

"Планировщик задач" в Т-банке

1. Определение цели проекта и оценки выполнения

Цель проекта:

Предоставить пользователям удобный планировщик задач, который позволит:

- Заказывать различные услуги (доставка продуктов, бронирование билетов и т. д.) через интеграцию с сервисами банка.
- Получать кэшбэк за использование этих сервисов.

Зададим количественные и качественные метрики для определения эффективности проекта.

Метрики:

- **Количественные:**
 - Количество пользователей, использующих планировщик задач (ежедневно, ежемесячно).
 - Увеличение использования банковских сервисов благодаря интеграции с планировщиком.
 - Увеличение выручки от предоставленных услуг и начисленных кэшбэков.
 - Количество пользователей, которые использовали предложение о кэшбэке, сумма кэшбэка по всем пользователям.
- **Качественные:**
 - Оценка пользовательского опыта (UX) через анкеты или интервью.
 - Показатели удовлетворенности пользователей (например, через Net Promoter Score).
 - Анализ отзывов на доступных платформах.

2. Описание конкретного пользователя

Персона 1: Обычный пользователь (Casual User)

- **Цели:**
 - Удобно планировать повседневные задачи (заказать еду, забронировать билеты, заправить машину).
 - Экономить деньги через кэшбэк.
- **Проблемы:**

- Множество приложений для разных типов задач.
- Отсутствие централизованного места для планирования задач и получения выгодных предложений.

Персона 2: Представитель компании (Business User)

- **Цели:**
 - Интегрировать корпоративные задачи в систему для упрощения работы.
 - Отслеживать задачи, связанные с бизнес-потребностями (например, организация деловых встреч, бронирование услуг).
- **Проблемы:**
 - Нет интеграции с корпоративными сервисами.
 - Неудобство в планировании задач и их отслеживании.

3. Описание функциональных требований

User Story, основные возможности:

1. Как пользователь, я хочу иметь возможность создавать задачи;
2. Как пользователь, я хочу иметь возможность отмечать выполненными задачи;
3. Как пользователь, я хочу иметь возможность удалять задачи;
4. Как пользователь, я хочу получать уведомления о приближающихся сроках выполнения задач, чтобы не пропустить важные события;
5. Как пользователь, я хочу просматривать документы к задаче (например, билеты в театр) для получения доступа к информации в плане и с целью не потерять важные документы по важным событиям;
6. Как пользователь, я хочу выделять важные задачи, чтобы не упустить главного;
7. Как пользователь, я хочу планировать совместные события для важных встреч, с целью напоминания для каждого участника;
8. Как пользователь, я хочу получить кэшбэк за выполнение задач через сервисы банка.

Я не добавила возможность редактирования, так как считаю, что можно ограничиться созданием и удалением задачи.

Модель статусов задач:

- **Создана** – задача только что добавлена.

- **В процессе** – задача в процессе выполнения.
- **Завершена** – задача выполнена.
- **Отменена** – задача отменена пользователем.

4. Приоритеты функциональностей

Можно использовать метод MoSCoW для расставления приоритетов функциональностей.

MoSCoW приоритеты:

Must have (Обязательно):

- Как пользователь, я хочу иметь возможность создавать задачи
- Как пользователь, я хочу иметь возможность отмечать выполненными задачи
- Как пользователь, я хочу иметь возможность удалять задачи
- Как пользователь, я хочу получать уведомления о приближающихся сроках выполнения задач
- Как пользователь, я хочу получить кэшбэк за выполнение задач через сервисы банка

Should have (Желательно)

- Как пользователь, я хочу просматривать документы к задаче (например, билеты в театр)
- Как пользователь, я хочу выделять важные задачи

Важные задачи требуют большего внимания, и эта функция улучшит их приоритетность.

Could have (Может быть)

- Как пользователь, я хочу планировать совместные события для важных встре

Won't have (Не нужно)

- Редактирование задач

Редактирование задач можно опустить на первом этапе, чтобы избежать излишней сложности в интерфейсе.

5. Нефункциональные требования

- **Производительность:** Быстрая загрузка и обновление задач в реальном времени.
- **Безопасность:** Защита персональных данных пользователей и безопасности транзакций.

- **Удобство использования:** Интерфейс должен быть понятным и простым в навигации и использовании.
- **Масштабируемость:** Система должна поддерживать большое количество пользователей и сервисов без потери качества работы.

