Самостоятельная работа №4

Выполнили:

Федулов Алексей Васильевич

Шульга Иван Николаевич

**Часть №1:**

Функциональные требования определяют, что система должна делать. Они описывают конкретные функции и требования системы.

Примеры:

* Система позволяет пользователям регистрироваться через номер телефона и социальные сети
* Система должна отправлять уведомления пользователю

Нефункциональные требования определяют, как система должна работать. Они описывают характеристики системы, такие как производительность, безопасность и тому подобные…

Примеры:

* Карты, телефонные номера и пароли должны быть защищены шифрованием
* Система должна обрабатывать не менее 1500 запросов с секунду

Правильное оформление необходимо сразу по нескольким причинам:

* **Чёткость и понимание**: Чётко выраженные требования помогают всем участникам проекта понимать, что и как нужно реализовать. Что значительно снижает риск возникновения ошибок
* **Качество продукта**: Чёткие требования обеспечивают высокое качество конечного продукта, так как разработчики точно знают, что им нужно реализовать
* **Снижение рисков**: Чёткое понимание требований снижает риск возникновения проблем на поздних этапах, когда их исправление может быть дорогостоящим

Бизнес-требования для приложения по бронированию столиков в ресторанах:

* **Увеличение количества брони**: Приложение должно увеличить количество бронируемых столиков в ресторанах на 20% в течение первого года после запуска
* **Повышение удовлетворённости клиентов**: Приложение должно повысить уровень удовлетворённости клиентов на 17% за счёт удобного и быстрого процесса бронирования

**Заголовок**: Как юзер, я хочу иметь возможность бронировать отель через мобильное приложение, чтобы сэкономить время и упростить процесс.

**Описание**:

* *Кто выполняет действие*: Юзер мобильного приложения
* *Что делает*: Юзер ищет отель по заданным критериям (дата, заезда, местоположение), просматривает доступные варианты, выбирает подходящий отель и бронирует номер
* *Зачем это нужно*: Чтобы сэкономить время на поиске и бронировании отеля через другие источники, а также получить удобный и быстрый доступ к информации о доступных отелях
* *Приемлемый результат*: Юзер сравнил цены конкурентов и успешно забронировал номер в отеле через мобильное приложение, получил подтверждение бронирования и уведомление о предстоящем заезде

**Часть №2:**

Функциональные требования:

* **Создание и редактирование задач**: Юзер должен иметь возможность создавать и изменять задачи, указывая их название, описание, дату и время, а также их приоритет
* **Установка напоминаний**: Юзер должен иметь возможность устанавливать и управлять уведомлениями
* **Статус задач**: Юзер должен иметь возможность отмечать задачи как выполненные. Выполненные задачи должны отмечаться другим цветом или символом
* **Фильтрация и сортировка задач**: Юзер должен иметь возможность отмечать задачи по дате, времени, выполнению и алфавиту
* **Синхронизация данных**: Приложение должно синхронизировать данные между устройствами через облачные хранилища. Синхронизация должна происходить автоматически при подключении к интернету

Нефункциональные требования:

* **Время отклика**: время отклика приложения в основных задачах не должно превышать 2-х секунд
* **Поддержка мобильных платформ**: Приложение должно быть на мобильных устройствах для удобства пользования
* **Безопасность данных**: Система должна обеспечивать безопасность данных юзеров через ключи шифрования

Бизнес-требования:

* **Увеличение числа юзеров**: количество новых пользователей должно увеличиться не менее чем на 30% за 2 квартала
  + *Реферальная программа*: Юзер должен получить бонусы за приглашение и активность нового юзера по его реферальной ссылке
* Повышение узнаваемости через ATL-рекламу: Приложение должно увеличить узнаваемость бренда на 12% в течение первого квартала

**Часть №3:**

Must have:

* Просмотр меню и выбор блюд
* Возможность оплаты через интернет
* Уведомление о статусе заказа

Should have:

* Отслеживание доставки

Cloud have:

* Поддержка кредитных карт и PayPal (карты МИР)

|  |  |
| --- | --- |
| Пользователь  Ресторан | Заказ |
| Имя: Строка  Название: Строка  Email: Строка  Адрес: Строка  Телефон: Строка  Меню: Лист | Номер: Целое число  Дата: Дата  Статус: Строка |
| Зарегистрироваться()  ДобавитьБлюдо()  Войти()  УдалитьБлюдо()  СделатьЗаказ()  ОбновитьМеню()  ПолучитьМеню() | СделатьЗаказ()  Оплатить()  ОтследитьДоставку()  ОтправитьУведомление() |
| ↓↓↓ | ↓↓↓ |
| Блюдо  Оплата | Доставка |
| Название: Строка  Сумма: Десятичное число  Описание: Строка  Цена: Десятичное число  Дата: Дата  Ресторан: Ресторан  Заказ: Заказ | Адрес: Строка  Статус: Строка  Заказ: Заказ  Курьер: Пользователь |

**Часть №4:**

Команда разработчиков может столкнуться с такими проблемами, как:

* Нечёткие требования по разработке ПО
* Сложность интеграции с различными сервисами (на пример, облачные хранилища)
* Недостаток опыта команды: Команда разработчиков может не иметь достаточного опыта в разработке подобного ПО
* Изменение требований заказчика: Заказчик может изменить свои требования по ПО, что приведёт к дополнительным затратам и трудностям
* Некомпетентность членов команды: Некоторые в команде могут быть недостаточно компетентны. К примеру, пишут код не по стандартам, что затруднит его читаемость другими разработиками

Фронтенд:

* **React.js**: это библиотека JavaScript для создания пользовательских интерфейсов
* **Redux**: это библиотека для управления состоянием приложения
* **TypeScript**: это надмножество JavaScript, которое добавляет статическую типизацию

Бэкенд:

* **Node.js**: это среда выполнения JavaScript на сервере
* **Express.js**: это минималистичный и гибкий фреймворк для Node.js, который предоставляет набор функций для веб-приложений
* **MongoDB**: это документо-ориентированная база данных NoSQL

Инструменты и сервисы:

* **Git и GitHub**: Git — это система контроля версий, а GitHub — это платформа для хостинга и совместной работы над проектами
* **Jest и Enzyme**: Jest — это фреймворк для тестирования JavaScript, а Enzyme — это библиотека для тестирования React-компонентов
* **Docker**: это платформа для разработки, доставки и запуска приложений в контейнерах