Министерство науки и высшего образования Республики Казахстан

Северо-Казахстанский университет им. М. Козыбаева

Дуанбай Бақтияр Болатұлы

Куминов Илья Игоревич

На тему «Разработка системы поддержки принятия решения для управления рестораном»

ДИПЛОМНЫЙ ПРОЕКТ

6B06102 – Информационные системы

Петропавловск, 2025 г.

Министерство науки и высшего образования Республики Казахстан

Северо-Казахстанский университет им. М. Козыбаева

|  |  |
| --- | --- |
| «Допущен к защите»  Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_ И.Г. Курмашев |  |

**ДИПЛОМНЫЙ ПРОЕКТ**

На тему: «Разработка системы поддержки принятия решения для управления рестораном»

ИКТ1ИС.10619059.19ДП

|  |  |
| --- | --- |
| Выполнили студенты  группы ИС-21 | Дуанбай Б.Б.  Куминов И.И. |
| Научный руководитель  кандидат технических наук,  доцент кафедры «Информационно-коммуникационные технологии» | Кухаренко Е.В. |
|  |  |

Петропавловск, 2025

Содержание

[СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ 3](#_Toc197980778)

[АННОТАЦИЯ 4](#_Toc197980779)

[ВВЕДЕНИЕ 5](#_Toc197980780)

[1 Анализ объекта исследования и постановка задачи проекта 7](#_Toc197980781)

[1.1 Процесс исследования и подготовки к разработке 7](#_Toc197980782)

[1.2 Автоматизация основных бизнес-процессов 8](#_Toc197980783)

[1.3 Интерфейсные решения проекта 9](#_Toc197980784)

[1.4 Функциональные и технические требования 12](#_Toc197980785)

[1.5 Технологии и инструментальные средства разработки 16](#_Toc197980786)

[2 Проектирование, разработка и внедрение системы 18](#_Toc197980787)

[2.1 Архитектура системы 18](#_Toc197980788)

[2.2 База данных системы 21](#_Toc197980789)

[2.3 Взаимодействие пользователей с системой 24](#_Toc197980790)

[2.4 Программная реализация 30](#_Toc197980791)

[2.5 Функции и алгоритмы системы 48](#_Toc197980792)

[ЗАКЛЮЧЕНИЕ 58](#_Toc197980793)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 60](#_Toc197980794)

# СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

В пояснительной записке к дипломному проекту используются следующие обозначения и сокращения:

СППР – Система поддержки принятия решений

API – Application Programming Interface (интерфейс программирования приложений)

UI – User Interface (пользовательский интерфейс)

UX – User Experience (пользовательский опыт)

JWT – JSON Web Token (токен веб-аутентификации)

CSV – Comma-Separated Values (значения, разделённые запятыми)

PDF – Portable Document Format (портативный формат документов)

SQL – Structured Query Language (язык структурированных запросов)

REST – Representational State Transfer (архитектурный стиль веб-сервисов)

CRUD – Create, Read, Update, Delete (создание, чтение, обновление, удаление)

PWA – Progressive Web App (прогрессивное веб-приложение)

ID – Identifier (идентификатор)

JSON – JavaScript Object Notation (формат обмена данными)

HTTP – HyperText Transfer Protocol (протокол передачи гипертекста)

CSS – Cascading Style Sheets (каскадные таблицы стилей)

SPA – Single Page Application (одностраничное приложение)

XSS – Cross-Site Scripting (межсайтовый скриптинг)

CSRF – Cross-Site Request Forgery (подделка межсайтовых запросов)

IDE – Integrated Development Environment (интегрированная среда разработки)

АҢДАТПА

Дипломдық жобаның тақырыбы – «Мейрамхананы басқаруға арналған шешім қабылдауды қолдау жүйесін әзірлеу».

Жүйе тапсырыстарды рәсімдеу, үстелдерді брондау, мәзірді басқару, персоналдың жұмысын ұйымдастыру және талдау жүргізу сияқты негізгі бизнес-процестерді қамтиды. Жоба аясында клиенттерге, даяшыларға және әкімшілерге арналған интерфейстер React және FastAPI технологиялары негізінде әзірленді. Жүйеге жиналған деректер негізінде мәзірді оңтайландыру, залдың жүктемесін болжау және қаржылық жоспарлауды жақсартуға мүмкіндік беретін талдау және болжау модульдері енгізілді. Жоба мейрамхана бизнесін цифрландыруға, операциялық шығындарды азайтуға және клиенттерге қызмет көрсету сапасын арттыруға бағытталған.

# АННОТАЦИЯ

Темой дипломного проекта является «Разработка системы поддержки принятия решения для управления рестораном».

Система охватывает основные бизнес-процессы: оформление заказов, бронирование столов, управление меню, работу персонала и проведение аналитики. В рамках проекта были реализованы интерфейсы для клиента, официанта и администратора, разработанные с использованием React и FastAPI. В систему также интегрированы модули аналитики и прогнозирования, позволяющие на основе накопленных данных оптимизировать меню, планировать загрузку зала и улучшать финансовое планирование. Проект направлен на цифровизацию ресторанного бизнеса, снижение операционных издержек и повышение качества обслуживания клиентов.

ANNOTATION

The topic of this graduation project is "Development of a Decision Support System for Restaurant Management".

The system covers key business processes such as order processing, table reservation, menu management, staff operations, and analytics. The project includes interfaces for clients, waiters, and administrators, developed using React and FastAPI. The system also integrates analytics and forecasting modules that use accumulated data to optimize the menu, plan table load, and improve financial planning. The project is aimed at digitalizing restaurant operations, reducing operational costs, and improving customer service quality.

# ВВЕДЕНИЕ

В условиях стремительного развития цифровых технологий автоматизация процессов в ресторанной индустрии становится неотъемлемой частью успешного управления и повышения качества обслуживания. Современные клиенты предъявляют всё более высокие требования к скорости и удобству получения услуг, а владельцы ресторанов — к эффективности использования ресурсов и прозрачности бизнес-процессов. Актуальность разработки системы поддержки принятия решений (СППР) для управления рестораном обусловлена необходимостью цифровой трансформации ресторанного бизнеса и повышением конкурентоспособности на рынке.

Рынок общественного питания переживает активные изменения. Возрастает потребность в аналитике, прогнозировании спроса, автоматизированной обработке заказов, управлении меню, бронированиями и персоналом. Разработка СППР предоставляет инструменты для централизованного контроля, анализа клиентской базы, формирования финансовой отчетности, построения прогностических моделей и интерактивных дашбордов.

Цель дипломного проекта заключается в разработке и внедрении системы поддержки принятия решений для управления рестораном, которая обеспечит автоматизацию ключевых бизнес-процессов, повысит оперативность управленческих решений и улучшит пользовательский опыт как сотрудников, так и клиентов.

Для достижения цели поставлены следующие задачи:

- провести анализ текущих бизнес-процессов ресторана;

- сформулировать функциональные требования к СППР;

- выбрать подходящие технологии и инструменты разработки;

- разработать архитектуру системы, базу данных и пользовательские интерфейсы;

- реализовать модули бронирования, заказов, меню, аналитики и прогнозирования;

- внедрить и протестировать алгоритмы финансовой аналитики и анализа клиентской базы;

- подготовить документацию и пользовательские инструкции по использованию системы.

Объект исследования — деятельность ресторана, включающая обслуживание клиентов, бронирование столиков, оформление и обработку заказов, работу персонала и анализ операционных данных.

Предмет исследования — информационные технологии управления рестораном, методы сбора и обработки данных, алгоритмы прогнозирования и интерфейсы взаимодействия с пользователями.

Научная новизна проекта заключается в создании интегрированной СППР, включающей средства визуализации данных, прогностические алгоритмы и автоматизированную обработку операций, реализованной на основе современных веб-технологий с фокусом на гибкость и расширяемость.

Практическая значимость проекта заключается в возможности использования СППР реальными ресторанами для улучшения управления, снижения издержек, повышения точности планирования и качества обслуживания. Система позволяет ресторанам адаптироваться к динамичным условиям рынка и использовать данные как основу для принятия решений.

Структура и объем работы определяются тематикой, целью и задачами дипломного проекта. Работа состоит из введения, двух разделов, заключения и списка литературы. В первой главе описывается предметная область, проводится анализ и формализация бизнес-процессов, определяются требования к системе. Вторая глава содержит детальное описание разработки СППР: архитектура, интерфейсы, алгоритмы, программная реализация и визуализация. В заключении подводятся итоги работы и формулируются основные выводы.

Работа выполнена в соответствии с методическими указаниями по написанию дипломных проектов бакалавриата направления "Информационно-коммуникационные технологии".

# 1 Анализ объекта исследования и постановка задачи проекта

## 1.1 Процесс исследования и подготовки к разработке

На первоначальном этапе проектирования и создания программного обеспечения для системы поддержки принятия решений (СППР) в ресторанной индустрии выявлена насущная проблема – низкий уровень эффективности и автоматизации процессов, связанных с организацией обслуживания клиентов. Объектом исследования стал ресторан среднего уровня, расположенный в городской среде, где высока конкуренция и значима клиентская лояльность. Проведённое предварительное исследование подтвердило, что значительное количество операций осуществляется вручную, что приводит к повышенной нагрузке на персонал, увеличению времени обслуживания клиентов и, как следствие, к снижению общего качества сервиса.

Подготовительный этап включал в себя проведение наблюдений за текущими операциями ресторана, анкетирование работников различных категорий (включая официантов, администраторов, поваров) и сбор отзывов от посетителей. Также выполнен конкурентный анализ уже существующих решений в данной области: R-Keeper, Poster, iiko и другие. Анализ показал, что перечисленные системы часто перегружены ненужным функционалом, требуют значительных финансовых затрат и длительного периода обучения персонала, что делает их неудобными для применения в заведениях малого и среднего формата. Кроме того, в большинстве из них отсутствует встроенный модуль поддержки принятия решений, ориентированный на анализ и прогнозирование ключевых бизнес-показателей.[22]

В результате этих исследований сформулирована основная цель проекта — разработать интуитивно понятную, простую в использовании, но при этом функционально насыщенную систему, способную охватывать основные процессы ресторанного бизнеса: приём и обработку заказов, онлайн-бронирование столиков, управление меню, анализ выручки и поведения клиентов, а также генерацию подсказок и рекомендаций для управления ассортиментом.

SWOT-анализ позволил детально выявить слабые и сильные стороны текущей ситуации. В числе сильных сторон — высокая мотивация персонала, возможность внедрения технологий без капитального ремонта инфраструктуры, наличие постоянной клиентской базы. В числе слабых — высокая зависимость от устного взаимодействия между сотрудниками, ограниченность в анализе данных, отсутствие автоматизации повторяющихся задач. Возможностями стали цифровизация и переход на облачные решения, а угрозами — высокая конкуренция и техническая неграмотность части персонала. Этот анализ стал основой для выработки технического задания и определения ключевых точек развития проекта.[4][25]

Кроме того, в процессе подготовки проводился сбор данных по типичным поведенческим паттернам посетителей: когда чаще всего совершаются заказы, какие блюда являются самыми востребованными в разные дни недели, как влияет погодный фактор на бронирование столиков. Также проанализированы маркетинговые кампании и сезонные акции, что позволило лучше понять реакцию клиентов и их предпочтения. На основе полученных результатов составлен список требований к модулю аналитики, а также к логике формирования рекомендаций.[21][23]

## 1.2 Автоматизация основных бизнес-процессов

Система разрабатывается с ориентацией на максимальное покрытие и оптимизацию ключевых бизнес-процессов, типичных для ресторанной сферы. Выделено три главных направления: автоматизация приёма заказов, управление бронированиями и реализация аналитического модуля для поддержки принятия управленческих решений.[27]

Приём заказов: Процесс заказа организован так, чтобы клиент мог без труда ознакомиться с меню, выбрать интересующие позиции, добавить их в корзину и оформить заказ. Особое внимание уделено пользовательскому опыту: реализована фильтрация по категориям, быстрый поиск, отображение цен в тенге, визуальное оформление карточек блюд. Заказ автоматически попадает в систему, где официанты и повара могут видеть его в реальном времени, отслеживать этапы приготовления и изменять статус.

Дополнительно реализованы уведомления для клиентов о смене статуса заказа, возможность оставлять комментарии к заказу (например, «без лука», «не остро»), а также отправка уведомлений официантам на планшет или ПК. Всё это существенно снижает вероятность недоразумений и ошибок, связанных с человеческим фактором. В будущем возможна интеграция с кухонными дисплей-системами, что позволит ещё более точно отслеживать выполнение заказов.

Бронирование столиков: Функциональность бронирования позволяет клиенту выбрать дату, время и количество гостей. Система проверяет наличие свободных мест и направляет заявку администратору на подтверждение. Такой подход уменьшает число ошибочных бронирований, позволяет лучше планировать загрузку зала и снижает необходимость в устных переговорах с клиентом.

Дополнительные функции модуля бронирования включают автоматическое напоминание клиенту за 2 часа до брони, возможность клиентом самостоятельно отменить бронь в личном кабинете, и отображение общей статистики по броням в админ-панели. Также предусмотрена возможность ограничения бронирований по времени в зависимости от дней недели и времени суток, что удобно для планирования смен.

Аналитика и СППР: Модуль аналитики включает в себя средства для сбора, обработки и визуализации данных. Предусмотрены следующие функции: анализ выручки по дням и неделям, определение наиболее популярных блюд, расчёт среднего чека, прогнозирование спроса в зависимости от сезонных факторов и статистики заказов. Также реализована подсистема рекомендаций, которая на основе накопленных данных предлагает администратору решения по изменению состава меню, повышению цен на определённые блюда или увеличению количества персонала в часы пик.[2]

Инструменты аналитики дополнены визуальными графиками, тепловыми картами посещаемости, сравнениями по периодам (например, выручка понедельников текущего и прошлого месяца), и экспортом в XLSX и PDF форматы. Прогнозные функции опираются на методы скользящего среднего и трендовой экстраполяции, позволяя с высокой степенью вероятности оценивать будущие пиковые нагрузки и принимать обоснованные решения.[29]

## 1.3 Интерфейсные решения проекта

Для эффективного управления системой поддержки принятия решений в ресторанной сфере требуется тщательно продуманный интерфейс, адаптированный под различные роли пользователей и соответствующий актуальным стандартам пользовательского опыта (UX) и дизайна пользовательских интерфейсов (UI). Интерфейс является не просто средством навигации, а полноценным инструментом визуализации данных, исполнения функций системы и оптимизации бизнес-процессов. В современной цифровой среде визуальный и логический облик интерфейса напрямую влияет на производительность сотрудников, удовлетворённость клиентов и, как следствие, на успешность функционирования предприятия. [1]

Разработка интерфейсов в данном проекте базировалась на принципе «интерфейс как средство усиления логики», где каждый визуальный элемент несёт функциональную нагрузку. Проведены исследования взаимодействия персонала с существующими аналогами систем, в том числе с решениями R-Keeper, Poster POS и iiko. Анализ показал, что перегруженность интерфейсов, множество вкладок и непрозрачная навигация значительно тормозят скорость обслуживания. В отличие от них, интерфейсы разработанной системы отличаются целенаправленным минимализмом и чётким разграничением зон ответственности.

В рамках проекта разработано несколько интерфейсов, каждый из которых детально спроектирован под конкретные задачи и поведенческие сценарии соответствующей роли. Система предполагает четыре основные роли: администратор, официант, клиент и гость. Каждая из ролей взаимодействует с системой через собственный специализированный интерфейс, спроектированный с учётом уровня доступа, обязанностей и предполагаемых задач пользователя.

**Панель администратора** представляет собой серверную часть системы, которая агрегирует, визуализирует и позволяет оперативно управлять основными бизнес-процессами. Она делится на три ключевых блока:

**Аналитическая панель** — направлена на интерпретацию статистических и операционных данных:

- Динамика продаж по категориям блюд, временным периодам;

- Популярность позиций, их рентабельность;

- Средний чек по дням недели, сменам и официантам;

- Частота заказов с повторяемостью клиентов (коэффициент возврата)

Все данные визуализированы в виде столбчатых, линейных и круговых диаграмм, таблиц, графиков скользящего среднего.

**Оперативная панель** отвечает за контроль в реальном времени:

- Панель мониторинга текущих заказов и бронирований;

- Инструмент переназначения заказов между официантами;

- Интерактивный план зала с цветовой индикацией занятости столов;

- Настройки доступности столов по времени, зонам и событиям (банкет, ремонт);

- Мгновенное редактирование элементов меню, состава, наценок.

Для обеспечения быстродействия реализована система websocket-соединений, позволяющая обновлять статус интерфейса без перезагрузки.

**Стратегическая панель** служит для долгосрочного управления:

- Сравнение показателей с предыдущими периодами;

- Построение прогнозов по загрузке и спросу на основе сезонности;

- Сценарное моделирование изменений (например, добавление нового блюда как повлияет на средний чек);

- Генерация шаблонных отчётов для руководства или инвесторов.

Данный модуль интегрирован с аналитическим и использует кэширование отчётов для снижения нагрузки.

Также панель администратора включает:

- Редактор пользователей: добавление, изменение, блокировка, просмотр логов;

- Управление правами доступа: разграничение по ролям и действиям;

- Конфигурация расписания сотрудников, учёт их активности;

- Система оповещений и уведомлений (по e-mail, внутри интерфейса).

**Интерфейс официанта** является рабочим инструментом, предназначенным для быстрого и надёжного взаимодействия с заказами. Визуально он состоит из нескольких зон:

- Очередь заказов с таймером ожидания;

- Статусы заказов с визуальной индикацией (новый, в работе, готов, завершён);

- История заказов с возможностью поиска;

- Инструмент изменения статуса блюда (например, задержка по готовке);

- Возможность добавления комментариев клиента ("без соли", "отдельно упаковка").

Панель официанта поддерживает фильтрацию заказов по залам, столам, времени. Также реализован механизм смены — каждый официант может войти под своим ID, после чего фиксируются все действия в отчёте по смене. Данные синхронизируются с админ-панелью, что позволяет отслеживать производительность персонала.

**Клиентский интерфейс** — это основной канал взаимодействия посетителя с системой. Интерфейс реализован в виде адаптивного веб-приложения с возможностью использования в PWA-формате. Основной функционал включает:

Просмотр интерактивного меню с фото, ценами в тенге, описанием и составом;

- Быстрый поиск и фильтрация (категория, цена, вегетарианское, острое и пр.);

- Оформление заказа через корзину с комментариями;

- Просмотр и управление бронированиями (дата, время, гости);

- История заказов, повтор заказа, избранное меню.

Каждое действие пользователя сопровождается микровзаимодействием: всплывающее подтверждение, анимация добавления в корзину, цветовая индикация статуса. Интерфейс полностью локализован, использует адаптивную вёрстку, поддерживает светлую и тёмную темы.

**Гостевой интерфейс** — облегчённая версия клиентского, позволяющая без регистрации:

- Ознакомиться с рестораном и меню;

- Добавлять блюда в корзину.

Скорость загрузки и минимализм — ключевые приоритеты. Использованы статические пререндеры, lazy-loading изображений и упрощённый рендер.

**Универсальные принципы проектирования:**

- Адаптивная сетка (Tailwind CSS) под мобильные, планшеты и десктопы;

- Многоуровневая структура навигации;

- Контрастность и типографика согласно WCAG;

- Поддержка тёмной темы для комфортной работы в вечерние смены;

- Локализация на русском, казахском и английском языках.

Проект протестирован с участием пилотной группы сотрудников и клиентов. Результаты юзабилити-тестов позволили скорректировать структуру меню, изменить цвета индикации ошибок и добавить подсказки к часто используемым функциям.

Таким образом, интерфейсные решения, реализованные в рамках данного проекта, направлены на создание не просто интерфейсов взаимодействия, а полноценных цифровых рабочих пространств. Благодаря соблюдению международных стандартов UI/UX, а также глубокой адаптации к локальным особенностям работы ресторанного бизнеса, система позволяет эффективно выполнять задачи всех уровней — от приёма заказа до стратегического анализа эффективности работы предприятия. Разделение интерфейсов по ролям, высокая степень кастомизации, асинхронность работы, а также удобство навигации позволяют значительно повысить производительность персонала и удовлетворённость клиентов. Внедрение предложенных интерфейсных решений обеспечивает гибкость в масштабировании и лёгкость в обучении новых сотрудников. Их архитектура спроектирована таким образом, чтобы поддерживать возможные будущие расширения, включая внедрение новых ролей, дополнительных точек продаж, подключение внешних сервисов аналитики и автоматизации логистики.

Дополнительно стоит отметить важность визуальной унификации и стилистической преемственности между всеми интерфейсами системы. Использование единой дизайн-системы облегчает восприятие, минимизирует время на освоение и снижает вероятность пользовательских ошибок. Каждый элемент, от кнопки до таблицы, имеет строго определённый стиль, функциональную нагрузку и место в визуальной иерархии. Это способствует когнитивной согласованности восприятия даже у новых пользователей.

Также стоит подчеркнуть адаптацию интерфейсов к различным уровням цифровой грамотности пользователей. Интерфейс клиента интуитивно понятен даже для пользователей, не обладающих высоким техническим опытом, в то время как интерфейс администратора предоставляет продвинутые инструменты для анализа и управления, требующие минимального обучения. Это различие реализовано через уровни контекстной сложности: от простых иконок и подсказок в пользовательских интерфейсах до многоступенчатой фильтрации, сегментации и отчётности в административных модулях.

Кроме того, во всех интерфейсах реализована система контекстной справки, помогающая пользователю понять назначение каждого элемента и избежать ошибок. Поддерживаются горячие клавиши, автофокус при работе с формами, адаптивная подгрузка данных и оптимизация под слабые устройства. Всё это значительно повышает стабильность работы системы в условиях различной технической оснащённости.