6. Моделирование бизнес-процессов

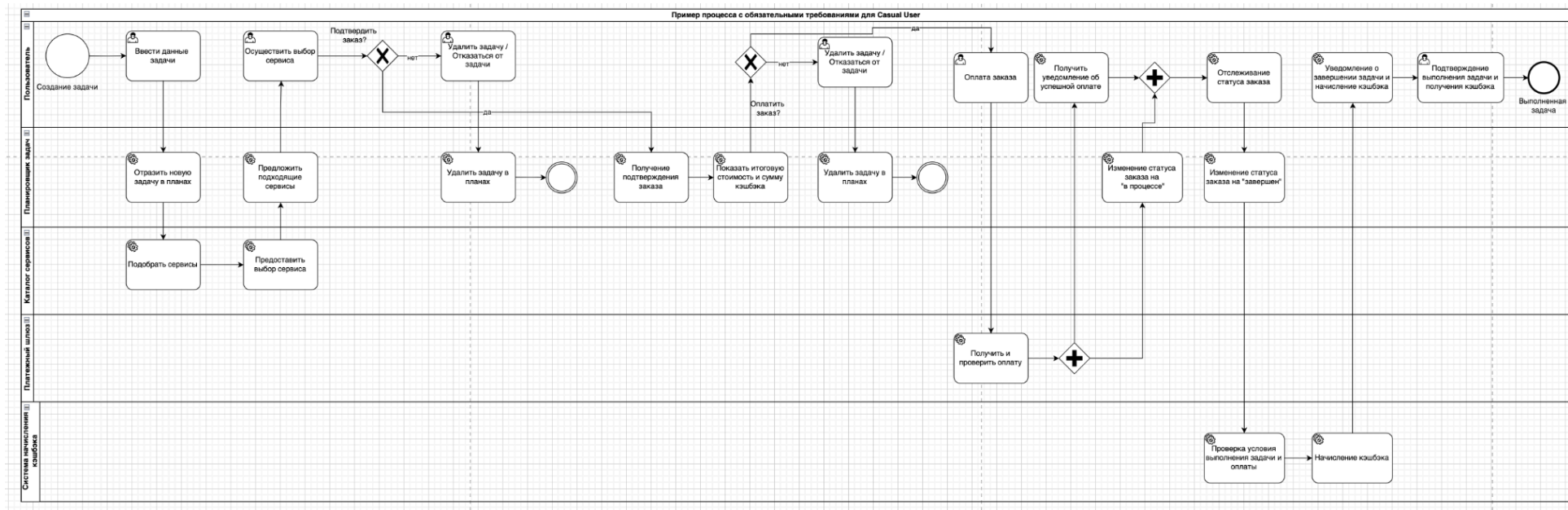
Для моделирования бизнес-процессов я использовала нотацию BPMN (Business Process Model and Notation), которая подходит для отображения шагов по созданию задачи, ее выполнению и начислению кэшбэка.

Пример процесса:

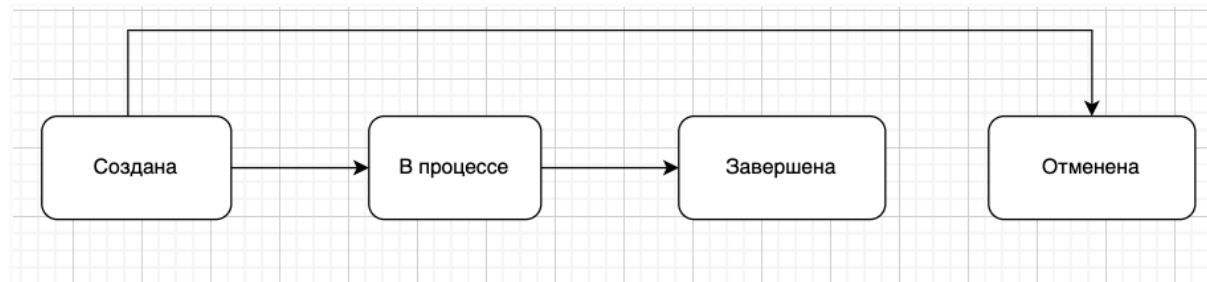
1. Пользователь создает задачу.
2. Планировщик ищет и предлагает подходящие сервисы.
3. Пользователь выбирает сервис, создает заказ.
4. После выполнения задачи пользователь получает кэшбэк.

Участники процесса:

- *Пользователь:* Обычный или бизнес-пользователь приложения.
- *Планировщик задач:* Модуль приложения, который управляет созданием и отслеживанием задач.
- *Каталог сервисов:* Система, которая предоставляет пользователю доступ к различным услугам (доставка еды, билеты и т.д.).
- *Платежный шлюз:* Система обработки платежей.
- *Система начисления кэшбэка:* Модуль банка, отвечающий за расчет и начисление кэшбэков.



7. Переход между статусами задач



Задача может быть отменена и “В процессе”, для краткости не прописывала это в схеме.

8. Прототипирование интерфейса

1. Главный экран:

- Список задач с кнопками "Добавить задачу", "Избранные", "Сервисы".
- Быстрый доступ к интегрированным сервисам (доставка, бронирование и т.д.).

2. Экран создания/редактирования задачи:

- Поля для ввода названия задачи, даты и времени.
- Кнопка для прикрепления документов.
- Опция отметить задачу как избранную.

3. Экран сервиса:

- Список доступных сервисов с возможностью фильтрации по категориям (доставка, бронирование и т.д.).
- Кнопка "Заказать" для быстрого доступа к услугам.

4. Экран кэшбэка:

- Отображение накопленного кэшбэка и истории транзакций.