

Лабораторная работа №3.

Трассировка требований

Тема: Бронирование столиков и формирование предварительного заказа в ресторане

Выполнили студенты группы М3311

Авсюкевич Анастасия

Михалев Никита

Цель:

Освоить методы трассировки требований и научиться связывать их с бизнес-целями и тестовыми сценариями

Матрица Трассировки:

*про приоритет:

H (high) - высокий приоритет

M (medium) - средний приоритет

L (low) - низкий приоритет

Требования	Этап разработки	Компонент системы	Тестовый сценарий	Бизнес-цели	Приоритет
В процессе создания заказа столик «замораживается» для других клиентов, с целью избежать двойного бронирования.	Разработка бизнес-логики заказа	Модуль блокировки столика, бронирование	Попытаться забронировать столик одновременно для двух клиентов. Проверить, что система блокирует второй запрос.	Система должна автоматически проверять доступность столиков.	H
Если клиент не отправляет данные на текущем шагу в течение 5 минут, то заморозка должна быть снята.	Реализация таймеров и логики заказа	Модуль блокировки столика, таймер ожидания	После 5 минут бездействия отменить блокировку стола и проверить, что заказ отменен.	Возможность изменять или отменять бронирование до определенного времени	H
При оформлении заказа система должна предоставлять клиенту фильтры для удобного выбора столов и блюд.	Разработка интерфейса и фильтров	Интерфейс пользователя, фильтры столов и блюд	Проверить, что фильтры для выбора столиков и блюд доступны и работают корректно.	Пользователь должен иметь возможность выбрать дату и время бронирования. Система дает возможность уточнить количество гостей и особые пожелания	M

В зависимости от указанного кол-ва гостей система автоматически предлагает подходящие варианты столиков.	Разработка логики рекомендаций	Рекомендательная система	Указать количество гостей и проверить, что система предлагает подходящие варианты столиков.	Система дает возможность уточнить количество гостей и особые пожелания	Н
Если пользователь сделал предварительный заказ еды на стол, списывается сумма за просроченную отмену.	Реализация механизма предоплаты	Механизм бронирования с предоплатой	Заказать стол с едой, отменить заказ позже установленного времени, проверить списание средств за просрочку.	Возможность изменять или отменять бронирование до определенного времени (например, за 2 часа).	М
Система фиксирует всю стадию изменения состояний отдельной брони/заказа и уведомляет пользователя.	Разработка логики изменения состояния	Логирование изменений, уведомления	Изменить состояние бронирования (например, отменить), проверить, что уведомление отправлено и состояние обновлено.	<p>Возможность изменять или отменять бронирование до определенного времени (например, за 2 часа).</p> <p>Система должна уведомлять пользователя о статусе изменения бронирования.</p> <p>Система должна отправить напоминание пользователю за 1–2 часа до бронирования.</p>	М
Автоматическое подтверждение бронирования через выбранный пользователем способ уведомления.	Разработка уведомлений	Модуль уведомлений	Завершить процесс бронирования и проверить, что подтверждение отправлено через выбранный канал уведомления.	Система должна уведомлять пользователя о статусе изменения бронирования.	Н

Автоматическая рассылка напоминаний и бронирования за 1-2 часа.	Реализация уведомлений	Модуль напоминаний	Проверить, что за 1-2 часа до бронирования отправляется напоминание о бронировании.	Система должна отправить напоминание пользователю за 1–2 часа до бронирования.	М
Визуальное отображение доступных столов на плане ресторана.	Разработка интерфейса отображения стола	Карта ресторана, визуализация доступных столов	Проверить, что карта ресторана отображает доступные столы в реальном времени.	Обязателен интуитивно понятный и удобный интерфейс для выполнения бронирования/заказа еды.	М
Возможность выбора по фильтрам (например, окна, терраса).	Разработка фильтров и интерфейса выбора	Фильтры столиков и блюд	Применить фильтры по типу столика и проверить, что результаты соответствуют выбранным фильтрам.	Система дает возможность уточнить количество гостей и особые пожелания	М/L
Корректная блокировка возможности бронирования уже занятых мест.	Проектирование логики блокировки	Модуль блокировок столиков, проверка состояния	Попытаться забронировать уже занятый столик и проверить, что система не позволяет сделать бронирование.	Система должна автоматически проверять доступность столиков.	Н
Ответ сервера на запрос бронирования не должен превышать 2–3 секунды.	Оптимизация производительности	API сервера, обработка запросов	Сделать запрос на бронирование и измерить время ответа сервера (не более 3 секунд).	У системы должна быть хорошая производительность при любых ситуациях.	Н

