|  |  |
| --- | --- |
| МИНОБРНАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  Федеральное государственное автономное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | |
| Физико-математический институт | |
| ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА | |
| **Разработка информационной системы для ресторанного бизнеса** | |
|  | Исполнитель  студент 4 курса бакалавриата  физико-математического института,  Направление: 09.03.02  Информационные системы и технологии  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.М. Радостев |
|  | Научный руководитель:  И. О. зав. кафедры ИТ, к.т.н., доцент  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.Н. Раевский |
| Пермь 2025 | |

**Аннотация**

Радостев Степан Михайлович, направление «Информационные системы и технологии», группа ИТХ 2 2021, очная форма обучения. Выпускная квалификационная работа «Разработка информационной системы для ресторанного бизнеса».

В работе рассматриваются существующие системы для ведения ресторанного бизнеса, проводится их анализ и определяется необходимые требования к проектируемой системе. Выбраны программные средства для проектирования, дизайна и разработки системы.

Во введении обосновывается актуальность темы, определяется предмет и объект исследования.

В первой главе проводится анализ существующих систем для ведения бизнеса, выявляются требования к разрабатываемой системе.

Вторая глава посвящена выбору средств для проектирования, разработки системы, а также выбраны средства для проектирования интерфейса системы.

В третьей главе представлены различные диаграммы, подробно описывающие функционирование системы, а также прототип пользовательского интерфейса системы.

В четвертой главе описана разработка информационной системы, представлен разработанный пользовательский интерфейс.

Данная работа написана для кафедры информационных технологий.

Количество страниц: 65, количество иллюстраций: 33, количество таблиц: 3

ОГЛАВЛЕНИЕ

[Введение 5](#_Toc198658608)

[Глава 1 Анализ предметной области 7](#_Toc198658609)

[1.1 Описание предметной области 7](#_Toc198658610)

[1.2 Анализ и сравнение существующих информационных систем 9](#_Toc198658611)

[1.3 Обоснование разработки 13](#_Toc198658612)

[1.4 Постановка задачи для проектирования 14](#_Toc198658613)

[1.5 Вывод 14](#_Toc198658614)

[Глава 2 Анализ и выбор средств проектирования и разработки информационной системы 15](#_Toc198658615)

[2.1 Анализ средств проектирования информационной системы 15](#_Toc198658616)

[2.2 Выбор системы управления базой данных 19](#_Toc198658617)

[2.3 Выбор языка программирования 20](#_Toc198658618)

[2.4 Выбор среды разработки кода 21](#_Toc198658619)

[2.5 Выбор средства для разработки интерфейса 22](#_Toc198658620)

[2.6 Вывод 23](#_Toc198658621)

[Глава 3 Проектирование информационной системы 24](#_Toc198658622)

[3.1 Моделирование логики информационной системы 24](#_Toc198658623)

[3.2 Проектирование базы данных 30](#_Toc198658624)

[3.3 Диаграмма размещения 32](#_Toc198658625)

[3.4 Пользовательский интерфейс 32](#_Toc198658626)

[3.4 Вывод 34](#_Toc198658627)

[Глава 4 Разработка информационной системы 35](#_Toc198658628)

[4.1 Выбор средства для создания настольного приложения 35](#_Toc198658629)

[4.2 Разработка интерфейса и логики ИС 35](#_Toc198658630)

[4.3 Вывод 49](#_Toc198658631)

[Заключение 50](#_Toc198658632)

[Библиографический список 51](#_Toc198658633)

[Приложение А Диаграмма прецедентов Администратора 53](#_Toc198658634)

[Приложение Б Программный код 54](#_Toc198658635)

[Приложение В Руководство пользователя 55](#_Toc198658636)

# Введение

Деятельность, связанная с организацией и предоставлением услуг общественного питания, занимает одно из ведущих мест в сфере обслуживания. В России ресторанный бизнес увеличивает оборот на 10‑20% каждый год и по прогнозам экспертов в данной области, это число не планирует снижаться [1]. Для того, чтобы оставаться конкурентоспособными и максимизировать свою прибыль, предприятия общественного питания вынуждены непрерывно совершенствовать свою деятельность. Одним из способов улучшения эффективности деятельности является автоматизация бизнес-процессов и грамотная бизнес-стратегия. Предприятия активно применяют информационные системы для решения различных задач, включая планирование и учет. Управленческий учет нужен собственнику для того, чтобы принимать взвешенные стратегические решения для дальнейшего развития своего бизнеса на основе текущей информации.

Актуальность разработки информационной системы для ресторанного бизнеса обусловлена несколькими факторами:

Во-первых, избавление администратора от введения бумажного учета, что позволит устранить ошибки, связанных с ручным вводом данных.

Во-вторых, на рынке программных продуктов существует множество решений для ведения бизнеса, но большинство из них являются дорогостоящими, либо не способны в полной мере удовлетворить потребности заказчика.

Объект исследования – программное обеспечение для ресторанного бизнеса

Предмет исследования – информационная система для ведения учета

Цель работы – разработка информационной системы для ресторанного бизнеса

Задачи, которые необходимо решить для достижения цели работы:

1. изучение предметной области;
2. анализ аналогичных существующих систем;
3. анализ и выбор программного обеспечения для разработки ИС;
4. построение модели ИС;
5. разработка ИС;
6. создание документации.

Цели и задачи работы определили ее структуру, которая состоит из теоретической, обзорной, практической частей и заключения.

В первой главе содержится анализ предметной области и объекта исследования. Был проведен обзор существующих решений на рынке программного обеспечения для ресторанного бизнеса и на основе анализа поставлены цели на проектирование.

Вторая глава посвящена анализу программных средств для проектирования и реализации информационной системы.

Третья глава посвящена проектированию информационной системы и прототипированию интерфейса.

Четвертая глава посвящена разработке информационной системе. Были реализованы все задачи, поставленные в теоретической части работы.

Заключение включает в себя основные выводы по проделанной работе.

# Глава 1 Анализ предметной области

В данной главе изучается предметная область, актуальность проектируемой информационной системы, рассматриваются существующие аналоги и на основе анализа определяются требования к разрабатываемой системе.

## 1.1 Описание предметной области

Одним из способов развития деятельности предприятия является автоматизация. С использованием подобных систем, руководителям предприятия будет предоставлена возможность следить за своим бизнесом, проводить анализ работы организации и обладать полной информацией. Внедрение информационной системы повысит качество управления рестораном и поможет сократить количество ошибок при ведении бизнеса, повысить прибыль, создавать привлекательные предложения для клиентов, что может привлечь новых посетителей и удержать старых. Однако, внедрение таких систем требует глубокого анализа процессов предприятия, что зачастую требует большого денежного вложения. Для понимания того, что требуется от информационной системы, необходимо оценить типы автоматизирующих систем, которые предназначены для решения определенного круга задач.

CRM‑система (Customer Relationship Management) – система, управляющая взаимоотношениями с клиентами [2]. Это означает, что система хранит данные о работе бизнеса с клиентами, позволяет взаимодействовать с ними.

Существует 3 вида CRM систем:

* операционная;
* коллективная;
* аналитическая.

Операционные CRM – программы, предназначенные для отдела продаж. В них хранится информация о клиентах и фиксируются все этапы взаимодействия с ними. Такие системы помогают менеджерам взаимодействовать с клиентами, следить за статусом сделок, отправлять клиентам SMS и электронные письма, записывать звонки и повышать качество обслуживания.

Коллективные CRM‑системы объединяют работу разных отделов, таких как техническая поддержка, продажи и маркетинг.

Аналитическая CRM предназначена для оптимизации внутренних рабочих процессов в компании. Она помогает анализировать, улучшать и расширять возможности работы сотрудников.

Другим видом систем для автоматизации работы предприятия являются ERP (Enterprise Resource Planning) – управление ресурсами предприятия. Такие системы предназначены для автоматизации ключевых процессов предприятия:

* управление ресурсами;
* оптимизация затрат;
* управление складом;
* планирование;
* управление персоналом.

ERP больше подходит для владельца бизнеса, ведь такие системы показывают информацию по развитию предприятия, что имеет большое значение для построения стратегии по управлению.

ERP направлены на внутренние процессы предприятия, повышения эффективности, а CRM направлена на повышение продаж и удержание клиентов. По отдельности системы не являются универсальными, но ERP сочетает в себе большее количество функций. Современные ERP могут совмещать в себе и CRM систему, но такие решения будут обходиться гораздо дороже.

Информационные системы для предприятий общественного питания состоят из двух компонентов: фронт‑офис и бэк‑офис. Фронт‑офис предназначен для обеспечения деятельности в зале и связан с обслуживанием клиентов. Часто такие приложения устанавливаются на смартфон или терминал, чтобы у официанта была возможность быстро обслужить гостей. Бэк‑офис предоставляет информацию для управляющего заведением и устанавливается на персональный компьютер администратора. Информация, предоставляемая бэк‑офисом, очень разнообразна и важна для повышения эффективности работы компании.

Бэк‑офис может обеспечивать выполнение:

* учета продаж;
* получение отчетности и аналитика;
* управление меню;
* управление складом;
* управление программой лояльности;
* планирование и прогнозирование.

Таким образом, информационные системы для управления рестораном сочетают в себе функции ERP и CRM систем и называются бэк-офисом. Такие системы позволяют автоматизировать основные процессы предприятия и повысить эффективность. Автоматизация предприятия позволит избежать убытков, но не является ее гарантом, система всего лишь поможет управляющему следить за работой и принимать верные решения на основе анализа работы предприятия.

## 1.2 Анализ и сравнение существующих информационных систем

При анализе и сравнении существующих информационных систем в ресторанном бизнесе необходимо обратить внимание на несколько важных критериев. К ним относятся:

* функциональность системы;
* удобство ее использования;
* возможность интеграции с другими системами;
* стоимость системы;

Существует большое количество систем для автоматизации процессов в ресторанном бизнесе и все они обладают различным функционалом. Рассмотрим самые популярные варианты.

1. Restik [3]

Restik – облачная система автоматизации ресторанного бизнеса, которая подходит для разных типов заведений. С помощью этой системы можно автоматизировать процесс оформления заказов, складской учет, печать чеков, анализировать статистику продаж и работать над программами лояльности. Restik интегрируется с онлайн‑кассами и позволяет ресторанам запустить доставку.

В приложении можно просматривать показатели бизнеса. Система анализирует продажи, чтобы руководство могло следить за работой предприятия.

Достоинства системы:

* программа лояльности;
* учет склада;
* отчетность о продажах;
* облачная система.

Недостатки:

* необходимо постоянное подключение к интернету;
* высокая стоимость дополнительных функций.

Цена месячной подписки на Restik с базовым функционалом онлайн кассы начинается от 1590 рублей.

2. Fusion POS [4]

Fusion POS – это облачная система для точки продаж. Бэк-офис предоставляет информацию о состоянии продаж, финансовый учет и актуальные данные о количестве гостей и заказах онлайн.

Программа позволяет управлять товарами: просмотр данных о сырье на складе, проводить инвентаризацию, загружать технологические карты. Программа способна рассчитывать себестоимость блюд.

Доступна программа лояльности. Администратор может создать базу постоянных клиентов, начислять им бонусы, разбивать на группы и предоставлять скидки.

Система поддерживает режим терминала, где сотрудники могут вносить данные о заказах. Сервис предоставляет возможность работы на различных платформах. Администраторская панель с отслеживанием всех показателей находится на серверах компании, поэтому необходим постоянный доступ в интернет.

Достоинства системы:

* облачная система;
* система лояльности;
* управление складом;
* отчетность.

Недостатки:

* случаются проблемы со стабильностью системы.

Цена месячной подписки на Fusion POS начинается от 1490 рублей, в которую входит онлайн касса.

3. Quick Resto [5]

Quick Resto – облачная система автоматизации ресторана. В приложении для руководителя можно просматривать показатели заведения, создавать собственные шаблоны отчетов, проверять остатки на складе, просматривать чеки.

Сервис предоставляет возможность складского учета. Доступно создание технических карт с модификаторами, которая автоматически рассчитывает себестоимость блюда. В системе имеется возможность создания приложения для гостей за дополнительную плату. В нем клиенты могут сделать предзаказ или выбрать доставку на дом.

Достоинства:

* отчетность;
* складской учет;
* гибкое меню для клиентов.

Недостатки:

* высокая стоимость за подключение дополнительных функций.

Цена месячной подписки на Quick Resto начинается от 2490 рублей за функционал онлайн кассы.

4. iiko [6]

Сервис предлагает функции автоматизации кассы, склада, учета персонала и финансов. Программа может работать в облаке или локально. Стоимость локальной версии для ресторана начинается от 90000 рублей в год.

Сервис предоставляет онлайн кассу с программой приема заказов. Для кухни доступны печать чеков с блюдами и экран для отслеживания заказов. Клиенты могут заказывать блюда с модификаторами, в которые входят дополнительные ингредиенты или соусы., после чего программа автоматически посчитает себестоимость блюда.

В программе администратор может просматривать отчеты, данные о продажах, складских остатках, списаниях. Доступно автоматическое формирование заказов поставщикам. Сервис может прогнозировать будущую заполняемость заведения на основе данных о продажах и планировать закупку ингредиентов.

Для администратора существует возможность создать программу лояльности, чтобы у клиентов была возможность пользоваться акциями и скидками.

Достоинства:

* отчетность;
* складской учет;
* система лояльности;
* автоматический заказ продуктов у поставщика.

Недостатки:

* высокая стоимость дополнительных функций.

Цена месячной подписки на iiko с базовым функционалом в качестве онлайн кассы начинается от 1490 рублей.

5. Yuma [7]

Yuma – это система, состоящая из нескольких сайтов и приложений для операционных систем Windows и Android. Программа устанавливается на терминал, который размещается в зале. Официанты могут формировать заказы через мобильное приложение или получать их в терминале.

Управляющий заведением имеет бэк-офис, размещенный на сайте. На этом сайте собираются данные о продажах, расходах, прибыли и состоянии склада. Есть возможность создавать блюда в меню с модификаторами, управлять ценами. Сервис имеет возможность отслеживать складские остатки, а также управляет закупками у поставщиков.

В системе есть учет работы персонала. Сотрудники отмечают свои смены через приложение, а также отмечаются о перерывах в работе. Программа позволяет рассчитывать зарплату и формировать ведомости для финансовой отчетности.

Достоинства системы:

* аналитика;
* управление складом;
* управление закупками у поставщиков;
* управление персоналом.

Недостатки:

* высокая стоимость дополнительных функций;
* сложности для подключения.

Цена месячной подписки на Yuma с базовым функционалом в качестве онлайн кассы начинается от 1490 рублей, но разработчик дает возможность оформить подписку на год за 39000 рублей.

Сведем проанализированные системы в таблицу, где отметим наличие у системы функционала для бэк‑офиса и системы регистрации заказов.

Таблица 1 – Сравнение

|  | Restik | Fusion POS | Quick Resto | iiko | Yuma |
| --- | --- | --- | --- | --- | --- |
| Отчетность | + | + | + | + | + |
| Аналитика | + | + | + | + | + |
| Складской учет | - | + | + | + | + |
| Автоматический заказ у поставщиков | - | - | - | + | + |
| Управление персоналом | - | - | - | + | + |
| Система лояльности | + | - | + | + | + |
| Размещение | Облако | Облако | Облако | Локальное | Облако |
| Система регистрации заказов | + | + | + | + | + |
| Цена | 1590 руб. в месяц | 1490 руб. в месяц | 2490 руб. в месяц | 90000 руб. в год | 39000 руб. в год |

## 1.3 Обоснование разработки

Рассмотренные информационные системы предоставляют широкий функционал для ведения бизнеса. Проблема существующих информационных систем заключается в том, что их внедрение требует значительных финансовых затрат и они охватывают большое количество функций, за которые необходимо платить. Для их использования необходимы глубокие знания системы, а в случае возникновения проблем требуется вмешательство специалистов по установке. Почти ни одна из проанализированных систем не имеют полноценного бэк-офиса, кроме iiko и Yuma, но их цена может быть отталкивающим фактором при выборе информационной системы для малого предприятия. Все рассмотренные системы имеют функционал системы POS (Point of Sale), в которой регистрируются заказы, однако данная система является базовой, т.е. только она входит в минимальную стоимость подписки за сервис. Актуальность данного проекта состоит в создании более простой в использовании и бесплатной автоматизированной системы для ведения учета.

Информационная система упростит наблюдение за рестораном, предоставит управляющему информацию для принятия решений. Анализируя данные о продажах, предпочтениях клиентов, уровне запасов и эффективности персонала, можно оптимизировать бизнес-процессы и разработать эффективную стратегию развития.

## 1.4 Постановка задачи для проектирования

На основе анализа существующих систем можно выделить те задачи и функции, которые необходимы в проектируемой системе:

* система регистрации заказов;
* отчеты о продажах и прибыли;
* формирование базы клиентов;
* учет выручки и затрат;
* формирование меню и технический карт;
* отслеживание запасов ингредиентов на складе;
* автоматическое формирование заявки поставщикам при необходимости.

## 1.5 Вывод

В этой главе был проведен анализ предметной области, анализ существующих информационных систем, обоснована актуальность и поставлены задачи на проектирование информационной системы.

# Глава 2 Анализ и выбор средств проектирования и разработки информационной системы

Вторая глава посвящена анализу и выбору средств проектирования, выбору языка программирования и системе хранения данных. Необходимо выбрать подходящие программные средства для достижения поставленных задач

## 2.1 Анализ средств проектирования информационной системы

Информационная система – это набор взаимосвязанных элементов, которые включают в себя информационные ресурсы и технологии. Они созданы для того, чтобы упростить и автоматизировать процессы сбора, сохранения, обработки и анализа данных в определённой предметной области [8].

