

Лабораторная работа №2.

Классификация и приоритезация требований

Тема: Бронирование столиков и формирование предварительного заказа в ресторане

Выполнили студенты группы М3311

Авсюкевич Анастасия

Михалев Никита

## Цель этапа

1. Научиться классифицировать требования и определять их приоритеты

## Задание для выполнения этапа

1. Возьмите набор требований из первой работы.
2. Классифицируйте требования на функциональные, нефункциональные, бизнес-требования и требования пользователей.
3. Проведите приоритизацию требований, используя метод MoSCoW или метод весовых коэффициентов
4. Обоснуйте выбранные приоритеты.

## Функциональные требования:

- 1) В процессе создания заказа столик «замораживается» для других клиентов, с целью избежать двойного бронирования. Если клиент не отправляет данные на текущем шагу в течение 5 минут, то заморозка должна быть снята, а текущий процесс заказа аннулируется.
- 2) При оформлении заказа система должна предоставлять клиенту фильтры для удобного выбора столов и блюд.
- 3) В зависимости от указанного кол-ва гостей система автоматически предлагает подходящие варианты столиков.
- 4) В случае, если пользователь сделал предварительный заказ еды на стол, то должно списываться определенное количество средств за просрок отмены бронирования (перед бронированием с заказом еды просим предоплату, которую будем использовать как гарантию того, что пользователь точно сообщит об отмене заказа заранее).
- 5) Система фиксирует всю стадию изменения состояний отдельной брони/заказа и уведомляет пользователя об этом.
- 6) Автоматическое подтверждение бронирования через **выбранный пользователем** способ уведомления.
- 7) Автоматическая рассылка напоминаний и бронировании за 1-2 часа.
- 8) Визуальное отображение доступных столов на плане ресторана.
- 9) Возможность выбора по фильтрам (например, окна, терраса).

## Нефункциональные требования:

- 1) Корректная блокировка возможности бронирования уже занятых мест.
- 2) Ответ сервера на запрос бронирования не должен превышать 2–3 секунды. (Система должна быть протестирована в штатном и критическом случае, где ответ от сервера будет стабилен и не превышал указанный диапазон времени, которое считается от отправки запроса, до получения ответа, включая в себя обработку и сохранение данных, а также формирование ответа).
- 3) Все персональные данные пользователей должны быть надежно защищены от несанкционированного доступа и утечек, с использованием шифрования данных (передаваемые данные должны быть защищены TLS, не ниже версии 1.2), все пароли должны быть зашифрованы стойкими алгоритмами (например Argon2d, который

используется на backend серверах и позволяет выполнять операции шифрования за достаточно быстрое время).

- 4) Платежные данные должны шифроваться и передаваться через защищенные каналы, а также их хранение на серверах системы запрещено. Все обработки должны осуществляться через сертифицированные платежные провайдеры.
- 5) Система должна поддерживать двухфакторную аутентификацию.
- 6) Система должна быть защищена от всех возможных видов атак (ddos, xss, csrf, sql-инъекции т.п.)
- 7) Система должна обеспечивать высокую доступность (не менее 99% времени).
- 8) Должна быть предусмотрена защита от сбоев (например, резервные копии данных).
- 9) Необходимо контролировать количество одновременных запросов на бронирование столиков, в штатном режиме система обязана поддерживать 200–400 запросов на бронирование одновременно, а в критичных ситуациях необходима возможность горизонтального масштабирования с использованием балансировки нагрузки.
- 10) Поддержка Chrome начиная с версии 127, Firefox начиная с версии 130, Safari начиная с версии 15.0, Yandex версии 24.5.
- 11) Поддержка iOS начиная с версии 15, Android начиная с версии 10.
- 12) Для слабых устройств должна быть поддержка lite версии сайта с минимальным количеством анимаций и более низким расширением изображений.