Таким образом, интерфейсные решения в системе СППР не являются вторичным элементом — напротив, они играют фундаментальную роль в обеспечении взаимодействия пользователей с цифровыми ресурсами ресторана. Интерфейсы обеспечивают не только стабильную работу, но и формируют имидж заведения, демонстрируя высокий уровень цифровой зрелости и ориентации на клиента.[3][5]

## 1.4 Функциональные и технические требования

На этапе проектирования системы поддержки принятия решений для управления рестораном сформулированы чёткие функциональные и технические требования, обеспечивающие соответствие создаваемой системы целям и задачам, поставленным в рамках дипломного проекта. Эти требования стали основой для подготовки технического задания и послужили базой для последующей архитектурной и программной реализации. Формулировка требований производилась на основе анализа существующих систем управления ресторанами, особенностей работы заведения, ролей сотрудников и потребностей клиентов.[7]

**Общие положения проекта:**

**- Наименование проекта:** «Разработка системы поддержки принятия решений для управления рестораном».

- Назначение**:** автоматизация процессов обслуживания, сбора аналитических данных, управления персоналом, повышение прозрачности работы и повышение уровня клиентского сервиса.

**- Форма реализации:** веб-приложение, содержащее административную панель, пользовательский интерфейс, аналитические модули и базу данных.

**- Целевая аудитория:** администраторы, официанты, зарегистрированные клиенты и незарегистрированные гости.

**Функциональные требования:**

1. **Административная панель** — основной инструмент управления, предназначенный для администраторов ресторана. Она предоставляет следующие возможности:

- Полное управление содержимым меню: создание, редактирование, деактивация блюд, добавление ингредиентов и указание калорийности;

- Настройка цен и управление скидками, временными предложениями и ценовыми категориями;

- Управление ролями пользователей, создание и удаление учётных записей сотрудников, привязка к сменам;

- Анализ заказов в реальном времени, просмотр текущих и завершённых транзакций;

- Модуль составления отчётности: генерация отчётов по дням, сменам, пользователям и позициям меню;

- Управление отзывами клиентов, модерация комментариев, реакции на негативную обратную связь;

- Панель уведомлений: настройка событий, при которых отправляются уведомления (например, при отмене заказа);

- Встроенный лог активности для отслеживания действий каждого пользователя с указанием IP-адреса и времени.

1. **Интерфейс официанта** служит как инструмент для взаимодействия с заказами и клиентами:

- Отображение активных заказов по категориям (новые, в процессе, завершённые);

- Быстрая смена статуса блюда (в очереди, готовится, подано);

- Возможность привязки заказов к конкретным столам и отслеживание очереди блюд;

- Добавление комментариев и пометок к заказу (например, «без лука», «срочно»);

- Ведение статистики работы официанта с отображением показателей эффективности;

- Возможность получения уведомлений о важных событиях (задержка на кухне, отзыв клиента и т.д.).

1. **Клиентский интерфейс:**

- Регистрация и вход в личный кабинет через e-mail или номер телефона;

- Возможность редактирования профиля: имя, контактные данные;

- Просмотр актуального меню с возможностью сортировки и фильтрации (по ингредиентам, цене, типу блюда);

- Формирование заказа и отправка с комментарием (например, «не острое», «двойная порция»);

- История заказов с функцией повторения и отметкой «избранное»;

- Просмотр начисленных бонусов и их использование;

- Возможность оставить оценку еде и обслуживанию.

1. **Интерфейс гостя:**

- Просмотр публичного меню без регистрации;

- Доступ к информации о заведении, часах работы, адресе и контактных данных;

- Возможность оставить заявку на бронирование с указанием времени и количества человек;

- Призыв к регистрации для оформления полноценного заказа.

**Технические требования:**

**- Клиентская часть:** разработана на React.js с использованием Tailwind CSS для стилизации и построения адаптивной сетки. Поддерживаются устройства с различными экранами: от смартфонов до мониторов 4K.

**- Серверная часть:** реализована с использованием Python и FastAPI. Архитектура REST API обеспечивает независимость между модулями и простоту масштабирования.

**- База данных:** SQLite — лёгкая реляционная база, встроенная в проект, подходит для локального и MVP-развёртывания. Структура БД оптимизирована под частые запросы, реализованы индексы.

**- Аналитика:** визуализация данных через Recharts, сохранение метрик по заказам, активности пользователей и отзывам. Поддержка экспорта отчётов в CSV и PDF.

**- Среда выполнения:** ОС Linux с установленным Python 3.10+, возможно использование Docker-контейнеров для быстрого развёртывания.

**Требования к безопасности и стабильности:**

- Аутентификация на базе JWT (JSON Web Tokens), защита маршрутов от несанкционированного доступа;

- Хэширование паролей с помощью bcrypt;

- Проверка данных на стороне клиента и сервера;

- Защита от XSS, CSRF и SQL-инъекций;

- Ведение логов с возможностью аудита действий пользователей;

- Система резервного копирования с периодичностью не менее одного раза в сутки.

**Условия эксплуатации:**

- Поддержка браузеров: Google Chrome, Mozilla Firefox, Microsoft Edge последних версий;

- Совместимость с ОС: Windows 10+, macOS, Android, iOS;

- Адаптивная верстка и мобильная оптимизация;

­ Поддержка светлой и тёмной тем;

­ Возможность масштабирования архитектуры при расширении бизнеса.

**Этапы реализации:**

1. Подготовка технического задания, сбор требований, согласование с заказчиком;
2. Проектирование архитектуры базы данных и схем взаимодействия модулей;
3. Разработка интерфейсов (администратор, официант, клиент, гость);
4. Реализация бизнес-логики на серверной стороне и связка с БД;
5. Тестирование: модульное, функциональное, интеграционное, UX;
6. Развёртывание MVP-версии системы и начальная эксплуатация;
7. Обратная связь, фиксация багов, оптимизация и доработка;
8. Подготовка инструкций, обучение сотрудников и окончательная сдача.

**Критерии приёмки:**

- Полное соответствие заявленным функциональным требованиям;

- Отсутствие критических ошибок при стандартных сценариях использования;

- Успешное прохождение тестирования и соответствие требованиям безопасности;

- Готовность к эксплуатации и наличие вспомогательной документации.

Дополнительно стоит отметить, что сформулированные требования учитывают не только текущие потребности ресторана, но и возможные сценарии масштабирования и дальнейшего роста системы. Например, в случае расширения сети ресторанов система может быть доработана для многопрофильного управления несколькими точками, с возможностью настройки индивидуальных параметров и политик для каждой из них. Также предусмотрена возможность интеграции с внешними бухгалтерскими и управленческими системами, что позволяет создать единую цифровую экосистему предприятия.

Кроме того, в систему заложены требования по масштабируемости интерфейса, благодаря чему возможно быстрое подключение новых модулей (например, доставка или управление бронированием мероприятий). Эти требования позволяют поддерживать систему в актуальном состоянии без необходимости полной переработки кода. Такой подход обеспечивает гибкость и адаптивность — важнейшие качества в условиях динамично меняющейся ресторанной среды.[18]

Отдельное внимание в функциональных и технических требованиях уделено вопросам устойчивости и отказоустойчивости. Применены методики резервного копирования, антикризисного восстановления, отслеживания сбоев и автоматического перезапуска критических процессов. Это гарантирует, что в случае любых непредвиденных ситуаций, система продолжит работать с минимальными потерями данных и без критических простоев.

Таким образом, данные функциональные и технические требования формируют не просто техническое описание кода, а стратегически продуманный каркас цифровой платформы, позволяющий эффективно управлять рестораном, анализировать данные, автоматизировать взаимодействие с клиентами и обеспечивать высокую производительность персонала.[1]

## 1.5 Технологии и инструментальные средства разработки

Выбор технологий и инструментальных средств для реализации системы поддержки принятия решений для управления рестораном продиктован не только актуальными тенденциями в разработке программного обеспечения, но и практическими задачами, которые стояли перед системой. Важнейшими критериями выбора стали: надёжность, масштабируемость, производительность, безопасность, а также активная поддержка сообществом разработчиков.

В качестве основного языка для реализации серверной части выбран **Python**. Это один из самых популярных языков программирования, который отличается читаемым синтаксисом, обширной экосистемой библиотек и фреймворков, а также поддержкой сообщества. Особое внимание уделено **FastAPI** — современному фреймворку для создания веб-приложений и REST API, обладающему высокой скоростью работы и простотой в разработке. FastAPI обеспечивает асинхронную обработку запросов, что критически важно для проектов с высокой нагрузкой и необходимостью обрабатывать десятки параллельных соединений.[16]

Благодаря FastAPI удалось реализовать чистую архитектуру API, обеспечить документацию Swagger автоматически, а также упростить написание модульных тестов. Python и FastAPI идеально подошли для реализации сложной бизнес-логики, построения отчётов, логирования действий пользователей и безопасной авторизации через JWT.

Для клиентской части выбрано решение на базе **React.js**. Этот JavaScript-фреймворк обеспечивает высокую отзывчивость интерфейса, компонентную архитектуру и отличную интеграцию с API. Использование **Tailwind CSS** позволило создавать адаптивный, модульный и современный интерфейс без избыточных CSS-файлов. Такой подход упростил поддержку стиля, ускорил разработку и обеспечил кроссбраузерную совместимость.[9][15]

React-приложение построено по принципам SPA (Single Page Application), что позволило достичь высокой скорости отклика при переходах между страницами, минимизировать загрузку и повысить удобство взаимодействия. Интерфейсы разработаны с учётом пользовательского опыта на различных устройствах: от смартфонов до широкоформатных дисплеев в офисах управления.[19]

В качестве СУБД (системы управления базами данных) использована **SQLite**. Этот выбор обусловлен простотой настройки, отсутствием необходимости в отдельном сервере и хорошей производительностью при малых и средних объёмах данных. База данных представляет собой файл, который можно переносить между системами без дополнительной конфигурации. Для целей дипломного проекта, а также при локальной эксплуатации в небольшом ресторане, SQLite является полностью достаточной и оптимальной.[20]

Структура базы данных включает таблицы пользователей, заказов, меню, отзывов, ролей, действий и событий. Использованы внешние ключи, индексы по часто запрашиваемым колонкам и триггеры для автоматизации внутренней логики.

Для построения визуальной аналитики применялась библиотека **Recharts**. Она позволила отображать на административной панели графики, диаграммы и показатели в реальном времени. Система визуализации охватывает данные по продажам, активности пользователей, эффективности смен и предпочтениям клиентов. Также реализована возможность выгрузки отчётов в формате CSV и PDF для последующего анализа и предоставления руководству.

Основная среда разработки — **Visual Studio Code (VS Code).** Это лёгкий, но мощный редактор, поддерживающий все используемые языки и фреймворки проекта. Благодаря множеству расширений, интеграции с Git и встроенному терминалу, VS Code стал универсальной платформой для ведения всего проекта. Использованы расширения для Python, React, Tailwind, а также плагины для работы с REST API и базы данных SQLite. [8][17]

Для управления версиями проекта применялся **Git**, с размещением репозитория на платформе **GitHub**. В процессе разработки велось ветвление под новые модули, фиксация багов и контроль за кодом. Использование Git обеспечило надёжное хранение исходного кода, возможность отката изменений и ведение истории коммитов.

Для тестирования использовались следующие методы:

**- Модульное тестирование** основных функций API;

**- Интеграционные тесты** клиент-серверного взаимодействия;

**- Юзабилити-тестирование** интерфейсов на пилотной группе пользователей;

- Проверка адаптивности и кроссбраузерной совместимости.

Для целей тестирования и демонстрации использовалось локальное развёртывание проекта. Система запускается на любой машине с установленным Python и SQLite, не требует отдельного сервера или облачной платформы. При необходимости может быть упакована в **Docker-контейнер**, что позволит легко перенести проект в любое окружение, в том числе на VPS или корпоративный сервер ресторана.

Таким образом, совокупность выбранных технологий и инструментальных средств позволила реализовать надёжную, гибкую, современную и масштабируемую информационную систему, обеспечивающую эффективную поддержку принятия решений в сфере ресторанного бизнеса.

# 2 Проектирование, разработка и внедрение системы

## 2.1 Архитектура системы

Архитектура системы поддержки принятия решений для управления рестораном реализована в виде трёхуровневой структуры, состоящей из клиентской части, уровня API и серверной логики, а также уровня базы данных. Такая схема обеспечивает модульность, простоту поддержки, гибкость внедрения новых компонентов и масштабируемость в случае роста нагрузки. Все компоненты взаимодействуют между собой через API Gateway.[11]

На клиентском уровне реализованы пользовательские интерфейсы для различных ролей пользователей:

**Публичный интерфейс** — предоставляет гостям доступ к просмотру меню без необходимости регистрации.

**Клиентский интерфейс** — предназначен для зарегистрированных пользователей: просмотр меню, оформление заказов, управление профилем, отзывы и история заказов.

**Интерфейс официанта** — предоставляет персоналу возможность обработки текущих заказов, изменения их статуса, а также взаимодействия с системой в режиме смены.

**Админ-панель** — даёт администратору доступ к управлению пользователями, меню, заказами, просмотр аналитики и отчётов.

Все интерфейсы обращаются к серверной части через **API Gateway**, обеспечивающий маршрутизацию, безопасность и разграничение прав доступа.

Серверная логика реализована на FastAPI и разделена на независимые функциональные модули, каждый из которых решает отдельные бизнес-задачи:

**Модуль пользователей** — регистрация, авторизация, профили, роли и аутентификация через JWT.

**Модуль заказов** — управление оформленными заказами, их статусами, привязкой к пользователям и официантам.

**Модуль меню** — добавление и редактирование блюд, категоризация и фильтрация позиций.

**Модуль аналитики и отчётов** — построение графиков, подсчёт ключевых метрик, экспорт статистики.

**Модуль бронирования** — формирование и учёт брони на столы (если используется).

**Модуль оформления заказов** — связывает клиента и официанта, оформляет финальные действия в процессе подачи заказа.

**Система поддержки принятия решений** — на базе данных анализирует спрос, средний чек, занятость персонала и выдаёт рекомендации.

На уровне базы данных сосредоточены таблицы и хранилища информации:

1. **Таблицы пользователей** — учётные записи, роли, токены;
2. **Таблицы заказов** — текущие, завершённые, отменённые заказы;
3. **Таблицы меню** — блюда, категории, изображения, цены;
4. **Таблицы аналитических данных** — показатели и отзывы.

Все данные хранятся в SQLite, что упрощает установку, поддержку и переносимость системы. Структура оптимизирована под частые запросы и аналитическую выборку.

Клиентские интерфейсы взаимодействуют с API Gateway, который перенаправляет запросы к соответствующим модулям FastAPI. Каждый модуль имеет чётко определённую область ответственности. Бэкенд обращается к базе данных, сохраняя и извлекая информацию по необходимости.[24][26]

Визуально архитектура представлена на рисунке 2.1:

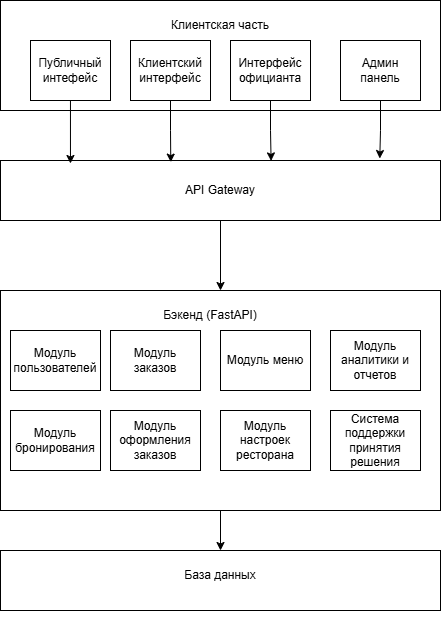


Рисунок 2.1 — Архитектура системы СППР для ресторана

Такая архитектура обеспечивает логичное разделение ответственности, ускоряет разработку, облегчает поддержку, а также создаёт условия для безопасного масштабирования системы под задачи реального бизнеса.

Проектирование, разработка и внедрение системы

Архитектура системы поддержки принятия решений для управления рестораном реализована в виде трёхуровневой структуры, состоящей из клиентской части, уровня API и серверной логики, а также уровня базы данных. Такая схема обеспечивает модульность, простоту поддержки, гибкость внедрения новых компонентов и масштабируемость в случае роста нагрузки. Все компоненты взаимодействуют между собой через API Gateway.

**Клиентская часть:**

На этом уровне реализованы пользовательские интерфейсы для различных ролей пользователей:

**Публичный интерфейс** — предоставляет гостям доступ к просмотру меню без необходимости регистрации.

**Клиентский интерфейс** — предназначен для зарегистрированных пользователей: просмотр меню, оформление заказов, управление профилем, отзывы и история заказов.

**Интерфейс официанта** — предоставляет персоналу возможность обработки текущих заказов, изменения их статуса, а также взаимодействия с системой в режиме смены.

**Админ-панель** — даёт администратору доступ к управлению пользователями, меню, заказами, просмотр аналитики и отчётов.

Все интерфейсы обращаются к серверной части через **API Gateway**, обеспечивающий маршрутизацию, безопасность и разграничение прав доступа.

**Бэкенд (FastAPI):**

Серверная логика реализована на FastAPI и разделена на независимые функциональные модули, каждый из которых решает отдельные бизнес-задачи:

**- Модуль пользователей** — регистрация, авторизация, профили, роли и аутентификация через JWT.

**- Модуль заказов** — управление оформленными заказами, их статусами, привязкой к пользователям и официантам.

- Модуль **меню** — добавление и редактирование блюд, категоризация и фильтрация позиций.

**- Модуль аналитики и отчётов** — построение графиков, подсчёт ключевых метрик, экспорт статистики.

**- Модуль бронирования** — формирование и учёт брони на столы (если используется).

**- Модуль оформления заказов** — связывает клиента и официанта, оформляет финальные действия в процессе подачи заказа.

**- Модуль настроек ресторана** — отвечает за параметры работы системы, расписание, валюту, НДС и другие параметры.

**- Система поддержки принятия решений** — на базе данных анализирует спрос, средний чек, занятость персонала и выдаёт рекомендации.

На уровне базы данных сосредоточены таблицы и хранилища информации:

**- Таблицы пользователей** — учётные записи, роли, токены;

**- Таблицы заказов** — текущие, завершённые, отменённые заказы;

**- Таблицы меню** — блюда, категории, изображения, цены;

**- Таблицы аналитических данных** — агрегированные отчёты, показатели и отзывы.

Все данные хранятся в SQLite, что упрощает установку, поддержку и переносимость системы. Структура оптимизирована под частые запросы и аналитическую выборку.

**Связь между слоями:**

Клиентские интерфейсы взаимодействуют с API Gateway, который перенаправляет запросы к соответствующим модулям FastAPI. Каждый модуль имеет чётко определённую область ответственности. Бэкенд обращается к базе данных, сохраняя и извлекая информацию по необходимости.[12]

## 2.2 База данных системы

База данных информационной системы поддержки принятия решений для ресторана построена на реляционной модели и реализована с использованием системы управления базами данных SQLite. Структура базы данных отражает все основные сущности, действия и настройки, необходимые для корректного функционирования проекта. Все таблицы логически разделены на группы: пользователи, настройки, меню, оформление заказов, аналитика и справочные данные.

**1. Таблицы пользователей и управления ролями:**

- users — основная таблица, содержащая данные обо всех зарегистрированных пользователях: идентификатор, адрес электронной почты, телефон, зашифрованный пароль, полное имя, роль (гость, клиент, официант, администратор), дата рождения и временные метки создания/обновления записи;

- Дополнительные поля, такие как birthday, age\_group, is\_active позволяют персонализировать опыт пользователей и реализовывать возрастные ограничения или приоритетные группы клиентов.

**2. Таблицы управления рестораном и конфигурацией системы:**

- settings — содержит глобальные параметры заведения: наименование, контактные данные, адрес, сайт, валюта и символ валюты (в системе используется тенге), процент налога, минимальная сумма заказа, стоимость доставки, условия бесплатной доставки. Также в этой таблице реализована поддержка JSON-полей для хранения рабочего расписания (working\_hours) и схемы размещения столов (tables).

- SMTP и SMS-настройки позволяют интегрировать рассылки уведомлений пользователям, подтверждения заказов и автоматическую отправку системных сообщений.

**3. Таблицы справочной информации и классификаторы:**

- categories — таблица с категориями блюд (например, горячие блюда, напитки, десерты). Каждая категория имеет название, описание и метки времени;

- allergens — список возможных аллергенов с их кратким описанием, используется для маркировки блюд;

- tags — используется для присвоения блюдам тегов (например, острое, веганское, без сахара), что позволяет клиентам фильтровать меню по предпочтениям;

- Эти таблицы используются интерфейсом администратора и клиентской частью для динамической фильтрации и настройки карточек блюд.

**4. Таблицы меню и связей между сущностями:**

- products — основная таблица с блюдами. Содержит название, описание, цену, вес, состав, калорийность и внешние ключи на категорию, аллерген и другие параметры;

- product\_tags — связь многие ко многим между products и tags;

- product\_allergens — связь многие ко многим между products и allergens;

- Эти таблицы формируют ядро клиентского интерфейса при формировании меню и заказа.

**5. Таблицы заказов и оформления покупок:**

- orders — таблица заказов, в которой фиксируются заказы клиентов: идентификатор пользователя, статус заказа (новый, в процессе, завершён, отменён), итоговая сумма, дата оформления и тип оплаты;

- order\_items — товары, включённые в каждый заказ, с указанием количества, цены на момент оформления, скидок и наценок;

- Такая структура обеспечивает полную прослеживаемость и поддержку аналитики для оценки работы персонала.

**6. Таблицы отзывов и пользовательской активности:**

- reviews — содержит оценки и текстовые отзывы от клиентов о блюдах или обслуживании. Связывается с таблицей users и может быть модифицирована администратором;

**7. Технические особенности базы данных:**

- База данных спроектирована с учётом принципов нормализации и обеспечения согласованности данных. Каждая таблица реализует строго определённую бизнес-сущность, а связи между ними обеспечивают высокую степень согласованности и удобства в обработке. Например, отношение «многие ко многим» реализовано с помощью вспомогательных таблиц (product\_tags, product\_allergens, order\_items), а поля с внешними ключами защищены каскадной связью, что упрощает удаление и обновление записей.