Проектирование информационной системы представляет собой процесс создания моделей и диаграмм, отражающие структуру, функции и поведение системы. Это позволяет лучше понять работу системы, выявить ее сильные и слабые стороны, а также предсказать ее поведение в различных ситуациях. Одним из основных инструментов для моделирования является язык UML.

UML (Unified Modeling Language) – это унифицированный язык моделирования, который используется для создания моделей программного обеспечения. Он предоставляет набор графических обозначений и диаграмм, которые помогают визуализировать структуру, поведение и взаимодействие системы, что значительно упрощает понимание, разработку, реализацию и документирование программных продуктов.

CASE-средства (Computer Aided Software Engineering) представляют собой комплекс программных инструментов, предназначенных для автоматизации процессов разработки информационных систем и программного обеспечения. Эти средства позволяют формализовать и структурировать этапы жизненного цикла информационных систем, включая идентификацию и анализ бизнес-процессов, моделирование их взаимодействия, оптимизацию инфраструктуры и другие ключевые аспекты. Основной целью использования CASE-средств является повышение эффективности разработки за счет сокращения временных и финансовых затрат, а также улучшения качества конечного продукта [9].

Существует множество популярных CASE-средств для проектирования, поэтому необходимо выбрать средства, соответствующие следующим критериям:

* поддержка UML;
* удобный и понятный интерфейс;
* бесплатное распространение или пробная версия;
* поддержка системы Windows 10;
* наличие широкого типов диаграмм.

Рассмотрим несколько популярных средств проектирования, поддерживающие UML:

1. StarUML [10]

StarUML – это инструмент для визуального моделирования с открытым исходным кодом. В программе есть возможность строить 11 видов диаграмм, поддерживает UML 2.0, позволяет импортировать и экспортировать данные, генерировать программный код на языках Java, C++ и C#. Программа имеет интуитивно понятный интерфейс, может расширить свой функционал за счет установления дополнительных плагинов, которые можно создавать самостоятельно.

StarUML доступен для всех современных операционных систем. Актуальная версия приложения – 6.3.2.

Достоинства:

* графическое моделирование схем и диаграмм;
* генерация программного кода;
* поддержка плагинов;
* проверка моделей;
* приятный и простой интерфейс;
* поддержка всех актуальных операционных систем;

Недостатки:

* интерфейс только на английском языке.

Программа не распространяется бесплатно, но есть пробная версия, в которой есть все необходимые функции для проектирования системы.

2. Astah [11]

Astah – это инструмент для моделирования и проектирования программного обеспечения, который позволяет создавать различные типы диаграмм, такие как диаграммы классов, последовательности, состояний, активностей и другие.

Программа может генерировать программный код на языках программирования, таких как Java, C++, C#.

Для студентов доступна бесплатная версия Astah, которая предоставляет базовые функции для построения диаграмм.

Достоинства:

* возможность построения широкого типа диаграмм;
* генерация программного кода;
* возможность совместной работы.

Недостатки:

* программа на английском языке;
* перегруженный интерфейс.

3. Автограф [12]

Автограф – это редактор для описания бизнес-процессов, создания технических планов и визуального проектирования. Автограф выступает в роли замены других популярных средств моделирования, в том числе в рамках политики импортозамещения в России. К функциям программы относятся построение графиков, диаграмм, таблиц. Автограф работает на современных системах Windows и Linux.

Достоинства:

* графическое моделирование схем и диаграмм;
* совместная работа;
* наличие готовых шаблонов.

Недостатки:

* высокая стоимость.
* малое количество функций в пробной версии

Лицензия на год для Автограф стоит 14300 рублей, но существует пробная версия на 30 дней с ограниченным функционалом.

4. Diagrams.net [13]

Diagrams.net – сервис для построения диаграмм и схем. Сервис является бесплатным и доступен в виде сайта. Доступен широкий список различных фигур и диаграмм.

Достоинства:

* бесплатный сервис;
* сохранение на локальном диске и в облаке;
* совместная разработка.

Недостатки:

* нагруженный интерфейс;

Сведем описанные выше CASE-средства в таблицу 2, где отметим наличие интересующих критериев. В данной таблице присутствует оценка интерфейса и функционала. Оценивание идет по 5-ти бальной шкале, где «5» наивысший балл, а «1» наименьший балл.

Для интерфейса это означает, что он максимально удобен и интуитивно понятен, а «1» означает, что приложение неудобно использовать. Оценки интерфейса являются субъективными и основываются на опыте использования приложения.

Для функционала тоже используется оценка по 5-ти бальной шкале:

* «5» ПО обладает всем необходимым функционалом для проектирования информационной системы;
* «4» ПО обладает всем функционалом, но часть функций ограничена;
* «3» имеется достаточно функций для моделирования системы, большая часть функций ограничена;
* «2» функционал ограничен, трудно выполнить работу в полном объеме;
* «1» невозможно выполнить работу.

Таблица 2 – Сравнение CASE-средств

| Название | Поддержка UML | Пробная версия | Платформы | Оценка интерфейса | Функционал |
| --- | --- | --- | --- | --- | --- |
| StarUML | + | + | Windows 10, Linux, MacOS | 5 | 5 |
| Astah | + | + | Windows 10, Linux, MacOS | 3 | 4 |
| Автограф | + | + | Windows 10, Linux | 3 | 1 |
| Diagrams.net | + | + | Web приложение | 2 | 5 |

Исходя из вышеперечисленных вариантов, был выбран StarUML. Программа обладает бесплатной версией, предоставляет широкий функционал, имеет интуитивно понятный интерфейс.

## 2.2 Выбор системы управления базой данных

База данных представляет собой структурированный набор данных, управляемый и поддерживаемый компьютерной системой. Она обеспечивает эффективный способ хранения, организации и извлечения информации, что позволяет осуществлять быстрый доступ к необходимым данным. Основной принцип работы разрабатываемой информационной системы – это активное взаимодействие с базой данных.

СУБД (система управления базами данных) – это программное обеспечение, предоставляющее инструменты для работы с базами данных, включая их создание, модификацию, удаление, а также операции добавления, извлечения, обновления и удаления данных. К популярным СУБД относятся MySQL, SQLite, PostgreSQL и другие. Рассмотрим некоторые из них.

SQLite – это встраиваемая реляционная система управления базами данных. Особенность данной СУБД заключается в том, что она не функционирует как отдельный сервер, а интегрирована в приложение, использующее базу данных [14]. SQLite использует один файл для хранения всей информации, включая схему базы данных, таблицы, индексы и данные. Данная СУБД не поддерживает создание хранимых процедур.

PostgreSQL – это объектно-реляционная СУБД, предоставляющая возможности для хранения и управления данными [15]. Данная СУБД поддерживает широкий спектр функций, включая сложные SQL запросы, транзакции, индексы, хранимые процедуры, репликацию данных и множество других функций. PostgreSQL имеет открытый исходный код и активное сообщество, что обеспечивает постоянное развитие и поддержку СУБД.

В качестве системы управления базой данных была выбрана PostgreSQL в виду того, что данная СУБД является бесплатной, поддерживает сложные SQL запросы, возможность создавать хранимые процедуры, что имеет преимущество для разрабатываемой системы.

## 2.3 Выбор языка программирования

C# – это объектно-ориентированный язык программирования, разработанный компанией Microsoft. C# применяется для создания различного программного обеспечения, включая приложения для Windows, веб-приложения, мобильные приложения, разработка игр [16]. Язык обладает строгой типизацией, автоматическим управлением памяти. C# является популярным языком программирования благодаря своей простоте, кроссплатформенности и широкой поддержке со стороны сообщества разработчиков.

Python – это интерпретируемый язык программирования высокого уровня [17]. Язык известен простотой синтаксиса, что делает его популярным выбором среди программистов. Python имеет динамическую типизацию и автоматическое управление памятью. Данный язык программирования применяется в различных областях, включая создание веб-приложений, автоматизацию задач с использованием скриптов, анализ данных и разработку систем машинного обучения.

C++ – это компилируемый язык программирования высокого уровня [18]. Он известен высокой производительностью, гибкостью и широким спектром применения, включая создание операционных систем, драйверов, разработку игр, приложений для настольных компьютеров и мобильных устройств. Однако, разработка программ на этом языке требует больше навыков и опыта, что может сильно замедлить процесс разработки.

В качестве языка программирования был выбран C#. Язык имеет большое количество библиотек и фреймворков, которые упростят разработку. Так же язык имеет активное сообщество и отличную документацию, переведенную на русский язык, которое может помочь с разработкой информационной системы.

## 2.4 Выбор среды разработки кода

IDE (интегрированная среда разработки) – это программное обеспечение, предоставляющее набор инструментов для разработки программных продуктов. IDE включает в себя функции текстового редактора, средства для компиляции или интерпретации кода, отладчик, автоматическое завершение кода, инструменты для управления версиями и другие функции, упрощающие разработку.

Microsoft Visual Studio 2022 – это интегрированная среда разработки для создания приложений для различных платформ [19]. Данная IDE пользуется большой популярностью за счет своих функций. Преимущества данной IDE:

* поддержка многих языков программирования: Visual Studio поддерживает широкий спектр языков программирования, таких как C++, C#, F#, Python, JavaScript, TypeScript и другие;
* наличие встроенных средств по разработке настольных приложений, таких как WPF и WinForms;
* пакетный менеджер NuGet позволяет быстро загрузить в разрабатываемый проект библиотеку;
* IntelliCode подсистема призванная ускорить написание кода благодаря подсказкам и завершения кода.

Visual Studio 2022 доступна в нескольких изданиях, включая Community (бесплатная версия для индивидуальных пользователей), Professional и Enterprise.

К недостатку данной IDE можно отнести высокий вес, занимаемый на жестком диске, но он компенсируется большим количеством функций, предоставляемый данной программой.

Основной причиной выбора данной IDE выступает то факт, что в качестве языка программирования был выбран C# и у программы имеется встроенные средства для разработки WPF и WinForms приложений.

## 2.5 Выбор средства для разработки интерфейса

Прототипирование интерфейса играет важную роль в разработке приложения, ведь от него зависит удобство и опыт использования пользователями.

Были выбраны популярные средства для разработки интерфейсов приложений и сайтов, среди них Figma [20], MockFlow [21], NinjaMock [22]. Сделаем их анализ и дадим оценку следующим критериям:

* Платформа: 1 – доступно только на персональных компьютерах или только веб-версия; 2 – доступен на персональном компьютере и имеется веб-версия.
* Доступность: 1 – платная версия, 2 – есть бесплатная версия.
* Функционал: 1 – мало функций для прототипирования; 2 – достаточно функций для прототипирования; 3 – много функций и есть готовые конструкции.
* Дизайн: 1 – неприятный интерфейс; 2 – интуитивно понятный, приятный интерфейс

Сведем результаты анализа в таблицу 3.

Таблица 3 – Сравнение средств прототипирования интерфейса

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Figma | MockFlow | NinjaMock |
| Платформа | 2 | 2 | 1 |
| Доступность | 2 | 2 | 2 |
| Функционал | 3 | 2 | 1 |
| Дизайн | 2 | 1 | 1 |
| Сумма | 9 | 7 | 5 |

Исходя из анализа сервисов для прототипирования, наибольшее количество баллов набрал сервис Figma, поэтому будем использовать его.

## 2.6 Вывод

В данной главе проанализированы популярные и доступные CASE-средства, СУБД, языки программирования и средство для прототипирования интерфейса. В результате анализа были выбраны следующие средства:

* для разработки прототипа и построения UML диаграмм был выбран StarUML, ПО имеет понятный и приятный интерфейс, обладает достаточным функционалом для проектирования системы;
* для управления и хранения данных был выбран PostgreSQL за его функционал и наличие бесплатной версии;
* в качестве языка программирования был выбран C#;
* в качестве среды разработки кода был выбран Microsoft Visual Studio 2022 за его удобство и простоту в использовании;
* в качестве средства для прототипирования интерфейса был выбран сервис Figma.

# Глава 3 Проектирование информационной системы

Данная глава посвящена проектированию информационной системы, включая описание работы моделей информационной системы, построение UML диаграмм и схемы структуры базы данных. Также был создан прототип интерфейса программы.

В прошлых главах была рассмотрена теоретическая часть работы, в которой были поставлены задачи для проектирования:

* система регистрации заказов;
* отчеты о продажах и прибыли;
* формирование базы клиентов;
* учет выручки и затрат;
* формирование меню и технический карт;
* отслеживание запасов ингредиентов на складе;
* автоматическое формирование заявки поставщикам при необходимости.

С помощью UML диаграмм продемонстрирован принцип работы проектируемой ИС.

## 3.1 Моделирование логики информационной системы

В соответствии с поставленными задачами были спроектированы диаграммы, раскрывающие функционал системы.

Для отображения возможностей Администратора заведения была создана диаграмма прецедентов. В системе разработан вход по логину и паролю, чтобы незарегистрированные пользователи не смогли пользоваться системой и изменять данные. Поэтому первым этапом пользования системой является авторизация (см. рис 1).

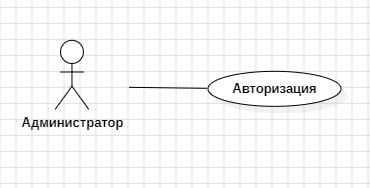


Рис. 1 – Авторизация

После успешной авторизации Администратору открывается полный доступ к системе, в которой можно добавлять всю служебную информацию о заведении. Функции спрятаны в отдельных страницах, чтобы разделить отдельные модули и сделать дизайн более приятным и менее нагруженным. На рисунке 2 изображен фрагмент диаграммы прецедентов для отображения возможностей по изменению меню заведения, просмотра сведения о столиках и просмотра состояния склада. Диаграмма получилось большой и для описания функций ее пришлось разделить. Полная диаграмма находится в приложении А.

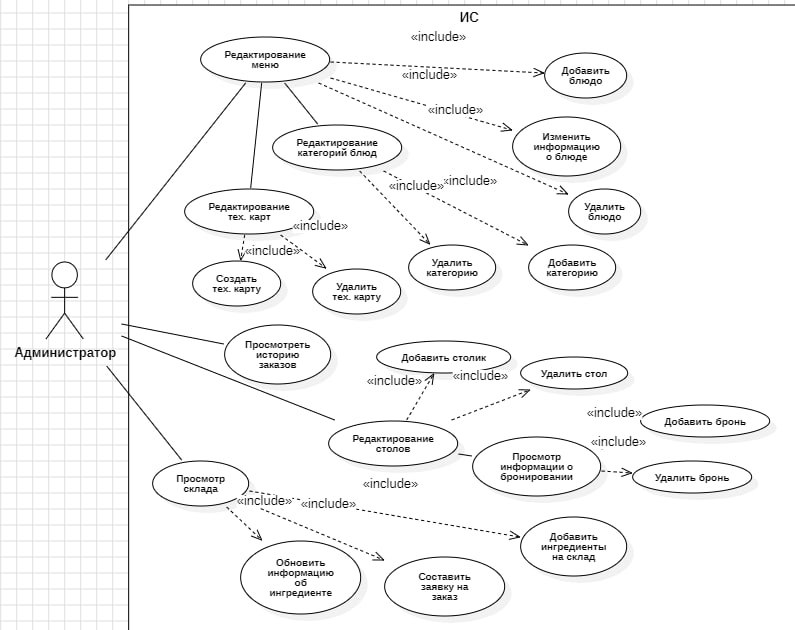


Рис. 2 – Диаграмма прецедентов для администратора

Редактирование меню представляет собой набор вкладок по функциям: редактор блюд, редактирование категорий блюд и редактирование технических карт.

Редактор блюд включает в себя возможности по добавлении нового блюда, изменении информации о существующем блюде и удалении существующего.

Редактор категорий блюд включает в себя раздел по добавлению, редактированию и удалению категорий блюд. Данный раздел нужен, чтобы соотносить готовые блюда к определенным категориям и выполнять быстрый поиск.

Редактор технических карт представляет собой раздел, в котором Администратор может указывать рецептуру каждого блюда. В будущем, при заказе блюда, у которого есть рецептура, то продукты, указанные в технической карте, автоматически спишутся со склада в указанном количестве. Данный функционал позволит оценивать близкие к реальности остатки продуктов и своевременной создать заявку на заказ продуктов у поставщиков.

Также на диаграмме изображены прецеденты по редактированию столов. В данной вкладке Администратор указывает существующие столы в заведении, а в случае, когда клиенты захотят забронировать стол на определенной время, то Администратор сможет сделать это в соответствующей вкладке.

Просмотр истории заказов на данный момент имеет в себе обзорный характер, в котором можно просмотреть историю всех заказов и информации о них. В будущей разработке этот раздел можно расширить и дополнить.

Просмотр склада подразумевает собой функции по просмотру содержимого склада, внесение или изменение информации о продуктах на складе и удаление отдельных продуктов. Также имеется функция по составлению заявки на пополнение склада поставщикам в случае, когда остатки отдельных продуктов ниже их лимита или выхода срока годности.

