ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

Домашняя работа №2 по дисциплине “Проектирование архитектуры программных систем”

Анализ задачи
Архитектурно-значимые атрибуты качества системы
Критерии оценки жизнеспособности архитектуры и контроля качества

Выполнили студенты группы БПИ223
Абдуллаев А.Ш.
Жалилов А.
Курманова А.

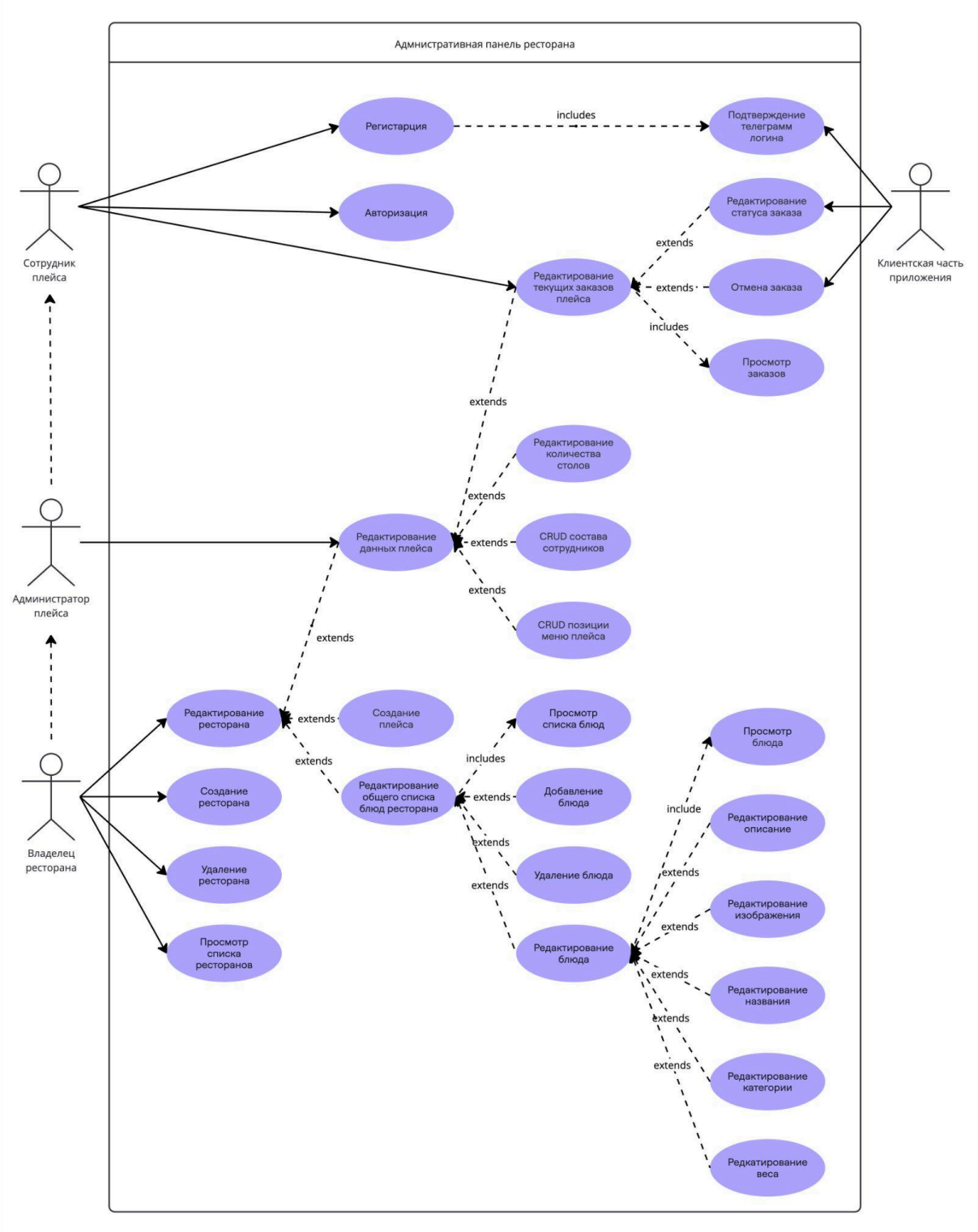
Москва 2025

Содержание

Ключевые функциональные требования

Административная панель

Диаграмма прецедентов

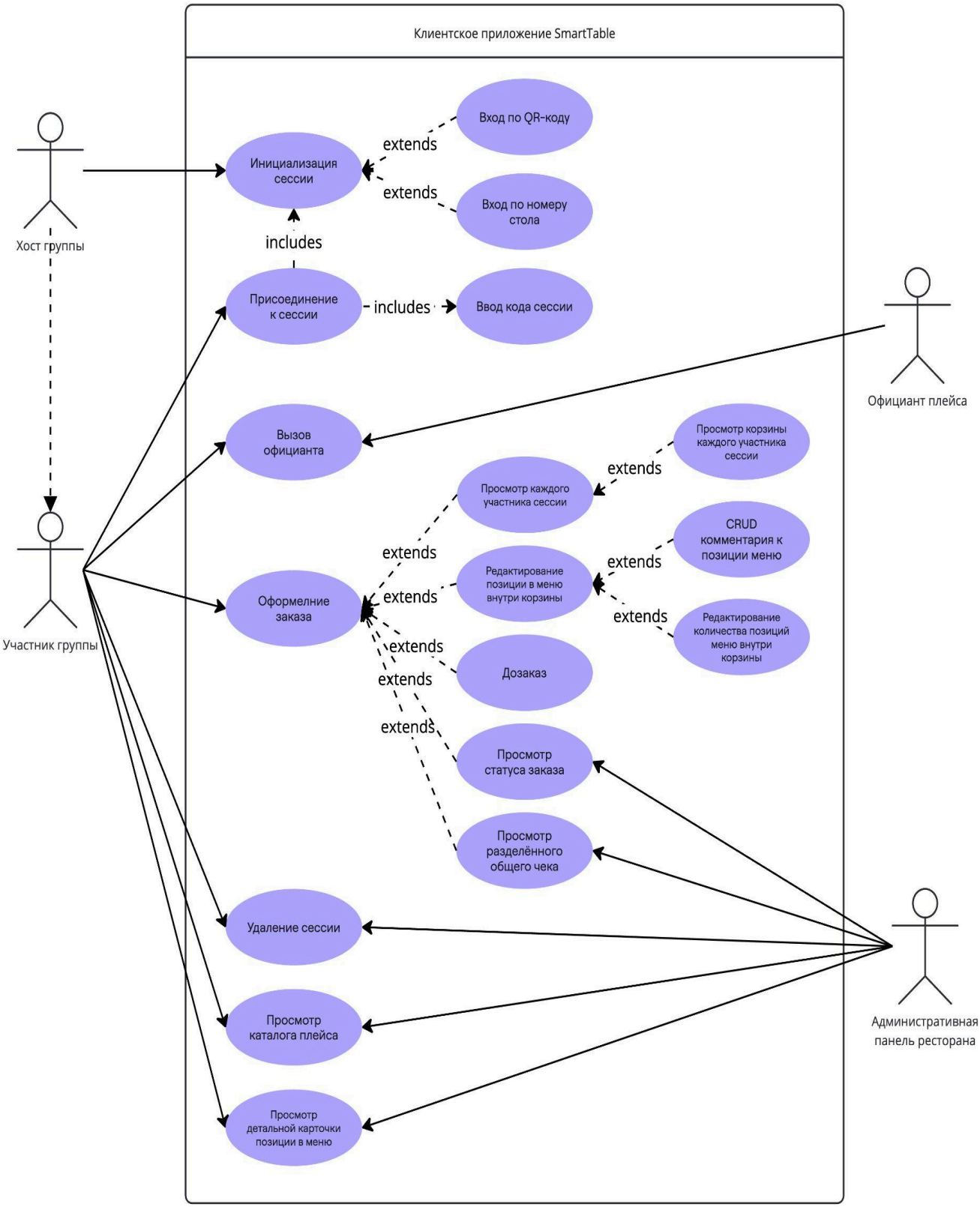


Список функциональных требований:

