|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Радиотехнический (РТ)

КАФЕДРА Системы обработки информации и управления (ИУ5)

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОМУ ПРОЕКТУ***

***НА ТЕМУ:***

**Автоматизированная система\_\_\_\_\_\_\_\_\_\_\_**

**«Учёт и планирование меню ресторана» \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_**

Студент РТ5-41Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Забурунов Л. В.

(Группа) (Подпись, дата) (Фамилия И.О.)

Руководитель курсового проекта **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_**Маслеников К.Ю.

(Подпись, дата) (Фамилия И.О.)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (Фамилия И.О.)

*2020г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия)

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ**

**на выполнение курсового проекта**

по дисциплине Базы Данных

Студент группы РТ5-41Б Забурунов Леонид Вячеславович

(Фамилия, имя, отчество)

Тема курсового проекта Автоматизированная система учёта и планирования меню ресторана

Направленность КП (учебный, исследовательский, практический, производственный, др.)

Учебная

Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения проекта: 25% к 3 нед., 50% к 9 нед., 75% к 12 нед., 100% к 15 нед.

**1. Задание**

А) Используя MS SQL и язык программирования C#, разработать АС учёта и планирования меню ресторана, отвечающую на запросы о сотрудниках ресторана, меню ресторана, складе ресторана и заказах от посетителей.

Б) В ходе курсового проектирования спроектировать функциональную, инфологическую и датологическую модели предметной области; создать приложение на основе спроектированных моделей; описать интерфейс пользователя и методику испытаний; разработать структурную схему системы, схему работы системы, граф диалога; создать руководство пользователя и техническое задание.

**2. Оформление курсовой работы**

Расчетно-пояснительная записка на 33 страницах формата А4.

Дата выдачи задания «14» февраля 2020 г.

**Руководитель курсового проекта**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Маслеников К. Ю.

(Подпись, дата) (Фамилия И.О)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Забурунов Л. В.

(Подпись, дата) (Фамилия И.О)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**Аннотация**

Курсовой проект направлен на разработку автоматизированной системы учёта и планирования меню ресторана. Данная система позволит упростить ведение отчётности по деятельности ресторана путём автоматизации основных узлов. В системе будут храниться данные о персонале ресторана, о составе меню, о складе ресторана (динамика ингредиентов, - расход и приход). Персонал ресторана может редактировать данные, а посетители – просматривать меню.

В процессе выполнения проекта была исследована предметная область; разработаны функциональная, инфологическая и даталогическая модель; разработан интерфейс к приложению. Итогом выполнения проекта стала АС в виде базы данных с приложением к ней.

Оглавление

[Введение 3](#_Toc43756604)

[1. Анализ предметной области 4](#_Toc43756605)

[1.1 Описание предметной области 4](#_Toc43756606)

[1.2 Изображение предметной области 5](#_Toc43756607)

[1.3 Категории пользователей 5](#_Toc43756608)

[1.4 Ограничения предметной области 6](#_Toc43756609)

[1.5 Входные документы и сообщения 6](#_Toc43756610)

[1.6 Выходные документы и сообщения 7](#_Toc43756611)

[2. Функциональная модель предметной области 8](#_Toc43756612)

[2.1 Функциональные задачи пользователей проектируемой системы 8](#_Toc43756613)

[2.2 Спецификационный вариант функциональной модели 8](#_Toc43756614)

[2.3 Модель предметной области (IDEF-0) 9](#_Toc43756615)

[2.4 Модель предметной области (DFD) 13](#_Toc43756616)

[3. Инфологическая модель предметной области 16](#_Toc43756617)

[3.1 Графическая модель 16](#_Toc43756618)

[3.2 Описание в спецификационном варианте 17](#_Toc43756619)

[4. Выбор СУБД 21](#_Toc43756620)

[5. Даталогическая модель предметной области 22](#_Toc43756621)

[6. Схема работы системы 25](#_Toc43756622)

[6.1 Графическая схема 25](#_Toc43756623)

[6.2 Описание схемы 26](#_Toc43756624)

[7. Структурная схема системы 27](#_Toc43756625)

[7.1 Графическая схема 27](#_Toc43756626)

[7.2 Описание схемы 28](#_Toc43756627)

[8. Граф диалога системы 29](#_Toc43756628)

[8.1 Графическая схема 29](#_Toc43756629)

[8.2 Описание схемы 30](#_Toc43756630)

[Заключение 32](#_Toc43756631)

[Используемая литература 33](#_Toc43756632)

[ПРИЛОЖЕНИЯ 34](#_Toc43756633)

# Введение

В рамках курсового проекта необходимо спроектировать автоматизированную систему учёта и планирования меню ресторана с нуля. В результате система должна автоматически обрабатывать поступающие данные и упрощать процесс управления рестораном. Ценность системы заключается именно в избавлении сотрудников от рутинных действий по ведению документации, которая так же сопровождается неизбежными из-за человеческого фактора ошибками.

Для успешного выполнения курсового проекта необходимо освоить: различные аспекты работы с Microsoft SQL Server (в том числе взаимодействие через Microsoft SSMS); создание приложения с привязкой к базе данных; проектирование системы на этапах функционального, инфологического и даталогического моделирования; оформление документации в виде схемы работы системы, структурной схемы системы и диалогового графа.

Итогом выполнения курсового проекта будет являться готовое к использованию приложение, отвечающее требованиям надёжности и упрощающее работу персоналу.

# Анализ предметной области

# Описание предметной области

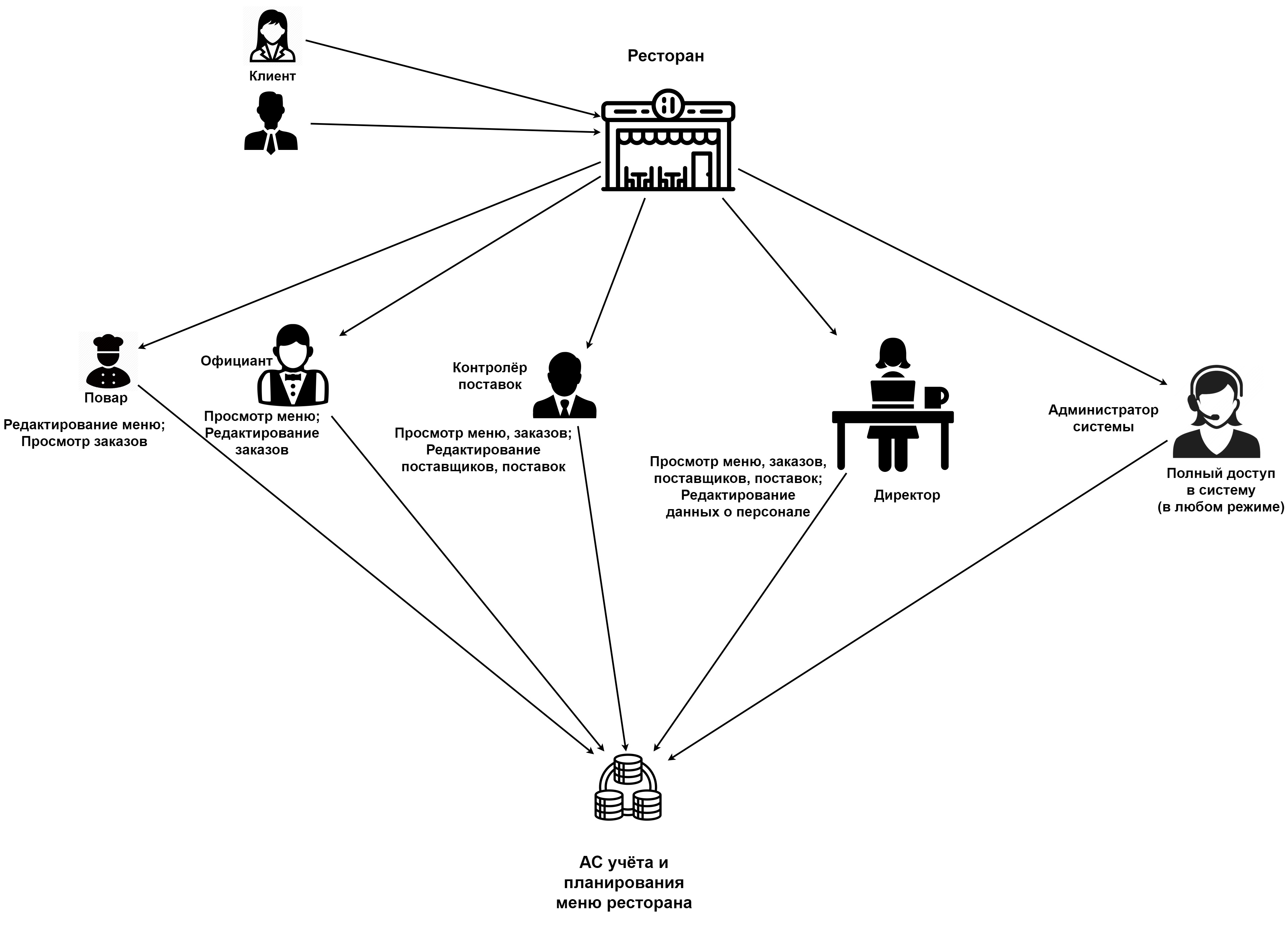
Контроль продуктов на складе для ресторана – критически важная задача, без которой невозможно его функционирование. При полностью ручном рабочем процессе человеческий фактор может сказываться на ошибках в учёте склада и на прибыли ресторана. Автоматизация процесса позволяет уменьшить вероятность возникновения ошибок на порядок, поскольку система будет знать всё о том, какие продукты находятся на складе, в каком количестве они там находятся, сколько они будут пригодны для использования и дальнейшего употребления в пищу и когда запланирована следующая поставка. При таком подходе сотрудникам требуется просто запросить в системе необходимую информацию и заказать недостающие продукты; поле для ошибок значительно уменьшено.