- Все таблицы содержат обязательные поля created\_at и updated\_at для отслеживания изменений и ведения версионности;

- Реализованы внешние ключи между таблицами для обеспечения целостности данных (например, orders → users, order\_items → products);

- Имеются индексы по полям, часто используемым в фильтрации и сортировке: email, status, created\_at и др.;

- Используются поля типа BOOLEAN, VARCHAR, JSON, DATETIME, обеспечивающие гибкость при хранении структурированных и полуструктурированных данных.

Визуализация структуры базы данных проекта представлена на рисунке ниже и охватывает все ключевые сущности системы, их атрибуты и связи между ними. Данная диаграмма отражает логику хранения и взаимодействия данных, обеспечивающих работу всех подсистем — от заказов и меню до настроек и пользовательской активности. На рисунке 2.2 представлена визуальная схема базы данных.

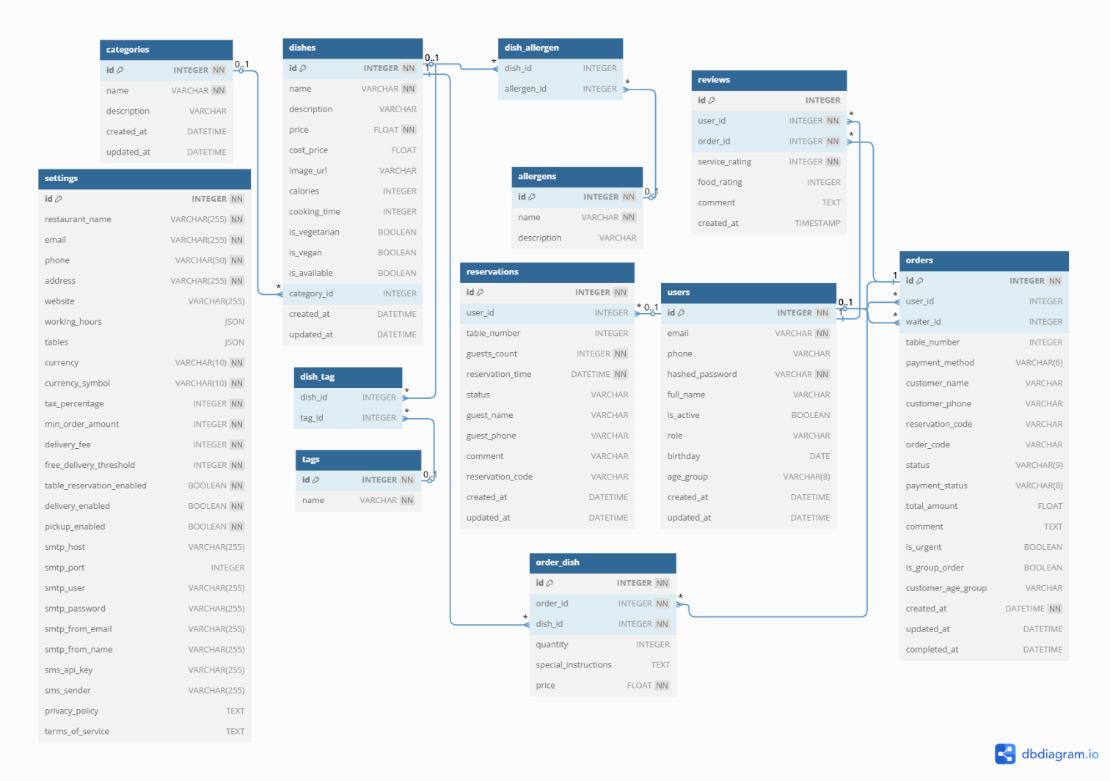


Рисунок 2.2 — Визуальная схема базы данных

Важным элементом проектирования базы данных также является обеспечение масштабируемости и адаптивности. Благодаря строгой логической структуре, новые функции или изменения в бизнес-логике могут быть реализованы путём добавления соответствующих таблиц, полей или индексов без изменения основного ядра данных. Например, если в будущем ресторан решит внедрить систему бонусов, таблицу лояльности или историю оплат, их можно будет легко встроить в существующую архитектуру.

База данных не только служит хранилищем данных, но и выступает в роли центра аналитики. На основе хранимой информации возможно формирование отчётов о популярности блюд, среднем чеке, частоте заказов и активности пользователей по времени суток. Эта информация может использоваться руководством ресторана для принятия управленческих решений: оптимизации меню, пересмотра ценовой политики или изменения состава смен персонала.

Кроме того, стоит отметить, что реляционная модель с чётко заданными ограничениями и связями помогает минимизировать ошибки на уровне данных. Например, невозможность создать заказ без привязки к существующему пользователю исключает логические нарушения в работе системы.

База данных проекта охватывает все аспекты работы цифровой ресторанной системы — от регистрации пользователя до получения отзыва о заказе. Благодаря модульной структуре, использование реляционных связей и гибкой схеме расширения, система обладает высокой адаптивностью. При необходимости возможно внедрение новых таблиц и полей без нарушения целостности. Благодаря использованию SQLite база данных легко переносится, не требует установки сервера и может быть развёрнута на любом устройстве, поддерживающем Python.

## 2.3 Взаимодействие пользователей с системой

Для системного понимания процессов взаимодействия между пользователями и системой поддержки принятия решений в ресторанной сфере использовались диаграммы вариантов использования (use case diagrams). Этот инструмент визуального моделирования из UML (Unified Modeling Language) позволяет представить, какие действия может выполнять каждая из ролей системы и как именно они взаимодействуют с основными функциями.[30]

Система разработана с учётом наличия четырёх основных типов пользователей. В этом разделе будет подробно рассмотрено взаимодействие с системой для каждой из ролей, начиная с самой базовой — **гостя**.

Гость — это неавторизованный пользователь, который впервые взаимодействует с системой. Его возможности ограничены, однако они играют важную роль в привлечении клиентов и демонстрации функционала системы. Основные сценарии взаимодействия:

**Ознакомление с меню:** гость может просматривать все доступные блюда, разбитые по категориям. Для удобства реализованы фильтры по аллергенам, тегам (вегетарианские, острые и т. д.) и ценовому диапазону. Каждое блюдо отображается с фотографией, описанием, ценой и составом.

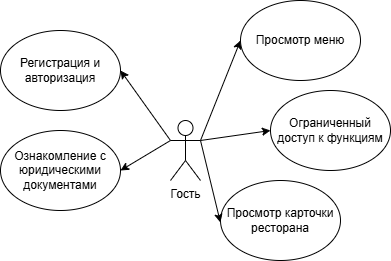
**Просмотр карточки заведения:** пользователь может ознакомиться с основной информацией о ресторане, включая название, описание концепции, график работы, контактные данные, местоположение на карте и способы связи (телефон, email, Telegram). Также отображаются логотип, краткое приветствие и фотогалерея интерьера/блюд.

**Регистрация и вход в систему:** чтобы оформить заказ или воспользоваться расширенными функциями, гость может зарегистрироваться. Для этого предусмотрена форма регистрации по email. После регистрации гость переходит в роль клиента.

**Ознакомление с юридическими документами:** гость может открыть страницы с политикой конфиденциальности, пользовательским соглашением и условиями возврата/оплаты. Это особенно важно для прозрачности и доверия со стороны потенциальных клиентов.

**Ограничение доступа к функциям:** гость может добавить блюда в корзину, но попытка оформить заказ или бронь приводит к всплывающему окну с предложением авторизоваться или зарегистрироваться. Такой подход мотивирует гостя к конверсии в клиента.

На диаграмме вариантов использования (use case diagram), гость обозначен как актер, взаимодействующий с системой через публичный интерфейс. Он может инициировать просмотр меню, просматривать информацию о ресторане и начать процесс регистрации. На рисунке 2.3 представлена Use Case диаграмма для роли «Гость».



**Рисунок 2.3 — Use Case диаграмма для роли «Гость»**

Этот сценарий учтён при разработке интерфейса главной страницы, навигации и логики ограничения доступа к защищённым действиям. В дальнейшем, часть данных, полученных от гостей (например, поисковые запросы, время на странице, просмотренные блюда), может использоваться для аналитики и оптимизации пользовательского пути. Таким образом, роль гостя — это первый шаг к превращению посетителя в постоянного клиента, и её важно реализовать удобно, информативно и без избыточного давления.

Клиент — это авторизованный пользователь, который получил доступ к полному функционалу системы после прохождения процедуры регистрации. Эта роль является центральной в пользовательском контуре платформы, так как именно через неё осуществляется основная активность: формирование заказов, взаимодействие с меню, просмотр истории и многое другое. Ниже приведены ключевые сценарии взаимодействия клиента с системой.

**Добавление блюд в корзину**: каждое блюдо можно добавить в корзину одним нажатием. При этом система учитывает модификации блюда и отображает их в составе заказа. Корзина обновляется в реальном времени, позволяя видеть итоговую сумму заказа, применённые скидки и стоимость доставки (если она активна).

Бронирование столика: клиент имеет возможность заранее забронировать столик через интерфейс сайта или мобильного приложения. Для этого он выбирает дату и время визита, количество человек и предпочтительное расположение. Система проверяет доступные столы и сохраняет заявку на бронь. Важно отметить, что подтверждение брони осуществляется администратором ресторана. До момента подтверждения клиенту отображается статус "ожидает подтверждения". В случае необходимости клиент может отменить или изменить бронь из своего профиля. Эта функция особенно важна в пиковые часы и позволяет планировать загрузку зала, избегать очередей и обеспечивать персонализированный сервис. После подтверждения пользователь получать код брони, который можно использовать на странице оформления заказа.

**Оформление заказа**: после выбора блюд клиент переходит к оформлению заказа. Клиент может вбить код брони для заказа блюда к определенному времени, чтобы когда клиент пришел, еда уже стояла на столе, или код, который дает официант, благодаря которому можно сделать заказ, указать комментарии к блюдам (например, "без лука"), выбрать тип оплаты (наличными, по карте).

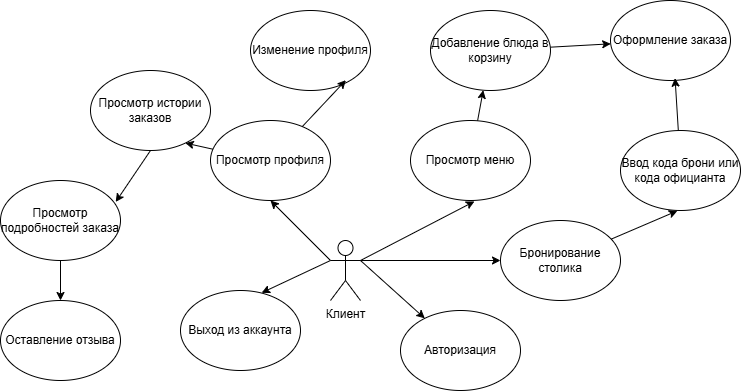
**Редактирование профиля**: клиент имеет доступ к разделу "Профиль", где может изменить своё имя, номер телефона, email, дату рождения. Также есть возможность установить аватар и изменить пароль.

**История заказов**: вся история заказов клиента сохраняется в отдельном разделе. Пользователь может просмотреть дату, состав и сумму каждого заказа, повторить любой заказ или оставить отзыв по каждому блюду из заказа.

**Оставление отзыва**: клиент может поставить оценку и написать отзыв на заказ, которые заказывал и обслуживание. Это даёт администрации обратную связь о качестве еды и сервиса. Отзывы отображаются в интерфейсе админ-панели.

**Безопасность**: клиент имеет возможность выйти из аккаунта, сбросить пароль и пройти повторную авторизацию.

На рисунке 2.4 представлена **Use Case диаграмма для роли «Клиент».**



**Рисунок 2.4 — Use Case диаграмма для роли «Клиент»**

Интерфейс клиента разработан с упором на простоту, понятность и быстроту взаимодействия. Особое внимание уделено доступности функций, визуальному представлению информации и адаптивности под мобильные устройства. Сценарии протестированы с учётом поведенческих паттернов пользователей, что позволило минимизировать количество шагов от выбора до оформления заказа.

Таким образом, клиент в системе — это основной пользователь, чьи действия определяют эффективность всей платформы. Грамотно проработанные сценарии позволяют сделать процесс заказа интуитивным, а интерфейс — дружественным и привлекательным для разных категорий посетителей ресторана.

Официант — это сотрудник ресторана, работающий с заказами, поступающими от клиентов, и отвечающий за передачу этих заказов на кухню, подачу блюд, а также взаимодействие с клиентами по ходу обслуживания. В системе официант является авторизованным пользователем с ограниченным доступом к функциональности, необходимой для выполнения своих обязанностей. Ниже приведены ключевые сценарии взаимодействия официанта с системой.

**Просмотр активных заказов**: при входе в систему официант получает доступ к панели, в которой отображаются все текущие активные заказы. Каждый заказ включает состав, статус (ожидает, готовится, готов к подаче), стол, на который оформлен заказ, время оформления и имя клиента. Также могут отображаться дополнительные метки, указывающие на срочность заказа или особые пожелания.

**Подтверждение получения заказа**: после того как клиент оформил заказ, он отображается в списке официанта как "Новый заказ". Официант может нажать "Принять", после чего заказ переходит в статус "В обработке". Этот этап фиксируется в системе и виден на панели администратора.

**Контроль готовности блюд**: официант следит за статусом блюд. Как только кухня помечает блюдо как "готово", официанту приходит уведомление. Он проверяет блюдо и подтверждает подачу. Уведомления приходят в режиме реального времени, что помогает снизить задержки в обслуживании.

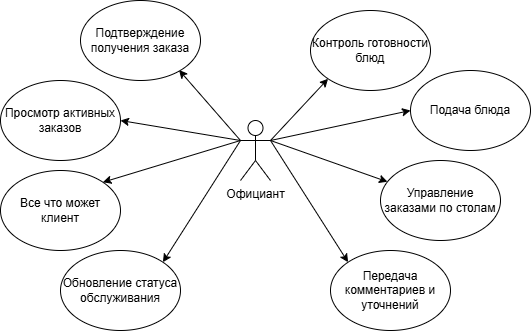
**Подача блюда**: после подтверждения, что блюдо подано, официант переводит заказ в статус "передан клиенту". Это позволяет системе отслеживать процесс исполнения и видеть общую нагрузку персонала в зале. Также это может служить основой для формирования отчетов о скорости обслуживания.

**Управление заказами по столам**: в интерфейсе отображаются все заказы, сгруппированные по столам. Это позволяет официанту быстро ориентироваться в зале, понимать к какому клиенту какой заказ относится, и избегать ошибок при подаче. Визуальное отображение заказов и занятых столов также упрощает взаимодействие с другими сотрудниками.

**Передача комментариев и уточнений**: в процессе оформления заказа клиент может оставить комментарий (например, "пожарить посильнее" или "принести две порции позже"). Официант видит эти комментарии в своём интерфейсе и может при необходимости отредактировать или уточнить их перед отправкой на кухню.

**Обновление статуса обслуживания**: в случае форс-мажоров (например, задержка блюда, замена ингредиента, отсутствие позиции) официант может отметить проблему в системе и уведомить администратора или клиента.

На рисунке 2.5 представлена **Use Case диаграмма для роли «Официант»**



**Рисунок 2.5 — Use Case диаграмма для роли «Официант»**

Сценарии взаимодействия официанта с системой разработаны с учётом высокой динамики работы персонала в зале. Интерфейс максимально упрощён и адаптирован под сенсорные устройства (планшеты), используемые на рабочем месте. За счёт автоматического обновления статуса заказов, быстрой навигации по столам и централизованного управления комментариями, официант может значительно сократить время на обслуживание и уменьшить количество ошибок.

Таким образом, система предоставляет официанту чёткий и структурированный набор инструментов, необходимых для эффективной и качественной работы. Всё взаимодействие фиксируется, что обеспечивает прозрачность и возможность последующего анализа качества обслуживания.

Администратор является ключевой служебной ролью в системе и обладает самым широким уровнем прав доступа. Именно администратор отвечает за настройку, обслуживание и координацию системы, управление пользователями и контентом, а также контроль ключевых метрик и параметров работы. Он выступает как технический и организационный центр, через который проходит вся операционная информация и настройки.

**Управление пользователями и ролями:** администратор имеет доступ к полному списку пользователей системы. Он может создавать новых пользователей, назначать им роли, деактивировать или удалять аккаунты.

**Обработка бронирований столиков:** при поступлении брони от клиента, она поступает в интерфейс администратора со статусом "ожидает подтверждения". Администратор проверяет доступность столов и подтверждает или отклоняет бронь. Также он может изменить параметры бронирования и связаться с клиентом при необходимости.

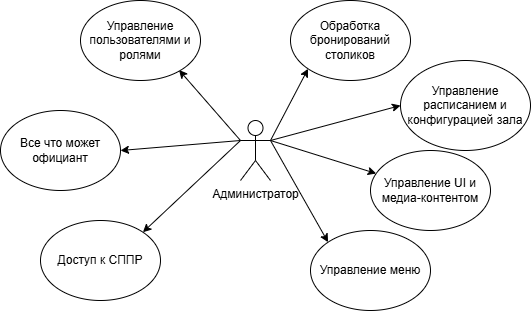
**Управление расписанием и конфигурацией зала:** в админ-панели доступен визуальный редактор схемы зала. Это позволяет администратору корректировать расположение столов, указывать их вместимость.

**Управление UI и медиа-контентом:** администратор может обновлять визуальные компоненты системы — логотип, контактную информацию, оформление главной страницы, изображения блюд и другие ресурсы, используемые в интерфейсе.

**Управление меню:** администратор имеет возможность редактировать меню ресторана, включая добавление, удаление и изменение блюд. Он может указывать состав, цену, вес, категорию блюда, наличие аллергенов, а также прикреплять изображения и задавать сортировку позиций. Кроме того, можно активировать или деактивировать блюда временно (например, в случае отсутствия ингредиентов), а также устанавливать параметры отображения по времени суток или дням недели. Интерфейс редактирования меню поддерживает быстрый поиск и фильтрацию позиций по категориям, что ускоряет обновление контента в случае изменений.

**Просмотр и анализ статистики:** система собирает данные о заказах, бронях, популярность блюд и их маржинальность, и т.д. Администратор имеет доступ к аналитике в виде таблиц, графиков и диаграмм. Это позволяет принимать обоснованные решения по оптимизации бизнес-процессов.

Доступ к СППР: администратор получает доступ к модулю системы поддержки принятия решений, в котором представлены агрегированные отчёты по выручке, количеству заказов, популярности позиций меню, активности пользователей, отказам, времени исполнения заказов и другим критическим метрикам. Он может использовать эти данные для анализа сезонности, построения прогнозов, выявления узких мест и подготовки предложений по оптимизации процессов ресторана. Система позволяет строить графики, сравнивать периоды, экспортировать отчёты и использовать фильтры по времени, типам пользователей или направлениям продаж. На рисунке 2.6 представлена **Use Case диаграмма для роли «Администратор».**



**Рисунок 2.6 — Use Case диаграмма для роли «Администратор»**

Таким образом, администратор выполняет стратегическую и операционную роль в системе, обеспечивая её непрерывную и эффективную работу. Система предоставляет инструменты для настройки каждого аспекта — от пользовательских форм до визуального отображения зала, что делает эту роль незаменимой для управления рестораном. Разделение задач и автоматизация процессов позволяют администратору быстро адаптировать систему под любые внутренние или внешние изменения.

## 2.4 Программная реализация

Программная реализация веб-системы поддержки принятия решений для управления рестораном основывается на модульной архитектуре и многоуровневом интерфейсе. Каждый раздел платформы спроектирован с учетом потребностей определённых ролей: гость, клиент, официант и администратор. Все компоненты реализованы с использованием современных веб-технологий, включая React.js, FastAPI, SQLite и Tailwind CSS, что обеспечило высокую производительность, адаптивность и расширяемость системы.[10]

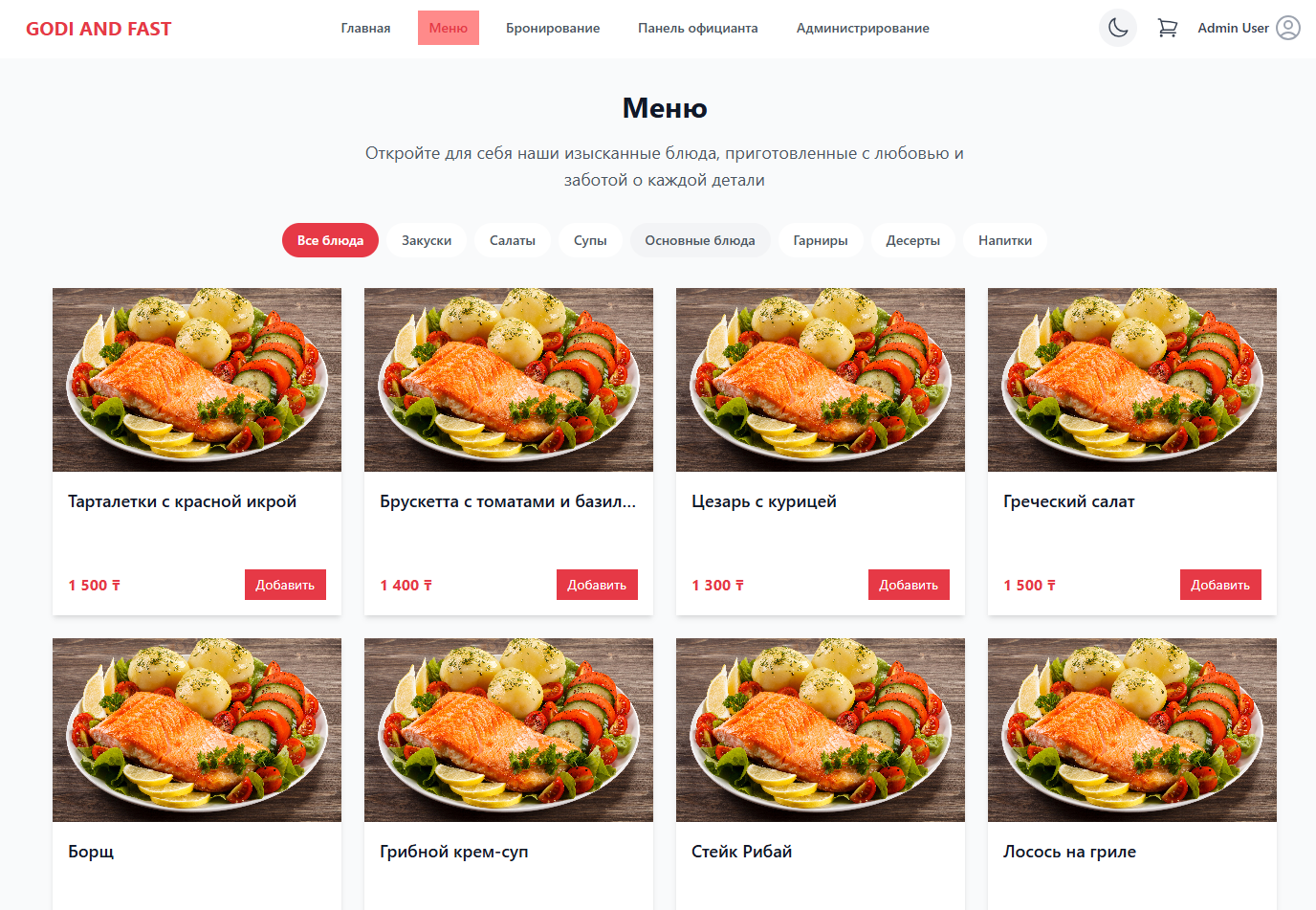
Раздел «Меню» представляет собой один из ключевых интерфейсных компонентов клиентской части системы. Он позволяет пользователю ознакомиться с полным ассортиментом блюд, предлагаемых рестораном, и служит стартовой точкой формирования заказа. Меню разделено на категории, такие как закуски, горячие блюда, напитки и десерты, что обеспечивает быстрый доступ к нужной группе позиций. Такой подход позволяет пользователю ориентироваться в меню без необходимости загружать избыточную информацию, снижая когнитивную нагрузку и улучшая восприятие структуры ассортимента.