1. Регистрация\Авторизация пользователя
2. Подтверждение телеграмм-логина
4. Отображение списка ресторанов пользователя
5. Создание ресторана
7. Удаление ресторана
8. Редактирование ресторана
    - 8.1. Отображение списка плейсов ресторана
    - 8.2. Создание плейса
    - 8.3. Удаление плейса
    - 8.4. Редактирование плейса
        - 8.4.1 CRUD позиции меню плейса
        - 8.4.2 CRUD состава сотрудников
        - 8.4.3 Редактирование количества столов
        - 8.4.4 Просмотр текущих заказов плейса
        - 8.4.5 Редактирование текущих заказов плейса
            - 8.4.5.1 Редактирование статуса заказа
            - 8.4.5.2 Отмена заказа
    - 8.5 Просмотр общего списка блюд ресторана
    - 8.6 Редактирование общего списка блюд ресторана
        - 8.6.1 Просмотр блюда
        - 8.6.2 Добавление блюда
        - 8.6.3 Удаление блюда
        - 8.6.4 Редактирование блюда
            - 8.6.4.1 Редактирование описания
            - 8.6.4.2 Редактирование изображения
            - 8.6.4.3 Редактирование названия
            - 8.6.4.4 Редактирование категории
            - 8.6.4.5 Редактирование веса

Клиентское приложение

Диаграмма прецедентов



Список функциональных требований:

1. Инициализация сессии
    - 1.1. Инициализация по QR коду
    - 1.2. Инициализация по номеру стола
2. Присоединение к сессии
    - 2.1. Ввод кода сессии
3. Вызов официанта
4. Оформление заказа
    - 4.1. Просмотр каждого участника сессии
        - 4.1.1. Просмотр корзины каждого участника
    - 4.2. Редактирование позиции в меню внутри корзины
        - 4.2.1. CRUD комментария к позиции меню
        - 4.2.2. Редактирование количества позиций меню внутри корзины
    - 4.3. Дозаказ
    - 4.4. Просмотр статуса заказа
    - 4.5. Просмотр разделенного общего чека
5. Удаление сессии
6. Просмотр каталога плейса
7. Просмотр детальной карточки позиции в меню

Нефункциональные требования и атрибуты качества

1. Согласованность

Требования:

- Мгновенное обновление статусов заказов у всех пользователей.
- Гарантированная синхронизация данных между клиентской и административной частями.
- Защита от одновременного редактирования данных разными пользователями.

SmartTable — это система, работающая с заказами в реальном времени. Все пользователи (клиенты, официанты, администраторы) должны видеть актуальные данные одновременно. Нам важно, чтобы при изменении заказа или статуса информации она обновлялась у всех пользователей. Несогласованные данные приводят к неверной подаче блюд, путанице при расчёте счета, недовольству клиентов и финансовым потерям ресторанов.

Обратные зависимости:

- Увеличение нагрузки на сервер: чрезмерная частота обновлений может замедлить работу сервиса, постоянная синхронизация данных (особенно при большом количестве пользователей) требует дополнительной обработки.

- Возможные конфликты при одновременном редактировании: если два пользователя редактируют корзину одновременно, система должна правильно обработать изменения.

Метрики:

- Время обновления статусов заказов

  Допустимое значение: ≤ 200 мс.

- Частота расхождений данных между клиентской и административной частями

  Допустимое значение: 0 (или ≤ 0.1% от общего числа заказов).