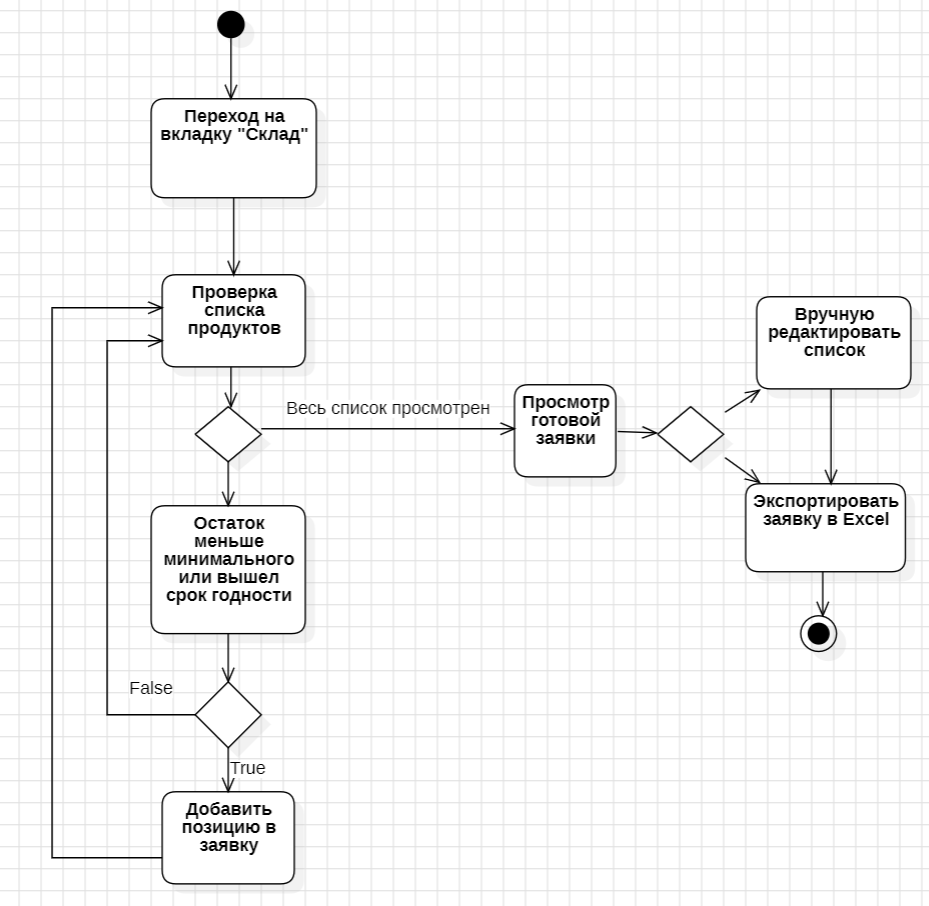


Рис. 3 – Диаграмма деятельности для автоматической работы склада

Логика заполнения заявки на пополнение склада продемонстрирована на рисунке 3. Данный модуль важен для понимания оставшегося запаса продуктов, так как работа кухни напрямую влияет на содержимое склада и необходимо его вовремя пополнять для непрерывной работы. Автоматизация склада позволяет оптимизировать процесс управления запасами и уменьшить время на обработку заказов.

На рисунке 4 изображен второй фрагмент диаграммы прецедентов Администратора.

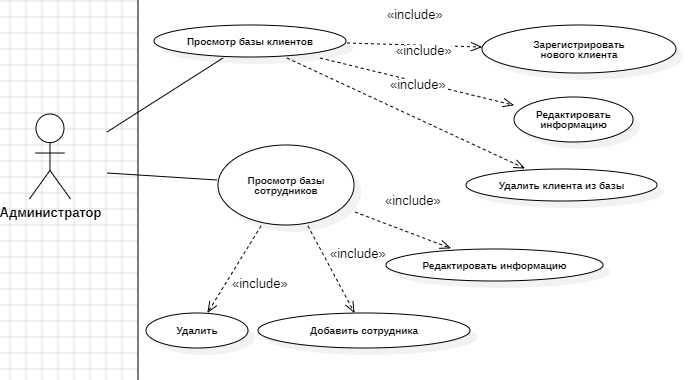


Рис. 4 – Второй фрагмент диаграммы прецедентов для администратора

В данных прецедентах описывается возможности по просмотру информации о сотрудниках заведения, изменении информации о них и удалении. Также описаны прецеденты по просмотру базы клиентов, в которой можно зарегистрировать нового клиента или изменить информацию о старом клиенте, если возникла ошибка.

Логика по заполнению информации о различных модулях подвергается общей последовательностью, изображенной на рисунке 5:

1. пользователь хочет ввести информацию;
2. приложение открывает модальное окно;
3. пользователь вводит информацию;
4. система обрабатывает введенную информацию на корректность;
5. приложение отправляет команду базе данных на добавление информации.

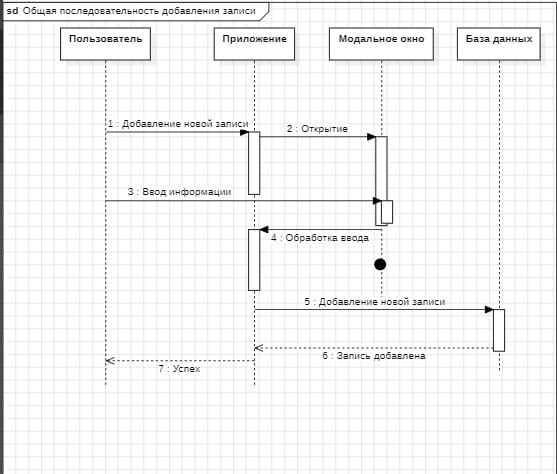


Рис. 5 – Общая последовательность добавления записи

Последним фрагментом диаграммы прецедентов для Администратора является описание функций по просмотру отчетов о работе заведения.

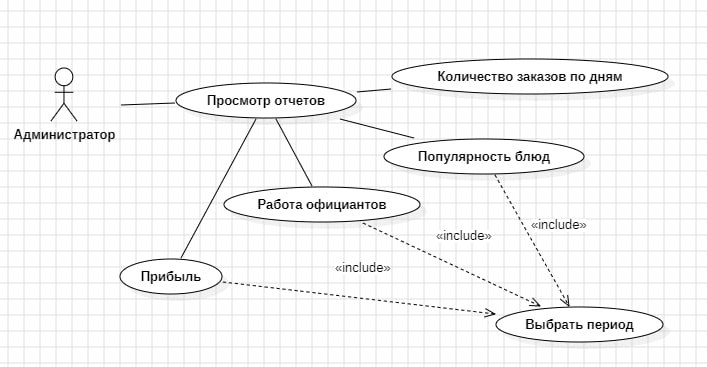


Рис. 6 – Третий фрагмент диаграммы прецедентов для Администратора

Отчетность предоставляет Администратору информацию о работе ресторана, которая может быть использована для принятия решений, планирования и анализа.

Отчет о продажах предоставляет информацию о продажах ресторана за определенный период времени или в конкретный день.

Отчет о работе официантов предоставляет информацию о работе определенный период времени. Он содержит информацию о количестве обслуженных клиентов. С его помощью администратор может назначать премии для активных сотрудников.

Отчет о популярности блюд позволит Администратору проанализировать популярность блюд, что позволит убрать блюда, не пользующиеся популярностью.

Отчетность помогает Администратору контролировать работу ресторана, оценивать эффективность бизнеса, оптимизировать расходы и принимать решения о дальнейшем развитии бизнеса. Отчетность может предоставлять информацию в виде графиков или диаграмм.

Для занесения данных о работе заведения необходима система, регистрирующая их. Для POS системы диаграмма прецедентов представлена на рисунке 7.

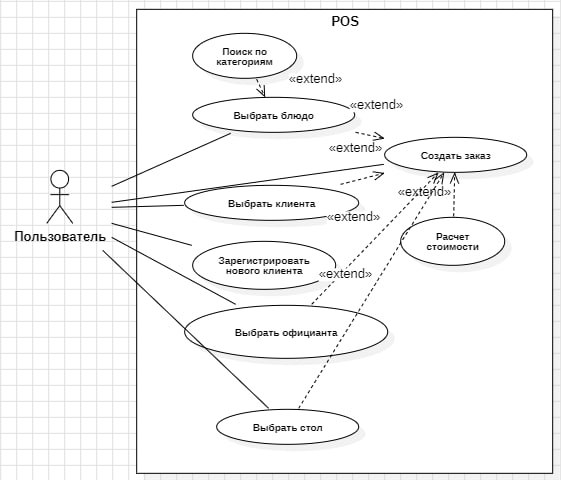


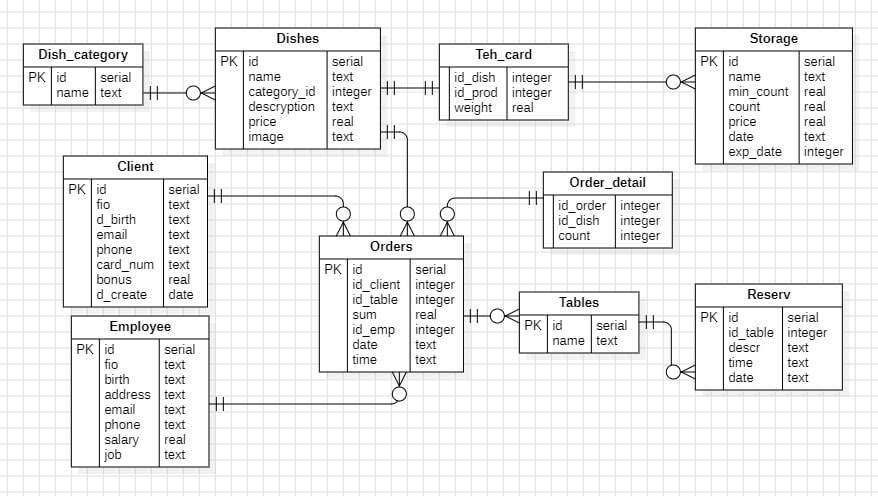
Рис. 7 – Система регистрации заказов (POS)

Пользователь данной системы может выбирать блюда из существующего меню, для быстрого поиска блюд предусмотрен поиск по категориям. Выбранные блюда будут заноситься в текущий заказ, и система рассчитает стоимость.

Пользователь может выбирать официанта, который будет обслуживать клиента, и выбрать стол, за которым находится клиент. Также присутствуют прецеденты, описывающие возможность выбора обслуживаемого клиента. Если клиента нет в системе, то существует возможность регистрации нового клиента в базе данных.

## 3.2 Проектирование базы данных

Для описания логики базы данных будем использовать ER-диаграммы. ER-диаграмма (Entity-Relationship diagram) – это графическое представление модели данных, которая описывает структуру базы данных и ее взаимосвязи. На рисунке 8 изображена схема базы данных.

Рис. 8 – Схема базы данных

Бэк-офис подразумевает хранение большого количества информации о заведении. Было создано 10 таблиц для хранения основных данных.

Таблица Employee представляет собой хранилище информации о сотрудниках. Содержит поля id – первичный ключ, fio – фамилия, имя и отчество сотрудника, birth – дата рождения, address – адрес сотрудника, email – электронная почта, phone – номер телефона, salary – заработная плата сотрудника, job – должность.

Таблица Client содержит информацию о клиентах, которые посетили заведение. Содержит поля: id – первичный ключ, fio – фамилия, имя и отчество клиента, d\_birth – дата рождения, email – электронная почта, phone – номер телефона, card\_num – номер бонусной карты, на по которой начисляются бонусы, bonus – бонусы, d\_create – дата регистрации.

Таблица Dishes содержит информацию о блюде. Содержит поля id – первичный ключ, name – название, category\_d – номер категории блюда, price –цена, image – путь к изображению блюда.

Таблица Dish\_category хранит информацию о категориях блюд. Содержит поля id – первичный ключ, name – название.

Таблица Storage хранит информацию о продуктах на складе. Содержит поля id – первичный ключ, name – название, min\_count – лимит остатка, после которого нужно заказать этот продукт, count – текущее количество на складе, price – стоимость 1 кг. продукта, date – дата добавления на склад, exp-date – срок годности в днях, после которого нужно сделать заказ этого продукта.

Таблица Teh\_card предназначена для хранения информации о технических картах блюд. Содержит поля id\_dish – id блюда, id\_prod – id ингредиента, weight – вес продукта.

Таблица Tables хранит информацию о столах в заведении. Всего два поля id – первичный ключ, name – название.

Таблица Reserv хранит информацию о бронировании стола. Содержит поля id – первичный ключ, id\_table – номер забронированного стола, desc –дополнительное описание бронирования, time – время бронирования, date – дата бронирования.

Таблица Orders хранит общую информацию о заказе. Содержит поля id -первичный ключ, id\_client – id клиента, id\_table – номер стола, sum – общая сумма заказа, id\_emp – id официанта, обслужившего стол, date – дата заказа, time – время заказа.

Таблица Order\_detail нужна для отображения деталей заказа. Содержит поля id\_order – id заказа, id\_dish – id блюда, count – количество заказанного блюда.

## 3.3 Диаграмма размещения

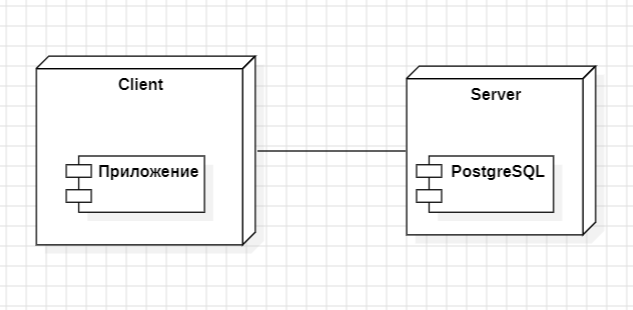


Рис. 9 – Диаграмма размещения

На диаграмме размещения (рис. 9) изображена схема, по которой видно, что информационная система не имеет встроенной базы данных и работает по системе «Клиент-Сервер». На сервере установлена СУБД PostgreSQL, которую выбрали в предыдущей главе. Клиент инициирует запросы, а сервер их обрабатывает и возвращает данные.

## 3.4 Пользовательский интерфейс

Проектируемая система должна иметь приятный и простой дизайн, понятный интерфейс, чтобы пользователь смог с удовольствием пользоваться приложением. Было решено вынести список страниц на левую часть и отделить блок другим цветом, чтобы пользователь смог быстро найти нужную страницу. За страницами спрятаны дополнительные функции, которые отображаются при переходе, чтобы не создавать визуальный шум большим количеством элементов. Сделан акцент на темно-зеленом, синем и белых цветах.

На рисунке 10 изображен прототип интерфейса для редактирования меню. Записи выполнены в виде таблицы, чтобы удобно просмотреть большое количество информации. У каждой строки таблицы есть кнопки редактирования и удаления записи.

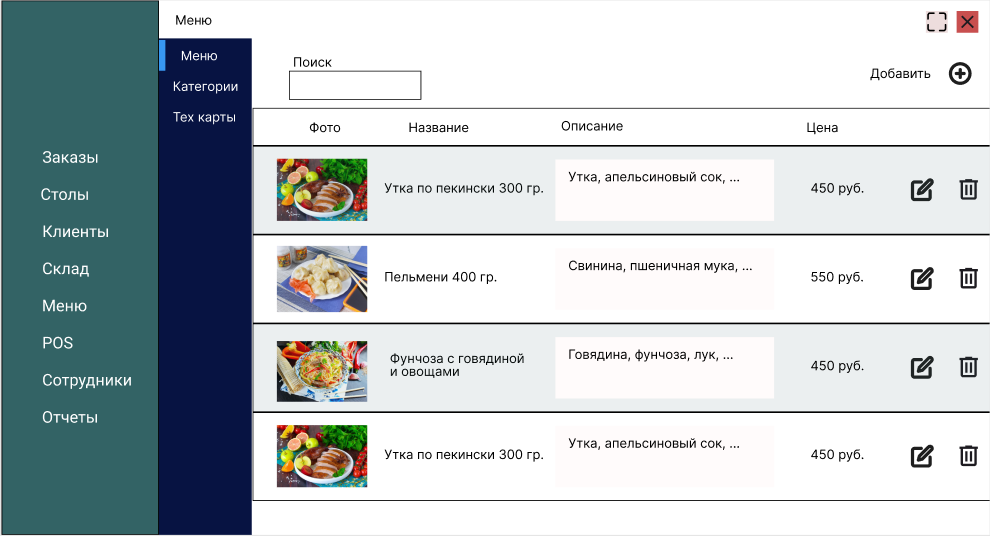
. 

Рис. 10 – Прототип интерфейса меню

Для добавления новой записи в базу данных создан прототип модального окна, который изображен на рисунке 11

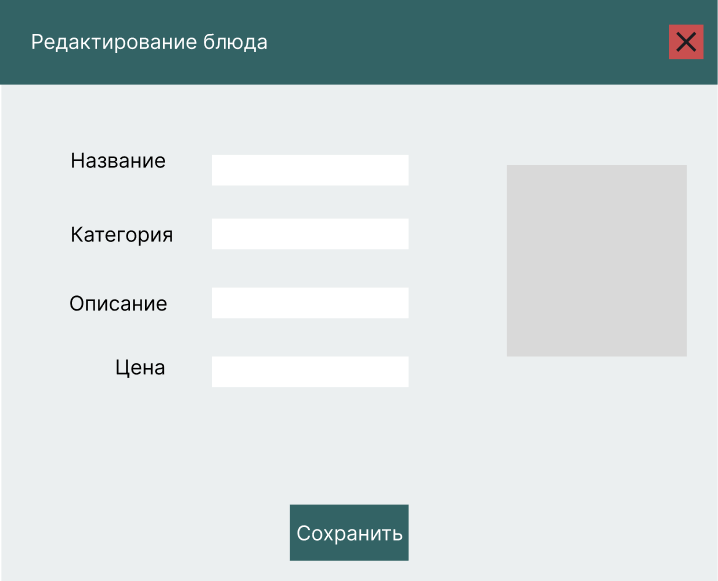


Рис. 11 – Модальное окно для редактирования блюда

Модальное окно имеет поля для ввода информации, а в случае с меню дополнительное место для отображения изображения блюда.

Подобным образом выглядят остальные разделы информационной системы, отличается лишь конкретное наполнение, а для модальных окон отображены соответствующие поля для ввода информации у конкретного раздела.