На странице используется карточный интерфейс: каждое блюдо отображается в виде отдельной карточки с изображением, названием, кратким описанием, стоимостью и кнопкой «Добавить в корзину». Такой подход обеспечивает высокую читаемость и визуальное восприятие информации. При клике на карточку открывается подробная информация о блюде — состав, калорийность, вес порции, а также возможность выбрать количество порций перед добавлением в корзину. Отображение данных реализовано с учетом адаптивной вёрстки, обеспечивая корректное отображение как на настольных, так и на мобильных устройствах.

Интерфейс реализует динамическую фильтрацию по категориям и строку поиска. Это упрощает навигацию по обширному меню и позволяет быстрее находить нужные позиции. Кроме того, предусмотрена возможность сортировки блюд по популярности и стоимости, что особенно полезно для постоянных клиентов, знающих свои предпочтения и желающих сэкономить время на выборе. Функция фильтрации сопровождается анимацией и мгновенной реакцией интерфейса без перезагрузки страницы, что достигается благодаря архитектуре SPA (Single Page Application).

Также реализована система визуальных индикаторов для обозначения новинок, острых блюд и вегетарианских позиций. Это позволяет сделать пользовательский опыт более интуитивным и персонализированным. Взаимодействие пользователя с карточками блюд сопровождается микроанимациями, акцентами при наведении и плавными переходами, что делает работу с системой более приятной и современной с точки зрения UX/UI.

На рисунке 2.7 представлен интерфейс страницы меню клиента.



**Рисунок 2.7 – Интерфейс страницы меню клиента**

Раздел «Меню» предоставляет пользователю возможность ознакомиться со всеми блюдами, доступными в ресторане. Интерфейс выполнен в виде плиточной сетки, где каждое блюдо представлено в формате карточки, содержащей изображение, название, краткое описание, цену, а также кнопку «Добавить в корзину».

На странице реализована система фильтрации по категориям: «Салаты», «Супы», «Горячие блюда», «Напитки», «Десерты» и др. Это позволяет пользователю быстро найти нужную позицию. Также предусмотрена строка поиска, которая выполняет фильтрацию по названию и составу блюда в реальном времени.

Карточки обновляются динамически в зависимости от текущей фильтрации. При наведении на карточку визуально отображаются дополнительные элементы управления — увеличение изображения, раскрытие полного состава и кнопка перехода на страницу детального описания блюда. На отдельной странице блюда пользователь может ознакомиться с подробным описанием, узнать об аллергенах, энергетической ценности, а также изменить количество порций перед добавлением в корзину.

Разметка страницы реализована с использованием адаптивной сетки, обеспечивающей корректное отображение как на десктопах, так и на мобильных устройствах. Список блюд подгружается с сервера при открытии страницы, используя REST API, что минимизирует время загрузки и повышает отзывчивость интерфейса. В случае отсутствия соединения или ошибок загрузки, предусмотрено отображение fallback-сообщений и анимаций загрузки.[13]

Система взаимодействует с административной панелью, откуда загружается актуальное меню. Все изменения, внесённые администратором (например, добавление новых блюд или изменение цен), отображаются для пользователя в режиме реального времени благодаря использованию клиентского кэширования и периодической синхронизации. Корзина реализована в виде отдельной страницы и плавающего компонента на всех страницах. Пользователь может изменять количество порций, удалять позиции, видеть итоговую стоимость и перейти к оформлению заказа. При авторизации корзина автоматически синхронизируется с сервером и сохраняется между сессиями.

Страница бронирования столика предназначена для организации посещений ресторана и позволяет клиенту заранее выбрать удобное время, дату и количество мест. Интерфейс реализован с акцентом на простоту и интуитивность взаимодействия, что особенно важно для пользователей, впервые взаимодействующих с системой.

На первом этапе пользователь выбирает дату и временной интервал из доступных. Система автоматически исключает нерабочие часы ресторана и временные интервалы, уже занятые другими бронями. Следующим шагом клиент указывает количество гостей и переходит к выбору столика на схеме зала.

Интерактивная схема зала представляет собой визуальное отображение всех доступных столиков. Каждый столик отображается с уникальным номером и статусом (свободен, занят, ожидает подтверждения, временно недоступен). Визуальная реализация схемы адаптирована под мобильные и десктопные устройства, обеспечивая удобство навигации вне зависимости от формата экрана. При наведении курсора или тапе на столик отображаются дополнительные сведения: вместимость, расположение (например, у окна, в зале для некурящих) и возможность выбора.

После выбора места пользователь переходит к форме ввода контактных данных: имя, номер телефона, а также поле для комментариев. Здесь можно указать пожелания к брони, такие как размещение рядом с детской зоной, предоставление детского стула или указание о праздновании события.

После отправки формы бронь попадает в раздел «Ожидающие подтверждения» административной панели. Администратор, в свою очередь, может либо подтвердить бронирование, либо отклонить его, указав причину. Пользователь получает соответствующее уведомление в интерфейсе и по SMS (если настроено). Вся информация сохраняется в базе данных и доступна для просмотра в разделе «Мои бронирования».

До подтверждения клиент может внести изменения в бронь или полностью её отменить. В случае одобрения, система направляет автоматическое уведомление с деталями визита, включая дату, время и номер столика. Также за 2 часа до события пользователь получает напоминание, минимизируя вероятность неявки.

Для повышения удобства реализован адаптивный дизайн страницы, а также оптимизация скорости отклика: бронирование не требует полной перезагрузки страницы, благодаря использованию асинхронных запросов.

На рисунке 2.8 представлен интерфейс страницы бронирования столика клиента.

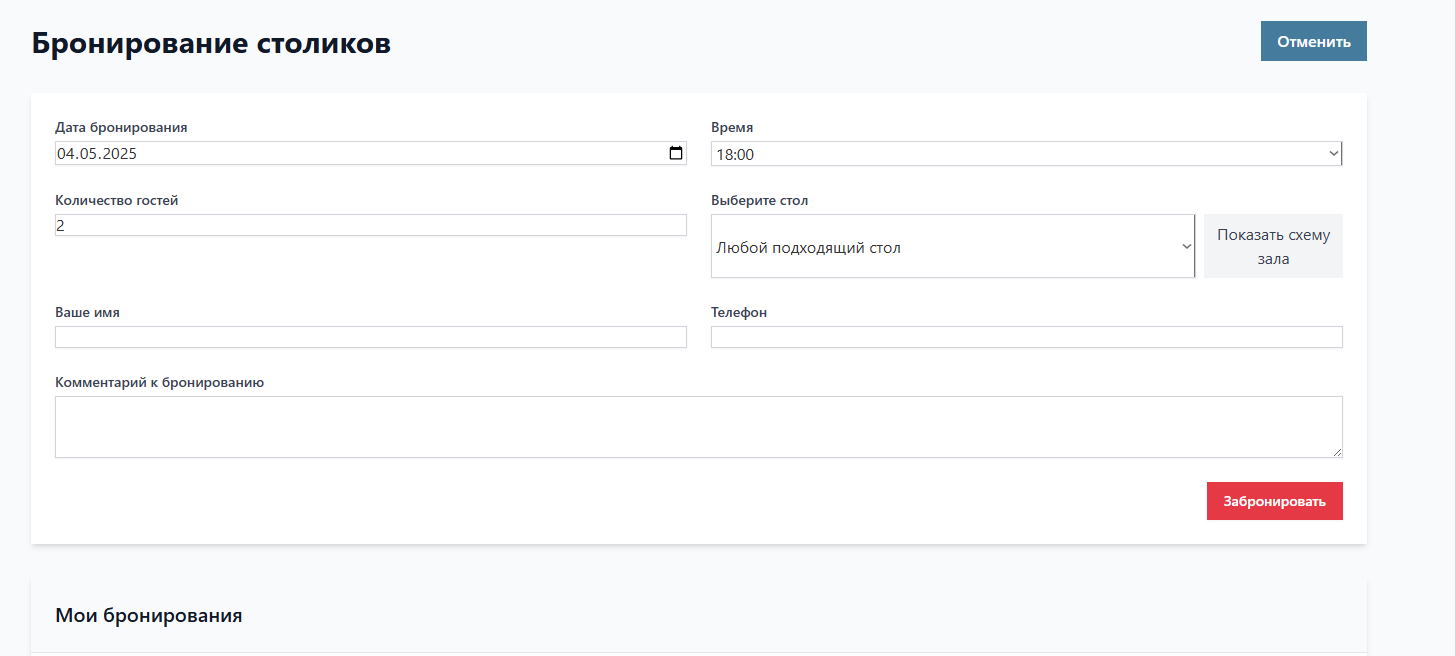


Рисунок 2.8 – Интерфейс страницы бронирования столика клиента

Страница корзины является важным связующим элементом между выбором блюд и оформлением заказа. Её функциональность направлена на предоставление пользователю удобного интерфейса для контроля и редактирования своего будущего заказа до его подтверждения.

На экране корзины отображаются все блюда, добавленные пользователем. Каждая позиция включает изображение, наименование, цену за единицу, количество и итоговую стоимость. Пользователь может изменить количество порций с помощью специальных кнопок или удалить ненужные блюда одним кликом.

Общая сумма заказа рассчитывается автоматически в нижней части экрана.

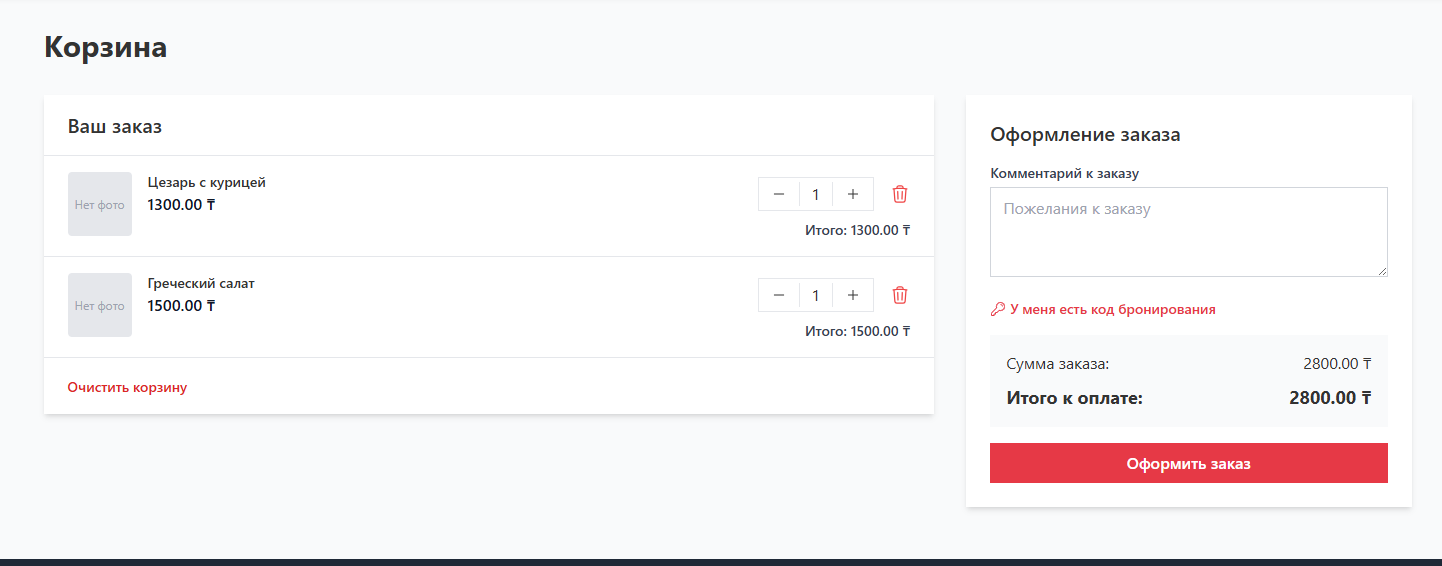
Реализовано с использованием локального хранилища и синхронизации с серверной частью при наличии активной сессии.

Кроме того, реализована валидация содержимого: если блюдо удалено или временно недоступно, пользователь получает уведомление и предложение заменить его или удалить. Это снижает количество ошибок при оформлении и повышает доверие к системе.

В случае, если пользователь пытается перейти к оформлению без авторизации, ему предлагается авторизоваться или зарегистрироваться, чтобы продолжить оформление.

Также предусмотрен визуальный индикатор прогресса — пользователь видит, что он находится на этапе «Корзина» перед следующим шагом «Оформление». Это способствует логическому восприятию пользовательского пути.

На рисунке 2.9 представлен интерфейс страницы корзины клиента.



**Рисунок 2.9 – Интерфейс страницы корзины клиента**

Страница оформления заказа служит заключительным этапом взаимодействия клиента с системой до момента отправки заказа в работу. Её основной задачей является сбор и подтверждение всей информации, необходимой для обработки заказа и его дальнейшей передачи на кухню и к официантам.

На странице отображается сводка текущего содержимого корзины — список блюд с указанием количества, ценой и итоговой суммой. Все данные доступны для финальной проверки перед оформлением. При необходимости пользователь может вернуться назад для редактирования содержимого корзины.

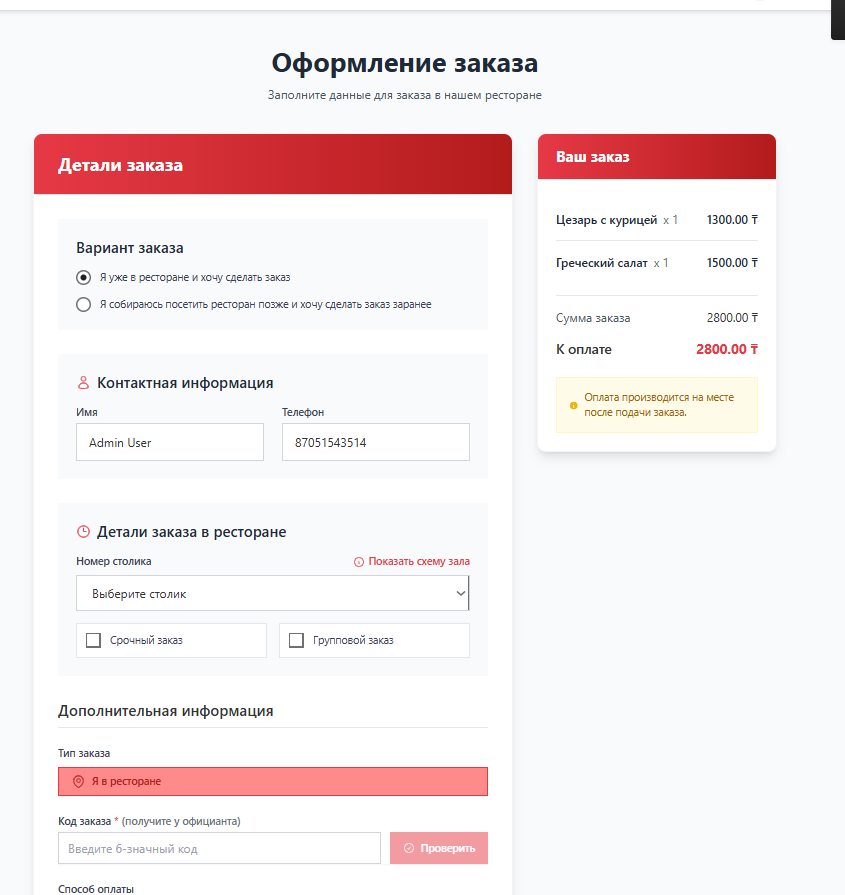
Ниже расположена форма ввода персональных данных: имя и номер телефона. Для удобства реализована автоматическая подстановка ранее введённых данных, если пользователь ранее оформлял заказы с авторизованного аккаунта.

Особое внимание уделено элементам валидации: поля, обязательные для заполнения, выделяются визуально. При попытке отправить форму с незаполненными обязательными полями система подсказывает пользователю, где именно необходимо внести данные. Это позволяет избежать ошибок и упрощает взаимодействие.

После заполнения формы пользователь нажимает кнопку «Оформить заказ». Заказ отправляется на сервер, сохраняется в базе данных со статусом «ожидает подтверждения» и становится доступным для дальнейшей обработки на стороне персонала. Клиенту отображается экран подтверждения с номером заказа и приблизительным временем ожидания.

Важной особенностью реализации является использование асинхронной отправки данных (AJAX), что позволяет избежать полной перезагрузки страницы и обеспечивает быструю реакцию интерфейса.[6]

На рисунке 2.10 представлен интерфейс страницы оформления заказа клиента.



**Рисунок 2.10 – Интерфейс страницы оформления заказа клиента**

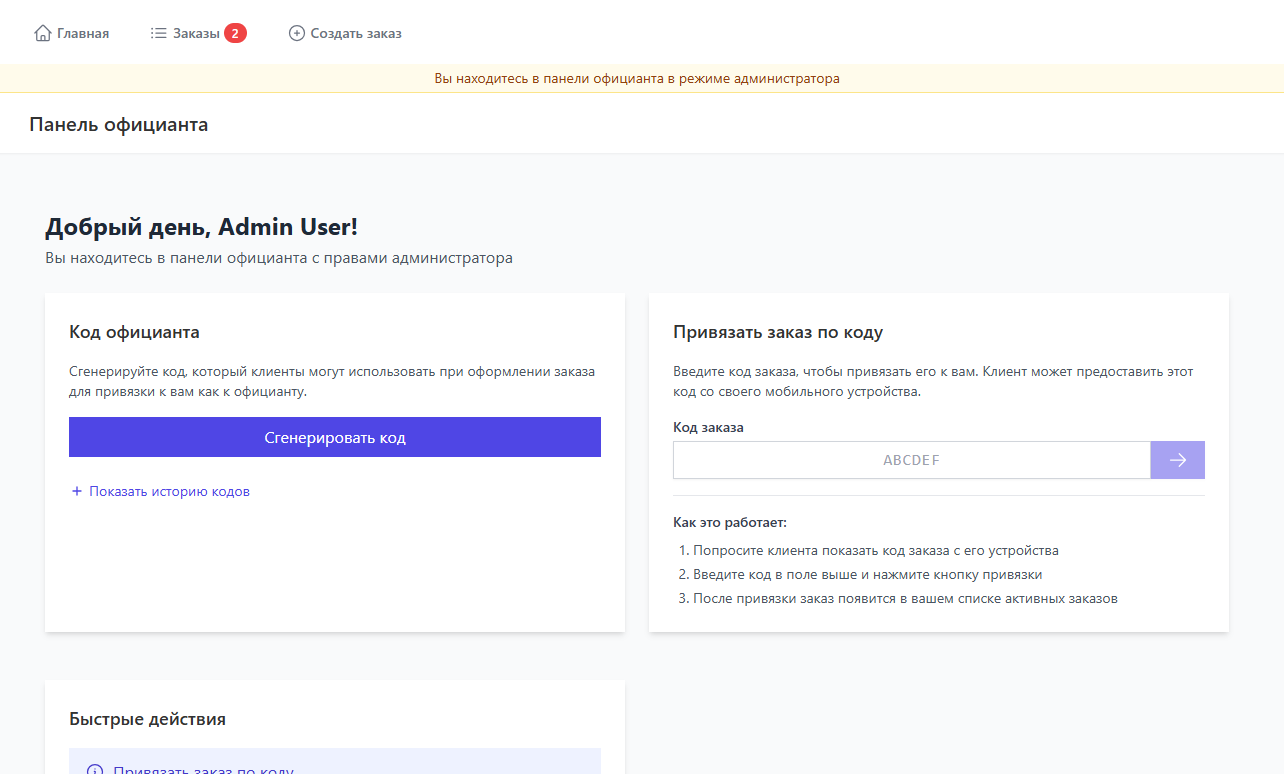
Главная страница панели официанта — это основной рабочий интерфейс, с которого начинается взаимодействие персонала с системой. Её ключевая функция — создание связей между клиентом и официантом, а также формирование начального заказа в зале ресторана.