Все персональные данные пользователей должны быть защищены от несанкционированного доступа и утечек.	Разработка безопасности данных	База данных, шифрование	Проверить, что персональные данные шифруются и передаются по защищенному каналу (TLS).	Все персональные данные пользователя должны быть защищены.	Н
Платежные данные должны шифроваться и передаваться через защищенные каналы.	Интеграция с платежными провайдерами	Платежная система	Проверить, что платежные данные передаются через защищенный канал (TLS) и не хранятся на сервере.	Все персональные данные пользователя должны быть защищены.	Н
Система должна поддерживать двухфакторную аутентификацию.	Разработка системы аутентификации	Модуль аутентификации	Выполнить вход с двухфакторной аутентификацией и проверить, что все этапы аутентификации пройдены корректно.	Все персональные данные пользователя должны быть защищены.	М
Система должна быть защищена от всех видов атак (ddos, xss, csrf, sql-инъекции и т.п.).	Разработка безопасности системы	Защита от атак, API безопасность	Выполнить тесты на уязвимости (SQL-инъекции, XSS и т.д.) и убедиться в отсутствии уязвимостей.	Все персональные данные пользователя должны быть защищены.	Н
Система должна обеспечивать высокую доступность (не менее 99%).	Разработка системы резервирования и масштабирования	Система резервного копирования, балансировка нагрузки	Провести стресс-тестирование и проверить, что система работает стабильно при 99% доступности.	У системы должна быть хорошая производительность при любых ситуациях.	Н
Поддержка 200–400 одновременных запросов на бронирование.	Оптимизация производительности и масштабирование	Балансировка нагрузки, серверы	Симулировать 400 одновременных запросов на бронирование и проверить, что система	Система должна быть способна обрабатывать одновременные запросы на бронирование без задержек и ошибок.	Н

			обрабатывает их без сбоев.		
Поддержка различных версий браузеров и мобильных ОС.	Разработка интерфейса для разных платформ	UI компоненты для разных платформ	Открыть сайт в разных браузерах и мобильных устройствах, проверить корректность отображения.	Приложение должно поддерживать все версии операционных систем популярных мобильных платформ.	М

Анализ полноты покрытия бизнес-целей и функциональности

В требованиях и тестовых сценариях хорошо покрыты основные бизнес-цели и функциональность системы с учетом всех ключевых аспектов безопасности, производительности и удобства использования. Например:

- *Бронирование столиков и управление предпочтениями* — Все основные функциональные требования, такие как выбор столика, проверка доступности, фильтры по предпочтениям, уведомления и напоминания, покрыты в тестовых сценариях.
- *Платежи и безопасность* — Система защиты данных, двухфакторная аутентификация и работа с платежными данными также хорошо покрыты тестами. Протестированы как общие аспекты безопасности, так и возможность интеграции с платежными провайдерами.
- *Оптимизация производительности и доступность* — Включены тесты на высокую доступность, защиту от атак, а также тесты на стрессовую нагрузку, что обеспечивает хорошие показатели производительности системы при высоком объеме запросов.
- *Мобильная версия и поддержка устройств* — Тестирование на мобильных устройствах и старых версиях браузеров также учтено, что соответствует нефункциональным требованиям.