### **Бизнес-требования:**

- 1) Пользователь должен иметь возможность выбрать дату и время бронирования.
- 2) Система должна автоматически проверять доступность столиков.
- 3) Система должна позволять учитывать предпочтения по ингредиентам (например, аллергии, вегетарианский выбор).
- 4) Система дает возможность уточнить количество гостей и особые пожелания (например, для вегетарианцев или без глютена).
- 5) Возможность изменять или отменять бронирование до определенного времени (например, за 2 часа).
- 6) Система должна уведомлять пользователя о статусе изменения бронирования.
- 7) Система должна отправить напоминание пользователю за 1–2 часа до бронирования.
- 8) Обязателен интуитивно понятный и удобный интерфейс для выполнения бронирования/заказа еды.
- 9) Система должна быть способна обрабатывать одновременные запросы на бронирование без задержек и ошибок.
- 10) У системы должна быть хорошая производительность при любых ситуациях.
- 11) Все персональные данные пользователя должны быть защищены.
- 12) Интерфейс системы должен быть адаптирован под различные типы мобильных устройств, а также корректно отображаться на популярных браузерах.
- 13) Приложение должно поддерживать все версии операционных систем популярных мобильных платформ.

### **Требования пользователей:**

- 1) Пользователь хочет забронировать столик на определенное время в конкретном ресторане.
- 2) Пользователь хочет выбрать конкретный столик (по типу или расположению).

- 3) Пользователь хочет видеть подтверждение бронирования/заказа
- 4) Пользователь хочет иметь возможность выбрать способ уведомления.
- 5) Пользователь хочет иметь возможность выбрать блюда из меню и заказать их заранее.
- 6) Пользователю полезно напомнить о брони за пару часов.
- 7) Пользователь хочет иметь возможность удобно и быстро совершать бронирование/делать заказ еды.
- 8) Пользователь хочет иметь возможность изменить время/дату/место бронирования или вовсе отменить бронирование.

### **Приоритизация требований с помощью метода MoSCoW**

Must-have (**M**) - Should-have (**S**) - Could-have (**C**) - Won't-have (**W**) (this time)

1. В процессе создания заказа столик «замораживается» для других клиентов, с целью избежать двойного бронирования. Если клиент не отправляет данные на текущем шагу в течение 5 минут, то заморозка должна быть снята, а текущий процесс заказа аннулируется.

**M** (обязательное требование для предотвращения двойного бронирования).

2. В процессе оформления заказа система учитывает и отдельно выделяет предпочтения пользователя по ингредиентам.

**M** (важно для персонализации заказа).

3. В зависимости от указанного кол-ва гостей система автоматически предлагает подходящие варианты столиков.

**M** (основная функциональность системы бронирования).

4. В случае, если пользователь сделал предварительный заказ еды на стол, то должно списываться определенное количество средств за просрок отмены бронирования.

**S** (желательно, но не критично для базовой функциональности).

5. Система фиксирует всю стадию изменения состояний отдельной брони/заказа и уведомляет пользователя об этом.

**S** (важно для прозрачности и комфорта пользователя, но не обязательно).

6. Автоматическое подтверждение бронирования через выбранный пользователем способ уведомления.

**M** (основная функциональность подтверждения брони).

7. Автоматическая рассылка напоминаний о бронировании за 1-2 часа.

**S** (удобство, но не обязательно в первой версии).

8. Визуальное отображение доступных столов на плане ресторана.

**S** (приятное улучшение, но не критично).

9. Возможность выбора по фильтрам (например, окна, терраса).

**C** (можно реализовать позже, если есть ресурсы).

10. Корректная блокировка возможности бронирования уже занятых мест.

**M** (необходимо для корректной работы системы).

11. Ответ сервера на запрос бронирования не должен превышать 2–3 секунды.

**M** (ключевая характеристика производительности системы).

12. Шифрование персональных данных.