- Конфликты при редактировании заказа

  Допустимое значение: ≤ 1% заказов.

Методики оценки:

- Логирование событий в базе данных → проверка задержки между изменением заказа и его отображением у всех пользователей.
- Тестирование конкурентных запросов → имитация одновременного редактирования заказа разными пользователями.
- A/B-тестирование → запуск системы с разными механизмами синхронизации.

2. Производительность

Требования:

- Время отклика API не более 500 мс.
- Операции с корзиной и заказами должны выполняться за 300 мс.
- Загрузка меню не более 1 секунды.

В заведениях скорость обслуживания напрямую влияет на прибыль. Если клиенту приходится ждать 10 секунд для загрузки меню или оформления заказа, это снижает удобство использования системы. Также критически важно, чтобы клиенты и сотрудники не ждали обновления информации и система могла обслуживать тысячи пользователей одновременно, не перегружая сервер.

Обратная зависимость:

- Конфликт с надежностью и согласованностью: чем быстрее система обрабатывает запросы, тем выше вероятность ошибок при обработке данных.

Метрики:

- Время отклика API

  Допустимое значение: ≤ 500 мс.

- Время загрузки меню

  Допустимое значение: ≤ 1 сек.

- Время обработки заказа

  Допустимое значение: ≤ 300 мс.

- Максимальное количество пользователей одновременно

  Допустимое значение: ≥ 10 000.

Методики оценки:

- Нагрузочное тестирование → проверка работы сервера при массовых запросах.
- Стресс-тестирование → эмуляция пиковых нагрузок.
- Профилирование кода → оптимизация баз данных и API-запросов для снижения времени обработки.

3. Удобство использования

Требования:

- Интерфейс должен быть простым и интуитивным.
- Минимальное число кликов для оформления заказа.
- Гибкая работа с корзиной (удаление, изменение количества, комментарии).

Клиенты ожидают интуитивно понятного интерфейса – сложные действия приводят к отказу от использования сервиса. Сотрудники также должны быстро ориентироваться в системе. Сложный интерфейс увеличивает время обработки заказов и снижает удовлетворенность пользователей.

Обратная зависимость:

- Ограничение функциональности: упрощённый интерфейс может привести к ограничению возможностей системы.

Метрики:

- Среднее время оформления заказа

Допустимое значение: ≤ 30 сек.

- Число кликов до оформления заказа

  Допустимое значение: ≤ 3.

- Процент пользователей, успешно завершивших заказ

  Допустимое значение: ≥ 95%.

- Среднее время обучения сотрудников

  Допустимое значение: ≤ 10 мин.

Методики оценки:

- UX-тестирование → проведение тестов с реальными пользователями, отслеживание их действий.
- Запись сессий пользователей → анализ реального поведения клиентов в приложении.
- Опросы и фидбек пользователей → сбор обратной связи об удобстве интерфейса.

4. Надежность

Требования:

- Данные заказов и платежей должны быть сохранены даже при сбоях.
- В случае ошибки данные автоматически восстанавливаются.
- Логирование всех изменений в заказах.

Если данные о заказах теряются или дублируются, это создаст хаос в ресторане, вызовет недовольство клиентов и сотрудников, а также финансовые потери.

Обратная зависимость:

- Сложность реализации и увеличение ресурсов: поддержка надежности требует резервного копирования и транзакционной обработки данных.

Метрики:

- Процент потерянных заказов

  Допустимое значение: 0%.

- Частота дублирования заказов

  Допустимое значение: ≤ 0.01%.

- Среднее время восстановления данных

  Допустимое значение: ≤ 1 мин.

- Процент завершенных транзакций

  Допустимое значение: ≥ 99.99%.

Методики оценки:

- Тестирование отказов → эмуляция сбоев базы данных, серверов и сети.
- Мониторинг логов ошибок → анализ неудачных операций с заказами.
- Имитация отказов платежных систем → проверка поведения системы при неудачных транзакциях.