Для системы регистрации заказов разработан прототип дизайна, изображенного на рисунке 12.

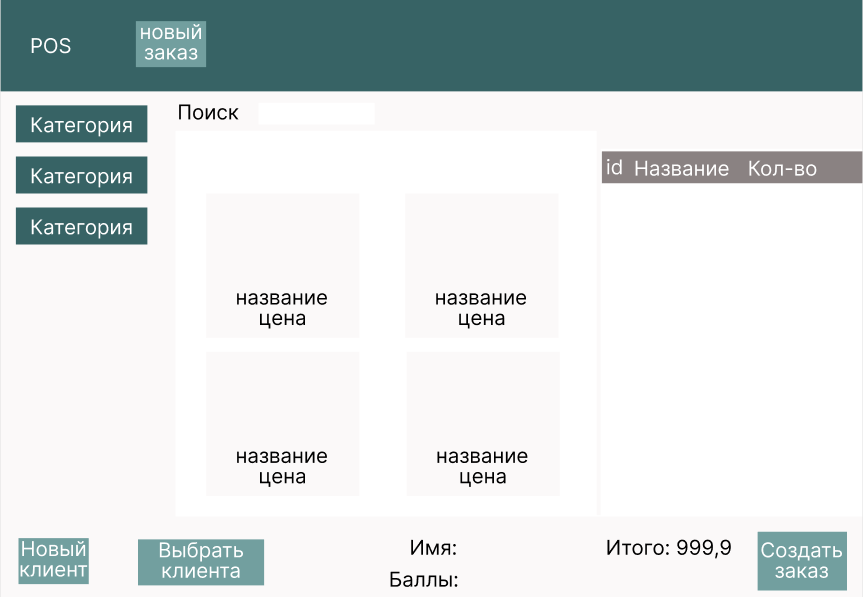


Рис. 12 – Прототип дизайна регистрации заказов

В данном прототипе указано, что пользователь может выбирать блюда из панели по середине, выполнять поиск по категориям или по названию, выбирать клиента или регистрировать нового. В правой панели изображен список текущего заказа, а под ним конечная стоимость и кнопка создания заказа, которая должна внести информацию в базу данных.

## 3.4 Вывод

В третьей главе было выполнено проектирование информационной системы и базы данных, созданы UML диаграммы для администратора, пользователя POS-системы и работы склада для лучшего понимания работы информационной системы. С помощью сервиса Figma был разработан прототип интерфейса системы.

# Глава 4 Разработка информационной системы

В данной главе описываются основные моменты по разработке информационной системы.

## 4.1 Выбор средства для создания настольного приложения

Для разработки настольного приложения на C# чаще всего используют WinForms или WPF.

WPF (Windows Presentation Foundation) и WinForms (Windows Forms) – это две основные технологии для разработки настольных приложений на платформе .NET. WinForms появилась раньше и представляет собой обёртку над стандартными элементами управления Windows, что делает её простой в освоении и быстрой в разработке. WPF предлагает более современный подход, основанный на использовании языка разметки XAML для описания интерфейса. Такой подход позволяет отделить визуальную часть от логики приложения, а также создавать сложные пользовательские интерфейсы с анимацией, шаблонами и привязкой данных.

Несмотря на то, что WinForms считается устаревшей технологией, она до сих пор используется во многих проектах в угоду своей простоты разработки.

Для разработки информационной системы подходит WinForms, так как для разработки подобных систем не требуется анимированный интерфейс, а скорость разработки значительно увеличится.

## 4.2 Разработка интерфейса и логики ИС

Для создания интерфейса использовалась библиотека Guna UI 2 для WinForms. Данная библиотека позволяет создавать современные и красивые интерфейсы для программ на WinForms. Библиотека устанавливается через встроенный пакетный менеджер Microsoft Visual Studio NuGet.

Была разработана форма авторизации в систему, изображенная на рисунке 12, в которой имеется два поля для ввода логина и пароля. Если в базе данных имеется соответствующая запись, то приложение откроет основную форму, иначе появится сообщение о неверном логине или пароле.

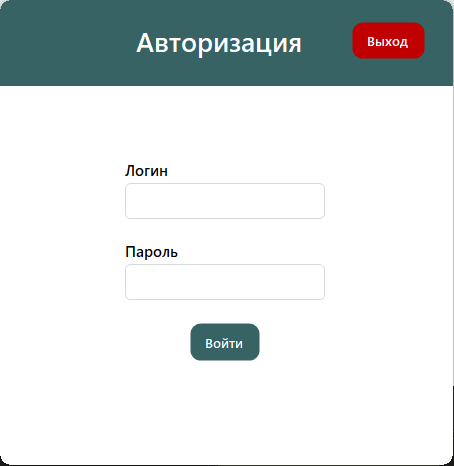


Рис. 13 – Окно авторизации

После успешной авторизации открывается форма главного меню (рис. 14), которое приветствует пользователя. На данный момент в этом окне имеется лишь одна запись. В левой части интерфейса расположены страницы с группированным функционалом: «Заказы», «Столы», «Клиенты», «Склад», «Меню», «POS», «Сотрудники», «Отчеты». У каждой страницы имеется своя иконка для явного разделения.

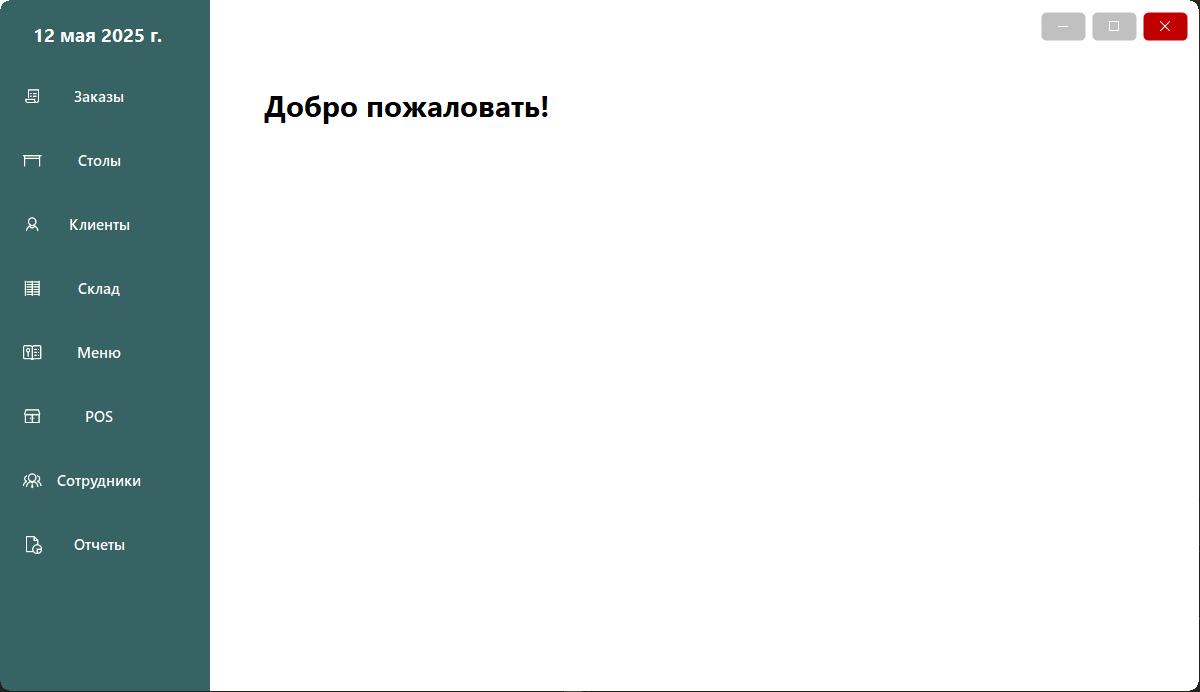


Рис. 14 – Главная страница

При нажатии на вкладку, у центральной панели заполняется на содержимое страницы. Первой по списку идет страница «Заказы», в которой отображен список всех заказов, выполненный в виде таблицы (рис. 15). В таблице указаны дата, время, имя клиента, стол, обслуживший официант и сумма заказа.

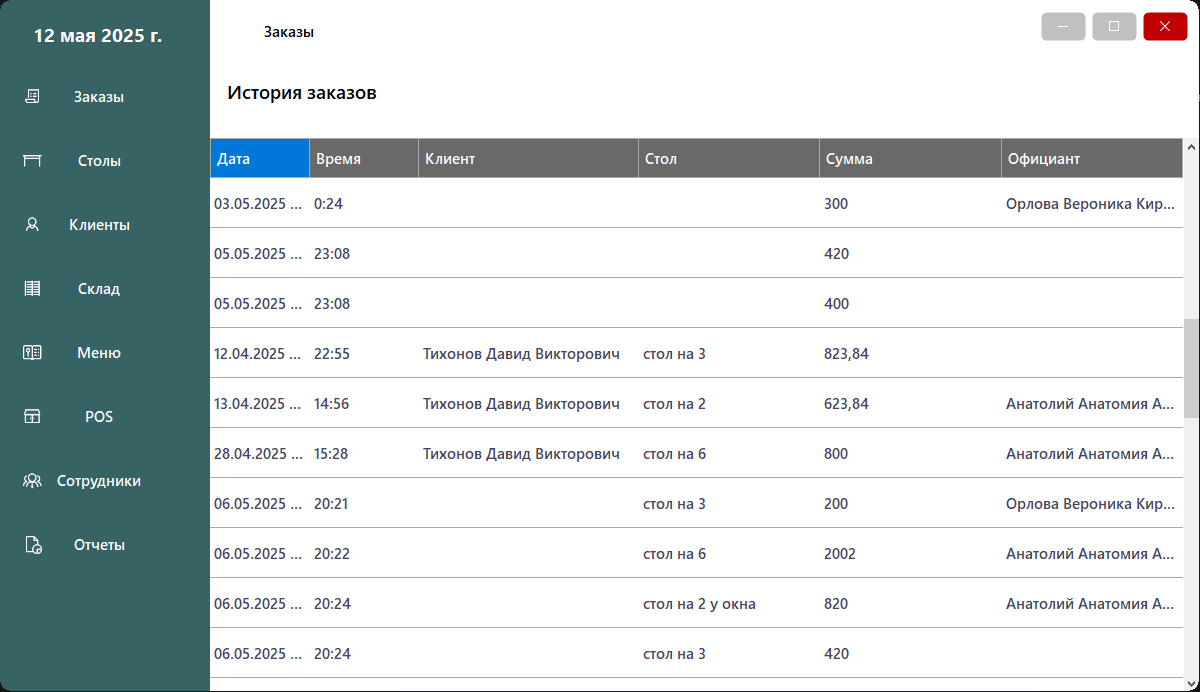


Рис. 15 – Заказы

Взаимодействие с базой данный происходит через ADO.NET. ADO.NET – это технология в составе платформы .NET, предназначенная для работы с данными в базах данных [23]. Она предоставляет набор классов для подключения к базам данных, выполнения запросов и обработки данных. Чтобы установить соединение с PostgreSQL, необходимо установить соответствующий пакет Npgsql, через пакетный менеджер NuGet, после чего станет доступна полноценная работа с базой данных.

Оперирование данными осуществляется через автономный уровень, в котором данные сохраняются в коллекцию на стороне клиента. Происходит это так:

1. подготавливается запрос на получение данных,
2. открывается соединение с базой данных,
3. выполняется команда,
4. получение данных;
5. соединение с базой данных закрывается.

Команда формируется через метод NpgsqlCommand, в котором пишется запрос и строка подключения к базе данных. Далее, через NpgsqlDataReader программа получает ответ от базы данных и заносит информацию, в данном случае в DataGridView. В случае возникновения ошибки, NpgsqlDataReader возвращает сообщение, в котором описана ошибка, и выполнение команды прекращается.

На рисунке 16 изображена страница «Столы», в которой имеется дополнительная вкладка «Бронь столов». На форме имеется поиск по названию столов, кнопка для добавления нового стола и таблица, содержащая в себе имеющиеся столы в базе данных. Напротив каждой записи есть кнопки, позволяющая редактировать запись, либо удалить ее.

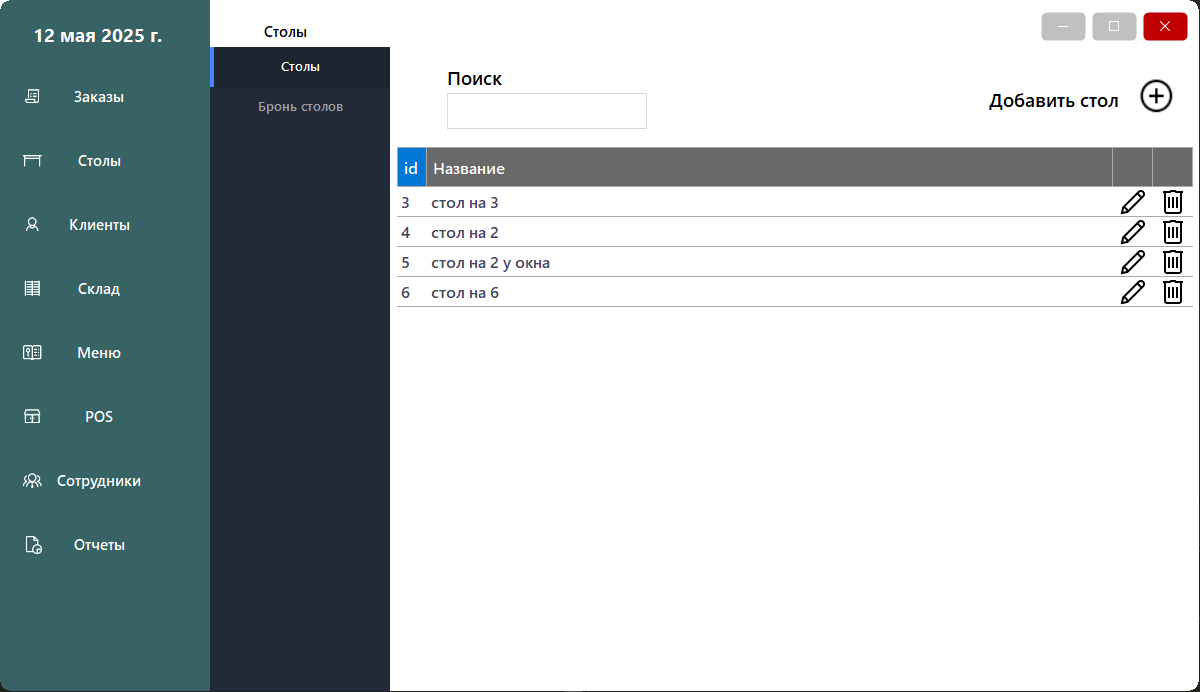


Рис. 16 – Столы

Так как данные хранятся в DataGridView и представляют из себя записи в таблице, то необходимо создать обработчик событий, который обрабатывает нажатие по таблице. Каждый столбец таблицы имеет отображаемое название, например «id» и «Название», и имя для разработки и взаимодействия с этим столбцом. Любое нажатие на таблицу обрабатывается событием CellContentClick, в котором можно указать различные функции. При нажатии на столбец с иконкой карандаша проверяется событие, что нажатие было сделано на столбец «edit», после чего открывается форма для редактирования, тот же процесс происходит для столбца удаления.

Для создания, редактирования и удаления создана одна форма модального окна (рис. 17), в которой меняются кнопки в зависимости от команды. При редактировании записи, поля заполняются информацией из таблицы, чтобы пользователю не пришлось переписывать существующую информацию. Кнопки «Сохранить» и «Изменить» становятся видимыми и действующими при определенной команде, полученной от обработки команды CellContentClick.

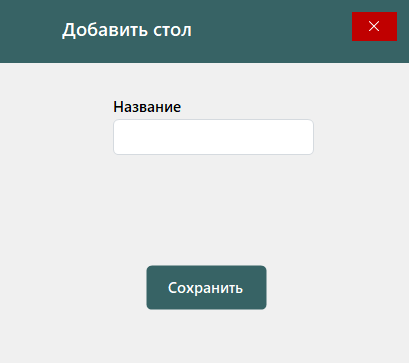


Рис. 17 – Форма добавления стола

Для того, чтобы сохранить введенную информацию, пользователь должен нажать кнопку «Сохранить», после чего формируется строка на добавление новой записи в базу данных. После формирования строки, в базу данных добавляется созданная запись.

Похожим образом выглядит соседняя вкладка «Бронь столов», содержащая информацию о забронированных столах под конкретную дату и время (рис. 18).