Первичный сценарий использования начинается с генерации уникального кода официантом. Этот код передаётся клиенту, и он может ввести его при оформлении заказа с личного устройства. Таким образом, создаётся связь между заказом клиента и конкретным официантом, обслуживающим стол.

Это решение позволяет автоматизировать процесс идентификации официанта, а также упрощает последующее распределение заказов по исполнителям. Визуально на странице отображается кнопка «Создать код» — при нажатии система формирует уникальную последовательность символов, которая моментально становится активной.

Также на главной странице официант видит список всех заказов, которые уже привязаны к нему по коду. Этот список обновляется в реальном времени и содержит сведения о статусе каждого заказа: «в процессе оформления», «ожидает подтверждения», «на кухне», «готов к подаче» и т. д.

На рисунке 2.11 представлен главная страница панели официанта.



**Рисунок 2.11 – Главная страница панели официанта**

Страница создания заказа является ключевым инструментом работы официант. Она позволяет оперативно формировать и отправлять заказы на кухню, учитывая пожелания клиента и внутренние процессы обслуживания.

Интерфейс страницы условно разделён на две части: список доступных блюд (меню) и состав текущего заказа. В левой части расположен список всех блюд, сгруппированных по категориям: закуски, горячее, напитки, десерты и т.д. Официант может быстро находить нужные позиции с помощью встроенного поиска и фильтрации по категориям.

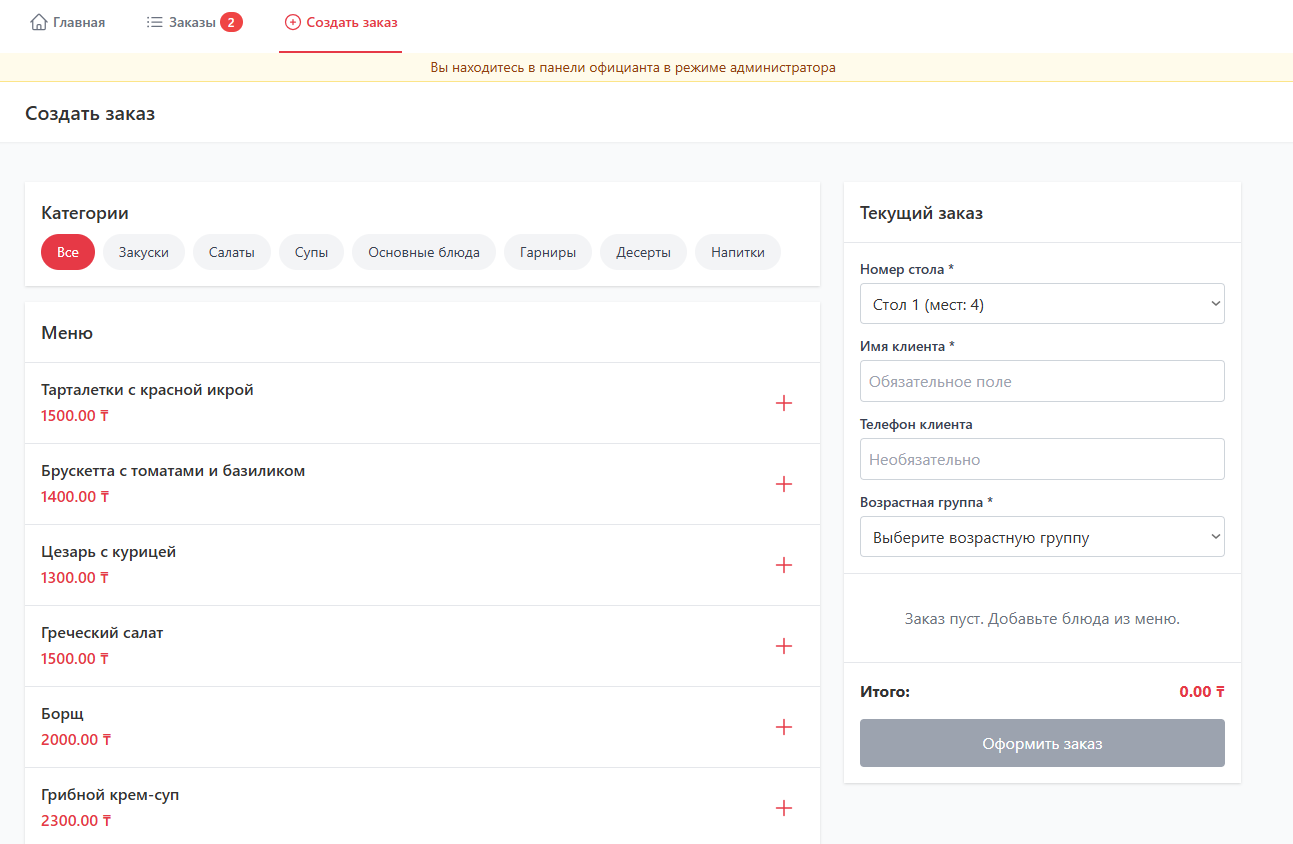
При выборе блюда оно добавляется в правую часть — текущий заказ. Каждое добавленное блюдо сопровождается возможностью указания количества порций и добавления комментариев, например: «без лука», «обжарить сильнее», «отдельно соус». Эти комментарии будут автоматически переданы на кухню и отобразятся поварам в интерфейсе обработки заказов. Также официант во время заказа должен определить возрастную группу. Это нужно будет для аналитики клиентской базы.

Также предусмотрена возможность удалить блюда из заказа или изменить количество уже добавленных позиций. В нижней части отображается итоговая стоимость заказа, автоматически пересчитываемая при каждом изменении. После завершения формирования заказа официант нажимает кнопку «Подтвердить» — заказ сохраняется в базе данных и переходит в статус «ожидает обработки».

Каждому заказу присваивается уникальный номер, по которому его можно отслеживать в интерфейсе. Кроме того, если официант обслуживает сразу несколько столов, система позволяет формировать и сохранять несколько заказов параллельно.

Это особенно важно в условиях высокой нагрузки, так как минимизирует вероятность ошибки и обеспечивает логическую разгрузку интерфейса.

На рисунке 2.12 представлен интерфейс создания заказа в панели официанта.



**Рисунок 2.12 – Интерфейс создания заказа в панели официанта**

Страница с подробной информацией о заказе является критически важным элементом интерфейса панели официанта. Она позволяет сотруднику оперативно отслеживать текущий статус заказа, взаимодействовать с кухней, а также учитывать индивидуальные пожелания клиента.

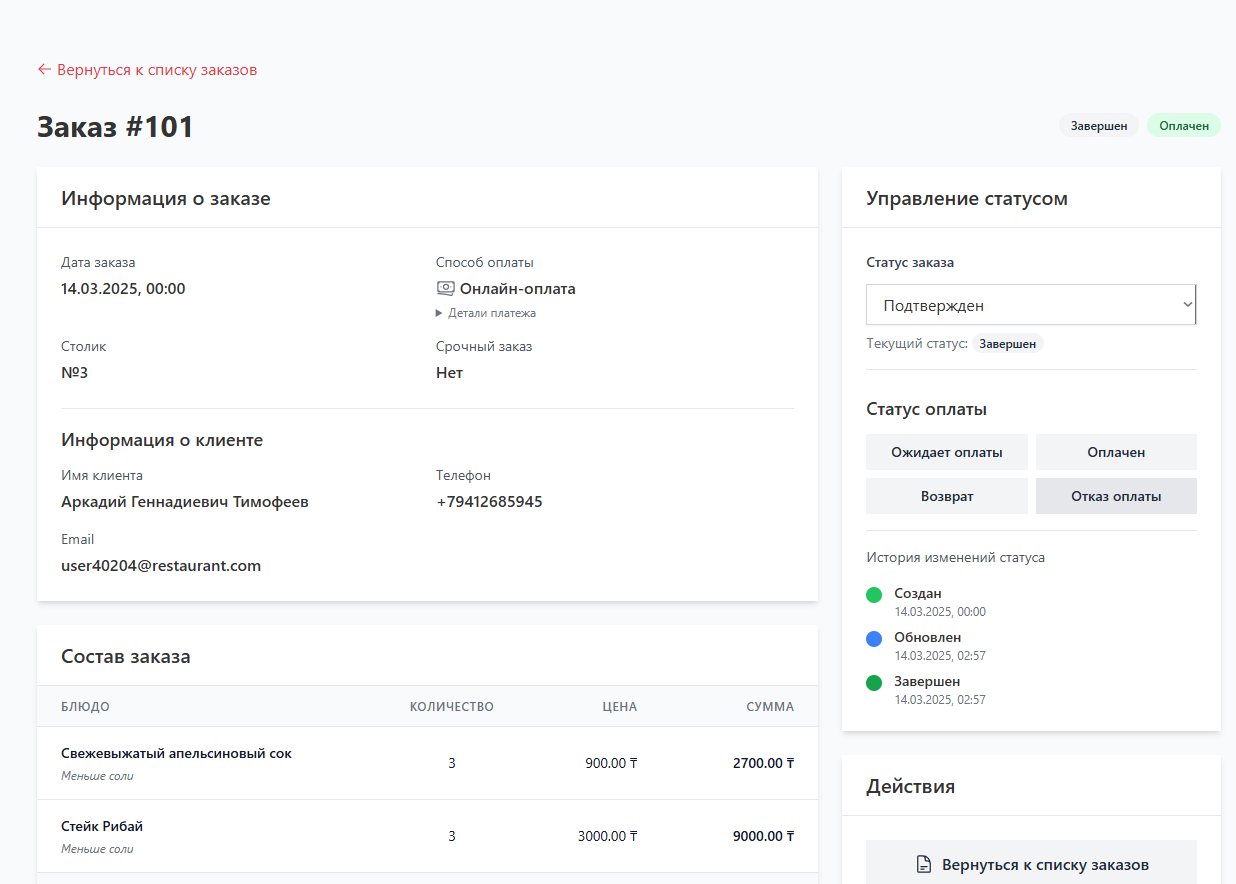
После выбора заказа из общего списка, официант переходит на экран, где отображается вся информация по конкретному заказу. В верхней части представлена общая сводка: уникальный номер заказа, имя клиента, время создания заказа и его текущий статус.

Ниже располагается список всех блюд, включённых в заказ. Каждое блюдо сопровождается данными о количестве, комментариях клиента. Комментарии, указанные при создании заказа (например, «без соли», «двойная порция», «не добавлять сыр»), отображаются сразу под наименованием блюда и выделяются визуально, чтобы избежать ошибок на этапе исполнения.

В боковой части интерфейса предусмотрены интерактивные кнопки управления, позволяющие официанту изменять статус заказа (например, «принят», «готовится», «готов», «подаётся») и фиксировать факт оплаты. При нажатии на соответствующую кнопку система обновляет статус заказа в базе данных и отображает изменения в реальном времени как для официанта, так и для других ролей, таких как администратор и кухня. Это обеспечивает прозрачность процессов и ускоряет обслуживание.

Также отображается общая сумма заказа и доступна кнопка «Печать», с помощью которой официант может распечатать заказ для передачи на кухню, если не используется автоматическая система передачи.

На рисунке 2.13 представлен интерфейс подробностей заказа в панели официанта



**Рисунок 2.13 – Интерфейс подробностей заказа в панели официанта**

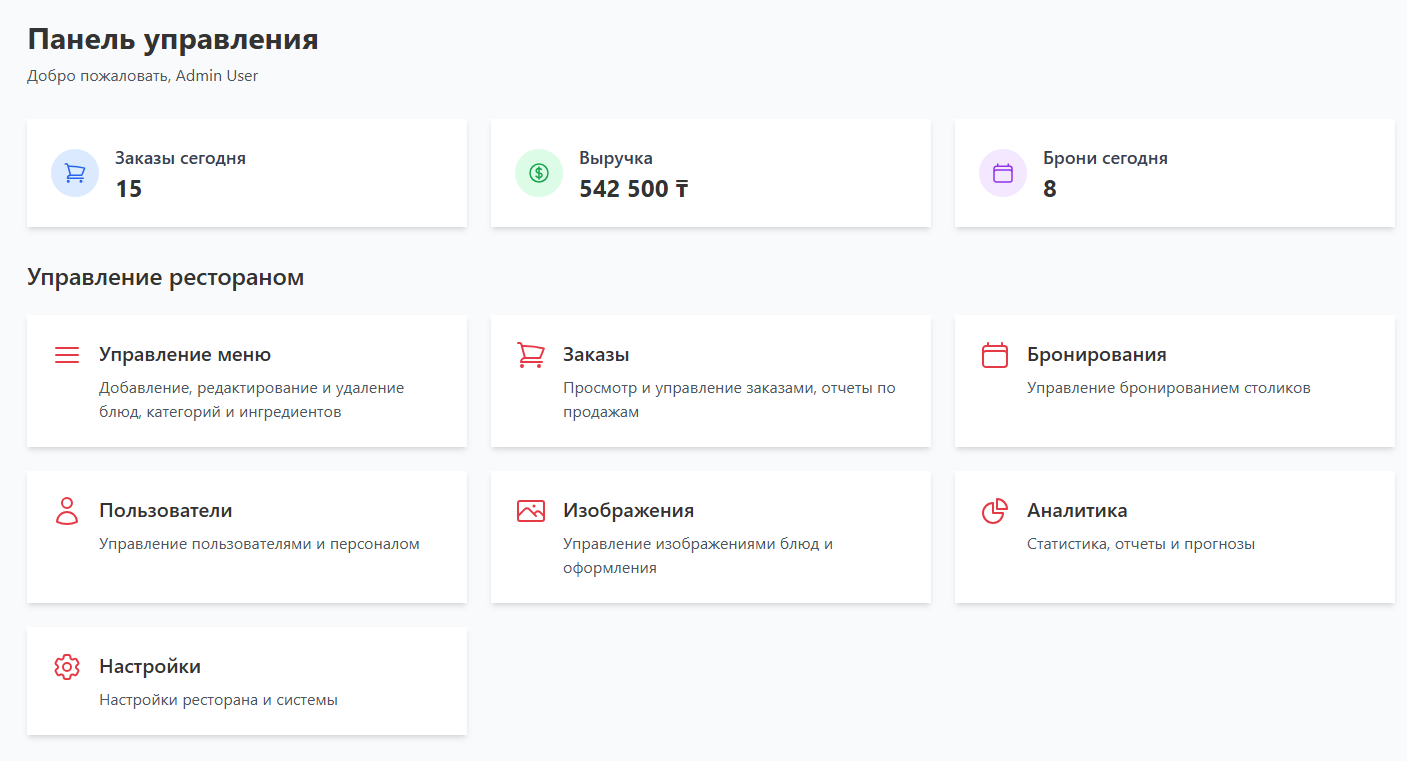
Главная страница панели администратора представляет собой центральную точку входа для управления всей системой. Здесь администратор получает доступ ко всем основным модулям и может отслеживать общее состояние работы ресторана.

В верхней части интерфейса отображаются сводные показатели: количество активных заказов, общая выручка за день и количество броней за день. Эти данные обновляются в режиме реального времени и позволяют администратору оперативно принимать управленческие решения.

Боковая панель навигации предоставляет доступ к следующим разделам: «Управление меню», «Бронирования», «Заказы», «Финансы», «Аналитика», «Пользователи». Каждый из этих пунктов ведёт на отдельную страницу, содержащую специализированные инструменты для настройки и контроля соответствующего раздела.

Для повышения эффективности работы предусмотрен быстрый доступ к панели редактирования профиля, настройкам системы и логам активности. Интерфейс выполнен в строгом, лаконичном стиле с акцентом на читаемость и функциональность.

На рисунке 2.14 представлен интерфейс главной страницы панели администратора.



**Рисунок 2.14 – Интерфейс главной страницы панели администратора**

Раздел «Управление меню» предназначен для администраторов ресторана и предоставляет им полный контроль над перечнем блюд, доступных для заказа клиентами через интерфейс.

На странице отображается таблица со всеми текущими блюдами. Каждое блюдо представлено в виде строки, содержащей название, категорию, цену, наличие (в наличии / временно недоступно), а также кнопки редактирования и удаления. Таблица поддерживает постраничную навигацию и поиск по ключевым словам, что особенно удобно при большом количестве позиций в меню.

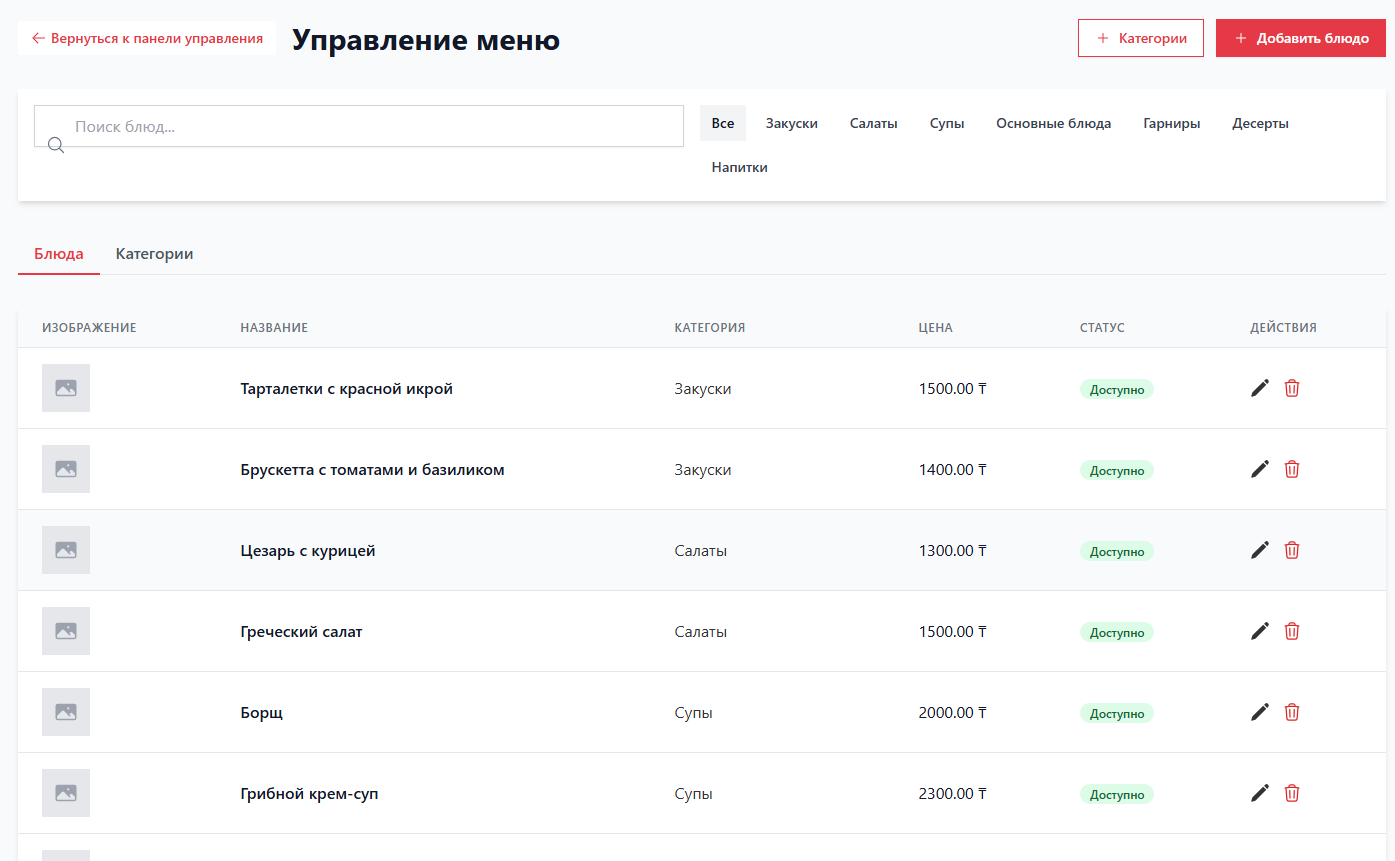
В верхней части интерфейса расположена кнопка «Добавить блюдо», при нажатии на которую открывается форма для создания новой позиции. В форме указывается: – название блюда, – категория (например, закуски, горячее, десерты, напитки и т.д.), – цена, – краткое описание, – изображение (загружается через встроенный загрузчик), – статус «в наличии» / «не в наличии».

После сохранения нового блюда оно автоматически появляется в общем списке и становится доступным для выбора клиентами и официантами. Аналогично, при нажатии на кнопку «Редактировать» рядом с конкретным блюдом, открывается форма редактирования с уже заполненными полями, где можно изменить любые данные.

Удаление блюда также осуществляется через соответствующую кнопку. При этом система запрашивает подтверждение, чтобы исключить случайное удаление. Удалённые блюда автоматически исключаются из клиентского интерфейса, но сохраняются в архиве, чтобы при необходимости можно восстановить.

Дополнительно, система поддерживает отображение блюд по категориям — администратор может быстро переключаться между ними, используя фильтры, размещённые над таблицей. Это особенно удобно при работе с сезонными предложениями или временным обновлением меню.

На рисунке 2.15 представлен интерфейс управления меню в панели администратора.



