шрифты и типографика**

* Использовать системный шрифт или универсальный без засечек.
* Размеры:
  + Заголовок экрана: 20 sp, полужирный.
  + Название задачи: 18 sp, полужирный или средний вес.
  + Описание расписания: 14 sp, регулярный вес, цвет приглушённый.
  + Мелкий текст (next occurrence): 12 sp, цвет более приглушённый.
  + Placeholder и подсказки: 16 sp, цвет #888888.
* Цвет текста: контрастный относительно фона (в тёмной теме — светлые).
* Интерлиньяж: достаточный для читабельности (1.2–1.4).

**5.5. Иконка/логотип**

* **Форма**: максимально простая, легко узнаваема при уменьшенном размере (например, 48×48 px).
* **Смысловой образ**: календарь/часы/таймер/галочка. Лучше: стилизованный календарь (две «шапки» сверху) либо круглый «часы»-контур, внутри яркая галочка.
* **Цвет**: фон тёмный или прозрачный, галочка яркая (зелёная или другой акцент). Можно несколько вариантов: монохромная (белая на тёмном фоне) и цветная для разных тем.
* **Избегать мелких деталей**: нет теней, градиентов с сильными переходами, мелкого текста. Лаконичный силуэт.
* **Примерное описание**: на тёмном квадратном фоне — контур календаря или круга, внутри смещённая слегка вправо-вниз зелёная галочка, чтобы ассоциировать «задача выполнена» или «напоминание».

**6. Нефункциональные требования**

**6.1. Производительность и ресурсопотребление**

* Приложение должно быстро стартовать и реагировать на действия пользователя без заметных задержек.
* Хранение данных локальное: использовать встроенный механизм хранения (например, key-value JSON или легковесную БД), минимизируя задержки чтения/записи.
* Управление уведомлениями должно быть оптимальным: при пересоздании уведомлений избегать избыточных операций, если возможно.

**6.2. Надёжность и устойчивость**

* Высокая устойчивость к ошибкам: проверка корректности данных (валидный формат времени, даты).
* Обработка отказа в разрешении уведомлений: приложение продолжает работать, просто уведомления не запланируются, пользователь информирован (необязательно всплывашка, но логично уведомить в интерфейсе).
* Корректная работа при смене системного времени или часового пояса: уведомления могут пересчитываться при перезапуске приложения (при старте подгружать сохранённые задачи и планировать заново).
* Защита от потери данных: данные должны сохраняться сразу при изменении, минимизируя риск потери при форсированном закрытии.

**6.3. Кроссплатформенность**

* UI/UX единый, но с учётом платформенных особенностей (стандартный вид селекторов времени/даты).
* Локальные уведомления корректно работают на обеих платформах (iOS/Android) с учётом требований к разрешениям.

**6.4. Юзабилити и доступность**

* Большие интерактивные зоны (кнопки, элементы списка).
* Контраст текста и фона должен соответствовать минимуму по доступности (WCAG AA) для тёмной темы.
* Поддержка масштабирования текста (если платформа позволяет).
* Чёткие метки, отсутствие неочевидных элементов.
* Ясные сообщения-подсказки, когда список пуст или при ошибке ввода (название задачи пустое).
* Поддержка RTL (опционально) в будущем: не жёстко захардкоженные отступы, а с учётом направления текста.

**6.5. Локализация**

* Структура должна поддерживать перевод строковых ресурсов: тексты заголовков, подсказок, кнопок. В текущем ТЗ русский язык, но предусмотреть возможность расширения на другие языки.

**6.6. Безопасность и конфиденциальность**

* Локальное хранение данных не содержит чувствительной информации: названия задач пользователь вводит сам.
* Не требуется сетевое взаимодействие по умолчанию, поэтому риски минимальны.
* При расширении функционала (синхронизация в облако) предусмотреть шифрование/авторизацию, но сейчас не входит в ТЗ.

**7. Хранение данных**

* **Структура хранилища**: локальное хранилище JSON-массив объектов задач. Ключ «plans» → JSON-массив.
* **Формат объекта**:
* {
* "id": 1633036800000,
* "title": "Выпить воду",
* "date": "2023-10-01", // строка в формате YYYY-MM-DD или null
* "time": "08:30", // строка "HH:mm"
* "recurring": true, // true для ежедневной задачи
* "lastDone": null // строка "YYYY-MM-DD" или null
* }
* **Операции**:
  + При добавлении: считать существующий JSON, добавить объект, записать обратно.
  + При удалении/отметке: изменить массив, записать.
  + При загрузке экрана: считать JSON, распарсить, затем обработать.
* **Миграции**: если в будущем структура изменится, предусмотреть версионирование хранилища или механизм миграции.

**8. Логика уведомлений**

* При старте приложения (или при первом отображении «Планы») делать:
  + Загрузить все задачи.
  + Для каждой задачи вычислить ближайшее срабатывание и запланировать уведомление (или после отмены старых).
* При изменении (добавление/удаление/отметка):
  + Отменить (точечно или массово) старые уведомления.
  + Запланировать уведомления для актуального списка.
* Проверять доступность/разрешение уведомлений:
  + Если разрешение отсутствует, при попытке планирования уведомлений показать лог или подсказку (не аварийная ошибка). Можно при первом создании задачи запрашивать разрешение.
* Для ежедневных задач: каждый раз после отметки обновлять lastDone и запланировать новое на следующий день.
* При смене даты (полночь) реальное событие может не произойти, если приложение закрыто: важно при следующем запуске/возвращении в приложение пересчитать уведомления. Поэтому при запуске/возврате в foreground нужно выполнять инициализацию уведомлений.