5. Отказоустойчивость

Требования:

- В случае сбоя сервер автоматически переключается на резервный.
- Данные автоматически сохраняются и восстанавливаются.
- Поддержка офлайн-режима с последующей синхронизацией.

Рестораны работают в режиме реального времени, и простои системы недопустимы. Без отказоустойчивости пользователи потеряют доступ к сервису. Если сервер выходит из строя, система должна автоматически переключаться на резервный.

Обратная зависимость:

- Рост затрат на инфраструктуру: поддержка отказоустойчивости требует дублирования серверов и резервного хранения данных.

Метрики:

- Время автоматического переключения на резервный сервер

  Допустимое значение: ≤ 5 сек.

- Процент доступности системы (Uptime)

  Допустимое значение: $\geq 99.99\%$.

- Время восстановления после сбоя

  Допустимое значение: ≤ 1 мин.

Методики оценки:

- Тестирование отказоустойчивости → симуляция отказа серверов и оценка времени их восстановления.
- Мониторинг uptime → использование сервисов для измерения доступности системы.
- Тестирование офлайн-режима → проверка работы приложения без подключения к интернету.

6. Масштабируемость

Требования:

- Гибкое добавление новых ресторанов и плейсов без перегрузки системы.
- Балансировка нагрузки между серверами для избежания задержек в работе.
- Репликация базы данных для распределения запросов.

SmartTable рассчитан на работу с большим количеством ресторанов и пользователей. Нам важно, чтобы система сохраняла стабильную скорость работы при увеличении нагрузки и расширении базы клиентов. Без масштабируемости рост системы приведет к ухудшению производительности и сбоям в обслуживании.

Обратная зависимость:

- Дополнительная сложность архитектуры: требуется балансировка нагрузки между серверами.

Метрики:

- Максимальное количество пользователей, работающих одновременно

  Допустимое значение: ≥ 10000.

- Количество ресторанов, работающих в системе

  Допустимое значение: ≥ 1000.

- Время масштабирования серверов

  Допустимое значение: ≤ 1мин.

- Балансировка нагрузки между серверами

  Допустимое значение: ≥ 95%.

Методики оценки:

- Тестирование горизонтального масштабирования → проверка автоматического добавления серверов.
- Мониторинг нагрузки → использование инструментов.
- Эмуляция пиковых нагрузок → проверка поведения системы при резком росте трафика.

Архитектурно-значимые требования

Нефункциональные требования

- Согласованность

  Обоснование: критично для обеспечения целостности данных в многопользовательском режиме. Влияет на выбор базы данных и механизмов транзакций.

- Производительность

  Обоснование: высокая нагрузка на сервер требует оптимизированной архитектуры, распределения нагрузки и кэширования данных.

Функциональные требования

Административная панель:

- Регистрация\Авторизация пользователя в админке

  Обоснование: данное требование критично, так как администраторы управляют системой. Требуется продуманная схема аутентификации и авторизации, обеспечивающая безопасность доступа.

- Редактирование текущих заказов плейса

  Обоснование: позволяет администраторам вносить изменения в активные заказы, что требует гибкой архитектуры базы данных и механизмов обновления информации в реальном времени.

Клиентское приложение:

- Присоединение к сессии

  Обоснование: важная функциональность, определяющая, как пользователи взаимодействуют с системой. Требуется надежный механизм управления сессиями.

- Удаление сессии

  Обоснование: необходимо для корректного завершения работы и освобождения ресурсов. Влияет на логику обработки данных и взаимодействие с сервером.

- Просмотр корзины каждого участника

  Обоснование: функционал требует хранения и отображения данных в реальном времени, что влияет на выбор архитектурного подхода.

- Дозаказ
  Обоснование: динамическое обновление заказов требует эффективной работы с базой данных и механизма обработки запросов в режиме реального времени.