**Рисунок 2.15 – Интерфейс управления меню в панели администратора**

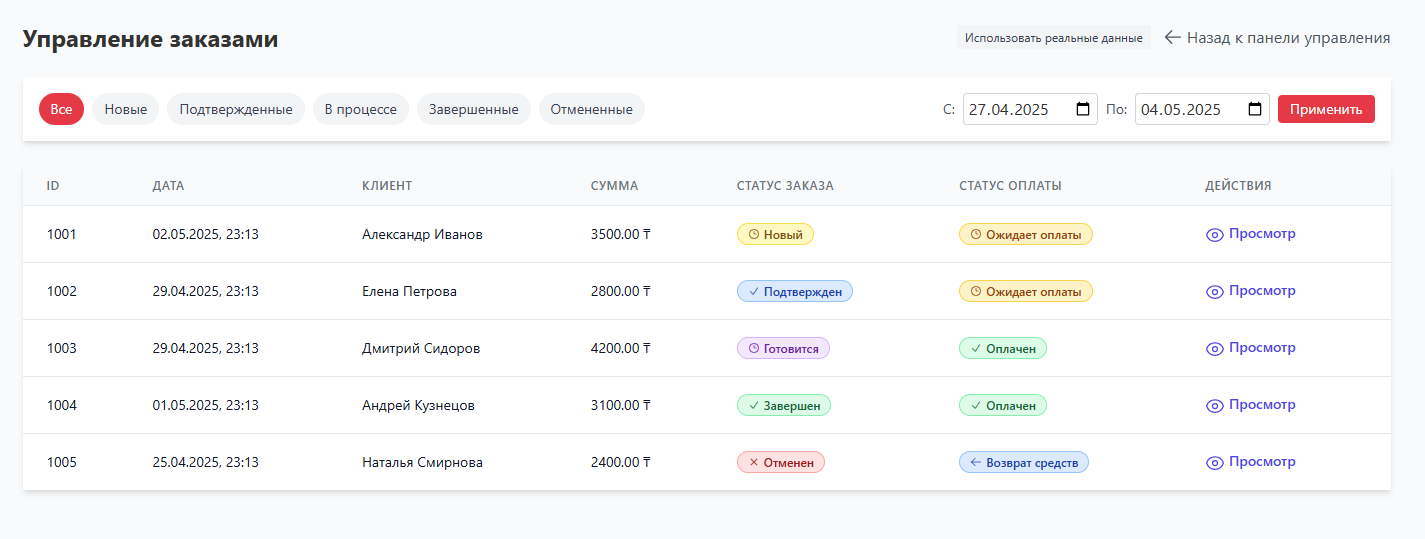
Раздел «Управление заказами» позволяет администратору в реальном времени контролировать все активные, завершённые и отменённые заказы. Этот интерфейс предоставляет полную картину происходящего в зале и на кухне, что критически важно для обеспечения стабильной работы ресторана.

Основной элемент страницы — таблица заказов, в которой отображаются ключевые параметры: номер заказа, имя клиента (если задано), дата и время создания, текущий статус (например, «в обработке», «готовится», «подан»), сумма заказа, ответственный официант и используемый столик. Таблица поддерживает фильтрацию по статусу и интервалу времени, а также поиск по ключевым словам или номеру заказа.

Каждую строку таблицы можно развернуть, чтобы просмотреть подробности: список заказанных блюд, их количество, комментарии клиента, историю изменений и статус оплаты. Это позволяет администратору проводить анализ по каждому заказу, выявлять отклонения и контролировать работу персонала.

Особое внимание уделено автоматическому обновлению данных. Интерфейс поддерживает автообновление, чтобы администратор всегда видел самые актуальные сведения без необходимости вручную перезагружать страницу. Это позволяет немедленно реагировать на проблемы, сбои или задержки.

На рисунке 2.16 представлен интерфейс управления заказами в панели администратора.



**Рисунок 2.16 – Интерфейс управления заказами в панели администратора**

Раздел «Управление бронями» предназначен для отслеживания и контроля всех заявок на бронирование столиков в ресторане. Этот модуль предоставляет администратору возможность в реальном времени просматривать все активные, завершённые и отменённые бронирования, а также управлять ими.

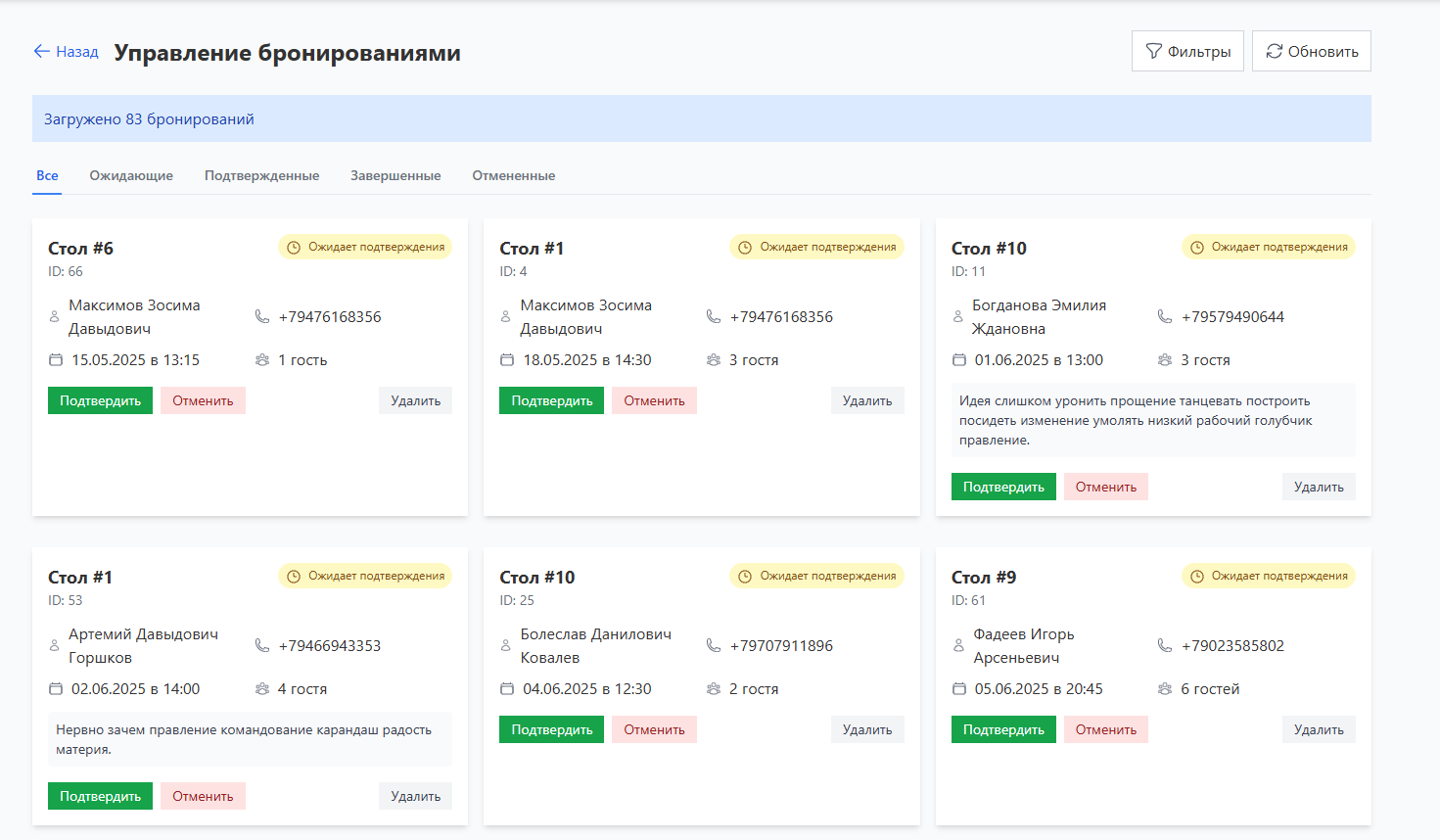
Интерфейс состоит из таблицы, в которой отображаются следующие параметры: имя клиента, дата и время брони, количество гостей, комментарии, статус бронирования («ожидает подтверждения», «подтверждено», «отклонено»), и присвоенный столик (если уже назначен). Также в таблице указывается, какой сотрудник подтвердил или отменил бронь.

Новые бронирования попадают в статус «ожидает подтверждения». Администратор может открыть карточку бронирования, ознакомиться с деталями и принять решение — подтвердить или отклонить заявку. После подтверждения столик закрепляется за клиентом, и запись переходит в соответствующий статус. При отклонении обязательно указывается причина отказа.

Для повышения эффективности предусмотрены фильтры по дате, статусу и имени клиента. Это позволяет быстро находить нужные записи и разгружает интерфейс при большом количестве броней.

Система также отображает уведомления о новых поступивших бронированиях, что помогает оперативно реагировать на входящие заявки.

На рисунке 2.17 представлен интерфейс управления бронями в панели администратора.



**Рисунок 2.17 – Интерфейс управления бронями в панели администратора**

Раздел «СППР – Дашборд» является ключевой частью системы поддержки принятия решений и предоставляет администратору централизованный обзор всех ключевых показателей работы ресторана. Здесь агрегируются и визуализируются данные, необходимые для анализа текущего состояния бизнеса и стратегического управления.

На главной панели отображаются следующие виджеты:

– Выручка за день, неделю и месяц;

– Количество заказов и броней за выбранный период;

– Наиболее популярные блюда;

– Средний чек;

– Количество новых и возвращающихся клиентов.

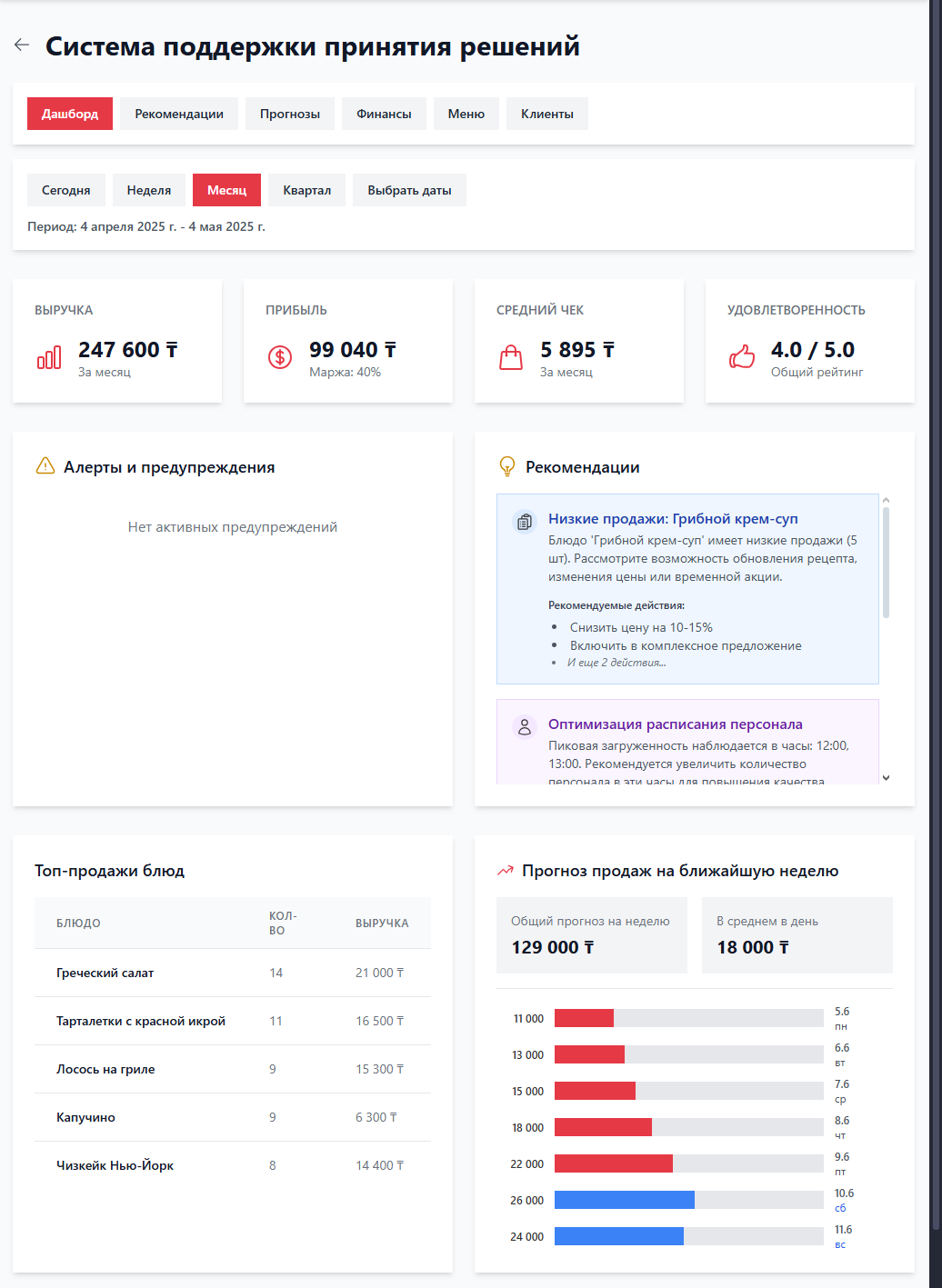
Данные на панели обновляются автоматически и отображаются в виде графиков, диаграмм и таблиц. Это позволяет администратору в один взгляд получить представление о состоянии дел в ресторане и быстро выявить отклонения или тренды.

Существует возможность выбора диапазона времени (день, неделя, месяц, произвольный интервал), что позволяет анализировать изменения во времени.

Кроме того, на дашборде представлены уведомления и рекомендации от модуля аналитики, например: «Снижение заказов в вечернее время — рассмотрите изменение акцентов в меню» или «Блюдо X теряет популярность — рассмотрите замену».

Таким образом, СППР-дашборд выполняет роль навигационного центра аналитики и позволяет принимать обоснованные управленческие решения на основе объективных данных.

На рисунке 2.18 представлен интерфейс раздела СППР – Дашборд.



**Рисунок 2.18 – Интерфейс раздела СППР – Дашборд**

Раздел «СППР – Рекомендации» предоставляет интеллектуальные советы по управлению рестораном на основе анализа накопленных данных. Эти рекомендации автоматически генерируются системой поддержки принятия решений на базе алгоритмов оценки поведения клиентов, динамики заказов и других статистических данных.

Основная цель раздела — помочь администратору повысить эффективность работы заведения, снизить издержки и увеличить выручку. Рекомендации сгруппированы по нескольким категориям:

– рекомендации по меню (удаление непопулярных блюд, добавление новых позиций на основе спроса);

– рекомендации по графику сотрудников (увеличение или снижение смен в зависимости от загруженности);

– рекомендации по посадке и бронированию столов (перераспределение бронирований для оптимального использования залов);

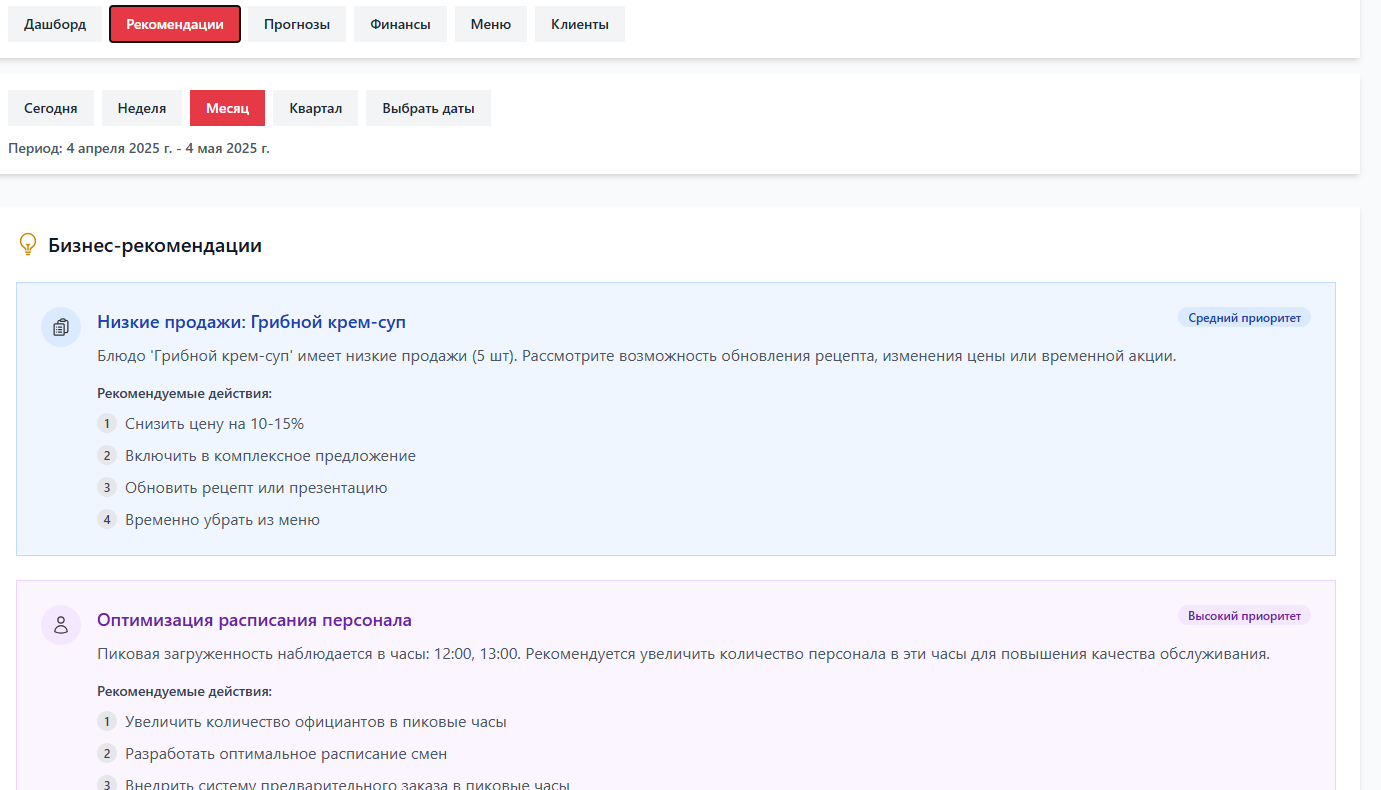
– рекомендации по времени проведения акций (предложения оптимального времени на основе спада заказов).

Каждая рекомендация сопровождается обоснованием: например, «Блюдо X имеет менее 3 заказов в неделю — рекомендуется удалить или заменить», либо «С понедельника по среду после 15:00 наблюдается спад заказов — рассмотрите введение спецпредложений в это время».

Интерфейс позволяет администратору просматривать рекомендации по дате, категории и степени приоритета. Также существует возможность отклонения или принятия рекомендаций, после чего система отслеживает их влияние на показатели.

Таким образом, данный раздел выступает не просто в роли отчётного инструмента, но как полноценный помощник в управлении рестораном, позволяя своевременно реагировать на изменения и оптимизировать бизнес-процессы.

На рисунке 2.19 представлен интерфейс раздела СППР - Рекомендации



**Рисунок 2.19 – Интерфейс раздела СППР – Рекомендации**

Раздел «СППР – Прогнозы» является ключевым элементом системы поддержки принятия решений, предоставляющим инструменты для анализа и прогнозирования бизнес-показателей. Он позволяет не только оценить текущие данные, но и строить предположения о будущих значениях на основе исторической информации.

Пользователь может выбрать интересующую метрику (выручка, загрузка, спрос на блюда), временной диапазон и дополнительные параметры. Все результаты визуализируются в виде графиков, диаграмм и таблиц.

Прогнозирование осуществляется с помощью алгоритмов машинного обучения и анализа временных рядов, что позволяет учитывать. Модели обучаются на данных ресторана, что повышает точность результатов.

Основной функционал раздела:

- Прогноз выручки: расчет ожидаемой выручки с учетом динамики, трендов и сезонности;

- Прогноз загруженности: оценка количества гостей по дням и часам для планирования смен и подготовки зала;

- Моделирование сценариев: анализ последствий управленческих решений через функции «что, если»;

- Система автоматически обновляет данные прогнозов. Результаты можно экспортировать для дальнейшего анализа.

Таким образом, «СППР – Прогнозы» является мощным инструментом для принятия стратегических решений и повышения эффективности управления рестораном.

На рисунке 2.20 представлен интерфейс раздела СППР-Прогнозы

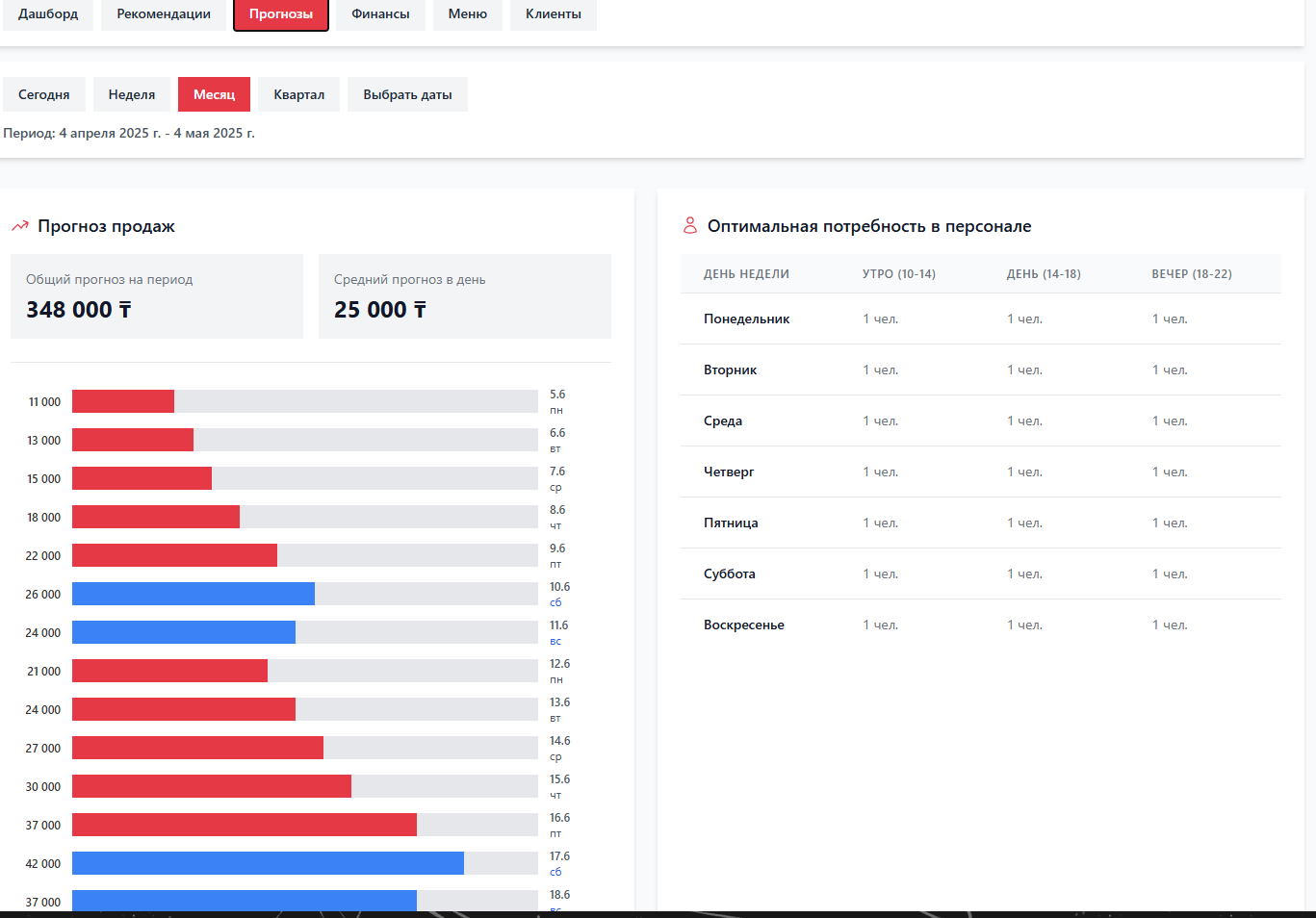


Рисунок 2.20 – Интерфейс раздела СППР – Прогнозы

Раздел «СППР – Финансы» предназначен для анализа финансовой активности ресторана и предоставления пользователю структурированных данных о доходах, расходах и ключевых финансовых метриках. Этот раздел помогает принимать обоснованные управленческие решения, связанные с оптимизацией затрат, повышением прибыльности и прогнозированием будущих финансовых результатов.

