Практическая работа

«Описание и анализ требований, выбор решений для разработки приложений»

Часть 1: Теория и анализ требований

1. Функциональные требования – это требования, который показывают, как должна работать система, а нефункциональные – какие задачи она выполняет. Пример функциональных требований:

* Система должна рассчитать налог с продаж путем умножения покупной цены на налоговую ставку;
* Программа должна иметь функцию поиска и фильтрации данных для удобного и быстрого доступа к нужной информации.

Пример нефункциональных требований:

* Программа должна быть совместима с операционными системами Windows, Mac OS и Linux;
* Программа должна иметь защиту от несанкционированного доступа и возможность резервного копирования данных.

2. Правильное оформление требований ведет к повышению качества продукта, производительности команды и удовлетворенности заказчика. Правильное оформление требований важно для:

* Четкость и понимание;
* Снижение рисков;
* Упрощение проверки.

3. 2 бизнес-требования для приложения, которое позволяет бронировать столики в ресторанах:

* Интеграция с календарем, который использует пользователь. Пользователь сможет видеть число свободных столиков на различные дни в календаре;
* Отзывы и рейтинги в реальном времени. Это поможет пользователю получить нужную информацию о ресторане от других посетителей.

4. User-Story: Как пользователь, я хочу выбрать номер в отеле и забронировать его в приложении, чтобы быстро подготовиться к отпуску.

Часть 2: Практическая работа с требованиями

1. Функциональные требования:

5 функциональных требований для приложения планировщика задач:

* Возможность сортировки и фильтрации задач по различным критериям, таким как статус выполнения, сроки, приоритеты и другие;
* Возможность автоматического уведомления ответственного лица о приближающемся сроке выполнения задачи;
* Возможность предоставления отчетов о выполнении задач и эффективности работы команды на основе собранных данных;
* Синхронизация данных между различными платформами и устройствами;
* Создание, редактирование и удаление задач.

2. Нефункциональные требования:

3 нефункциональных требований для приложения планировщика задач:

* Время загрузки приложения не должно превышать 5 секунд;
* Интерфейс должен быть понятным интуитивно для всех пользователей;
* Приложение должно корректно работать на различных устройствах и платформах, включая мобильные (Android, IOS).

3. Бизнес-требования:

2 бизнес-требования для приложения планировщика:

* Доход от подписок на премиум-версию может составлять не менее 50,000 долларов, с ростом на 25% каждые 6 месяцев.
* Приложение должно привлечь не менее 10,000 новых пользователей, с темпом роста не менее 15% ежемесячно.

Часть 3: Приоритизация требований и диаграммы

1. Приоритизация с использованием метода MOSCOW:

* Просмотр меню и выбор блюд;
* Поддержка кредитных карт и PayPal;
* Возможность оплаты через интернет;
* Уведомления о статусе заказа;
* Отслеживание доставки.

2. Создание диаграмм:

* Класс «Пользователь» содержит характеристики: логин, пароль и электронную почту;
* Класс «Заказ» содержит информацию о просмотренных и добавленных в корзину товарах, их стоимости и статусе доставки;
* Класс «Ресторан» содержит информацию о предоставляемом пользователям меню, наличии конкретных блюд, направленности заведения.

Часть 4: Анализ рисков и выбор технологического стека

1. Анализ рисков при создании приложения для управления расписанием:

* Приложение может содержать ошибки и недочеты, что может привести к негативному опыту пользователей;
* Если заказчик не предоставляет необходимую обратную связь, команда разработки может столкнуться с затягиванием сроков и неопределенностью в требованиях;
* Могут возникать проблемы с безопасностью личных данных пользователей.

2. Выбор технологического стека для разработки веб-приложения для управления задачами:

* Фронтенд: HTML и CSS;
* Бэкенд: Python
* База данных: PostgreSQL

Выполнили: Юсупова Камола,

Яковлева Анастасия,

Группа 319