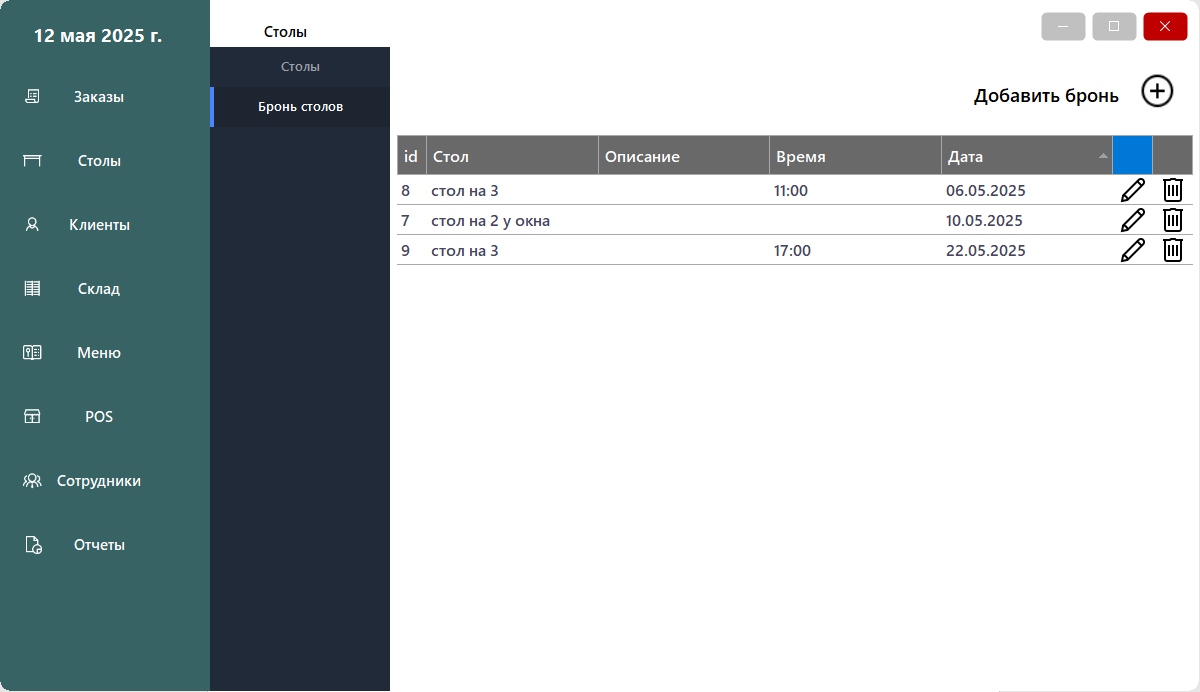


Рис. 18 – Бронь столов

Администратор может создавать новую бронь, редактировать существующую, либо удалить запись.

Страница «Клиенты» отображает всех зарегистрированных клиентов (рис. 19). Функционал схож со страницей «Столы». Для Администратора доступны функции по добавлению клиента, поиску, редактированию и удалению.

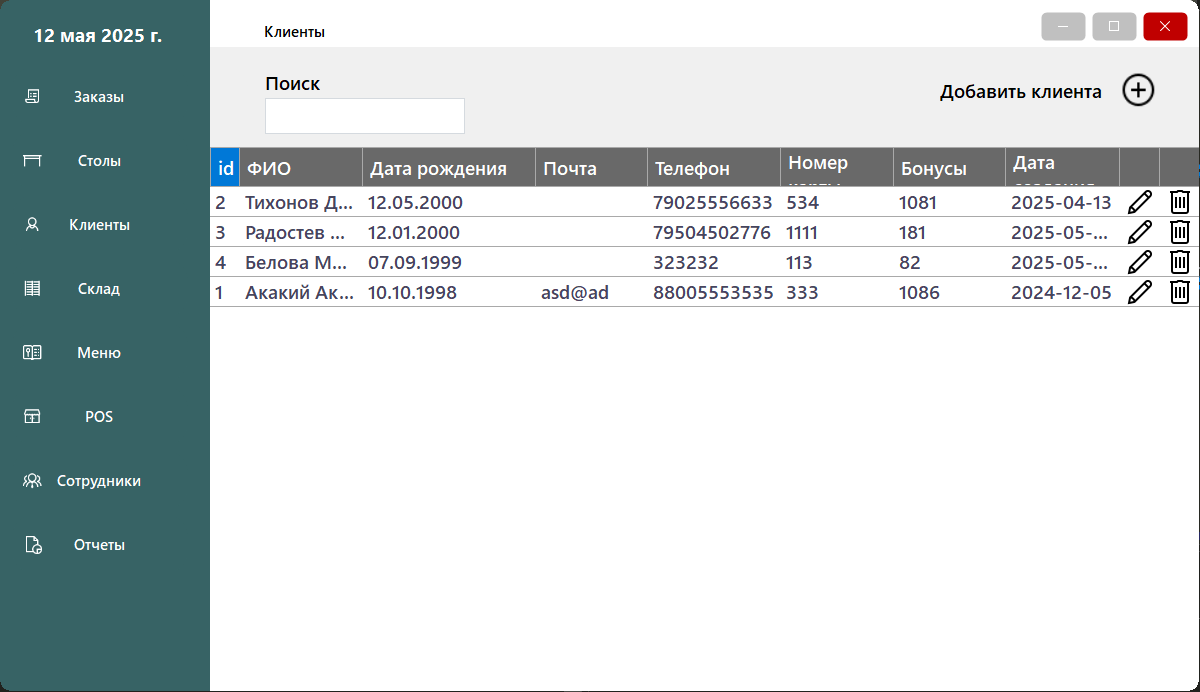


Рис. 19 – Клиенты

Страница «Склад» (рис. 20) представляет собой несколько вкладок, в которых можно просмотреть текущее состояние складских запасов и составить заявку на пополнение склада. Первая вкладка «Ингредиенты» хранит информацию о текущем состоянии склада. При загрузке данных из базы данных происходит проверка на соответствие минимальному количеству и срокам годности. Если у конкретного продукта текущее количество будет меньше минимального, то текст ячейки окрасится в красный цвет, чтобы пользователь смог акцентировать на этом внимание. Тоже самое происходит для проверки сроков годности. К дате изменения прибавляется количество дней, которое делает продукт годным к приготовлению, и если полученная дата при сравнении с текущей выходит из срока годности, то текст даты изменения становится красным.

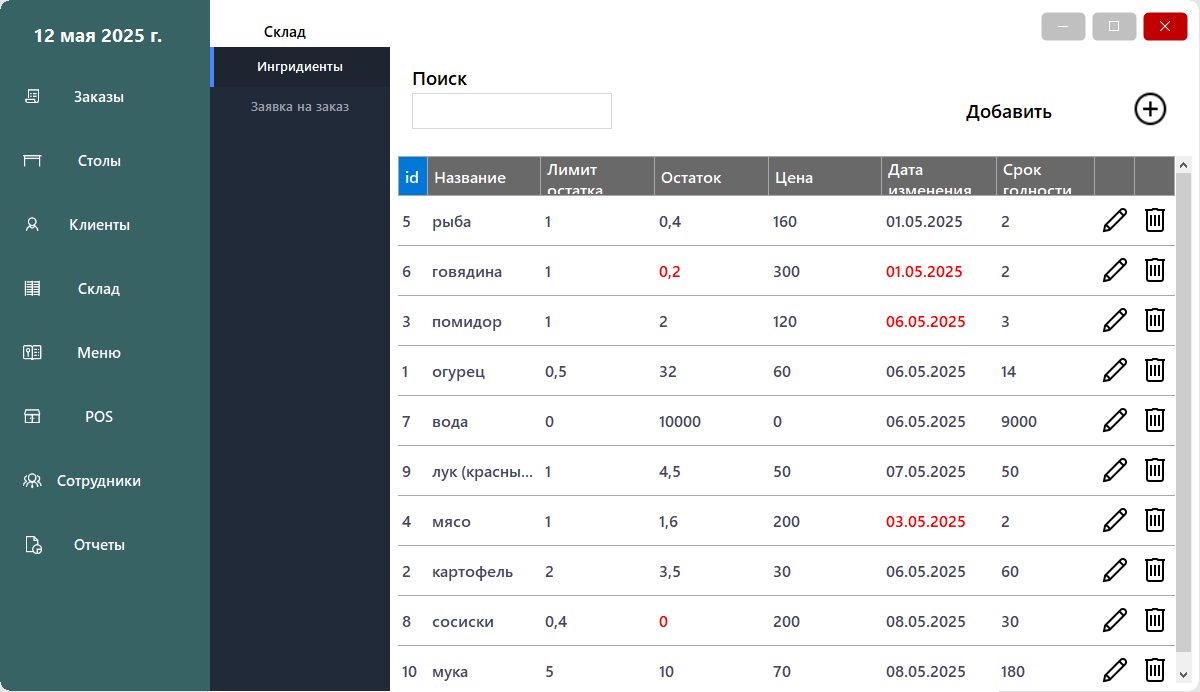


Рис. 20 – Ингредиенты

В модальном окне для добавления продукта происходят те же процессы, что и для других модальных окон (рис. 21). Для того, чтобы добавить продукт в базу данных, необходимо заполнить все поля. Для полей «Остаток в кг», «Минимальный остаток», «Цена за кг», «Срок годности в днях» проводится проверка на ввод числа, чтобы избежать ошибок при внесении информации в базу данных. В таблицу Storage в столбец date базы данных вносится дата, когда этот продукт был добавлен, от этого дня начинается отсчет для вычисления срока годности. При изменении продукта, дата изменения автоматически меняется на текущий день.

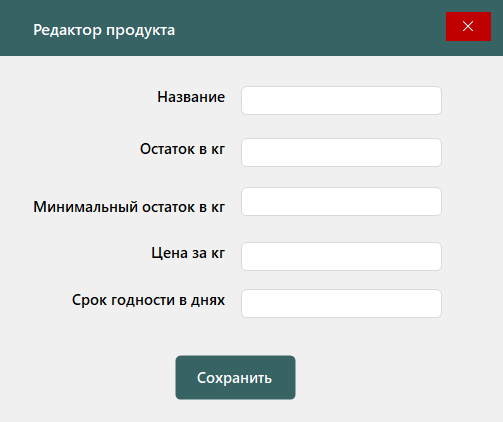


Рис. 21 – Редактор продуктов

Во вкладке «Заявка на заказ» (рис. 22) происходит автоматическое заполнение таблицы по текущему состоянию склада. Логика заполнения была описана в диаграмме деятельности, но была добавлена возможность пользователю заносить свои данные в последнюю строку таблицы, после заполнения которой автоматически появится еще одна строка.

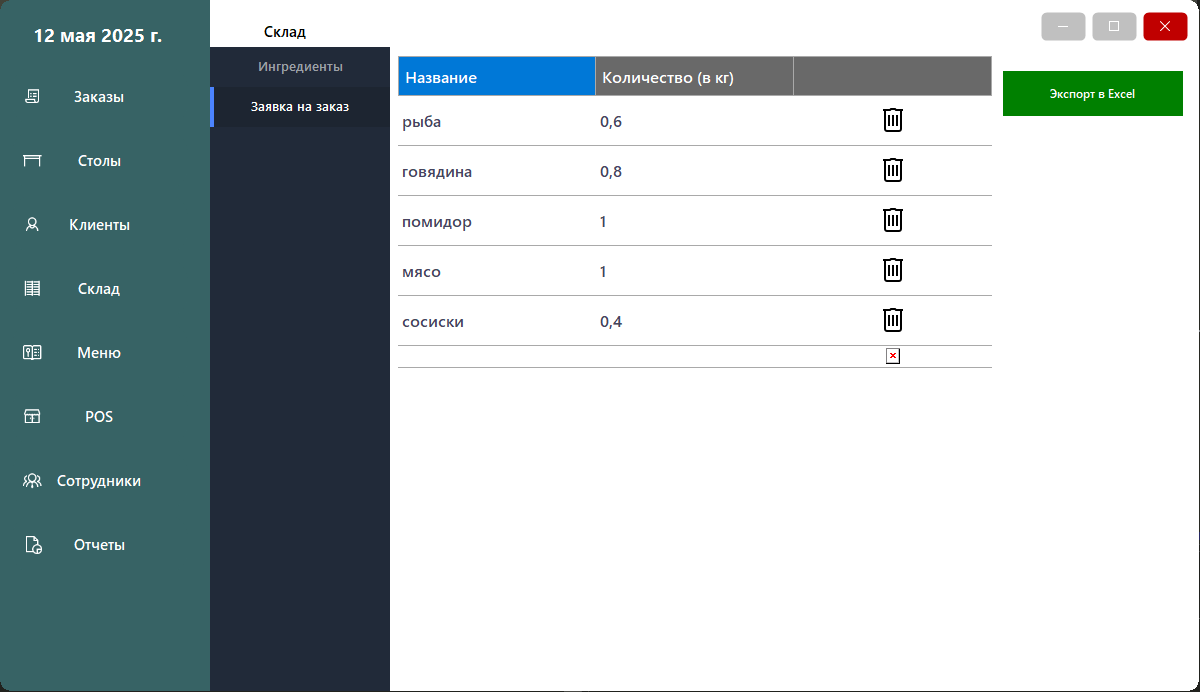


Рис. 22 – Заявка на заказ

После заполнения таблицы, пользователь может нажать кнопку «Экспортировать в Excel», после чего откроется окно, в котором ему нужно указать путь для сохранения файла. С помощью пакета Microsoft.Office.Interop.Excel формируется документ, который можно настроить, указывая настройки для ячеек и записывая в них информацию. Данный пакет требует наличия установленной версии Microsoft Excel. В дальнейшем пользователь может самостоятельно отредактировать документ, если это необходимо. После формирования документа список не очистится, так как берется текущая информация из базы данных, которая не была обновлена.

Страница «Меню» содержит в себе 3 вкладки: «Меню», «Категории», «Тех карты» (рис. 23). Для блюд предусмотрено отображение фотографий в таблице. Для этого в базе данных хранятся ссылки к изображениям, которые хранятся на компьютере пользователя. В случае, если изображение изменит свое местоположение, либо перестанет существовать, на месте фотографии отобразиться изображение «заглушка» в виде креста. Изображение строится через поток байт.

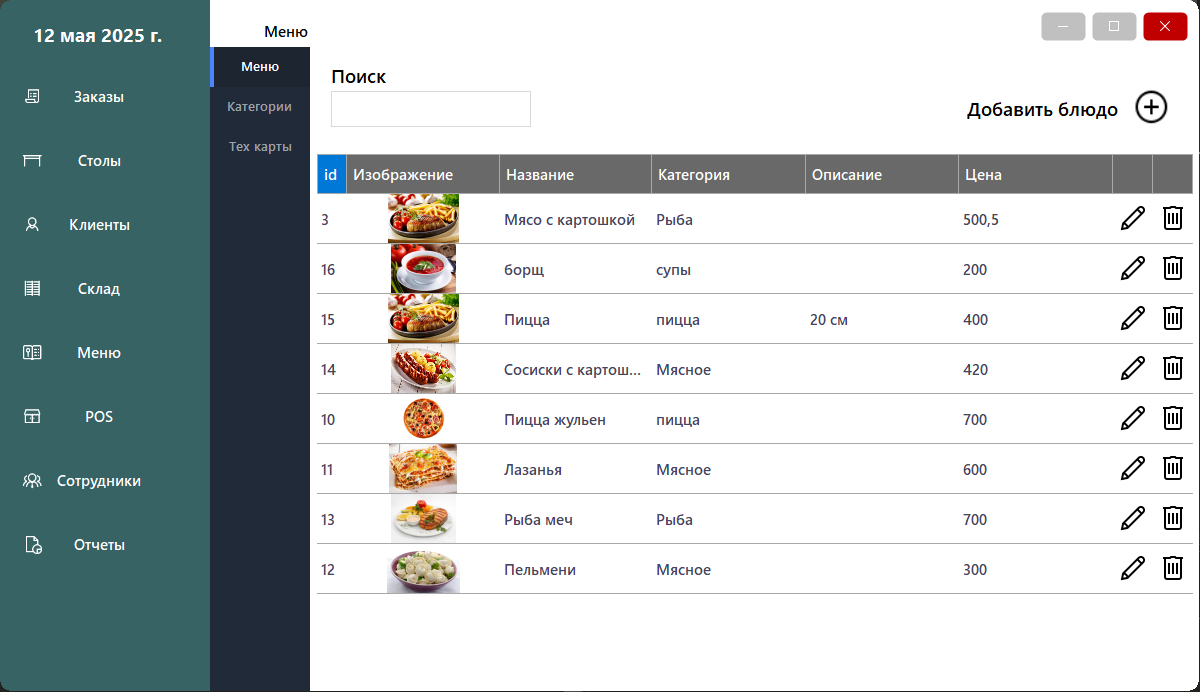


Рис. 23 – Меню

Редактор блюд выглядит следующим образом (рис. 24). Пользователь может ввести название блюда, указать ему категорию, добавить описание и указать цену, для которой проводиться проверка на число, чтобы избежать ошибок при внесении записи в базу данных. Также пользователь может загрузить изображение, нажав на иконку под предпросмотром изображения. Принимаются изображения с форматами .jpg, .jpeg. png.

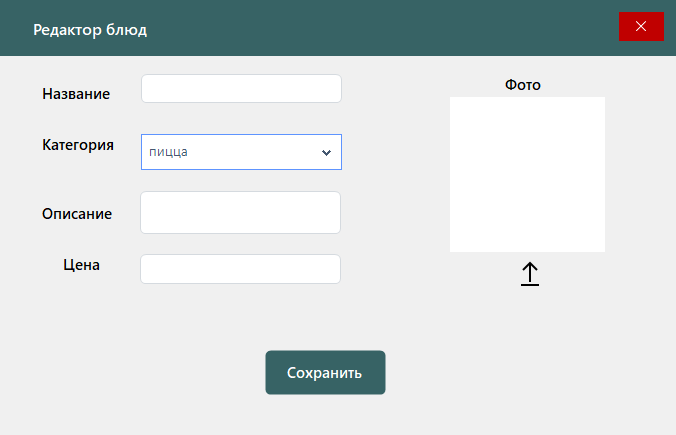


Рис. 24 – Редактор блюд

Вкладка «Категории» имеет тот же функционал по поиску, добавлению, редактированию и удалению записей (рис. 25). Редактируется единственный параметр «название».