**M** (безопасность данных обязательно).

13. Платежные данные должны шифроваться и передаваться через защищенные каналы, а их хранение на серверах системы запрещено.

**M** (безопасность данных обязательно для финансовых операций).

14. Система должна поддерживать двухфакторную аутентификацию.

**S** (важно для безопасности, но не критично в первой версии).

15. Система должна быть защищена от всех возможных видов атак (DDoS, XSS, CSRF, SQL-инъекции и т.п.).

**M** (обязательное требование для безопасности системы).

16. Система должна обеспечивать высокую доступность (не менее 99% времени).

**M** (критично для надежности работы системы).

17. Должна быть предусмотрена защита от сбоев (например, резервные копии данных).

**M** (необходимо для отказоустойчивости).

18. Необходимо контролировать количество одновременных запросов на бронирование столиков, в штатном режиме система обязана поддерживать 200–400 запросов на бронирование одновременно.

**M** (необходимо для работы с большим числом пользователей).

19. Поддержка Chrome начиная с версии 127, Firefox начиная с версии 130, Safari начиная с версии 15.0, Yandex версии 24.5.

**S** (желательно, но можно начать с более ограниченной совместимости).

20. Поддержка iOS начиная с версии 15, Android начиная с версии 10.

**S** (желательно для широкой аудитории).

21. Для слабых устройств должна быть поддержка lite версии сайта с минимальным количеством анимаций и более низким расширением изображений.

**C** (можно отложить до более поздних этапов).

22. Пользователь должен иметь возможность выбрать дату и время бронирования.

**M** (основная функциональность).

23. Система должна автоматически проверять доступность столиков.

**M** (основная функциональность).

24. Система должна позволять учитывать предпочтения по ингредиентам.

**M** (персонализация).

25. Система дает возможность уточнить количество гостей и особые пожелания.

**M** (важно для корректного выполнения брони).

26. Возможность изменять или отменять бронирование до определенного времени.

**M** (основная функциональность).

27. Система должна уведомлять пользователя о статусе изменения бронирования.

**S** (удобство, но не критично).

28. Система должна отправить напоминание пользователю за 1–2 часа до бронирования.

**S** (удобно, но не критично).

29. Обязателен интуитивно понятный и удобный интерфейс для выполнения бронирования/заказа еды.

**M** (основная функция удобства для пользователя).

30. Система должна быть способна обрабатывать одновременные запросы на бронирование без задержек и ошибок.

**M** (необходимо для стабильной работы).

31. У системы должна быть хорошая производительность при любых ситуациях.

**M** (критично для работы).

32. Все персональные данные пользователя должны быть защищены.

**M** (необходимо для безопасности).

33. Интерфейс системы должен быть адаптирован под различные типы мобильных устройств, а также корректно отображаться на популярных браузерах.

**М** (необходимо для удобства пользователей).

34. Приложение должно поддерживать все версии операционных систем популярных мобильных платформ.

**S** (желательно, но можно начать с более ограниченной поддержки).

35. Пользователь хочет забронировать столик на определенное время в конкретном ресторане.

**М** (основная функция).

36. Пользователь хочет выбрать конкретный столик (по типу или расположению) из доступных для бронирования в текущем ресторане.

**М** (приятное дополнение, но не критично).

37. Пользователь хочет видеть подтверждение бронирования/заказа.

**М** (основная функция).

38. Пользователь хочет иметь возможность выбрать способ уведомления.

**S** (удобно, но не критично).

39. Пользователь хочет иметь возможность выбрать блюда из меню и заказать их заранее.

**М** (основная функциональность).

40. Пользователю полезно напомнить о брони за пару часов.

**S** (удобно, но не критично).

41. Пользователь хочет иметь возможность удобно и быстро совершать бронирование/делать заказ еды.

**М** (основная цель).

42. Пользователь хочет иметь возможность изменить время/дату/место бронирования или вовсе отменить бронирование.

**М** (основная функциональность).