Цель проекта: Создать замкнутое пространство услуг, чтобы пользователь не покидал приложение в поисках функционала, а покрывал все потребности внутри одного продукта

Метрики:

* Как часто пользователь заходил в похожие разделы. Например, корреляция использования «доставка продуктов на дом» и «покупка билетов онлайн»
* Как часто пользователь возвращается к нашим услугам (количество заходов в неделю/месяц и тд)
* Проблемы, которые пользователь хочет решить после внедрения функционала:
* Помимо приложения календаря (списка задач/дел), надо найти способ решить задачу – еще одно приложение
* Необходимость иметь приложение для каждой отдельной задачи: доставка, такси, банк и тд

User story:

* Покрыть базовые функции планировщика задач:
  + Возможность создавать/удалять и редактировать записи
  + Возможность ставить дедлайн по задаче
  + Возможность визуально отмечать выполненные задачи, например, зеленый-красный
  + Выполненные задачи не должны «маячить» перед пользователем, но юзер должен иметь возможность посмотреть их
  + Возможность создавать звуковые напоминания, уведомления, чтобы пользователь не забывал о своих задачах
* Связь с другими сервисами:
  + Добавить отдельным разделом в приложении все доступные сервисы:
    - Такси, доставка, покупка вещей онлайн и тд
  + Добавить поиск по ключевым словам, во время создания записи, по этим ключевым предлагать использование сервисов. Например, при создании записи «список продуктов: …» во всплывающем окне предлагать сервис по доставке еды.
  + Все услуги должны быть в одном пространстве, приложение должно запоминать действия пользователя. Например, если пользователь сохранил банковскую карту при оплате такси, нельзя требовать повторный ввод данных карты при оплате во всех остальных сервисах.
* Полезное сетевое окружение:
  + Делиться списком дел с другими пользователями
  + Создавать общие списки дел
* Фоновый процесс:
  + приложение должно отслеживать приближение дедлайна по задаче, отправлять уведомление
  + Уведомление о состоянии текущего заказа (например, как далеко курьер)

Нефункциональные требования:

* Общий список дел обязан обновляться в течение минуты у всех пользователей, которым он доступен
* Необходимо получить разрешение каждого пользователя на обработку их личной истории заказов и записей
* Приложение не может весить больше 1гб памяти, не должно тратить больше 10% заряда устройства при работе в фоновом режим

Нефункциональные требования определяют характеристики системы, которые не связаны напрямую с конкретными функциями, а касаются качества, производительности и других аспектов. Для проекта "Удобный планировщик задач" в Т-банке можно выделить следующие нефункциональные требования:

▎1. Производительность

• Время отклика: Система должна обеспечивать время отклика не более 2 секунд для выполнения основных операций (добавление, редактирование, удаление задач)

• Обработка нагрузки: Приложение должно поддерживать одновременное использование не менее 1000 пользователей без значительного ухудшения производительности.

▎2. Надежность

• Доступность: Система должна быть доступна не менее 99.5% времени в месяц, исключая запланированные технические работы.

• Восстановление после сбоев: В случае сбоя система должна восстанавливать данные за последние 24 часа без потерь.

▎3. Безопасность

• Аутентификация и авторизация: Должна быть реализована многофакторная аутентификация для защиты учетных записей пользователей.

• Шифрование данных: Все данные, передаваемые между клиентом и сервером, должны быть зашифрованы с использованием протокола HTTPS.

• Защита от атак: Система должна быть защищена от распространенных уязвимостей (например, SQL-инъекций, XSS, CSRF).

▎4. Удобство использования

• Интуитивно понятный интерфейс: Интерфейс должен быть простым и понятным для пользователя, с минимальным количеством кликов для выполнения основных действий.

• Доступность: Приложение должно соответствовать стандартам доступности (например, WCAG), чтобы быть удобным для пользователей с ограниченными возможностями.

▎5. Поддержка и масштабируемость

• Масштабируемость: Система должна быть спроектирована так, чтобы легко масштабироваться при увеличении числа пользователей и объема данных.

• Легкость интеграции: Возможность интеграции с другими системами и сервисами (например, API для внешних поставщиков услуг) должна быть предусмотрена на этапе проектирования.

▎6. Совместимость

• Кросс-браузерная поддержка: Приложение должно корректно работать на основных браузерах (Chrome, Firefox, Safari, Edge).

• Мобильная адаптация: Интерфейс должен быть адаптирован для мобильных устройств и планшетов.

▎7. Документация

• Техническая документация: Должна быть предоставлена полная техническая документация для разработчиков и администраторов системы.

• Пользовательская документация: Должны быть созданы руководства и справочные материалы для пользователей.

▎8. Тестирование и качество

• Автоматизированное тестирование: Необходима реализация автоматизированных тестов для проверки ключевых функций приложения.

• Качество кода: Код должен соответствовать стандартам кодирования и проходить регулярные ревью на предмет качества и читаемости.

Эти нефункциональные требования помогут обеспечить высокое качество работы приложения "Удобный планировщик задач" и удовлетворить ожидания пользователей.