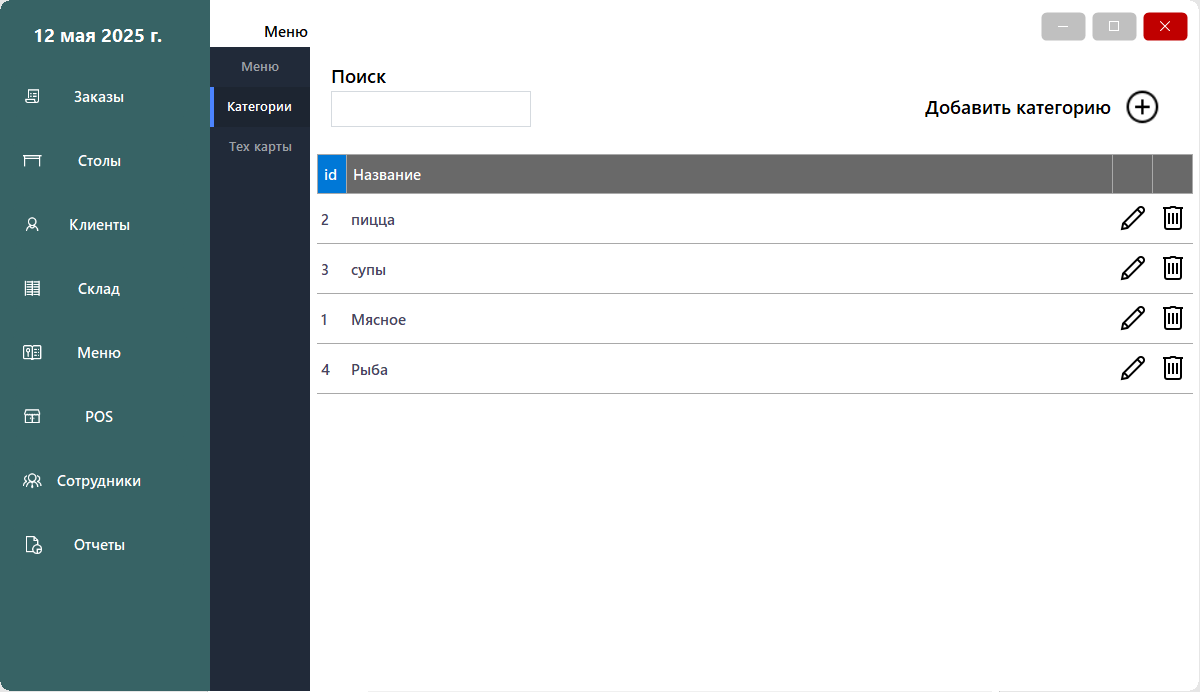


Рис. 25 – Категории

Вкладка «Тех карты» (рис 26) имеет тот же функционал по поиску, редактированию, удалению. Таблица заполняется только существующими блюдами из базы данных, для которых можно составить техническую карту. При нажатии на кнопку удаления, из базы данных удаляются все записи о технической карте для этого блюда. При нажатии на кнопку редактирования открывается редактор тех карты, указывая в заголовке название редактируемого блюда.

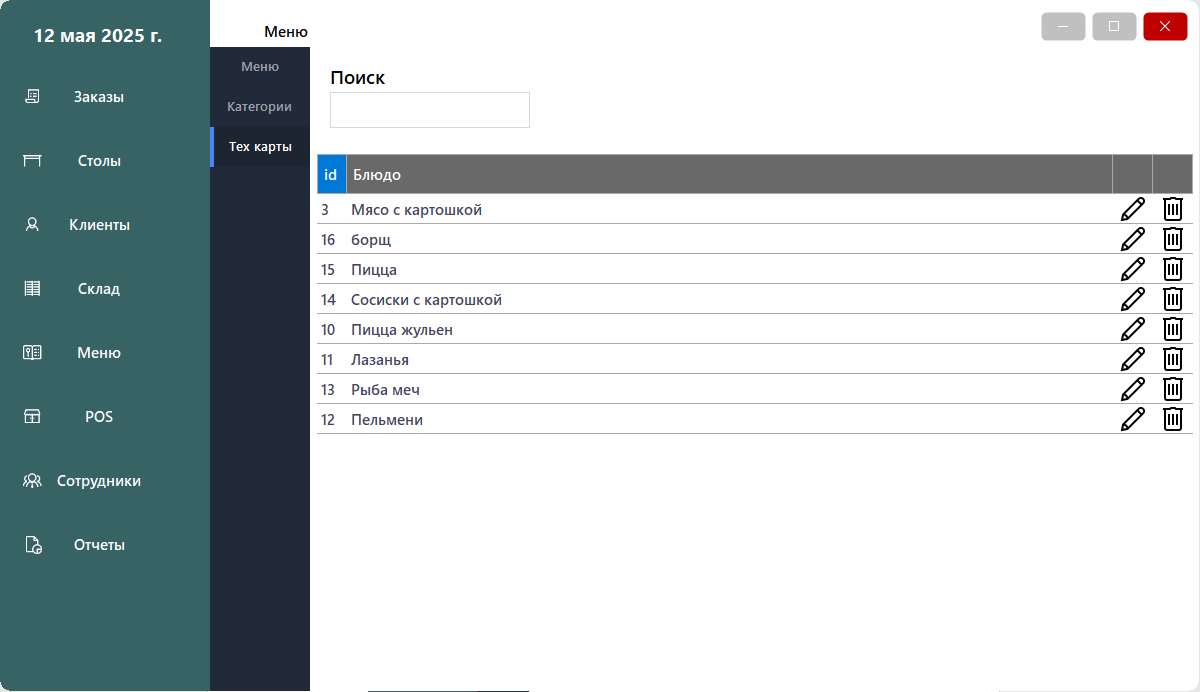


Рис. 26 – Технические карты

Для того, чтобы составить техническую карту, проводится сканирование склада на существующие продукты. На основе этой информации создается класс Product, хранящий в себе поля id, name, price. Заполняется список, состоящий из элементов класса Product, который становится источником данных для ComboBox, расположенной в таблице. Пользователь выбирает продукт из выпадающего списка, указывает вес в килограммах для этого продукта и может перейти на следующую строку (рис. 27). Чтобы рассчитать себестоимость блюда, предусмотрена кнопка, которая вычисляет стоимость по продуктам и их весу в блюде. После завершения составления тех карты, пользователь нажимает кнопку «Сохранить», таким образом записывая информацию в базу данных. Для каждого отдельного ингредиента формируется своя запись в базу данных.

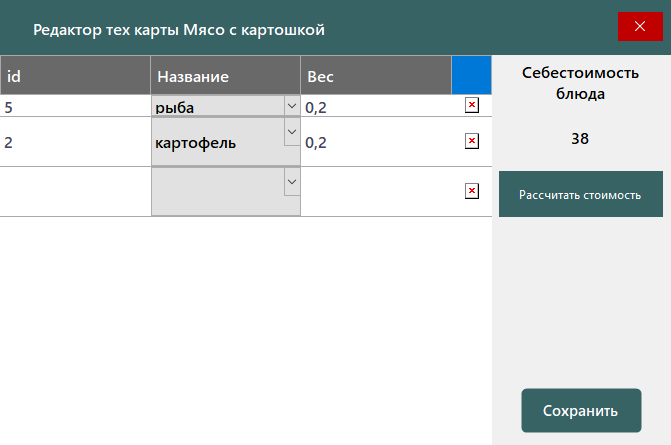


Рис. 27 – Редактор технической карты

Страница «Сотрудники» отображает информацию о сотрудниках заведения (рис. 28). Имеется функционал по поиску, добавлению, редактированию и удалению сотрудника. В редакторе сотрудника можно присвоить ему одну из должностей: повар, шеф-повар, су-шеф, официант, бармен, уборщик.

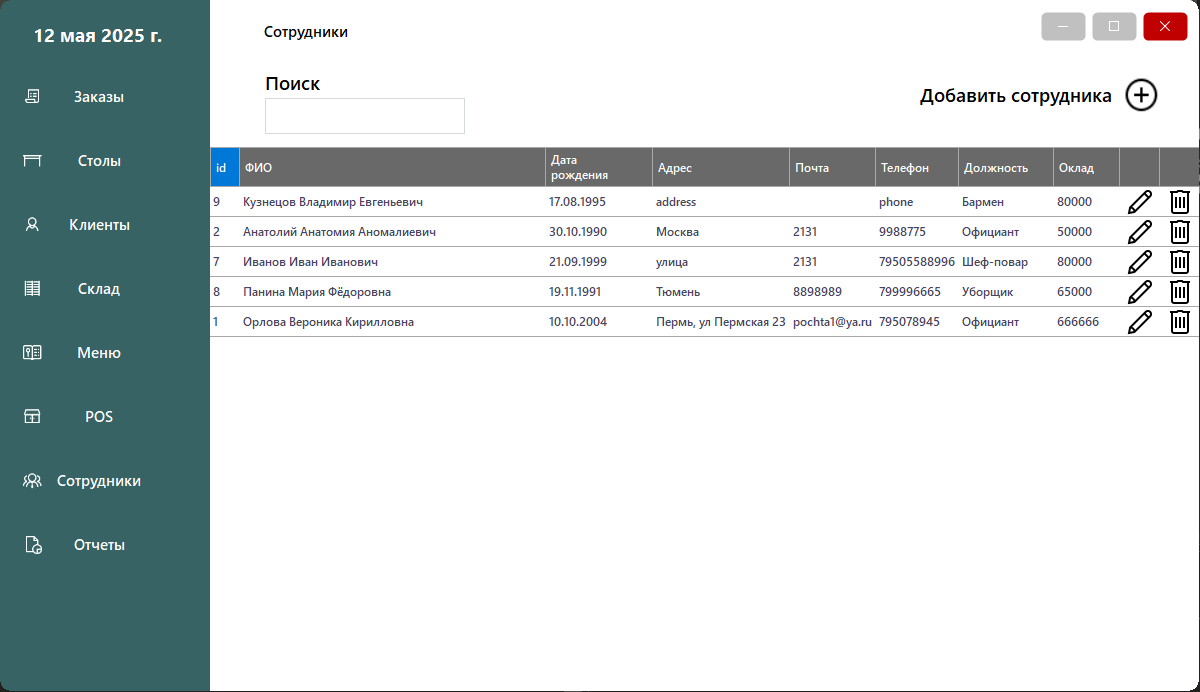


Рис. 28 – Сотрудники

Страница «Отчеты» отображает детали для аналитики работы заведения. В первой вкладе «Заказы» отображается количество созданных заказов по дням, чтобы Администратор смог сделать выводы, по каким дням наибольшее количество заказов. Отображение в виде графика для наглядного представления, а при наведении на точку графика отобразится точное количество.

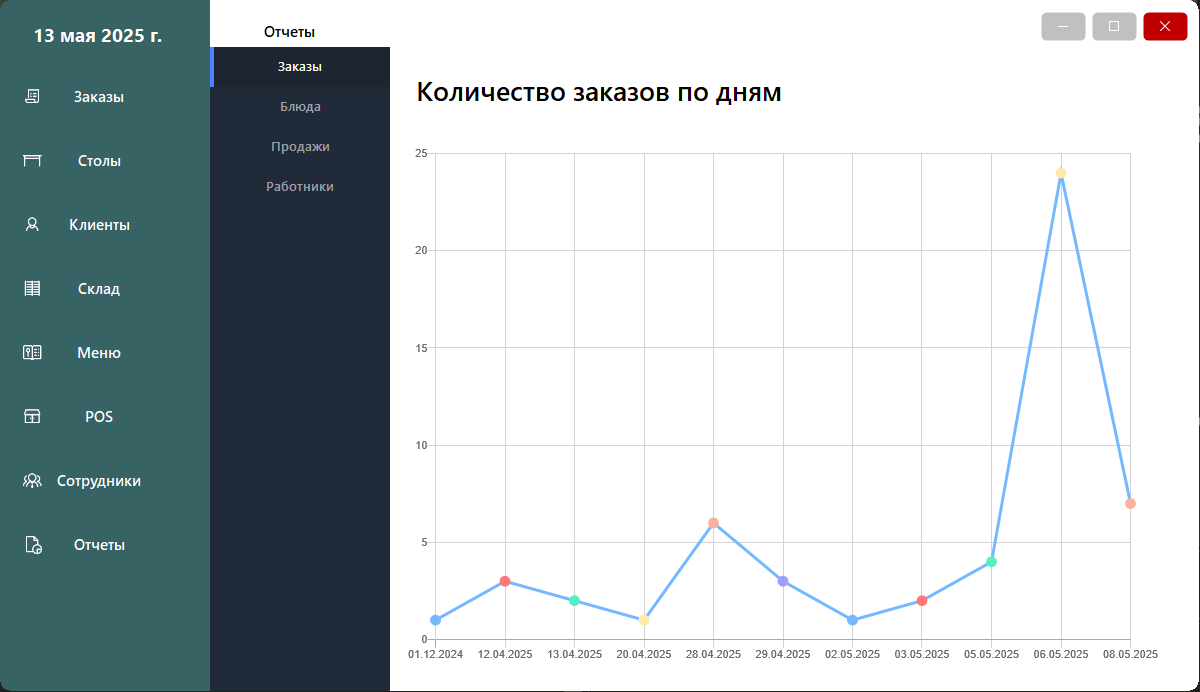


Рис. 29 – Количество заказов

Вкладка «Блюда» отображает количество заказанных блюд в виде столбчатой диаграммы (рис 30). Пользователь может указать период, для которого отобразится диаграмма. При наведении мыши на столбец отобразится точное количество для конкретного блюда.

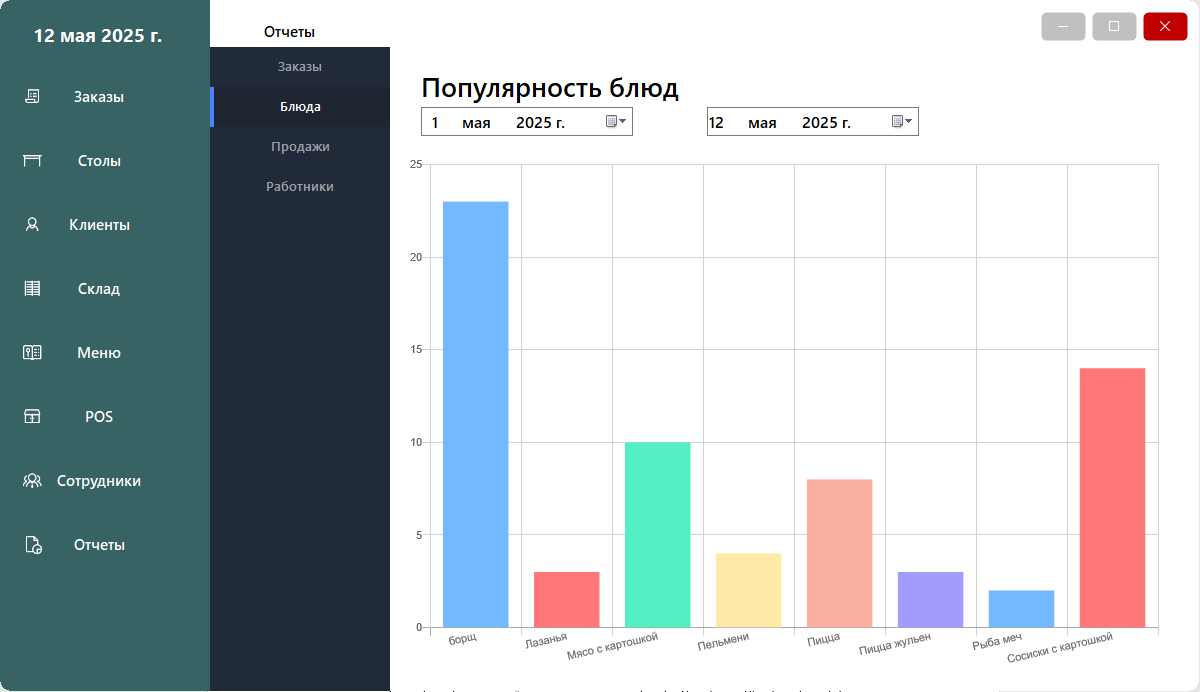


Рис. 30 – Популярность блюд

Вкладка «Продажи» предлагает администратору просмотреть заработок в выбранный период (рис. 31). Диаграмма отобразит заработок в виде столбчатой диаграммы, при наведении на столбец покажется точная сумма за конкретный день. Справа от графика отображена заработанная сумма за период.

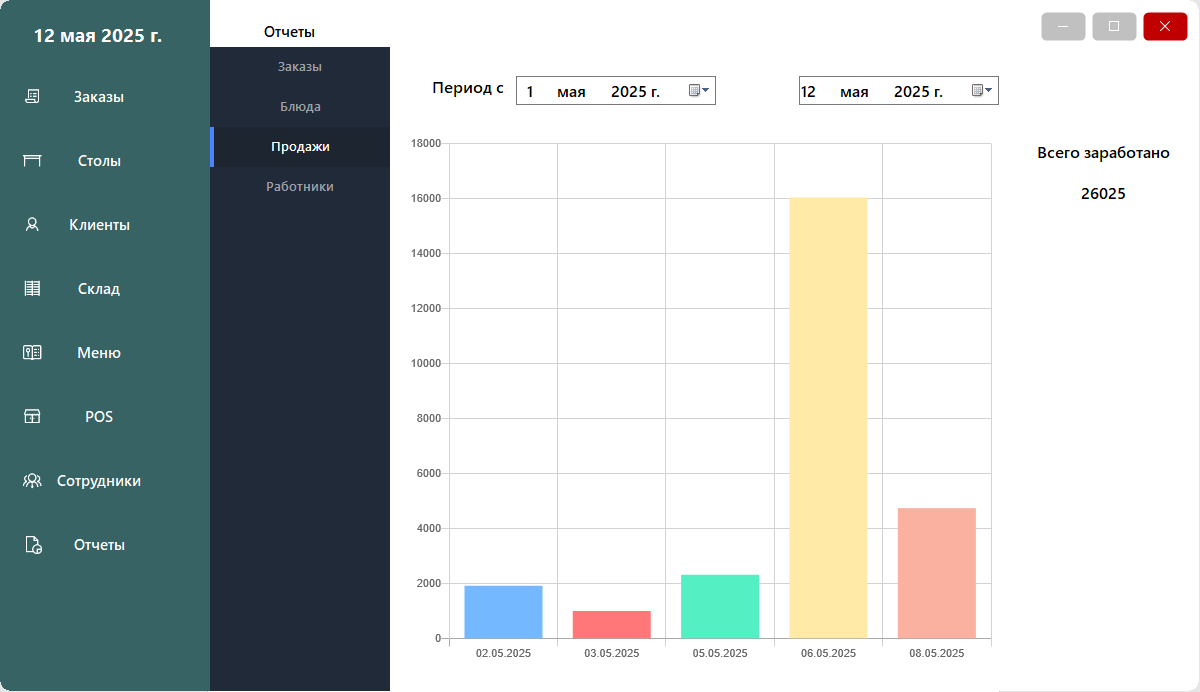


Рис. 31 – Продажи

Вкладка «Работники» отображает количество обслуженных клиентов по всем официантам за определенный период времени (рис. 32). Статистика отображена в виде столбчатой диаграммы, при наведении на столбец отобразится точное число обслуженных клиентов каждым официантом. Самые активные работники могут быть дополнительно награждены.