Сердцем разрабатываемой системы будет механизм обработки заказов: посетитель просматривает меню (которое так же доступно в системе), официант оформляет заказ (внося его в систему), повар обрабатывает заказ (предварительно отметив это в системе) и посетитель получит готовые блюда. Сотрудник ресторана, отвечающий за поставки продуктов, сможет видеть динамику расхода продуктов и доступных поставщиков, и на основе имеющихся данных заказать необходимые продукты. Соответственно, данная система также показывает круг ответственности каждого работника.

АС будет собирать имеющуюся информацию о работниках, блюдах, ингредиентах, поставках и составлять полноценные отчёты о деятельности ресторана. В числе таких отчётов будет статистика заказов по блюдам или по другим категориям, доходы и расходы по блюдам и входящим в состав продуктам, полная сводка о доходах и расходах и прочие. Управляющий рестораном сможет в любой момент получать доступ к этим сводкам и выявлять слабые звенья в работе ресторана с целью увеличения прибыли.

Таким образом, система облегчит процесс развития ресторана как бизнеса и позволит улучшить численные показатели при том, что сама система не является чрезвычайно сложной и не требует постоянной усиленной поддержки.

## Изображение предметной области



## Категории пользователей

1. Посетитель

Имеет возможность просматривать меню.

1. Официант

Имеет возможность вносить заказываемые блюда в систему и получать соответствующие сообщения о возможности приготовления запрашиваемого блюда (например, отсутствие какого-либо ингредиента); также выводится полную информацию о составе блюда (например, наличие продуктов, на которых распространена аллергия).

1. Повар

Имеет возможность просматривать информацию о блюде и взять себе поступивший заказ.

1. Контролёр поставок

Имеет возможность изменять данные о поставках, просматривать текущие запасы продуктов питания в ресторане и поступающие заказы (для сбора статистики и заблаговременного запроса на поставку).

1. Администратор системы

Имеет полный доступ ко всем элементам системы, а также право наделять работников полномочиями.

1. Директор ресторана

Имеет доступ к статистике (различного рода отчётам).

## Ограничения предметной области

1. Каждый сотрудник имеет только одну категорию доступа.

## Входные документы и сообщения

1. Поступающий заказ.
2. Сведения о поставке.
3. Сведения о поставщике.
4. Сведения о персонале.
5. Сведения о блюде.
6. Сведения об ингредиентах.

## Выходные документы и сообщения

1. Отчёт о продуктах на складе.
2. Отчёт по заказам (статистика по блюдам, по ингредиентам и т.д.).
3. Отчёт о поставках.
4. Отчёт о работниках.
5. Полная информация о блюдах и составляющих (меню).

# Функциональная модель предметной области

## Функциональные задачи пользователей проектируемой системы

1. Посетитель: просмотр информации о блюдах.
2. Официант: просмотр информации о блюдах; внесение заказов.
3. Повар: просмотр и редактирование информации о блюдах и их составляющих; просмотр заказов.
4. Контролёр поставок: просмотр информации о блюдах и их составляющих; внесение поставщиков и поставок.
5. Администратор системы: просмотр информации о блюдах и их составляющих, о поставщиках и поставках; внесение работников и распределение полномочий в системе.
6. Директор ресторана: просмотр информации о блюдах и их составляющих, о поставщиках и поставках, о работниках и их полномочиях.

## Спецификационный вариант функциональной модели

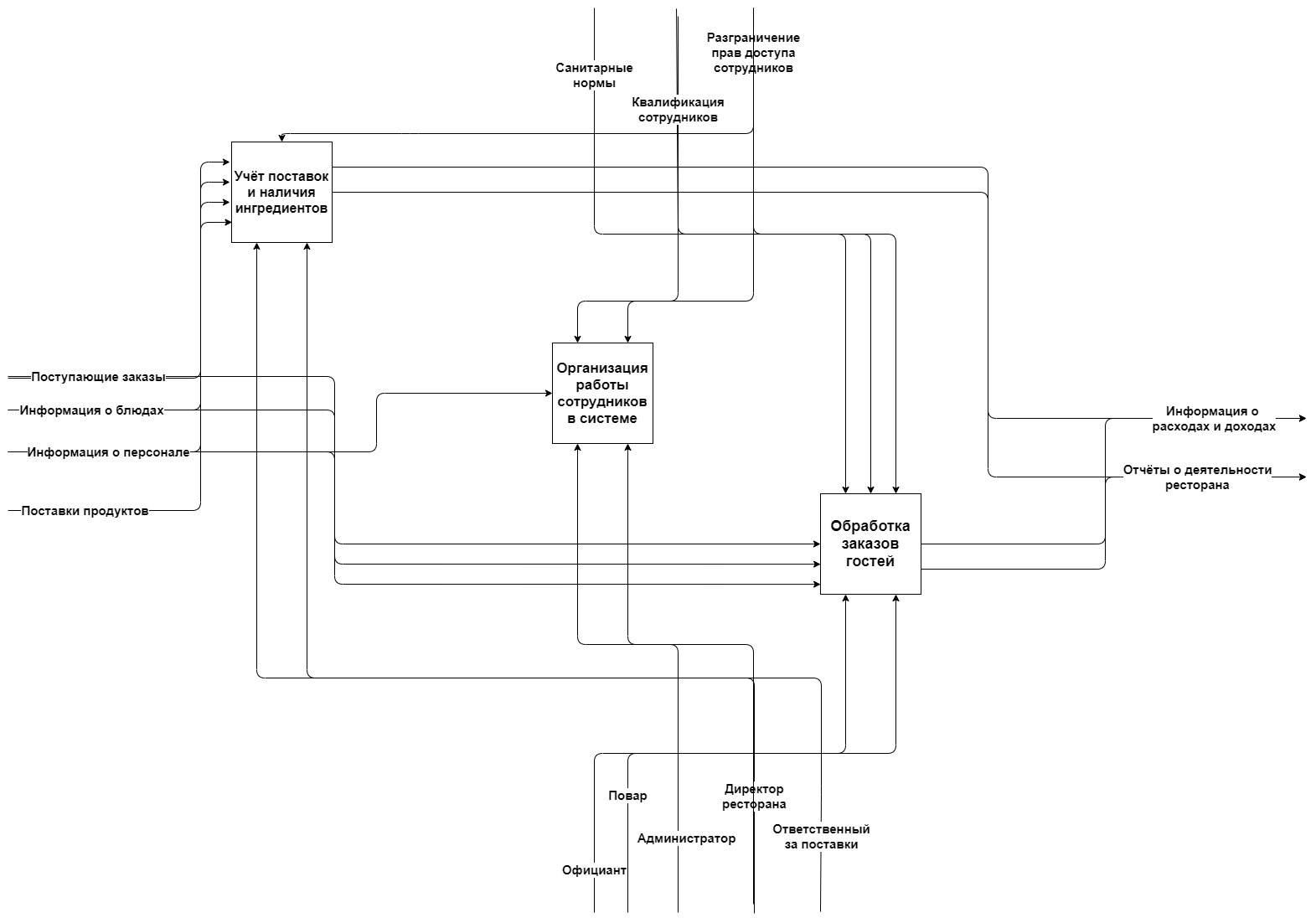
1. Ввод сведений о:  
   а) сотрудниках;  
   б) блюдах;  
   в) ингредиентах;  
   г) поступающих заказах;  
   д) поставщиках;  
   е) поставках;  
   ё) посетителях.
2. Редактирование сведений о:  
   То же самое, что и в предыдущем пункте.
3. Экранные формы для внесения и изменения сведений о:  
   То же самое, что и в пункте 1.
4. Запросы о:  
   То же самое, что и в пункте 1.
5. Отчёты о:  
   То же самое, что и в пункте 1.

## Модель предметной области (IDEF-0)

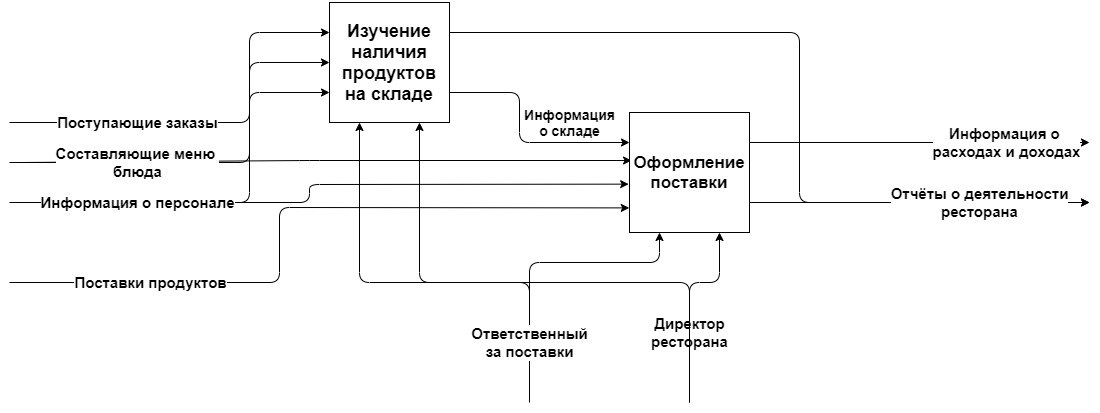
А) **графическая модель:**



Декомпозиция первого уровня:  
  
Декомпозиция второго уровня, блок «Обработка заказов гостей»:



Декомпозиция второго уровня, блок «Учёт поставок и наличия ингредиентов»:



Б) **спецификационный вариант модели:**

1. АС по учёту и планированию ресторанного меню:  
   *Вход:* поступающие из зала ресторана заказы; блюда, входящие в меню; данные о сотрудниках ресторана; информация о поставках.