Интерфейс раздела состоит из блоков визуализации и графиков. Пользователь может выбирать временные диапазоны для сравнения текущих значений с аналогичными периодами в прошлом.

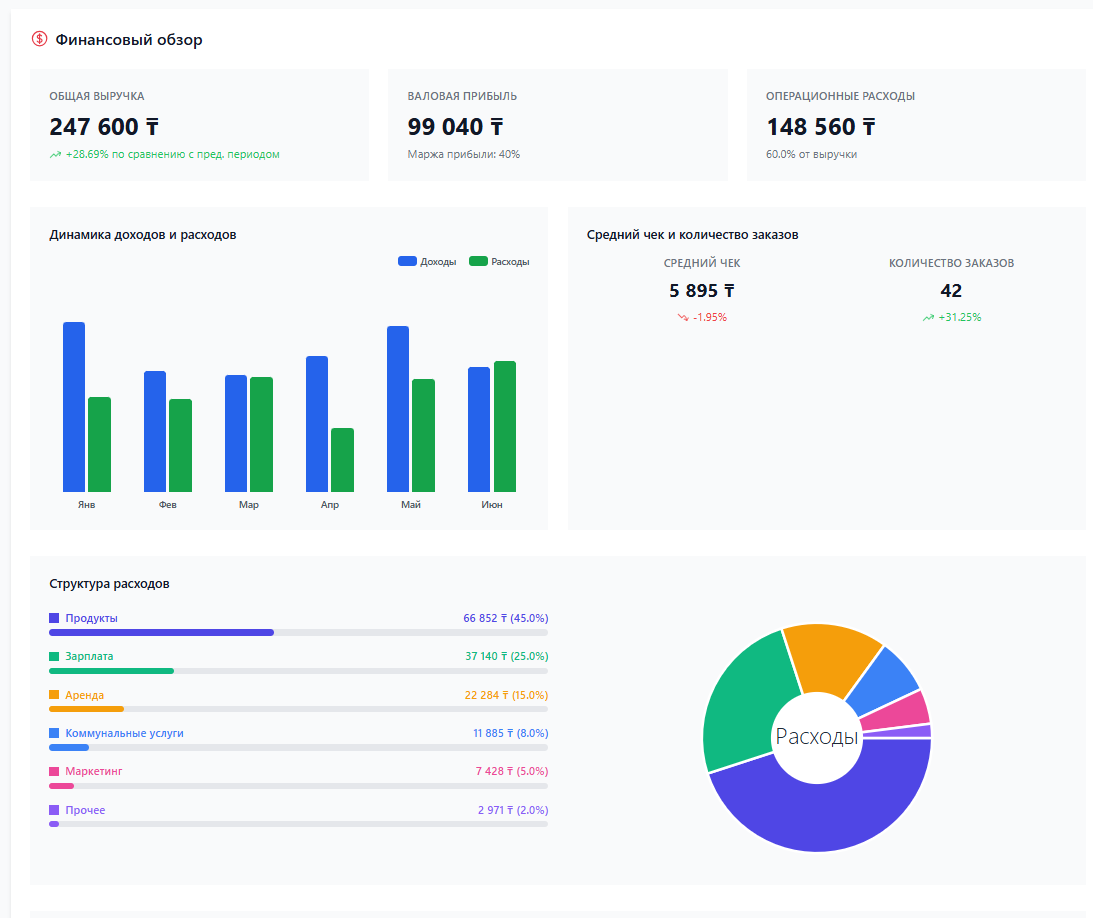
**Ключевые возможности раздела включают:**

1. Графики и таблицы по ежедневной, недельной и месячной выручке
2. Анализ затрат по категориям (продукты, персонал, аренда и т.д.).
3. Расчет валовой и чистой прибыли.
4. Симуляции финансовых сценариев на основе изменений в ценах, посещаемости или составе меню.

Раздел содержит также аналитическую панель с ключевыми метриками: средний чек, прибыль на клиента, уровень затрат по сегментам и соотношение доходов к расходам.

Раздел «СППР – Финансы» предоставляет управленцам инструменты для отслеживания финансовой устойчивости ресторана и выявления проблемных зон. Своевременный доступ к этим данным помогает улучшать рентабельность бизнеса и адаптироваться к изменениям рыночных условий.

На рисунке 2.21 представлен интерфейс раздела СППР – Финансы.



**Рисунок 2.21 – Интерфейс раздела СППР – Финансы**

Раздел «СППР – Меню» предоставляет пользователю набор аналитических инструментов, предназначенных для оценки эффективности позиций меню и принятия обоснованных решений по его оптимизации. Используя статистику заказов и данные об обороте, система помогает выявить наиболее и наименее популярные блюда, определить маржинальность позиций и составить рекомендации по их актуализации.

Интерфейс страницы содержит несколько ключевых блоков: таблица с блюдами и их параметрами, визуализация в виде матрицы ABC-анализа, а также фильтры для выбора периода анализа.

Основной функционал включает:

- Анализ популярности: отображение количества заказов по каждому блюду, выявление лидеров и аутсайдеров по продажам.

- Финансовые метрики: расчёт выручки, прибыли и средней цены по каждой позиции меню.

- Фильтрация и сравнение: возможность отфильтровать данные по категориям (например, напитки, закуски, горячие блюда) и сравнивать показатели в разные периоды.

- Рекомендации по меню: система предлагает убрать или переосмыслить неэффективные блюда, повысить цены на самые популярные или ввести аналогичные продукты.

На рисунке 2.22 представлен интерфейс раздела СППР – Меню.

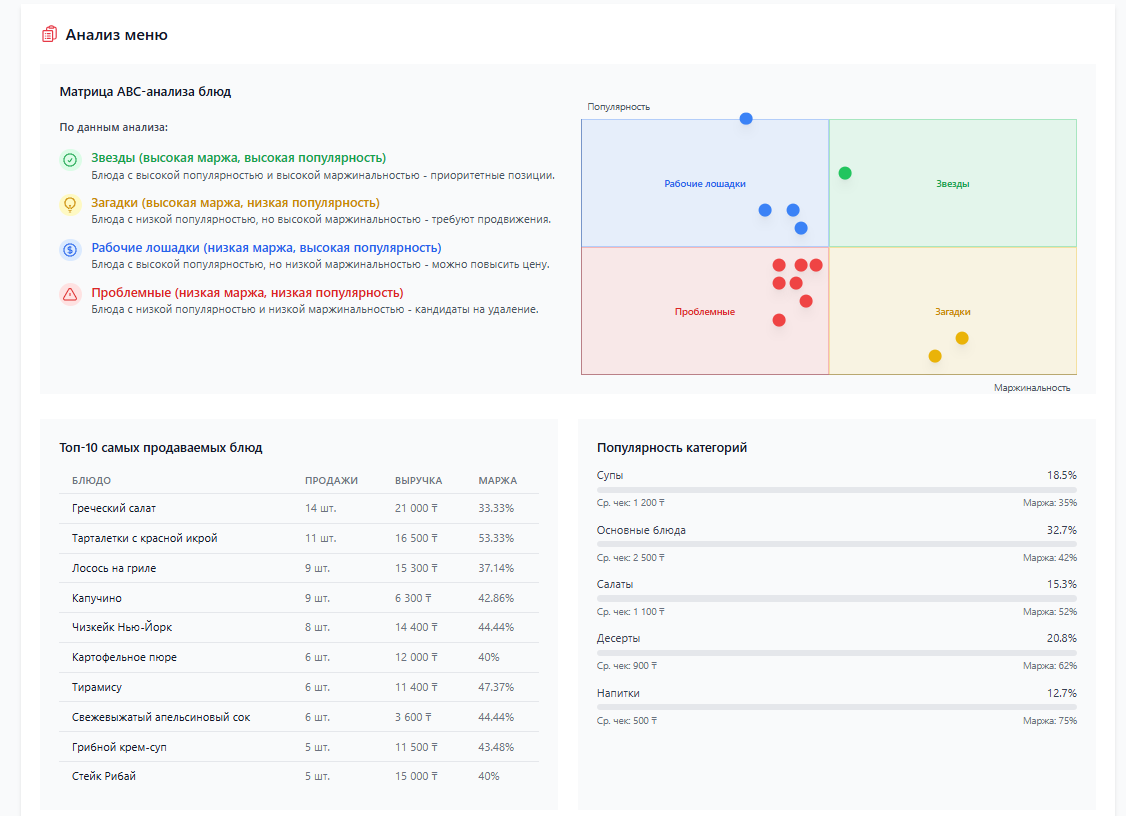


Рисунок 2.22 – Интерфейс раздела СППР – Меню

Раздел «СППР – Анализ клиентской базы» предоставляет администраторам и маркетологам инструменты для изучения поведения клиентов, оценки лояльности и выявления ключевых пользовательских сегментов. Этот раздел помогает принимать решения по улучшению сервиса, персонализации предложений и выстраиванию стратегий удержания гостей.

Интерфейс раздела содержит дашборд с основными метриками, таблицы с данными по клиентам, а также визуализации в виде графиков и диаграмм, отражающих поведенческие и статистические характеристики.

Основной функционал включает:

- Сегментация клиентов: распределение гостей по категориям на основе частоты визитов и объема заказов;

- Оценка лояльности: расчёт метрик возвратности, среднее количество посещений на клиента, доля повторных заказов;

- Анализ отзывов: если в системе реализован механизм сбора обратной связи, раздел позволяет анализировать оценки и отзывы клиентов, выявляя проблемные точки сервиса;

- Динамика аудитории: визуализация роста или снижения клиентской базы, выявление закономерностей и сезонных колебаний.

Пользователю доступна фильтрация данных по временным отрезкам, ролям клиентов и другим параметрам. Также реализована возможность экспорта данных в CSV или Excel для более глубокой работы вне системы.

Этот раздел даёт возможность системно подходить к пониманию аудитории заведения и принимать решения, направленные на повышение лояльности, улучшение пользовательского опыта и увеличение прибыли.

На рисунке 2.23 представлен интерфейс раздела СППР – Анализ клиентской базы.

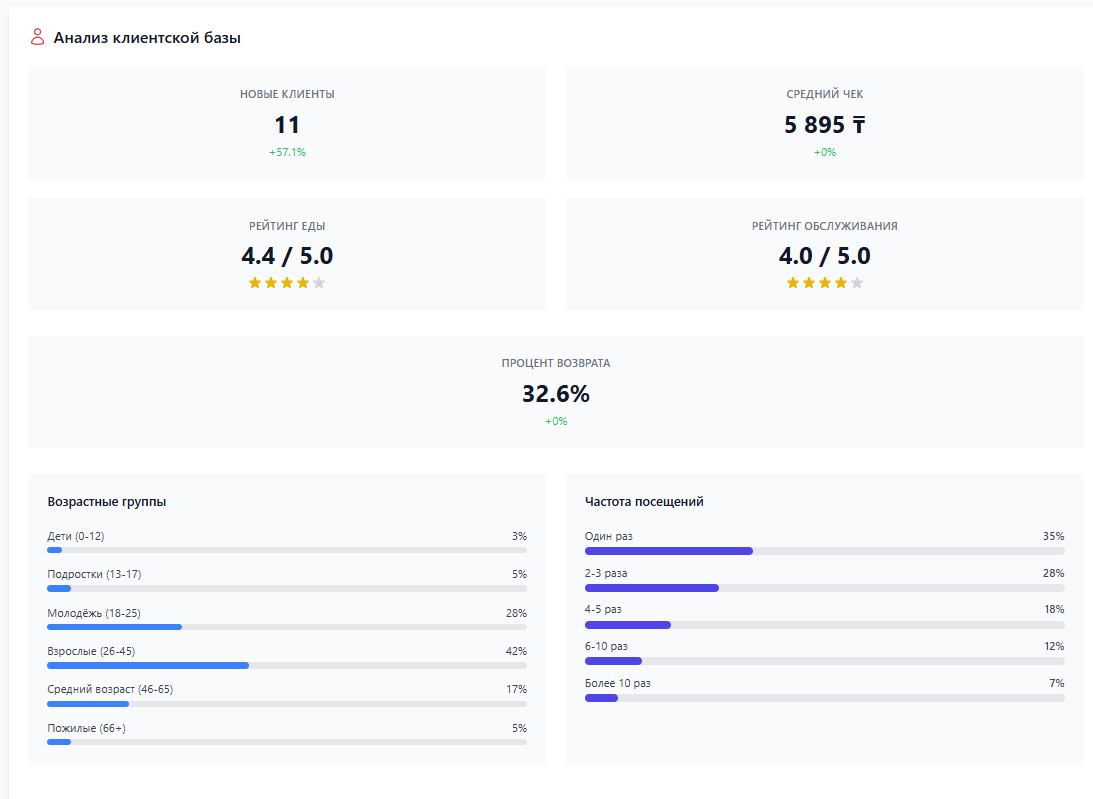


Рисунок 2.23 – Интерфейс раздела СППР – Анализ клиентской базы

## 2.5 Функции и алгоритмы системы

Разработка и внедрение интеллектуальных функций в систему поддержки принятия решений (СППР) играет ключевую роль в обеспечении точного анализа, прогноза и автоматизации управления ресторанным бизнесом. Одним из важнейших направлений программной реализации системы является модуль прогнозирования, построенный на базе алгоритмов машинного обучения и анализа временных рядов. Этот модуль реализован в разделе СППР – Прогнозы, доступном через административную панель системы.[28]

Функция прогнозирования выполняет важнейшие аналитические задачи: предсказание выручки по дням недели и часам работы ресторана, оценка предполагаемой загрузки по количеству посетителей, выявление циклических и сезонных паттернов поведения клиентов, а также моделирование поведения посетителей в зависимости от внешних факторов (праздники, погодные условия, дни недели и т.п.).

Прогнозная система базируется на накопленных исторических данных, поступающих из базы заказов, посещаемости и финансовых транзакций. Алгоритмы обучаются на этих данных, формируя модели, которые позволяют предсказывать ключевые метрики на будущие периоды. Это дает возможность администраторам и управляющим рестораном принимать обоснованные решения о количестве персонала на смене, запасах продуктов и степени подготовки кухни и зала.

Этапы работы алгоритма прогнозирования

1. **Сбор исторических данных:**

- Извлечение информации из таблиц заказов, посещаемости и финансовой отчетности.

- Группировка данных по дням недели, часам, месяцам и иным признакам.

1. **Очистка и предобработка данных:**

- Удаление аномалий, пропущенных значений, некорректных записей.

- Преобразование дат в необходимые временные интервалы.

1. **Формирование обучающей выборки:**

- Разделение данных на обучающую и тестовую выборки.

1. **Выбор и обучение модели:**

- Использование алгоритмов: линейная регрессия, XGBoost, Prophet и др.

- Обучение моделей на исторических данных.

1. **Оценка качества модели:**

- Метрики оценки: MAE, RMSE, MAPE.

- Анализ точности прогнозов на тестовой выборке.

1. **Построение прогноза:**

- Прогноз выручки и количества посетителей на будущие периоды.

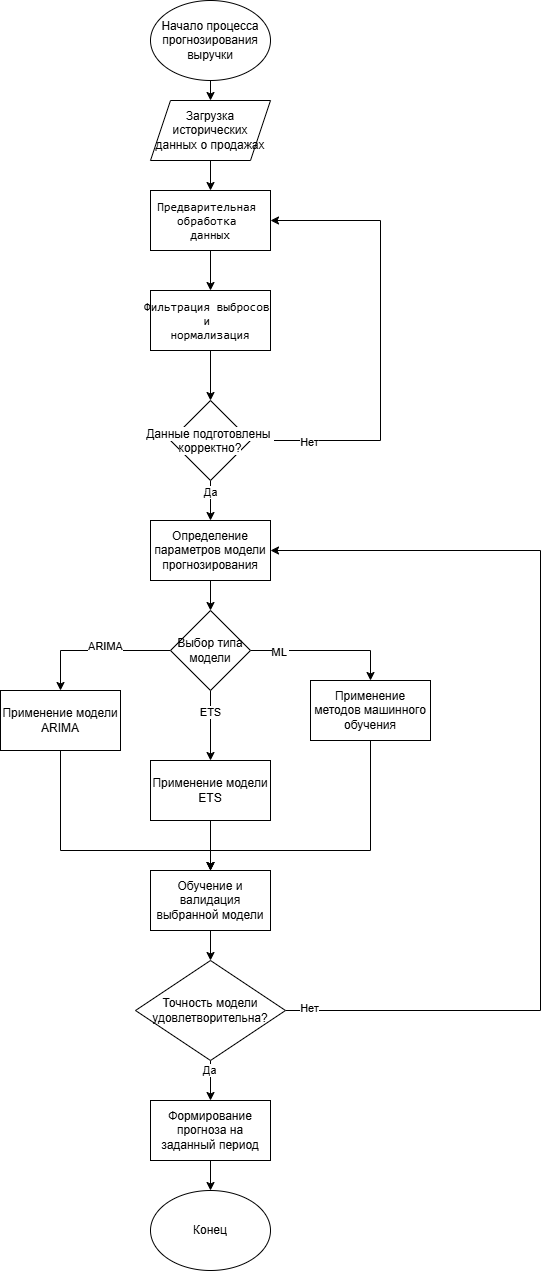
- Отображение результатов в виде графиков и таблиц в панели администратора.

1. **Визуализация и применение:**

- Представление прогноза в разделе СППР – Прогнозы.

- Использование прогнозных данных для формирования рекомендаций по закупкам, персоналу и меню.

Интерактивная схема работы модуля прогнозирования представлена на рисунке 2.24.



**Рисунок 2.24 – Схема алгоритма прогнозирования в СППР**

Таким образом, модуль прогнозирования в рамках СППР предоставляет мощный инструмент для принятия обоснованных решений на основе предиктивной аналитики. Он позволяет заранее реагировать на изменения в поведении клиентов, обеспечивать бесперебойную работу ресторана в пиковые часы, а также сокращать издержки, связанные с неэффективным планированием. Это особенно важно для ресторанов, работающих в условиях высокой сезонности или нестабильного потока клиентов, поскольку позволяет адаптировать внутренние процессы к реальному спросу и повысить устойчивость бизнеса.

Функция аналитики клиентской базы реализована с целью сбора, систематизации и анализа пользовательской активности и данных, оставленных в процессе взаимодействия с системой. Эта аналитика позволяет администратору получать расширенное представление о поведении клиентов, их предпочтениях, паттернах активности, а также помогает выявить проблемные зоны, определить целевые аудитории и принимать обоснованные управленческие решения. Внедрение такого функционала способствует формированию ориентированной на клиента стратегии обслуживания, что повышает уровень удовлетворенности и лояльности.

Основные источники данных, используемые в аналитике:

- данные о бронированиях (частота, количество гостей, среднее время нахождения, отмены);

- информация о заказах (наборы блюд, категории, предпочтения, средний чек);

- поведенческая активность (время взаимодействия с системой, страницы, которые чаще всего посещаются);

- обратная связь клиентов (оценки, отзывы, комментарии);

- история взаимодействий с официантами и обслуживанием.

Функция реализована на базе встроенного аналитического модуля, использующего предобработанные данные из базы данных, которые затем обрабатываются с помощью алгоритмов агрегации и сегментации.

Ключевые функциональные возможности:

- формирование клиентских сегментов по поведению (регулярные посетители, разовые гости, групповые клиенты);

- анализ предпочтений клиентов на уровне категорий и конкретных блюд;

- расчет среднего чека и его динамики в разрезе периодов времени;

- определение наиболее активных временных промежутков посещения ресторана;

- построение тепловых карт по активности и заказам;

- выявление «узких» мест и точек оттока пользователей;

- расчет коэффициента удержания клиентов (retention rate);

- отображение статистики в визуализированном виде: графики, таблицы, диаграммы.

Визуальный интерфейс модуля аналитики клиентской базы позволяет администраторам фильтровать информацию по ролям, датам, источникам заказов, а также экспортировать данные в удобных форматах (CSV, Excel, PDF) для дальнейшего анализа или отчетности. Кроме того, система предоставляет возможность формирования отчетов по заданным шаблонам, автоматизируя процесс отчетности для управляющего персонала.

Схема алгоритма аналитики клиентской базы представлена в соответствии с рисунком 2.25.