Рис. 32 – Работники

Все графики были отрисованы с помощью библиотеки Guna.Charts.WinForms. Данная библиотека позволяет отображать красивые и информативные графики с анимацией. Для отображения диаграммы с базы данных выбираются записи, помещаются в хранилище данных и на основе типа таблицы отображается определенный тип диаграммы. Библиотека предлагает 16 типов диаграмм, поддерживается построение графиков в реальном времени без принудительного обновления формы.

На рисунке 33 изображена система POS (Point of Sale) – система, которая регистрирует все продажи в заведении. Такие системы автоматизируют действия, связанные с заполнением заказа клиента, сохранением в базу данных информации о заказе, списание продуктов со склада и многое другое. POS системе необходимо отображать текущие блюда и создавать заказ, для создания информации о работе заведения.

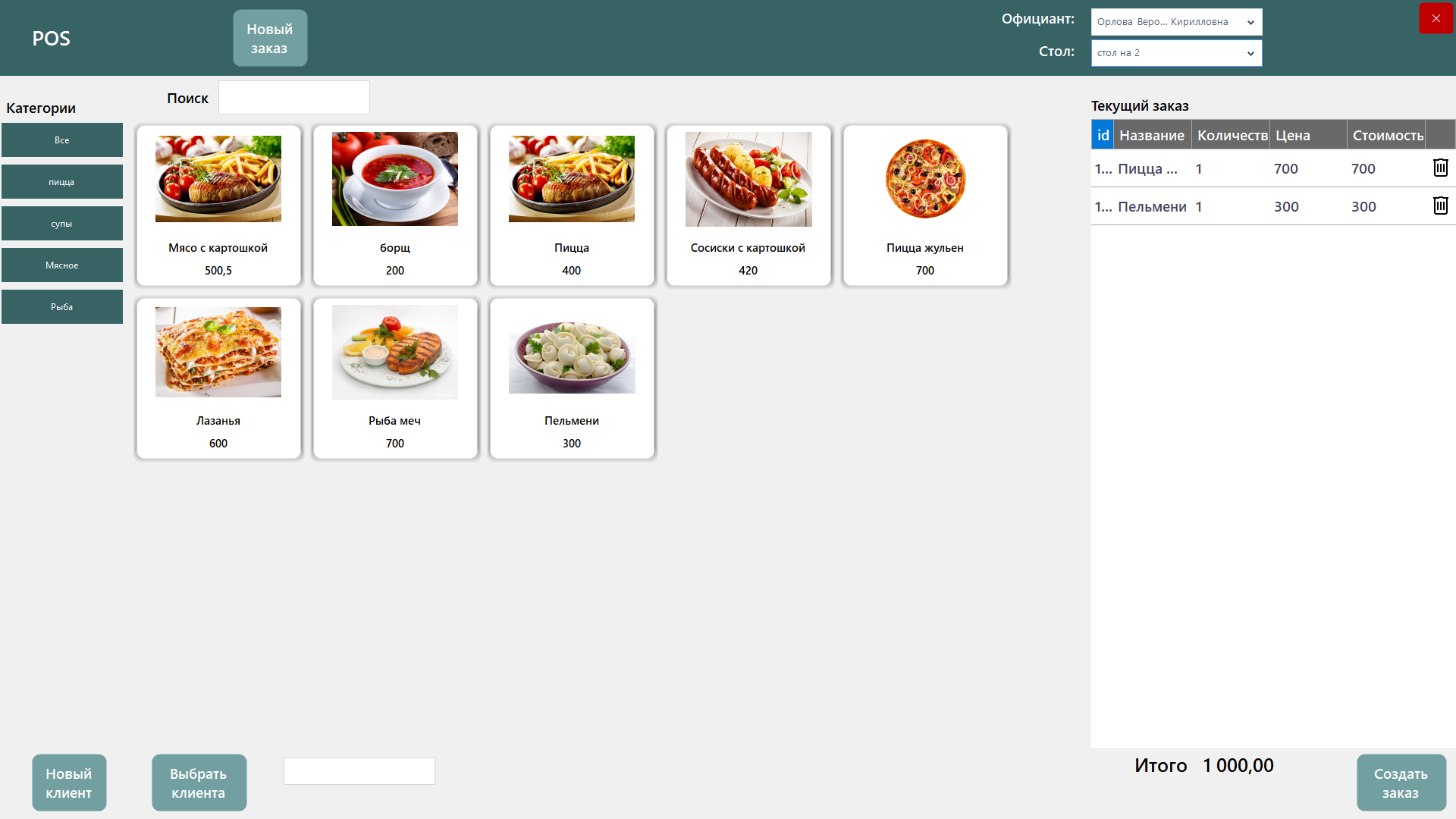


Рис. 33 – POS

Центральная панель представляет собой список из существующих блюд, при нажатии на изображение в заказ добавится одна позиция данного блюда. Чтобы быстро выполнить поиск определенной позиции, были создан поиск по категориям и названию.

Для отображения блюд на центральной панели был создан компонент, содержащий информацию о названии, цене и ссылки изображение блюда.

Пользователь выбирает блюдо из панели, после чего эта позиция добавляется в список текущего заказа. Блюдо может быть выбрано несколько раз, и эта информация отобразится в соответствующем столбце, в случае ошибки можно удалить позицию из заказа. Итоговая стоимость отображается под текущим заказом возле кнопки «Создать заказ».

После нажатия кнопки «Создать заказ» происходит заполнение деталей заказа в базу данных, а итоговый заказ заносится в таблицу Orders. Для списания продуктов по техническим картам была создана триггерная функция в СУБД PostgreSQL (рис.34), которая автоматически списывает продукты со склада при внесении новых записей в таблицу с деталями заказа, если для данного блюда существует техническая карта.

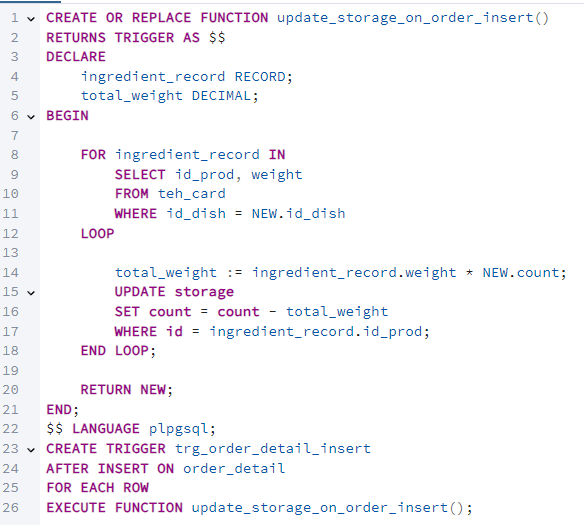


Рис. 34 – Триггерная функция

Также был добавлен функционал по поиску клиента по его номеру карты, а если клиент пришел в первый раз, то его можно сразу зарегистрировать. Для этого используется ранее созданная форма для добавления клиентов из главного меню. Для текущего заказа есть возможность выбрать официанта, который будет обслуживать данного клиента, и выбрать стол, за который можно посадить клиента.

## 4.3 Вывод

В данной главе были описаны основные моменты по разработке информационной системы, выбраны средства разработки, описан процесс создания графического интерфейса, использование библиотек для облегчения разработки, описана логика работы информационной системы. Были реализованы все задачи, поставленные при проектировании. Программный код информационной системы доступен по ссылке, расположенной в приложении Б. В приложении В представлено созданное руководство пользователя.

# Заключение

В результате проделанной работы была разработана информационная система для ресторанного бизнеса. Данная система должна облегчить работу администратора заведения.

Анализ предметной области показал, что подобные системы являются актуальными и востребованными, но имеют широкий функционал и требуют больших финансовых затрат для их использования.

В ходе выполнения работы были проделаны следующие этапы:

* проведен анализ существующих информационных систем и были выделены особенности функционирования для разрабатываемой информационной системы;
* проведен анализ и выбрано инструментальное программное обеспечение для проектирования и разработки информационной системы;
* спроектирована информационная система, разработан прототип графического интерфейса;
* разработана информационная система;
* создана документация.

Таким образом, были выполнены все поставленные задачи. Следовательно, цель данной работы достигнута.

# Библиографический список

1. Россияне пошли по ресторанам. Трафик в общепите вырос [Электронный ресурс] // Konkurent – URL: <https://konkurent.ru/article/67225> (дата обращения 01.05.2025)
2. CRM-системы: что это простыми словами, как работает, примеры, внедрение и интеграция [Электронный ресурс] // kp – URL: <https://www.kp.ru/money/biznes/chto-takoe-crm-sistemy/> (дата обращения 04.05.2025)
3. Простая и понятная система автоматизации для кафе, ресторанов и общепита | Restik [Электронный ресурс] // restik – URL: <https://restik.com/automation/> (дата обращения 04.05.2025)
4. Программа для общепита Fusinon POS: автоматизация кафе, рестораны, бары [Электронный ресурс] // fusionpos – URL: <https://fusionpos.ru/> (дата обращения 04.05.2025)
5. Программы для общепита – программное обеспечение для ресторанов и кафе | QuickResto [Электронный ресурс] // quickresto – URL: <https://quickresto.ru/> (дата обращения 04.05.2025)
6. iiko – система автоматизации для кафе, баров, службы доставки, ресторанов [Электронный ресурс] // iiko – URL: <https://iiko.ru/> (дата обращения 04.05.2025)
7. YUMA – Система автоматизации ресторанов, кафе, баров, доставки еды [Электронный ресурс] // yumapos – URL: [https://yumapos.ru/#](https://yumapos.ru/) (дата обращения 04.05.2025)
8. Шаврин С.М., Лядова Л.Н., Чуприна С.И. Моделирование и проектирование информационных систем: учебно-методическое пособие. – Пермь: Пермский государственный университет, 2007. – 7 с.
9. CASE средства [Электронный ресурс] // kpms – URL: <https://www.kpms.ru/Automatization/CASE_tools.htm> (дата обращения 03.05.2025)
10. StarUML [Электронный ресурс] // staruml – URL: <https://staruml.io/> (дата обращения 03.05.2025)
11. Powerful and Fast UML Diagramming Software – Astah [Электронный ресурс] // astah – URL: <https://astah.net/products/astah-uml/> (дата обращения 03.05.2025)
12. Автограф – графический редактор схем и чертежей [Электронный ресурс] // avtograf.pro – URL: <https://avtograf.pro/> (дата обращения 03.05.2025)
13. Draw.io [Электронный ресурс] // architect-design – URL: https://www.drawio.com/ (дата обращения 13.05.2025)
14. SQlite Documentation [Электронный ресурс] // sqlite – URL: <https://www.sqlite.org/docs.html> (дата обращения 03.05.2025)
15. PostgreSQL: Documentation [Электронный ресурс] // postgresql – URL: <https://www.postgresql.org/docs/> (дата обращения 03.05.2025)
16. Обзор C#: почему он актуален и какие у него перспективы [Электронный ресурс] // tproger – URL: <https://tproger.ru/articles/aktual-nost-c-i-vybor-ego-kak-pervogo-yazyka-programmirovaniya> (дата обращения 03.05.2025)
17. Python [Электронный ресурс] // python – URL: https://www.python.org/ (дата обращения 03.05.2025)
18. Документация по C++ [Электронный ресурс] // learn.microsoft – URL: https://learn.microsoft.com/ru-ru/cpp/cpp/?view=msvc-170 (дата обращения 03.05.2025)
19. Visual Studio 2022 Community Edition [Электронный ресурс] // Microsoft – URL: <https://visualstudio.microsoft.com/ru/vs/community/> (дата обращения 03.05.2025)
20. Figma: The Collaborative Interface Design Tool [Электронный ресурс] // figma – URL: https://www.figma.com/ (дата обращения 04.05.2025)
21. MockFlow – Online Wireframing and Product Design Tool [Электронный ресурс] // mockflow – URL: https://mockflow.com/ (дата обращения 04.05.2025)
22. NinjaMock online wireframe and mockup tool [Электронный ресурс] // ninjamock – URL: https://ninjamock.com/home/index (дата обращения 04.05.2025)
23. Обзор ADO.NET [Электронный ресурс] // professorweb – URL: https://professorweb.ru/my/ADO\_NET/base/level1/1\_1.php (дата обращения 12.05.2025)

# Приложение А Диаграмма прецедентов Администратора

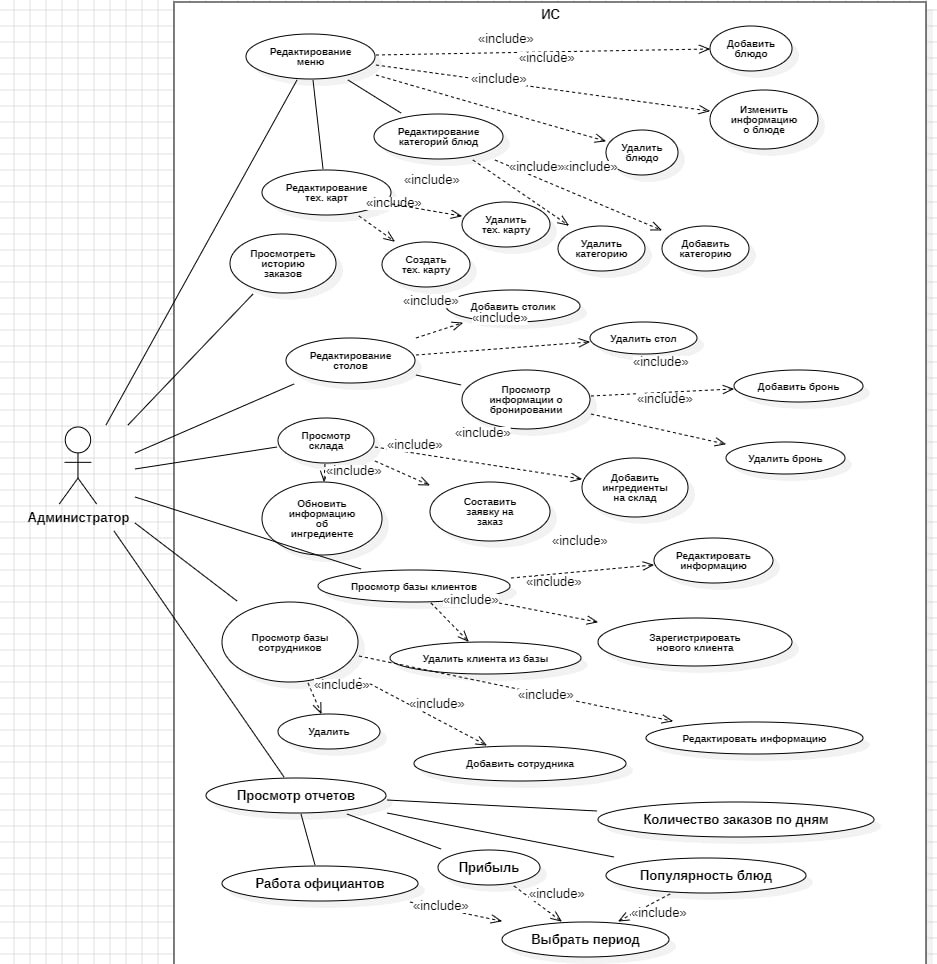


Рис. А.1 – Диаграмма прецедентов Администратора

# Приложение Б Программный код

Программный код доступен в публичном репозитории по ссылке: https://github.com/Buz1Z1/VKR

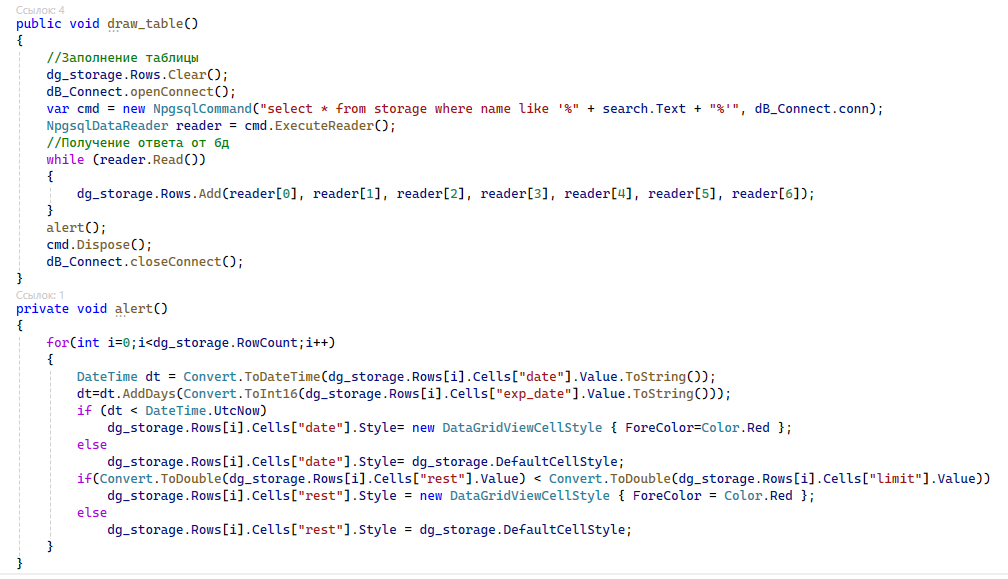


Рис. Б.1 – Пример заполнения страницы «Склад»