*Выход:* подача блюд гостям; формирование отчётов; информация о доходах ресторана и расходах на него.

*Управление:* санитарные нормы для ресторанов; разделение прав между сотрудниками; квалификация сотрудников.

*Механизм:* официант; повар; ответственное за поставки лицо; директор ресторана; администратор системы.

1. Декомпозиция первого уровня:
   1. Учёт поставок и наличия ингредиентов:

*Вход:* поступающие из зала ресторана заказы; блюда, входящие в меню; данные о сотрудниках ресторана; информация о поставках.

*Выход:* формирование отчётов; информация о доходах ресторана и расходах на него.

*Управление:* разделение прав между сотрудниками.

*Механизм:* ответственное за поставки лицо; директор ресторана.

* 1. Организация работы сотрудников в системе:

*Вход:* данные о сотрудниках ресторана.

*Управление:* разделение прав между сотрудниками; квалификация сотрудников.

*Механизм:* директор ресторана; администратор системы.

* 1. Обработка заказов гостей:

*Вход:* поступающие из зала ресторана заказы; блюда, входящие в меню; данные о сотрудниках ресторана.

*Выход:* формирование отчётов; информация о доходах ресторана и расходах на него.

*Управление:* санитарные нормы для ресторанов; разделение прав между сотрудниками; квалификация сотрудников.

*Механизм:* официант; повар.

1. Декомпозиция второго уровня:
   1. Обработка заказов гостей:
      1. Принятие заказа в зале:

*Вход:* поступающие из зала ресторана заказы; блюда, входящие в меню; данные о сотрудниках ресторана.

*Выход:* формирование отчётов; информация о доходах ресторана и расходах на него; передача заказа.

*Механизм:* официант.

* + 1. Приготовление блюда на кухне:

*Вход:* передача заказа; блюда, входящие в меню; данные о сотрудниках ресторана.

*Выход:* подача блюд гостям.

*Управление:* санитарные нормы для ресторанов; квалификация сотрудников. *Механизм:* повар.

* 1. Учёт поставок и наличия ингредиентов:
     1. Изучение наличия продуктов на складе:

*Вход:* принятие заказа в зале; блюда, входящие в меню; данные о сотрудниках ресторана.

*Выход:* информация о складе; формирование отчётов.  
*Механизм:* ответственное за поставки лицо; директор ресторана.

* + 1. Оформление поставки:

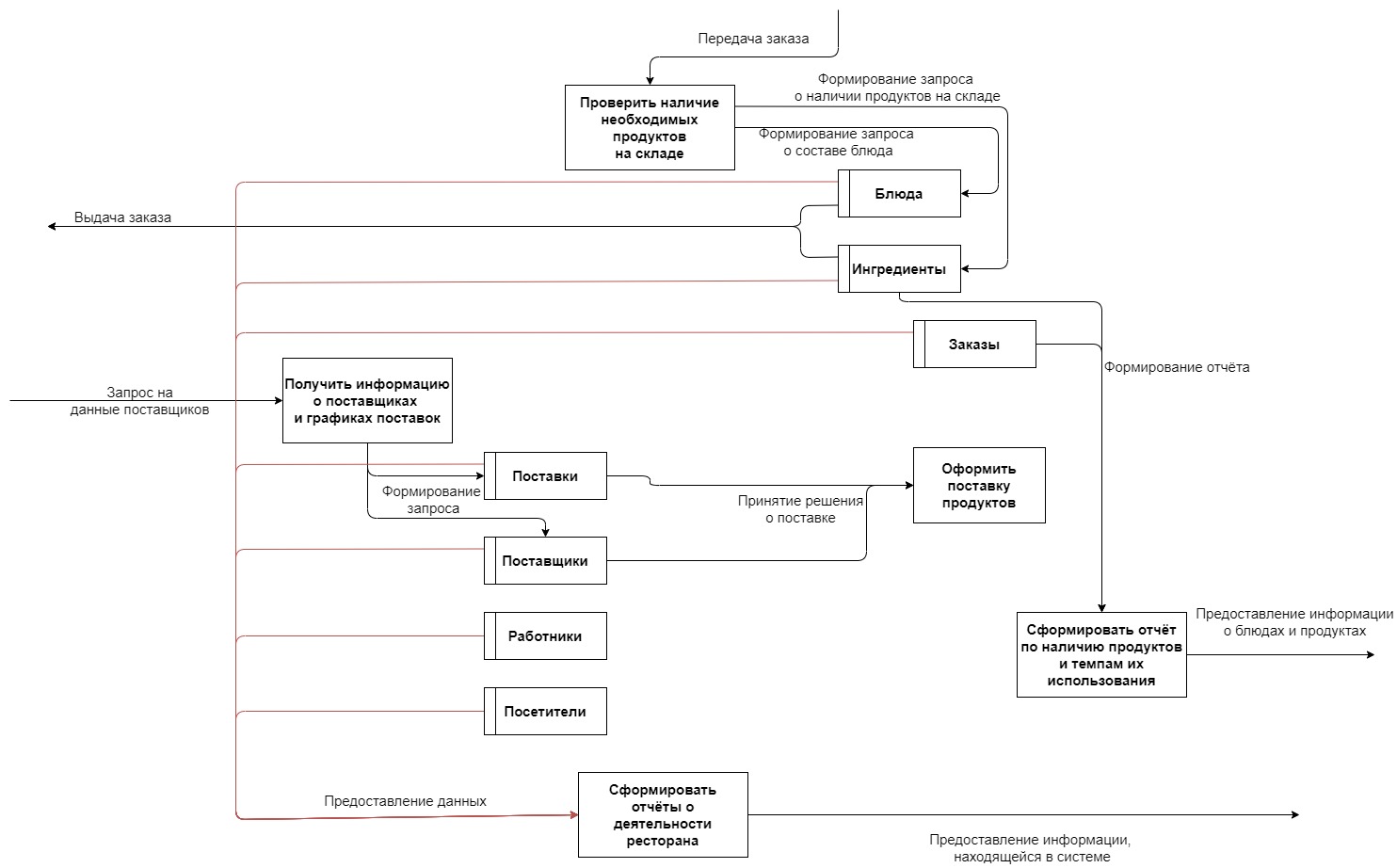
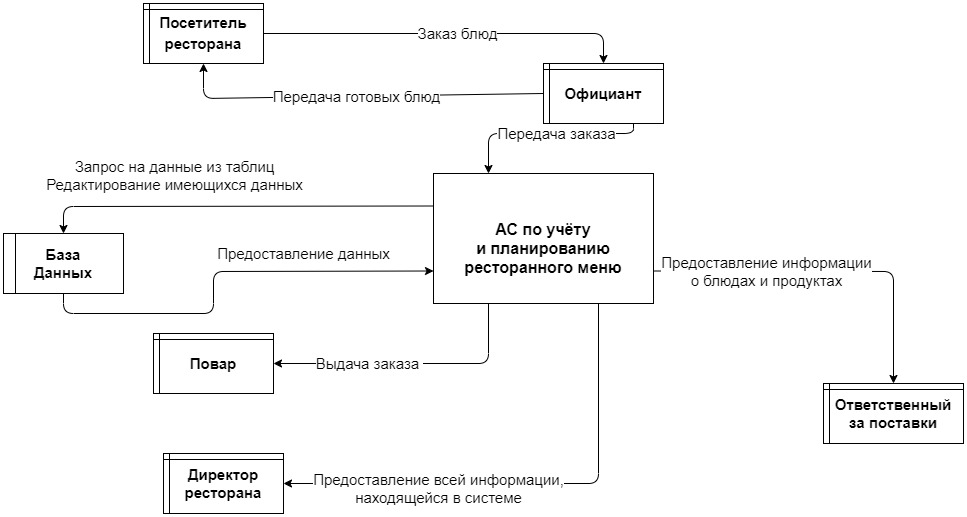
*Вход:* информация о складе; блюда, входящие в меню; данные о сотрудниках ресторана; информация о поставках.

*Выход:* информация о доходах ресторана и расходах на него; формирование отчётов.

*Механизм:* ответственное за поставки лицо; директор ресторана.

## Модель предметной области (DFD)

А) **графическая модель:**  
  
Декомпозиция первого уровня:   
Б) **спецификационный вариант модели**:



Внешние сущности:

1. Посетитель ресторана - просматривает информацию о блюдах (состав, наличие, время приготовление и так далее);
2. Официант – оформляет заказ и приносит готовые блюда к столу посетителя;
3. Повар – получает заказы и сообщает официанту о готовности;
4. Контроллёр поставок – наблюдает за динамикой товаров на складе и осуществляет своевременные поставки;
5. Директор ресторана – контроллирует деятельность работников.

Процессы:

1. Проверка наличия ингредиентов на складе для дальнейшего приготовления блюд из заказов;
2. Передача готового блюда официанту;
3. Формирование отчётов для директора;
4. Формирование отчётов по складу для работника, отвечающего за поставки продуктов;
5. Получение информации о поставщиках для работника, отвечающего за поставки продуктов.