**9. API и компоненты**

* **Компонент/модуль хранения**: предоставляет методы:
  + loadAllPlans(): Plan[]
  + saveAllPlans(plans: Plan[]): void
* **Компонент/модуль уведомлений**: предоставляет методы:
  + initNotifications(plans: Plan[]): Promise<void> — проверка разрешений, отмена старых, планирование новых.
  + scheduleNotifications(plans: Plan[]): Promise<void> — запланировать уведомления без отмены (если нужно).
  + cancelNotification(id: number): Promise<void> — отменить одно уведомление.
  + cancelAllNotifications(): Promise<void> — отменить все.
* **Экран «Планы»**:
  + Метод loadPlans() вызывает loadAllPlans(), обновляет локальную переменную, вызывает пересортировку и отображение.
  + Метод deletePlan(id) изменяет массив, вызывает saveAllPlans(), инициализирует уведомления.
* **Экран «Сегодня»**:
  + Метод loadPlans(), filterToday(): отфильтровать по today; метод markDone(plan) изменяет lastDone или удаляет и вызывает saveAllPlans(), инициализирует уведомления и обновляет представление.
* **Экран «Добавить»**:
  + Метод savePlan(data) вызывает callback в родительском контроллере или напрямую loadAllPlans(), saveAllPlans(), initNotifications(), и вызывает обновление представлений «Планы» и «Сегодня».
* **Навигация**:
  + Нижняя вкладочная навигация: переключение между экранами «Планы», «Сегодня», «Добавить». При добавлении переключение на «Планы».

**10. Тестирование и критерии приёмки**

1. **Добавление задачи**:
   * При вводе имени, выборе времени и даты/ежедневного режима новая задача добавляется.
   * Сразу после добавления она появляется в списке «Планы».
   * Если дата задачи = сегодня или это ежедневная, она сразу появляется в «Сегодня».
   * Планируется уведомление.
2. **Просмотр «Планы»**:
   * Список упорядочен по ближайшему времени (учёт даты).
   * Между карточками виден отступ ~10 dp.
   * Кнопка удаления 40×40 dp, при нажатии удаляет задачу и уведомление отменяется.
3. **Просмотр «Сегодня»**:
   * Список содержит только актуальные задачи.
   * Сортировка по времени.
   * Кнопка «✓» 40×40 dp; при нажатии ежедневная задача помечается выполненной (удаляется из «Сегодня»), одноразовая удаляется полностью.
   * После отметки уведомление отменяется, а для ежедневных планируется следующее.
4. **Уведомления**:
   * Если разрешения даны, уведомления приходят в заданное время (для ежедневных: повторяются правильно после отметки или на следующий день).
   * Если разрешений нет, приложение не падает, но уведомления не показываются.
   * При перезапуске приложения уведомления пересоздаются корректно.
5. **UI**:
   * Нет лишнего отступа сверху (собственный header сразу под статус-баром).
   * Все кнопки имеют правильный размер (40×40 dp для «✓»/«✕»).
   * Элементы списков корректно расположены и не накладываются друг на друга.
   * Нижняя навигация работает корректно: три вкладки.
   * Тёмная тема соблюдается на всех экранах.
6. **Юзабилити**:
   * Быстрый отклик на нажатия.
   * Чёткие тексты подсказок при пустом списке.
   * Поля ввода валидируются: нельзя добавить пустое название.
7. **Работа при смене даты/часового пояса**:
   * Если приложение оставлено открытым при смене даты, после возвращения в foreground пересчитываются уведомления.
8. **Тест на разных экранах**:
   * Проверить на устройствах с разными разрешениями, плотностями, чтобы отступы/размеры кнопок оставались адекватны.
   * Проверить поведение при ориентации (если поддерживается) или зафиксировать портретный режим.
9. **Локализация**:
   * Тексты вынесены в ресурсные файлы, легко переводятся.
10. **Код и архитектура**:
    * Модули хранения и уведомлений покрыты модульными тестами (опционально), логика сортировки и фильтрации протестирована.
    * UI-компоненты отделены от бизнес-логики, чтобы в будущем можно было расширять функционал.

**11. Документация**

* **README**: Описание целей, инструкции по сборке и запуску (без упоминания конкретных технологий в ТЗ, но при передаче разработчикам указываются технологии).
* **Описание компонентной архитектуры**: какие модули (хранение, уведомления, экраны), как они взаимодействуют.
* **Руководство по стилю**: цвета, размеры, отступы, шрифты, ресурсы (иконки).
* **Файл спецификации логотипа**:
  + Формат векторный (SVG) и растровый (PNG) разных размеров для иконки приложения (минимум: 48×48 px, 72×72, 96×96, 144×144, 192×192).
  + Описание элементов: силуэт календаря или круглой формы + галочка.
  + Цветовая схема логотипа: варианты для светлой/тёмной темы (монохромный или с акцентом).
* **Тест-кейсы**: сценарии добавления, удаления, отметки, работы уведомлений, UI-валидаторы (проверка отсутствия смещений, корректности отступов), проверка пустых состояний.