

PS Задание было выполнено в первую неделю прохождения курса от VK

Описание задачи (на примере job story):

Когда пользователи планируют свои задачи, они **хотят** видеть в нашем приложении удобный планировщик задач, **чтобы** своевременно получать удобные и выгодные решения их проблем из нашего каталога сервисов.

Список систем/сервисов, с которыми можно будет интегрировать планировщик задач:

1. Система интернет-банкинга.
2. Сервис по работе с бизнесом.
3. Сервис по работе с инвестициями.
4. Система мобильной связи.
5. Сервис для планирования путешествий.
6. Сервис уведомлений.
7. Система (авто)платежей.
8. Прочие сервисы Тинькофф: кино, рестораны, театры и концерты, уборка, цветы, косметика, красота, шопинг, спорт, товары, топливо.

Схема интеграции планировщика задач с системами и сервисами:

комментарий: знакомиться с этим вопросом мы начали только сегодня на занятии, поэтому не стану пока заполнять его. буду готов сделать это позднее.

Примеры функциональных требований:

1. Пользователь может добавлять, редактировать и удалять задачи.
2. Система автоматически напоминает пользователю о задачах с помощью уведомлений.
3. Пользователь может настроить способ уведомления о задачах — задать интервал/время для напоминаний, выбрать удобный сервис для уведомлений (почта/push/смс).
4. Пользователь может запускать приложение на операционных системах Windows7 и выше, а также MacOS (на Android и IOS для мобильной версии).
5. Система может предложить пользователю оптимальное решение проблемы.
6. Пользователь может установить сроки выполнения задачи.
7. Пользователь может отказаться от предложенного системой решения.
8. Система должна запрашивать подтверждение пользователем каких-либо денежных операций.
9. Пользователь может выбрать английский язык для работы с планировщиком.
10. Система должна запрашивать у пользователя соответствующие разрешения для работы планировщика, если они еще не были даны приложению.

11. Пользователь может загружать в планировщик файлы форматов .doc, .jpg, .pdf, .xls.
12. Пользователь может запланировать периодически повторяющиеся задачи.
13. Пользователь может поделиться доступом к задаче из планировщика с другим пользователем, выбрав при этом его уровень доступа.
14. Система может предложить пользователю пройти инструкцию по работе с планировщиком.
15. Система может перенести или отложить задачу, если она не была решена в назначенные сроки.
16. Пользователь может расставлять приоритеты между задачами.
17. Пользователь может экспортировать данные планировщика в формат .xml.

Примеры нефункциональных требований:

1. Интерфейс планировщика должен быть простым и удобным для пользователя.
2. Все сбои в работе планировщика должны предусматривать соответствующую обработку.
3. ПО планировщика должно быть написано на том же ЯП, что и приложение.
4. Разработка ПО должна вестись на определенной платформе.
5. Продукт планировщика должен быть легко поддерживаемым.
6. Дизайн планировщика должен сочетаться с дизайном приложения.
7. Планировщик должен быть безопасным для работы с приватными данными.
8. Ожидание выполнения любого действия пользователя не должно превышать 1 с.
9. Данные пользователей и их задач должны храниться в БД.
10. Мобильная и десктопная версии планировщика должны синхронизироваться.