Хранилища данных:

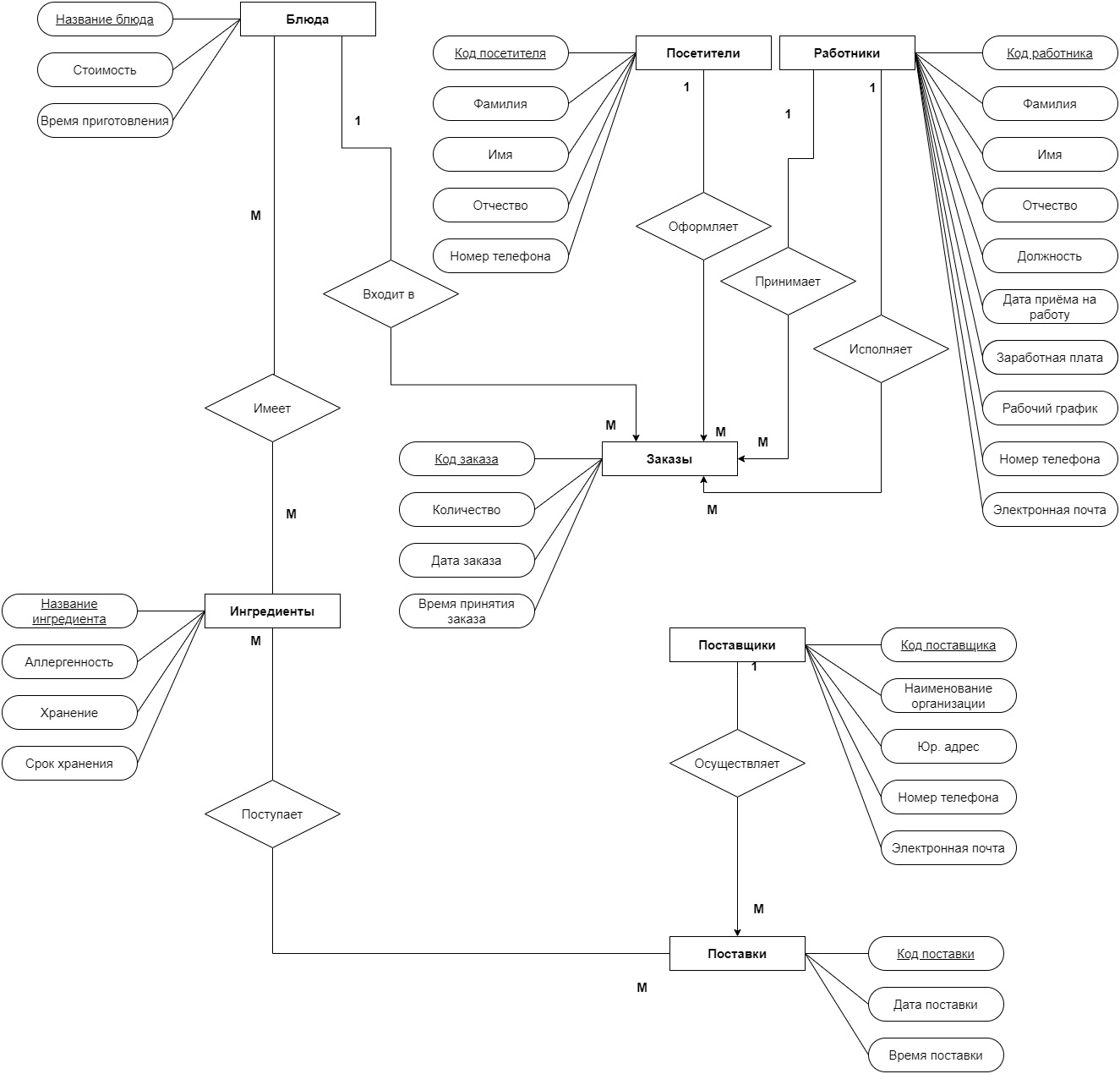
* 1. Посетители ресторана;
  2. Заказы;
  3. Блюда;
  4. Ингредиенты;
  5. Работники;
  6. Поставки;
  7. Поставщики.

Потоки данных:

1. Передача заказа;
2. Выдача заказа;
3. Отчёт о составах блюд;
4. Отчёт о продуктах на складе;
5. Отчёт о поставщиках;
6. Отчёт о поставках;
7. Отчёты для директора (по факту используются все данные, имеющиеся в системе).

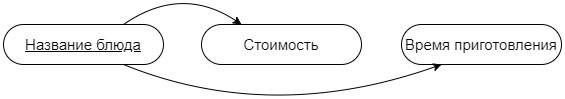
# Инфологическая модель предметной области

* 1. Графическая модель

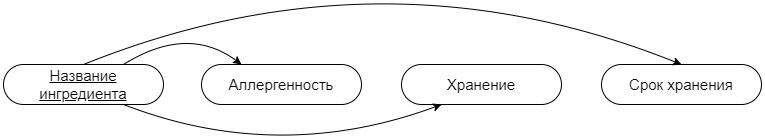


## Описание в спецификационном варианте

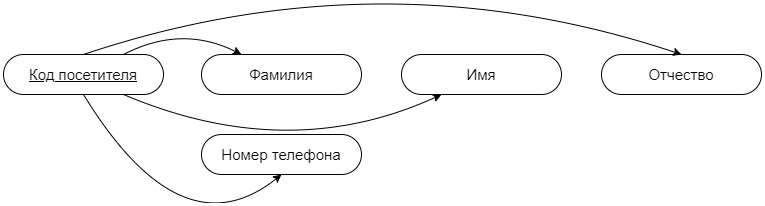
1. Сущности:
   1. Заказы (Код заказа, Блюдо, Количество, Официант, Повар, Дата заказа, Время принятия заказа);
   2. Работники (Код работника, Фамилия, Имя, Отчество, Должность, Дата приёма на работу, Заработная плата, Рабочий график, Номер телефона, Электронная почта);
   3. Посетители (Код посетителя, Фамилия, Имя, Отчество, Номер телефона);
   4. Блюда (Название блюда, Стоимость, Время приготовления, Состав);
   5. Состав блюда (Код состава, Блюдо, Ингредиент);
   6. Ингредиенты (Название ингредиента, Аллергенность, Хранение, Срок хранения);
   7. Поставщики (Код поставщика, Наименование организации, Юридический адрес, Контактный телефон, Электронная почта);
   8. Поставки (Код поставки, Дата поставки, Время поставки, Поставщик);
   9. Приход (Код, Поставка, Ингредиент, Количество).
2. Атрибуты:
   1. Код заказа: numeric(6) (Primary Key);
   2. Код работника: numeric(3) (Primary Key);
   3. Код посетителя: numeric(5) (Primary Key);
   4. Название блюда: varchar(20) (Primary Key);
   5. Код состава: numeric(4) (Primary Key);
   6. Название ингредиента: varchar(15) (Primary Key);
   7. Код поставщика: numeric(2) (Primary Key);
   8. Код поставки: numeric(6) (Primary Key);
   9. Код прихода: numeric(8) (Primary Key);
   10. Блюдо: varchar(20);
   11. Количество: tinyint;
   12. Официант: numeric(3);
   13. Повар: numeric(3);
   14. Дата заказа: date;
   15. Время принятия заказа: time;
   16. Фамилия: varchar(25);
   17. Имя: varchar(15);
   18. Отчество: varchar(20);
   19. Должность: varchar(12);
   20. Дата приёма на работу: date;
   21. Заработная плата: int;
   22. Рабочий график: varchar(50);
   23. Номер телефона: numeric(10);
   24. Электронная почта: varchar(25);
   25. Стоимость: smalint;
   26. Время приготовления: time;
   27. Состав: numeric(5);
   28. Ингредиент: varchar(15);
   29. Аллергенность: bool;
   30. Хранение: varchar(20);
   31. Срок хранения: tinyint;
   32. Наименование организации: varchar(20);
   33. Юр. адрес: varchar(30);
   34. Дата поставки: date;
   35. Время поставки: time;
   36. Поставщик: numeric(2);
   37. Поставка: numeric(6);
3. Связи между сущностями:
   1. Входит в (Блюдо, Название блюда): связь типа 1:М от «Блюда» к «Заказы».
   2. Оформляет (Посетитель, Код посетителя): связь типа 1:М от «Посетители» к «Заказы».
   3. Принимает (Официант, Код работника): связь типа 1:М от «Работники» к «Заказы».
   4. Исполняет (Повар, Код работника): связь типа 1:М от «Работники» к «Заказы».
   5. Осуществляет (Поставщик, Код поставщика): связь типа 1:М от «Поставщики» к «Поставки».
   6. Имеет: связь типа М:М от «Блюда» к «Ингредиенты»
   7. Пополняет: связь типа М:М от «Поставки» к «Ингредиенты»
4. Связи атрибутов:
   1. «Блюда»:



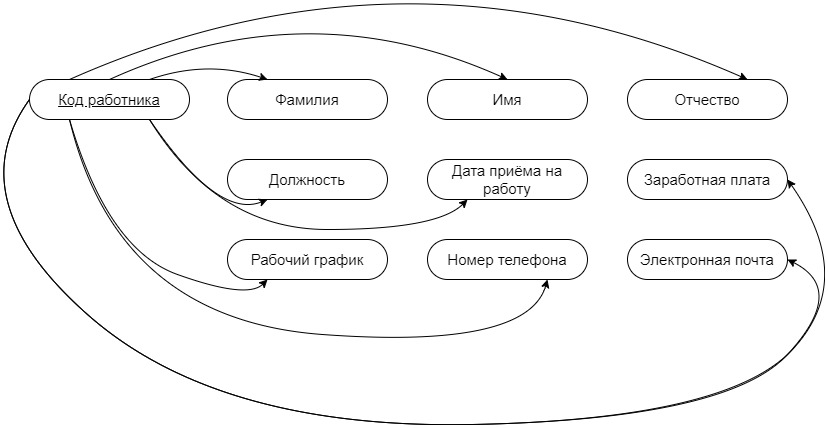
* 1. «Ингредиент»:



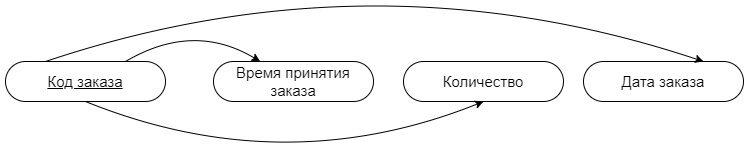
* 1. «Посетители»:



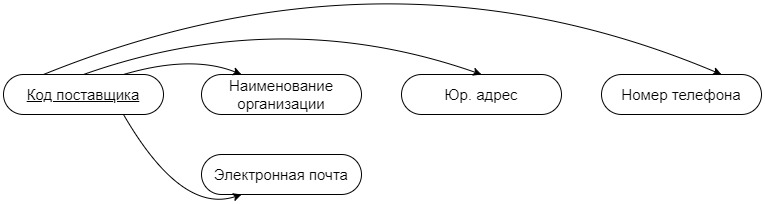
* 1. «Работники»:



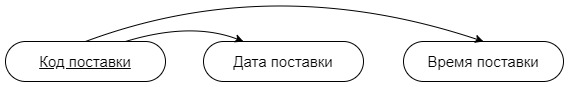
* 1. «Заказы»:



* 1. «Поставщики»:



* 1. «Поставки»:



# Выбор СУБД

Для реализации проекта будет использовать Microsoft SQL Server 2019 и Microsoft SSMS версии:

*SQL Server Management Studio 15.0.18330.0*

*SQL Server Management Objects (SMO) 16.100.37971.0*

*Клиентские средства служб Microsoft Analysis Services 15.0.19040.0*

*Компоненты доступа к данным (MDAC) 10.0.18362.1*

*Microsoft MSXML 3.0 4.0 6.0*

*Microsoft .NET Framework 4.0.30319.42000*

*Операционная система 10.0.18363*

Для создания приложения к проекту – проект C# Windows Forms в MS Visual Studio версии:  
*Microsoft Visual Studio Community 2019*

*Версия 16.5.4*

*VisualStudio.16.Release/16.5.4+30011.22*

*Microsoft .NET Framework*

*Версия 4.8.03752*

# Даталогическая модель предметной области

Создадим базу данных в SQL Server Management Studio с помощью следующего запроса:

USE KRBD\_Restaraunt

create table Dish(

dish\_id numeric(3) identity primary key,

\_name varchar(20),

\_price smallint not null,

\_time smallint,

);

create table Component(

comp\_id numeric(5) identity primary key,

\_name varchar(25),

allergic char(1),

storage varchar(25),

storage\_period smallint not null,

);

alter table Component add constraint Allerg\_yn check(allergic = '+' or allergic = '-');

create table Dish\_Structure(

struc\_id numeric(5) identity primary key,

dish numeric(3),

component numeric(5),

);

alter table Dish\_Structure add constraint dish\_struc\_fk foreign key (dish) references Dish(dish\_id);

alter table Dish\_Structure add constraint comp\_struc\_fk foreign key (component) references Component(comp\_id);

create table Customer(

cust\_id numeric(5) identity primary key,

last\_name varchar(25) not null,

first\_name varchar(15) not null,

patronymic varchar(20),

phone\_number numeric(10),

unique(last\_name, first\_name)

);

create table Employee(

empl\_id numeric(3) identity primary key,

last\_name varchar(25) not null,

first\_name varchar(15) not null,

patronymic varchar(20),

position varchar(15),

empl\_date date not null,

paycheck int not null,

schedule varchar(50),

phone\_number numeric(10),

mailbox varchar(25),

unique(last\_name, first\_name)

);

alter table Employee add constraint positions\_ check(position = 'Официант' or position = 'Повар');

create table Orders(

order\_id numeric(6) identity primary key,

dish numeric(3) not null,

amount tinyint not null,

customer numeric(5) not null,

waiter numeric(3),

chef numeric(3),

order\_date date not null,

receive\_time time(0),

);

alter table Orders add constraint order\_dish\_fk foreign key (dish) references Dish(dish\_id);

alter table Orders add constraint order\_cust\_fk foreign key (customer) references Customer(cust\_id);

alter table Orders add constraint order\_empl\_fk\_1 foreign key (waiter) references Employee(empl\_id);

alter table Orders add constraint order\_empl\_fk\_2 foreign key (chef) references Employee(empl\_id);

create table Providers(

prov\_id numeric(2) identity primary key,

org\_name varchar(20) not null,

\_address varchar(30),

phone\_number numeric(10) not null,

mailbox varchar(25)

);

create table Supply(

supply\_id numeric(6) identity primary key,

supply\_date date not null,

supply\_time time(0),

\_provider numeric(2) not null,

);

alter table Supply add constraint supply\_prov\_fk foreign key (\_provider) references Providers(prov\_id);

create table Refill(

refill\_id numeric(8) identity primary key,

supply\_id numeric(6) not null,

component numeric(5) not null,

amount tinyint not null,

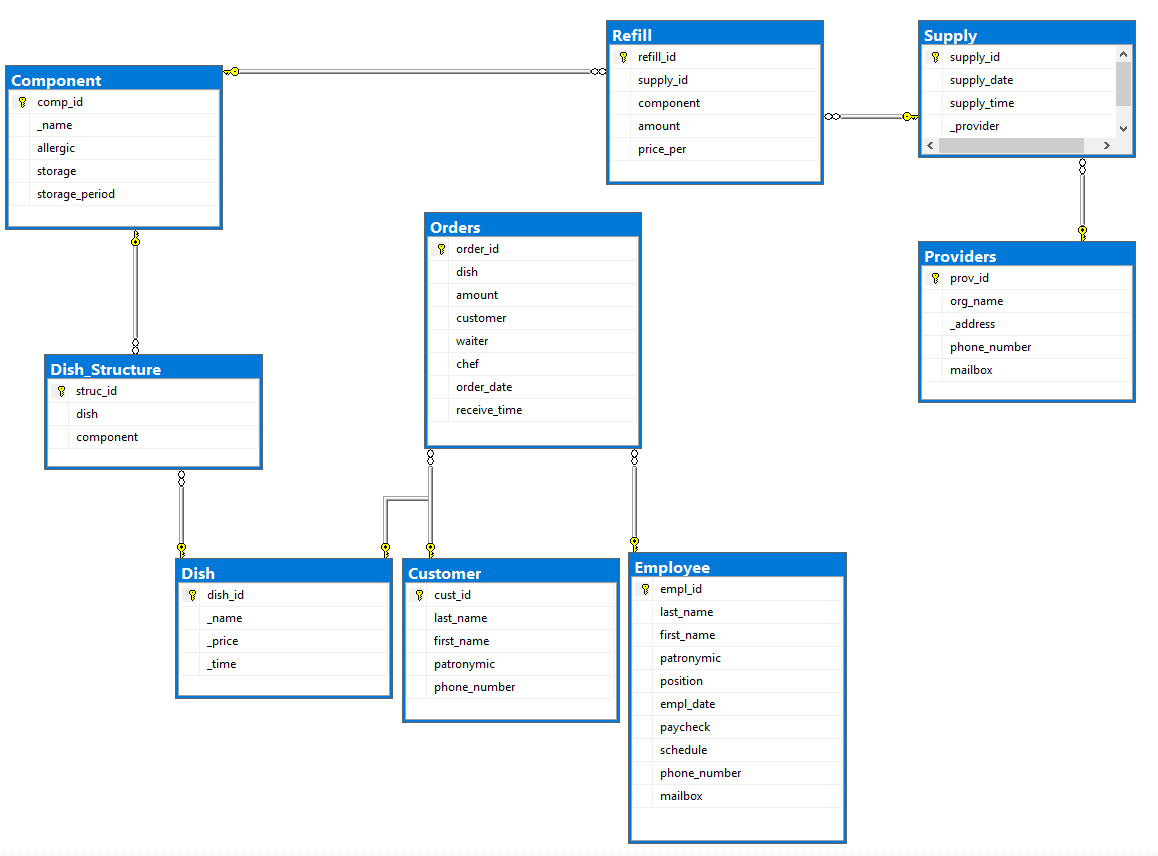
price\_per smallint not null,

);

alter table Refill add constraint refill\_supply\_fk foreign key (supply\_id) references Supply(supply\_id);