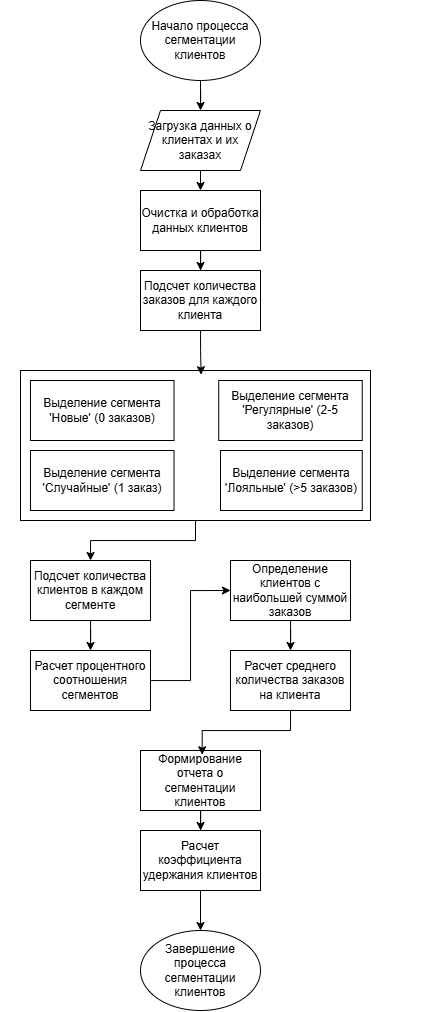


Рисунок 2.25 Схема алгоритма аналитики клиентской базы

Анализ клиентской базы позволяет своевременно реагировать на изменения в поведении аудитории, корректировать маркетинговые стратегии, оптимизировать меню и политику ценообразования. Также он играет важную роль в стратегическом планировании и развитии бизнеса, обеспечивая обоснованные данные для принятия решений.

Финансовая аналитика в системе поддержки принятия решений позволяет детально отслеживать доходы, расходы и общую финансовую эффективность работы ресторана. Алгоритм сбора и обработки финансовых данных реализован таким образом, чтобы предоставлять администраторам полную картину денежного потока в рамках заданного временного интервала, а также выполнять сравнительный анализ между различными периодами.

Финансовый алгоритм работает на основе данных о заказах, оплате, количестве посетителей, себестоимости блюд, налогах, зафиксированных в системе. Он обрабатывает следующие категории информации:

**- Выручка** — суммарная стоимость всех оплаченных заказов, за вычетом отменённых;

**- Себестоимость** — рассчитывается на основе встроенных данных о затратах на ингредиенты каждого блюда;

**- Налоги** — вычисляются согласно установленной налоговой ставке в настройках системы;

**- Прибыль** — формируется как разница между выручкой и себестоимостью с учётом налогообложения;

**- Количество заказов** — число всех завершённых заказов за выбранный период;

**- Средний чек** — рассчитывается как отношение выручки к количеству заказов;

**- Повторные клиенты** — доля заказов, совершённых зарегистрированными пользователями более одного раза.

Алгоритм включает несколько этапов: фильтрацию некорректных транзакций, агрегацию данных по временным отрезкам (день, неделя, месяц), расчёт ключевых показателей и визуализацию результатов в панели аналитики. Пользователю доступны функции выбора периода анализа, экспорта отчётов, а также сравнительный анализ нескольких временных отрезков.

Интерфейс страницы отображает все ключевые метрики в виде карточек, графиков и таблиц, обеспечивая быстрый доступ к финансовым данным. Это позволяет администраторам принимать решения на основе точной и актуальной информации, отслеживать динамику продаж, выявлять слабые и сильные периоды и своевременно реагировать на изменения в работе ресторана.

Процесс оформления и обработки заказа в системе автоматизации ресторана реализован с учетом всех ключевых этапов взаимодействия между клиентом, официантом и системой. Основная цель алгоритма — обеспечить простоту для клиента, оптимизировать работу официанта и гарантировать точность исполнения заказа с полной прозрачностью действий.

Оформление заказа начинается с клиента, который взаимодействует с электронным меню через пользовательский интерфейс. На этой странице отображаются все доступные блюда, отсортированные по категориям. Клиент может просмотреть описание каждого блюда, его стоимость и при необходимости задать количество порций. Выбранные блюда добавляются в корзину.

Следующим шагом клиент переходит в раздел оформления заказа. Здесь необходимо указать основную информацию: контактные данные, количество гостей, желаемый столик (если он не назначен автоматически), а также возможные комментарии к заказу (например, особые пожелания или указания по приготовлению блюда). После подтверждения, заказ сохраняется в системе и получает уникальный идентификатор. Однако он еще не считается активным, поскольку требуется подтверждение от официанта.

Официант авторизуется в системе и через свою панель может привязать заказ, введя или просканировав специальный код, отображенный у клиента. Эта процедура позволяет официанту закрепить заказ за собой, чтобы контролировать дальнейшее выполнение. В интерфейсе официанта отображаются все данные по заказу, включая список блюд, количество персон, номер столика и комментарии. Дополнительно официанту отображается информация о времени оформления заказа и примерном времени его исполнения, что позволяет планировать подачу более эффективно.

После привязки заказа официант может при необходимости изменить его структуру: добавить или удалить блюда, изменить количество порций, скорректировать комментарии. Такая гибкость позволяет оперативно учитывать предпочтения клиента и адаптироваться к изменениям даже после первоначального оформления. Когда заказ окончательно подтвержден, официант передает его на кухню, где повара получают полную информацию о блюдах, времени подачи и индивидуальных пожеланиях. Это также включает приоритет подачи, если заказ оформлен как срочный, или включает специальные блюда, требующие длительного приготовления.

Система фиксирует передачу заказа и присваивает ему статус "в работе". На этом этапе в интерфейсе поваров отображается список активных заказов, сгруппированных по приоритету и времени подачи. После приготовления официант отмечает готовность, и информация об этом немедленно отображается в панели официанта.

Официант получает уведомление о готовности и после доставки блюд к столу подтверждает выполнение заказа. Статус автоматически обновляется на "завершен". Все этапы — от оформления до завершения — сопровождаются логированием, включая временные метки и ответственных сотрудников.

Таким образом, алгоритм охватывает весь жизненный цикл заказа и минимизирует вероятность ошибок за счет четкого разграничения ролей и автоматизированного контроля. Он обеспечивает быстрый отклик, синхронность между кухней и залом, а также формирует аналитическую базу для последующего анализа производительности. Благодаря прозрачности всех действий и четко отслеживаемым этапам исполнения, система способствует улучшению обслуживания и снижает нагрузку на персонал.

Пошаговый алгоритм оформления и обработки заказа представлен в соответствии с рисунком 2.26.

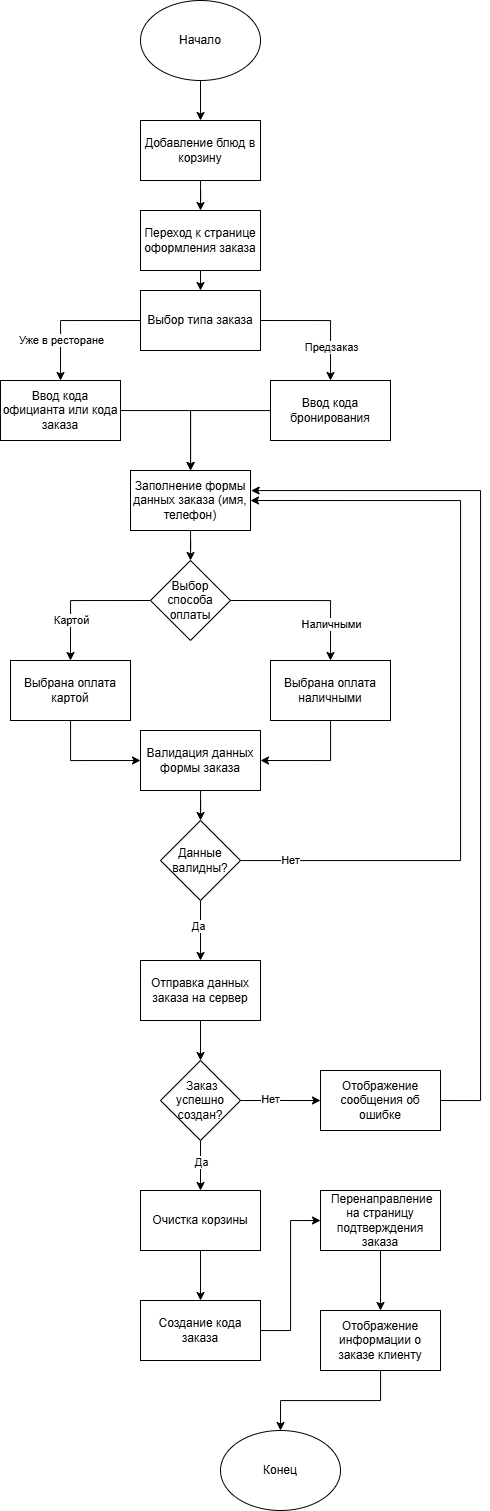


Рисунок 2.26 Алгоритм оформления и обработки заказа

Одним из ключевых алгоритмов системы поддержки принятия решений в рассматриваемом проекте является алгоритм анализа продаж и популярности блюд. Целью данного алгоритма является получение достоверной информации о том, какие блюда пользуются наибольшим спросом у клиентов, а также выявление неэффективных позиций меню для последующего их анализа и возможной замены или корректировки. Такой анализ позволяет не только оптимизировать ассортимент, но и принимать более взвешенные решения по закупке ингредиентов, разработке новых позиций и организации промо-акций.

Этапы работы алгоритма:

1. **Сбор статистических данных:**

- Система агрегирует информацию о заказах за выбранный период.

- Сохраняются сведения о номере заказа, дате, времени, блюдах, количестве и стоимости.

- Также учитываются данные о клиенте (если он авторизован), что позволяет связать поведение клиентов с предпочтениями.

1. **Группировка по позициям меню:**

- Заказы разбиваются по наименованиям блюд.

- Подсчитывается общее количество продаж каждой позиции, а также общая выручка, полученная от нее.

1. **Формирование рейтинга:**

- На основе количества продаж и выручки формируется рейтинг популярных блюд.

- Блюда, находящиеся в нижней части рейтинга, автоматически попадают в список на рассмотрение. Это может указывать на низкий спрос, проблемы с подачей или стоимостью.

1. **Вывод аналитических метрик:**

- Среднее количество продаж в день/неделю/месяц по каждой позиции.

- Средняя стоимость блюда и его доля в общем доходе.

- Динамика изменения спроса: алгоритм строит временные ряды по каждой позиции и анализирует тенденции (рост/падение интереса).

1. **Формирование визуализаций:**

- На дашборде отображаются графики и диаграммы, отражающие уровень продаж.

- Можно фильтровать по дням недели, времени суток, типу клиента (новый/постоянный).

- Выделяются наиболее стабильные и наиболее колеблющиеся позиции.

1. **Формирование рекомендаций:**

- Система предлагает удалить или заменить блюда с consistently низкими показателями.

- Также предлагаются идеи по продвижению успешных блюд (например, установка на главную страницу меню или отображение как "рекомендуемое").

1. **Оценка сезонности и влияния акций:**

- Учитывается, участвовали блюда в акциях или специальных предложениях.

- Система анализирует, как меняется популярность до, во время и после проведения акций.

Алгоритм реализован с применением стандартных средств обработки статистики в Python с использованием библиотек pandas, matplotlib и seaborn для визуализации, а также с использованием встроенных SQL-запросов для агрегации данных.

Схема работы алгоритма представлена на рисунке 2.27.

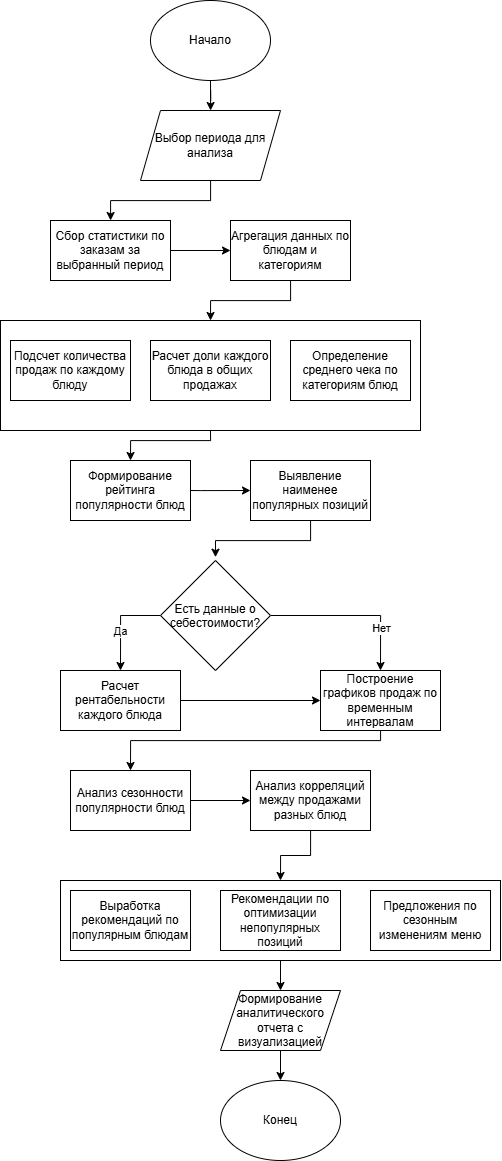


Рисунок 2.27 – Схема алгоритма анализа продаж и популярности блюд

В заключение, можно сказать, что каждый используемый алгоритм, играет свою важную роль и вместе они образуют сложную и гармонично работающую систему, способную удовлетворить потребности самых разных пользователей и обеспечить эффективное функционирование сайтом.

# ЗАКЛЮЧЕНИЕ

В рамках выполнения дипломного проекта спроектирована, реализована и внедрена комплексная система поддержки принятия решений для управления рестораном. В процессе разработки изучены ключевые аспекты функционирования ресторанного бизнеса, проведен анализ бизнес-процессов и информационных потоков, а также выявлены проблемные зоны, требующие автоматизации и оптимизации. Полученные данные позволили построить архитектуру системы, максимально соответствующую текущим и перспективным задачам компании.

Основной задачей дипломного проекта стало создание интеллектуальной цифровой платформы, способной поддерживать ежедневную операционную деятельность ресторана и предоставлять инструменты для принятия обоснованных управленческих решений. Система ориентирована как на сотрудников ресторана (официантов, администраторов, поваров), так и на конечных пользователей — клиентов, обеспечивая прозрачность процессов и удобство взаимодействия для всех ролей.

В процессе работы над проектом решены следующие задачи:

- проведен анализ существующих решений и требований рынка в области ресторанного управления;

- сформулированы функциональные и технические требования к системе;

- спроектирована масштабируемая архитектура и реализована база данных, охватывающая все ключевые сущности ресторана (меню, заказы, бронирования, роли сотрудников и др.);

- реализованы интерфейсы для различных пользователей системы — официантов, администраторов, клиентов;

- разработаны отдельные модули системы поддержки принятия решений (СППР), включая прогнозирование выручки, загруженности, финансовую аналитику, анализ клиентской базы и оценку популярности блюд;

- обеспечена возможность масштабирования, добавления новых функций и интеграции с внешними сервисами;

- внедрены алгоритмы, направленные на повышение эффективности работы ресторана, сокращение издержек и улучшение клиентского опыта.

Система СППР позволила автоматизировать критически важные аспекты деятельности ресторана: от оформления заказов и управления меню до построения прогнозов и проведения финансового анализа. Благодаря внедренным алгоритмам, ресторан получил возможность наглядно отслеживать тенденции спроса, оценивать эффективность работы персонала, прогнозировать загруженность и оптимизировать закупки.

Важной частью разработки стала реализация административной панели, которая обеспечивает централизованное управление системой. В ней администратор может управлять меню, заказами, бронированиями столов, просматривать статистику, принимать решения на основе аналитических данных и гибко настраивать работу ресторана. Панель официанта, в свою очередь, упрощает обслуживание клиентов и ускоряет процесс приема и обработки заказов.

Использование современных технологий и модульной архитектуры позволило создать решение с высоким уровнем надежности, производительности и защищенности данных. Система легко адаптируется под растущие объемы данных и может быть доработана под нужды конкретного заведения, в том числе интеграцией с кассовыми аппаратами, CRM-системами и другими внешними сервисами.

Ключевым результатом проекта стало не только создание программного продукта, но и повышение цифровой зрелости бизнеса. Благодаря автоматизации и инструментам СППР ресторан может принимать более обоснованные стратегические и тактические решения, снижая риски и увеличивая прибыльность бизнеса.

Кроме того, работа над дипломным проектом способствовала значительному росту профессиональных навыков, включая системный анализ, проектирование архитектуры, работу с базами данных, реализацию сложных алгоритмов, создание адаптивных интерфейсов и эффективное взаимодействие с командой.

В целом, система поддержки принятия решений для управления рестораном, разработанная в рамках дипломного проекта, доказала свою эффективность как инструмент повышения качества обслуживания, оптимизации бизнес-процессов и поддержки управленческих решений. Внедрение такой системы предоставляет ресторану устойчивое конкурентное преимущество и является важным шагом в сторону цифровой трансформации бизнеса.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1) Трухан, Ю. Л. Оценка эффективности бизнес-процессов логистических центров: сравнительный анализ существующих подходов и рекомендации по развитию / Ю. Л. Трухан // Вестник ВГТУ. – 2020. – № 1 (38). – С. 231–243.

2) Медведев, В. А. Информационные системы и технологии в логистике и управлении цепями поставок / В. А. Медведев, А. С. Присяжнюк. – Санкт-Петербург: Университет ИТМО, 2016. – 183 с.

3) Алиев, Г. А. Информационная система управления бизнес-процессами на предприятии / Г. А. Алиев, Ю. П. Каратеев // Сборник научных статей по материалам всероссийской научно-практической конференции. – Санкт-Петербург: Издательство "Агрус", 2019. – С. 67–70.

4) Репин, В. В. Бизнес-процессы. Моделирование, внедрение, управление / В. В. Репин. – Москва: МИФ, 2013. – 470 с.

5) Шабанов, Р. М. Интеллектуальная информационная система поддержки принятия решений / Р. М. Шабанов, Н. А. Микушин // Молодой исследователь Дона. – 2019. – Т. 4, № 19. – С. 91–97.

6) Лукинский, В. В. Оценка эффективности логистической деятельности компании на основе ключевых показателей / В. В. Лукинский, Т. Г. Шульженко // Аудит и финансовый анализ. – 2011. – № 4. – С. 1–8.

7) Ермаков, С. С. Проектирование пользовательского интерфейса: теория и практика / С. С. Ермаков. – СПб.: Питер, 2021. – 240 с.

8) Колчанова, С. А. Сравнение фреймворков Flutter и React Native, используемых в разработке гибридных приложений / С. А. Колчанова // E-SCIO. – 2022. – № 4 (67). – С. 558–565.

9) Сравнение фреймворков для кроссплатформенной мобильной разработки: React Native, Flutter, Ionic, Xamarin и PhoneGap [Электронный ресурс]. – Режим доступа: URL: <https://tproger.ru/translations/cross-platform-frameworks-for-mobile-development> (23.04.2023).

10) Шуваев, Я. А. UX/UI дизайн для создания идеального продукта. Полный и исчерпывающий гид / Я. А. Шуваев. – Москва: Бомбора, 2022. – 240 с.

11) ГОСТ 34.602-2020 Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы: дата введения 2022-01-01. – Москва: 1, 2021. – 12 с.

12) Головин, И. Г. Языки и методы программирования: учебник для вузов. / И. Г. Головин, И. А. Волкова. – М.: Академия, 2012. – 304 с.

13) D’Angelo, A. Learning React: Modern Patterns for Developing React Apps / Alex Banks, Eve Porcello. – O’Reilly Media, 2022. – 350 p.

14) Коряковцев, В. Ю. Практикум по проектированию и разработке баз данных: учебное пособие / В. Ю. Коряковцев. – Самара: Самарский университет, 2020. – 154 с.

15) Закас, Н. JavaScript. Оптимизация производительности / Н. Закас. – СПб.: Символ, 2012. – 256 с.

16) Tiangolo, S. FastAPI: Modern Web Framework for Python [Электронный ресурс]. – Режим доступа: https://fastapi.tiangolo.com (01.05.2024).

17) Ермаков, А. Ю. Visual Studio Code: быстрое освоение / А. Ю. Ермаков. – Москва: НИЦ ИНФРА-М, 2022. – 128 с.

18) O’Hara, M. Full Stack FastAPI and React / M. O’Hara. – Lean Publishing, 2023. – 204 p.

19) Глушаков, А. В. Разработка web-приложений с использованием React и REST API / А. В. Глушаков. – Казань: Издательство КФУ, 2021. – 132 с.

20) Конолли, Т. Базы данных: проектирование, реализация и сопровождение. Теория и практика / Т. Конолли, К. Бегг. – Москва: Вильямс, 2018. – 1439 с.

21) Немнюгин, С. А. Информационные технологии управления: учебное пособие / С. А. Немнюгин. – М.: Форум, 2019. – 288 с.

22) Андреев, С. Ю. Современные методы анализа данных: от теории к практике / С. Ю. Андреев. – СПб.: Питер, 2022. – 320 с.

23) Глушков, В. М. Основы системного анализа: учебник / В. М. Глушков. – М.: URSS, 2017. – 392 с.

24) Бройдо, А. И. Архитектура информационных систем / А. И. Бройдо. – М.: БХВ-Петербург, 2020. – 208 с.

25) Седунов, Ю. В. Методы искусственного интеллекта: учебное пособие / Ю. В. Седунов. – М.: Директ-Медиа, 2021. – 276 с.

26) Пресняков, А. В. Проектирование информационных систем. Практикум / А. В. Пресняков. – М.: КНОРУС, 2022. – 244 с.

27) Синицын, С. А. Принципы построения клиент-серверных приложений / С. А. Синицын. – М.: Лань, 2020. – 220 с.

28) Воронцов, К. В. Машинное обучение и анализ данных / К. В. Воронцов. – М.: МЦНМО, 2019. – 384 с.

29) Титов, В. С. DevOps: автоматизация процессов разработки / В. С. Титов. – М.: Диалектика, 2021. – 272 с.

30) Брукс, Ф. Мифический человеко-месяц. Очерки о разработке программного обеспечения / Ф. Брукс. – М.: Питер, 2020. – 336 с.