# Приложение В Руководство пользователя

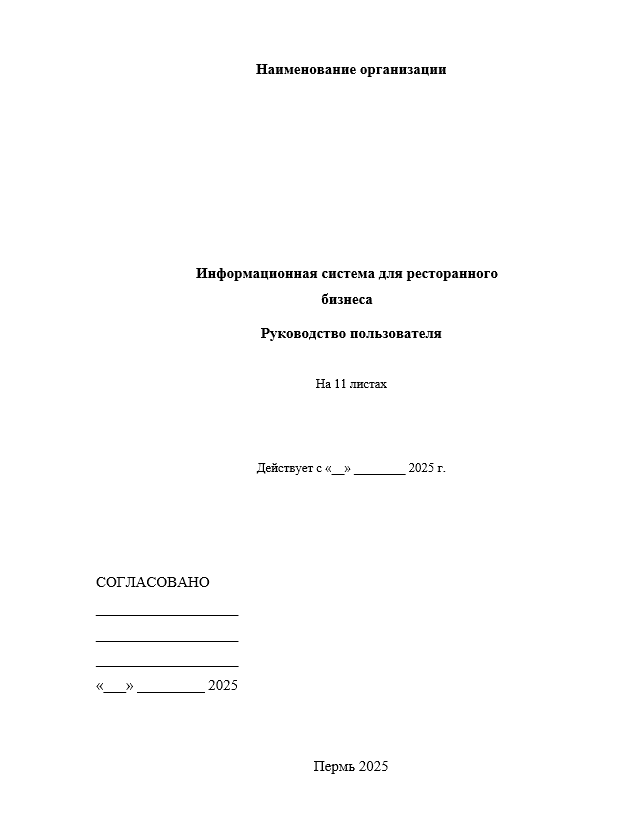


Рис. В.1 – Титульная страница

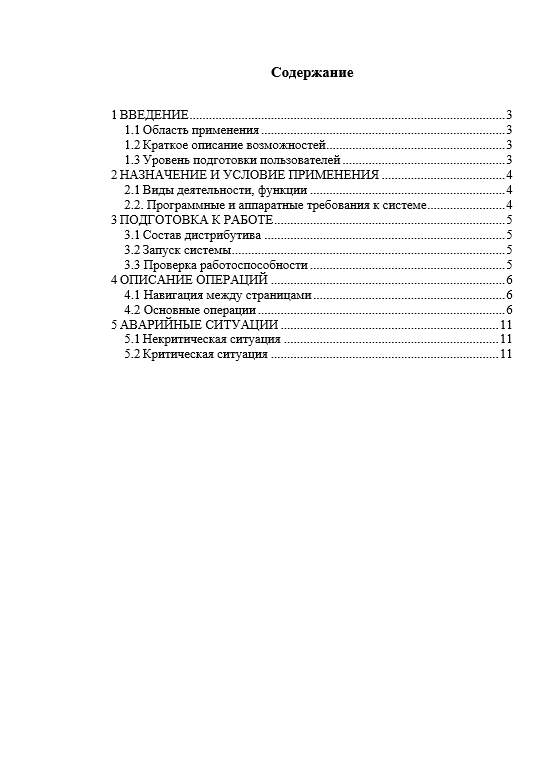


Рис. В.2 – Страница 2 «Содержание»

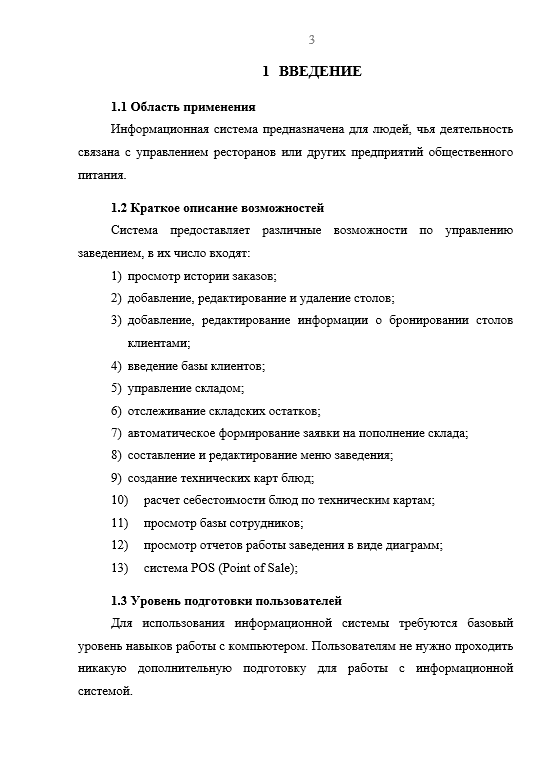


Рис. В.3 – Страница 3 «Введение»

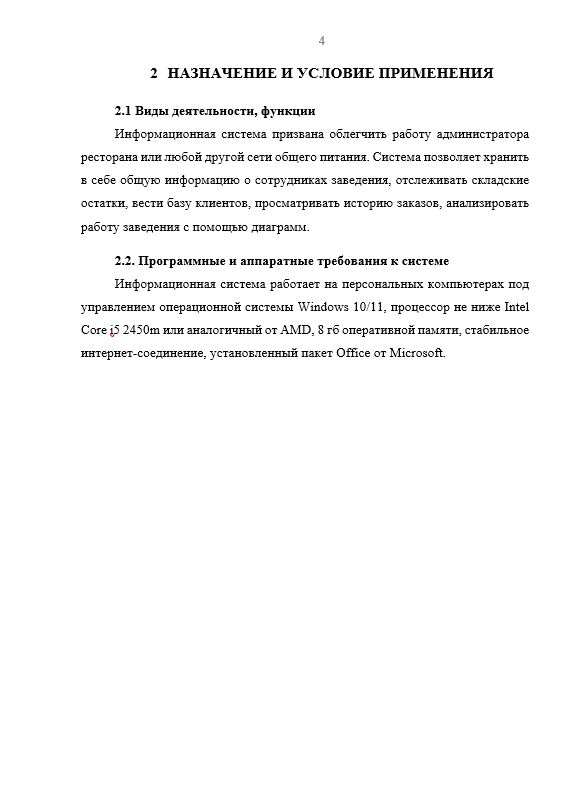


Рис. В.4 – Страница 4 «Назначение и условие применения»

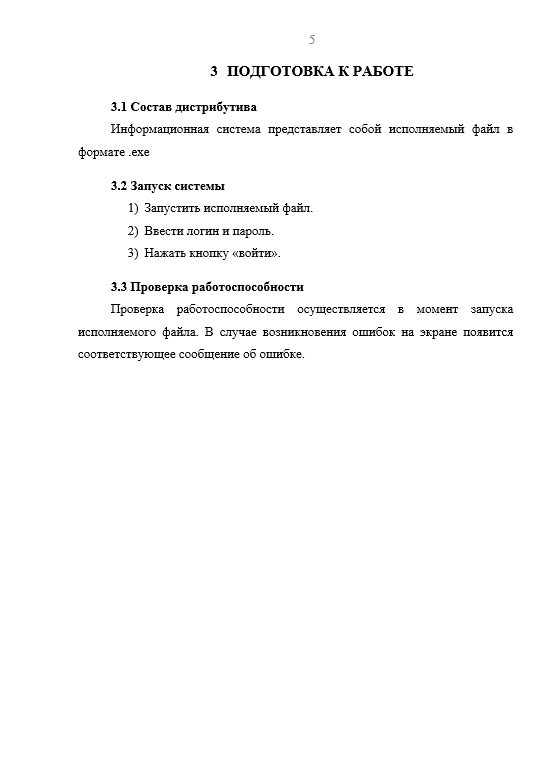


Рис. В.5 – Страница 5 «Подготовка к работе»

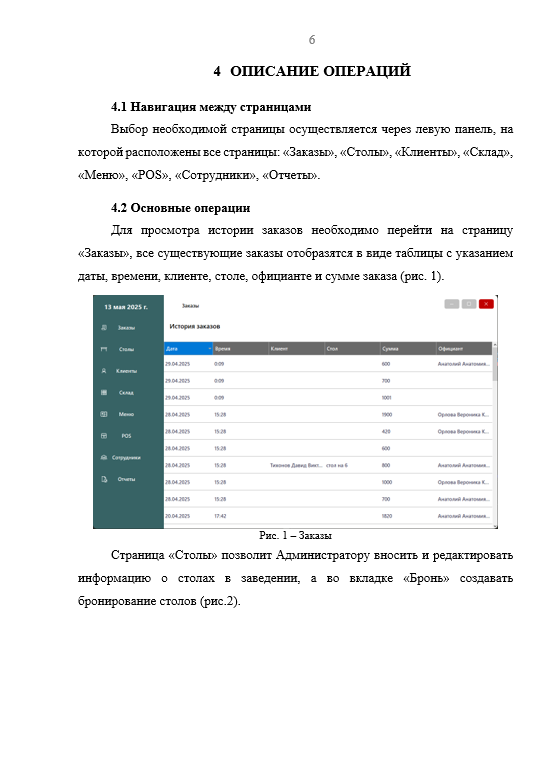


Рис. В.6 – Страница 6 «Описание операций»

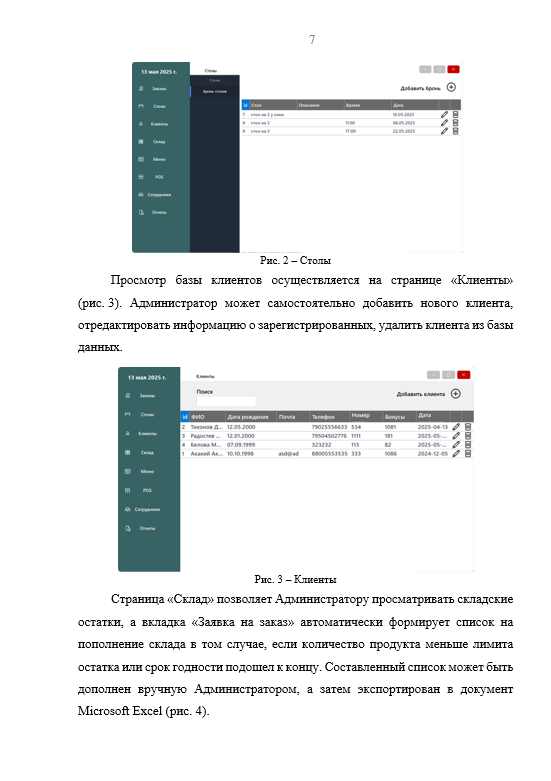


Рис. В.7 – Страница 7 «Описание операций»

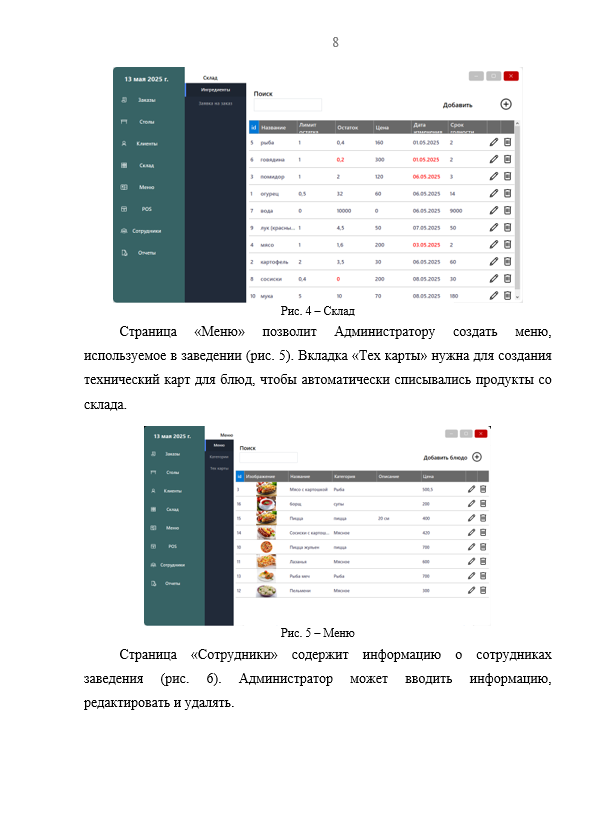


Рис. В.8 – Страница 8 «Описание операций»

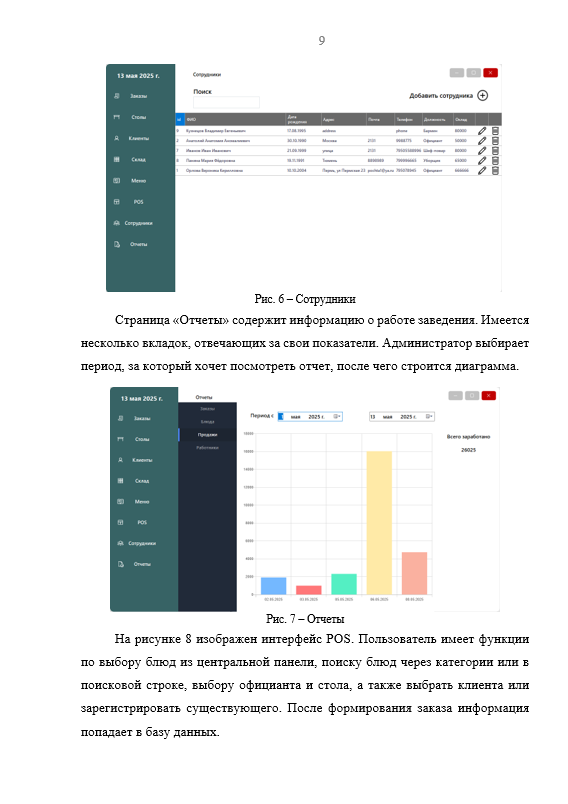


Рис. В.9 – Страница 9 «Описание операций»

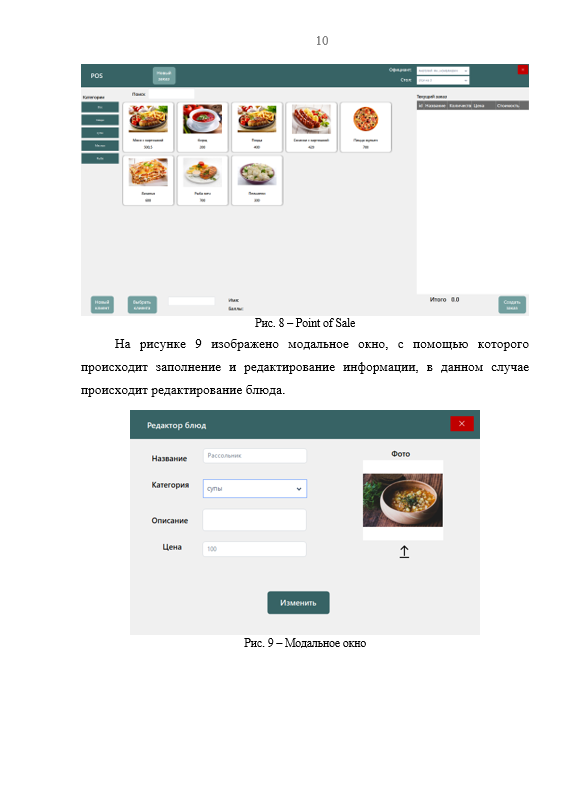


Рис. В.10 – Страница 10 «Описание операций»

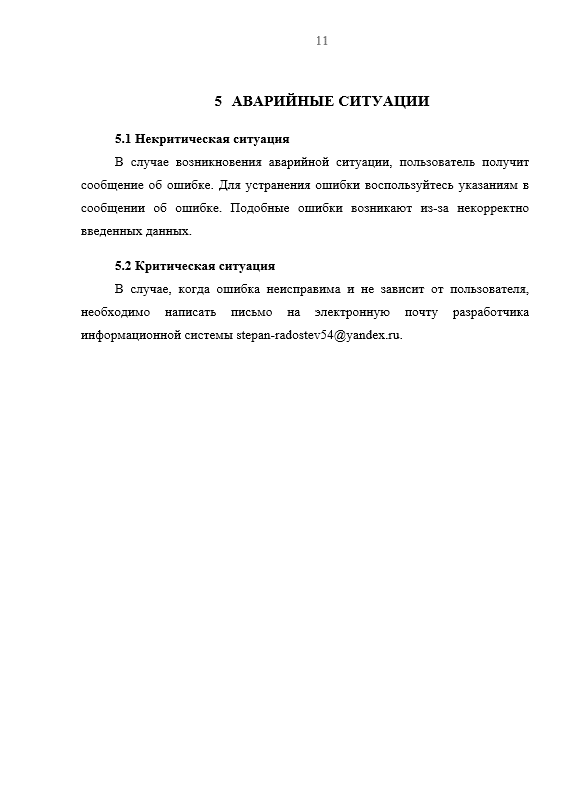


Рис. В.11 – Страница 11 «Аварийные ситуации»