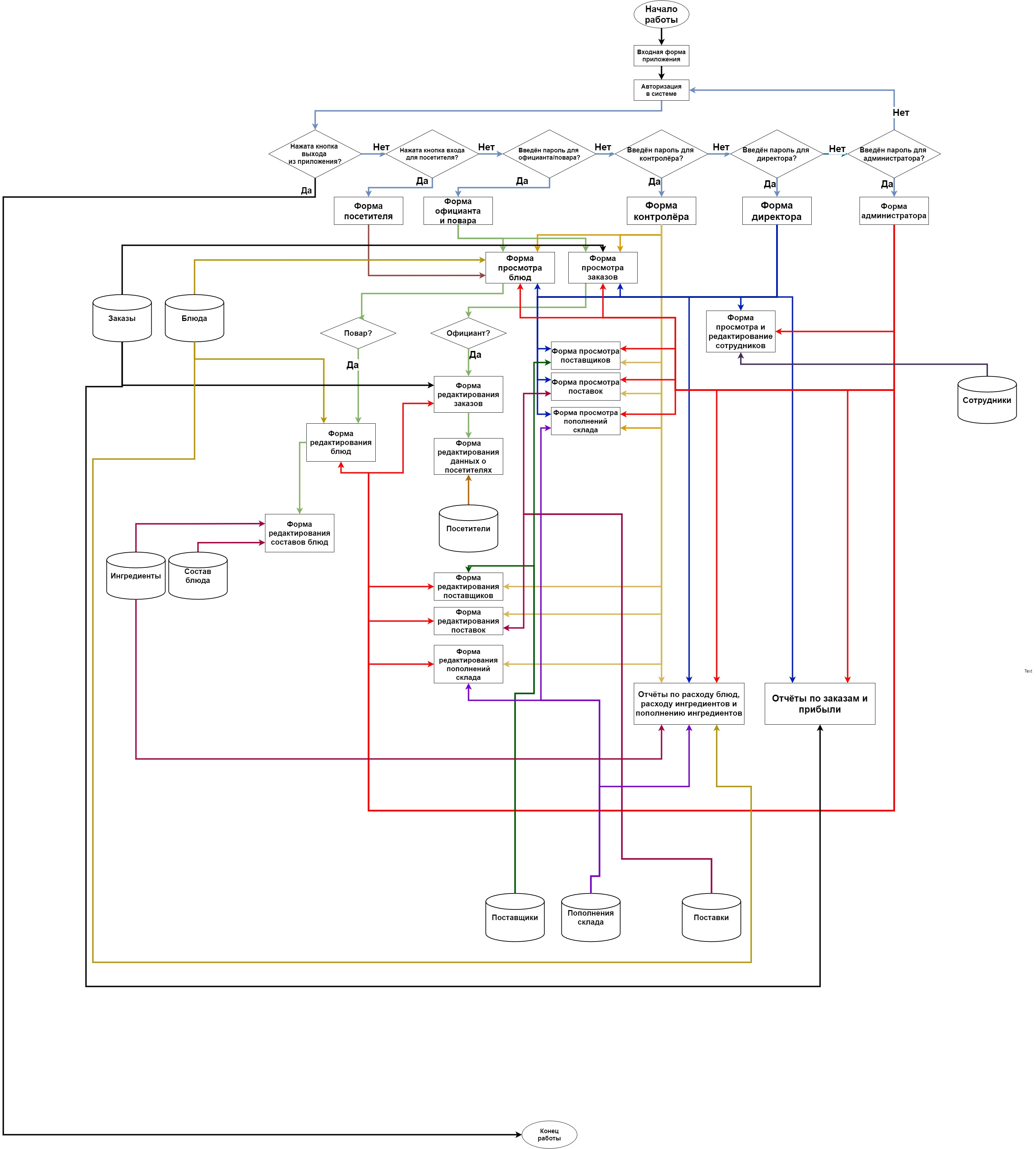
alter table Refill add constraint refill\_comp\_fk foreign key (component) references Component(comp\_id);

Графическое изображение даталогической модели:



# Схема работы системы

## Графическая схема



## Описание схемы

Приложение запускается со входной формой, на котором есть поле для ввода пароля, кнопка авторизации, кнопка неавторизованного входа в режиме посетителя и кнопка выхода из приложения.

При попытке авторизации приложение сверяет пароль и либо открывает форму для сотрудника, либо выдаёт сообщение о неверном пароле. В случае ввода правильного пароля для каждого сотрудника открывается форма, в которой предлагается выбрать любую из доступных ему форм:

А) официант может попасть в поле просмотра заказов, далее перейти к редактированию заказов, в конце – перейти к редактированию данных о посетителях. Так же официанту доступен список блюд;

Б) Повар при попадании на форму для редактирования сведений о блюдах может перейти в отдельную форму для работы с информацией о составах блюд; так же доступен просмотр заказов;

В) Ответственное за поставки лицо может просмотреть блюда и заказы, имеет доступ к редактированию данных о поставщиках, поставках и складе, а также к отчётности по динамике расхода ингредиентов;

Г) Директор имеет возможность просмотреть все описанные выше формы без доступа к редактированию, имеет полный доступ к форме сотрудников, а так же к отчётам по прибыли ресторана;

Д) Администратор системы имеет полный доступ ко всем формам в системе.

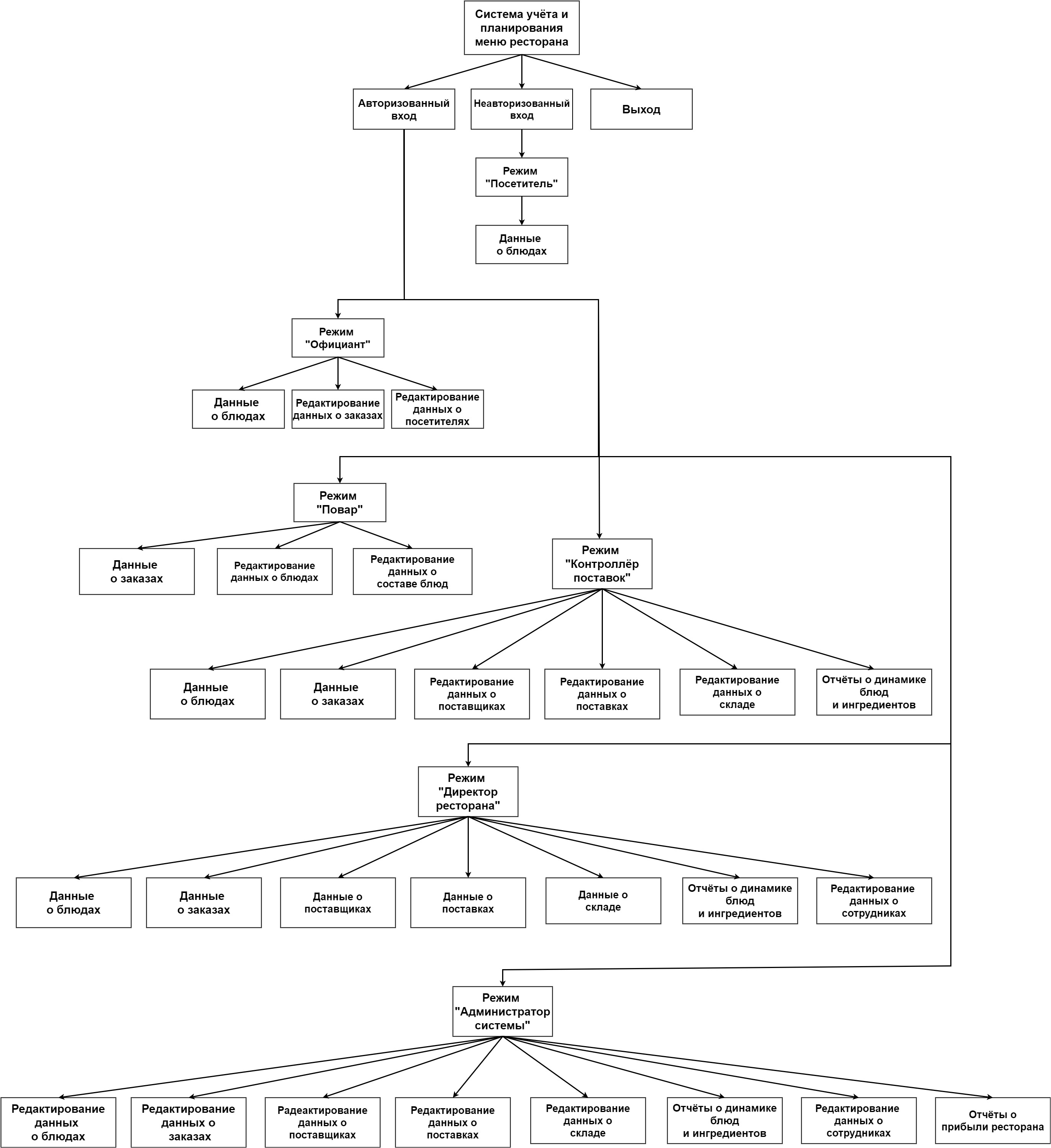
Один цвет стрелок на схеме соответствует одной категории доступа (администратор может «пройти по любому цвету»); так же отдельным цветом выделены все стрелки, ведущие из таблиц.

При неавторизованном входе откроется единственная форма просмотра блюд.

По кнопке выхода приложение закроет все открытые формы и завершит работу.

# Структурная схема системы

## Графическая схема

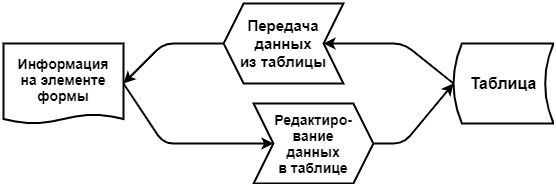


При этом каждый тупиковый блок функционирует в соответствии с шаблоном:

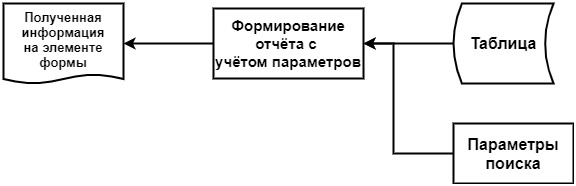
* + 1. «Данные о …»:



* + 1. «Редактирование данных о …»:



* + 1. «Отчёты о …»:



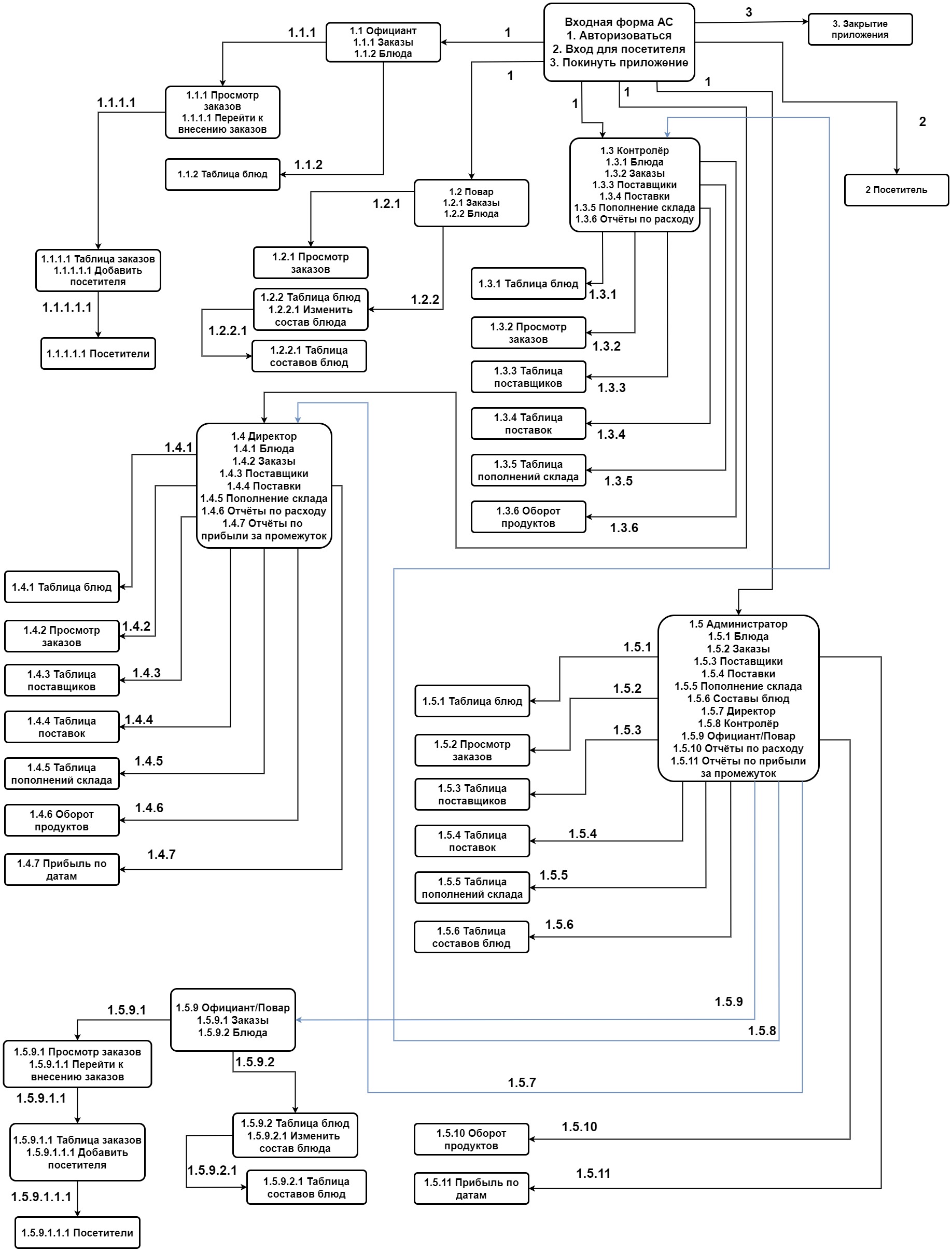
## Описание схемы

Схема представляет собой последовательное разворачивание всех блоков. Изначально можно авторизоваться с имеющимся паролем и получить соответствующий уровень доступа к тупиковым блокам схемы, либо не авторизовываться и быть клиентом.

Для разгрузки схемы декомпозиция тупиковых блоков была вынесена в отдельные подпункты, находящиеся сразу после схемы; данные дополнительные схемы отражают механизм работы этих тупиковых блоков всех трёх возможных типов.

# Граф диалога системы

## Графическая схема



## Описание схемы

Приложение встречает пользователя входной формой с тремя опциями: можно авторизоваться (1) , просмотреть меню для посетителей (2) и выйти из приложения (3) .

Режимы авторизации:

* 1. Официант:
     1. Заказы:
        1. Перейти к внесению заказов:
           1. Добавить посетителя;
     2. Блюда;
  2. Повар:
     1. Заказы;
     2. Блюда:
        1. Изменить состав блюда;
  3. Контролёр:
     1. Блюда;
     2. Заказы;
     3. Поставщики;
     4. Поставки;
     5. Пополнение склада;
     6. Отчёты по расходу;
  4. Директор:
     1. Блюда;
     2. Заказы;
     3. Поставщики;
     4. Поставки;
     5. Пополнение склада;
     6. Отчёты по расходу;
     7. Отчёты по прибыли за промежуток;
  5. Администратор:
     1. Блюда;
     2. Заказы;
     3. Поставщики;
     4. Поставки;
     5. Пополнения склада;
     6. Составы блюд;
     7. Директор (функционал – пункт 1.4);
     8. Контролёр (функционал – пункт 1.3);
     9. Официант/Повар:
        1. Заказы:
           1. Перейти к внесению заказов:

Добавить посетителя;

* + - 1. Блюда:
         1. Изменить состав блюда;
    1. Отчёты по расходу;
    2. Отчёты по прибыли за промежуток.

# Заключение

В данном проекте была разработана система, автоматизирующая основную деятельность ресторанов. Система контролирует расходы и доходы ресторана, работу сотрудников; сохраняет данные обо всех основных процессах в самом ресторане. По сути, всё сводится к пополнению склада и реализации продуктов, и система отражает эти функции в отчётах. Система отвечает базовым требованиям и при необходимости может быть доработана. Единственным минусом конечного продукта является ресурсоёмкость, прямо следующая из возможных размеров базы данных и использования приложения с Windows-формами на C#. Для реализации данного проекта были освоены основы SQL-запросов (+ процедуры и триггеры) и возможности соединения БД с приложением на формах.

# Используемая литература

1. Ревунков Г.И., Ковалева Н.А., Силантьева Е.Ю., Виноградова М.В., Маслеников К.Ю. / «МЕТОДИЧЕСКИЕ УКАЗАНИЯ К КУРСОВОЙ РАБОТЕ ПО ДИСЦИПЛИНЕ "БАЗЫ ДАННЫХ"»
2. Документация по C# [Электронный ресурс] / <https://docs.microsoft.com/ru-ru/dotnet/csharp/>
3. Справочник по Transact-SQL [Электронный ресурс] / <https://docs.microsoft.com/ru-ru/sql/t-sql/language-reference?view=sql-server-ver15>
4. Знакомство с нотацией IDEF0 и пример использования [Электронный ресурс] / <https://habr.com/ru/company/trinion/blog/322832/>
5. Что такое DFD (диаграммы потоков данных)

[Электронный ресурс] / <https://habr.com/ru/company/trinion/blog/340064/>

1. Microsoft® SQL Server® 2012 Создание запросов. Учебный курс Microsoft: Пер. с англ. / И. Бен-Ган, Д. Сарка, Р. Талмейдж. — М.: Издательство «Русская редакция», 2014 — 720 с.э

# ПРИЛОЖЕНИЯ

# ПРИЛОЖЕНИЕ 1. Техническое задание

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия)

« \_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**Курсовой проект по курсу «Базы данных»**

**АС «Учёт и планирование ресторанного меню»**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

|  |  |
| --- | --- |
| Руководитель проекта: |  |
| Маслеников К. Ю. | (Подпись)  «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |
|  |  |
| Исполнитель проекта:  Забурунов Л. В., РТ5-41Б | (Подпись)  «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

Москва, 2020 г.

[1. Полное наименование системы 38](#_Toc43756427)

[2. Основания для проведения работы 38](#_Toc43756428)

[3. Назначение разработки 38](#_Toc43756429)

[4. Исполнитель 38](#_Toc43756430)

[5. Технические требования к системе 38](#_Toc43756431)

[6. Требования к входным и выходным данным 39](#_Toc43756432)

[7. Требования к серверному обеспечению 39](#_Toc43756433)

[8. Требования к программному обеспечению 39](#_Toc43756434)

[9. Требования к техническому обеспечению 40](#_Toc43756435)

[10. Требования к лингвистическому обеспечению 40](#_Toc43756436)

[11. Требования к условиям эксплуатации 40](#_Toc43756437)

[12. Требования к надежности 40](#_Toc43756438)

[13. Требования к документации 40](#_Toc43756439)

[14. Стадии и этапы разработки 41](#_Toc43756440)

[15. Порядок контроля и приема задания 41](#_Toc43756441)

[16. Дополнительные условия 41](#_Toc43756442)

# Полное наименование системы

Автоматизированная система учёта и планирования меню ресторана.

# Основания для проведения работы

Учебный план кафедры РТ-5 «Системы обработки информации и управления», МГТУ им. Н. Э. Баумана.

# Назначение разработки

Разрабатываемая система позволит упростить процесс управления рестораном путём автоматизации, а также увеличить его надёжность.

Автоматизированная система будет хранить все необходимые данные, связанные с рестораном, и создавать отчёты о деятельности ресторана, необходимые для оформления документации и дальнейшего анализа этого ресторана как бизнеса.

# Исполнитель

Забурунов Леонид Вячеславович, РТ5-41Б-2020, МГТУ им. Баумана.

# Технические требования к системе

Разрабатываемая система должна иметь дружелюбный интерфейс и позволять пользователям вносить, редактировать и удалять данные. Приложение должно быть создано на высокоуровневом языке C# с использованием Windows-форм, а база данных для этого приложения – на базе Microsoft SQL Server.

В системе должны быть предусмотрены категории пользователей со следующими возможностями:

1. Официант: позволяется просматривать блюда и его составляющие, а так же вносить изменения в заказы из зала ресторана;
2. Повар: позволяется просматривать заказы из зала, а так же вносить изменения в меню ресторана;
3. Ответственное за поставки лицо: доступ к просмотру меню и заказов, а так же к изменениям данных, связанных с поставками продуктов в ресторан;
4. Директор ресторана: доступ к просмотру меню, заказов, данных о поставках, а так же к изменению данных о сотрудниках;
5. Администратор системы имеет полный доступ в систему.

# Требования к входным и выходным данным

В первую очередь система должна обрабатывать данные о поступающих из зала заказах. Помимо основного требования, в систему поступают данные о меню ресторана, о работниках ресторана, о поставщиках продуктов и о совершаемых поставках продуктов.

На выходе система представляет полученные данные в виде (виртуальных) таблиц, а так же формирует отчёты о деятельности ресторана.

# Требования к серверному обеспечению

1. Платформа .NET Framework 4.7.2;
2. SQL Server 2019.

# Требования к программному обеспечению

**Для разработчиков:**

1. Microsoft SQL Server Management Studio;
2. Среда разработки Microsoft Visual Studio 2019 Community Edition с пакетом «Классические приложения C#»;
3. Операционная система Microsoft Windows 10.

**Для пользователя**

1. Операционная система Microsoft Windows 10.

# Требования к техническому обеспечению

1. Компьютер с монитором;
2. Клавиатура, мышь.

Для плавной работы приложения рекомендуется использовать компьютер, имеющий процессор по мощности эквивалентный процессору, имеющему 2 физических и 4 логических ядра и построенному на архитектуре AMD64 (или вытекающей из неё Intel x86-64).

# Требования к лингвистическому обеспечению

Интерфейс приложения пользователя должен быть выполнен на русском языке.

# Требования к условиям эксплуатации

АС должна работать корректно при эксплуатации пользователем.

# Требования к надежности

Система должна сохранять данные без искажений и потерь.

# Требования к документации

Для приема работы необходимо предоставить следующие документы:

* 1. Разрабатываемые технические и эксплуатационные документы;
  2. Техническое задание на разработку АИС;
  3. Расчетно-пояснительная записка со следующими графическими материалами:
  4. Диаграмма DFD функциональной модели предметной области;
  5. Диаграмма IDEF0 функциональной модели предметной области;
  6. Структурная схема АИС;
  7. Инфологическая модель предметной области (графическая диаграмма);
  8. Даталогическая модель предметной области (графическая диаграмма);
  9. Граф диалога системы;
  10. Схема работы системы;
  11. Интерфейс пользователя (формы, запросы, отчеты);
  12. Руководство пользователя;
  13. Программа и методика испытаний.

# Стадии и этапы разработки

* 1. Исследование предметной области – 2-ая неделя 4 семестра;
  2. Разработка инфологической модели – 4-ая неделя 4 семестра;
  3. Разработка датологической модели – 8-ая неделя 4 семестра;
  4. Разработка базы данных – 10-ая неделя 4 семестра;
  5. Программирование приложения – 12-ая неделя 4 семестра;
  6. Тестирование приложения – 12-ая неделя 4 семестра;
  7. Оформление документации – 14-ая неделя 4 семестра;
  8. Защита курсового проекта – 16-ая неделя 4 семестра.

# Порядок контроля и приема задания

Прием и контроль программного изделия «АИС Фитнес-центр» производится в соответствии с программой и учебным планом.

# Дополнительные условия

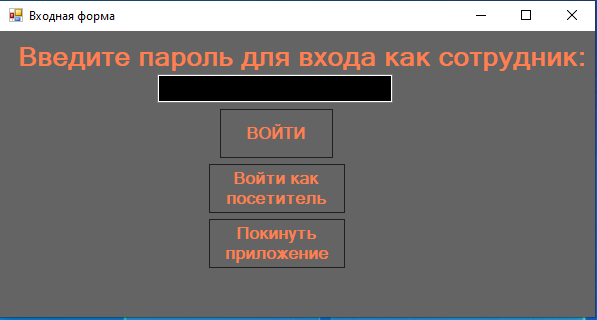
Техническое задание может уточняться в соответствии с установленным порядком.

# ПРИЛОЖЕНИЕ 2. Интерфейс системы

# Формы

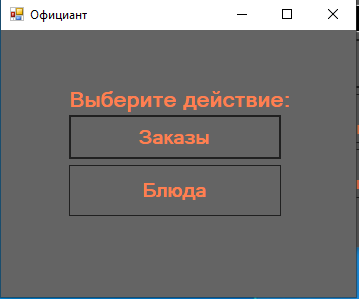
* 1. Входная форма

Позволяет всем желающим получить необходимую информацию в соответствии с категорией доступа



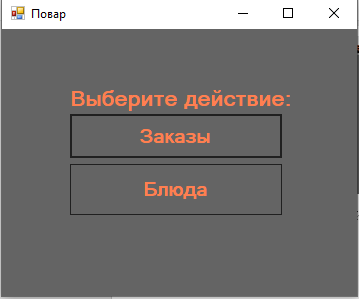
* 1. Базовая форма «Официант»

Позволяет выбрать официанту дальнейшее действие в системе



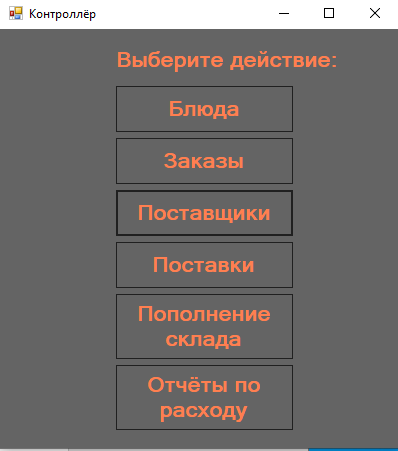
* 1. Входная форма «Повар»

Позволяет выбрать официанту дальнейшее действие в системе



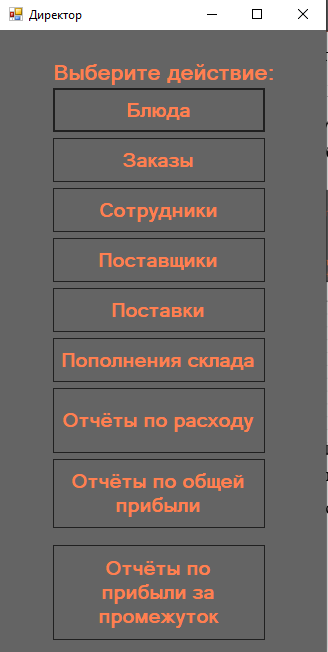
* 1. Входная форма «Контролёр»

Позволяет выбрать контролёру дальнейшее действие в системе



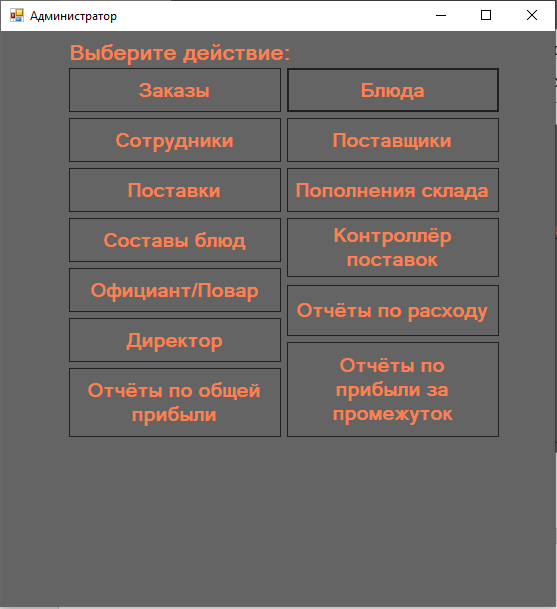
* 1. Входная форма «Директор»

Позволяет выбрать директору дальнейшее действие в системе



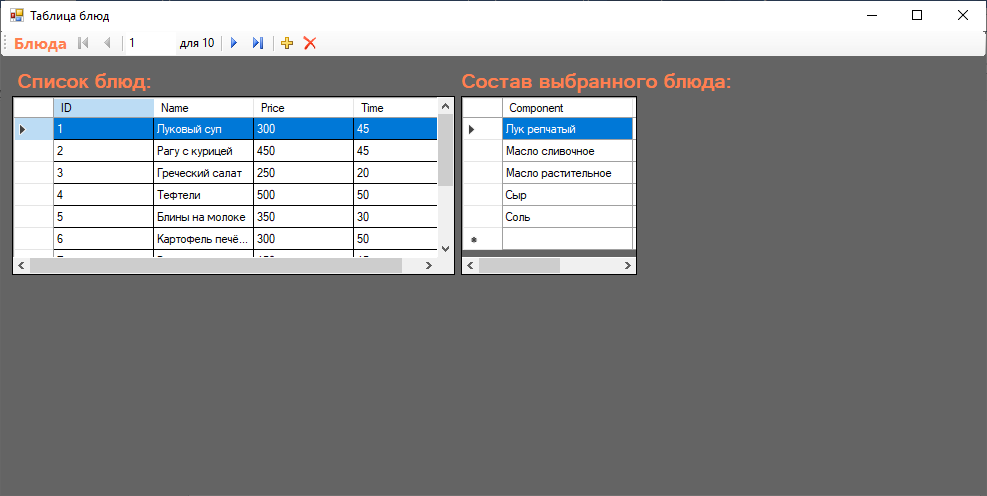
* 1. Входная форма «Администратор»

Позволяет выбрать администратору дальнейшее действие в системе



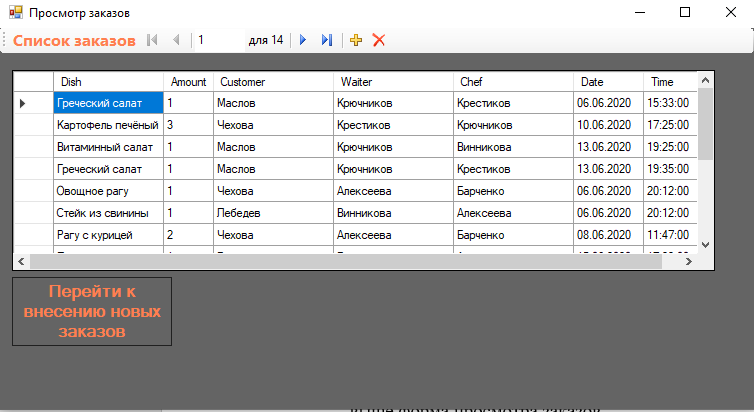
* 1. Форма «Таблица блюд»

Позволяет просматривать меню ресторана



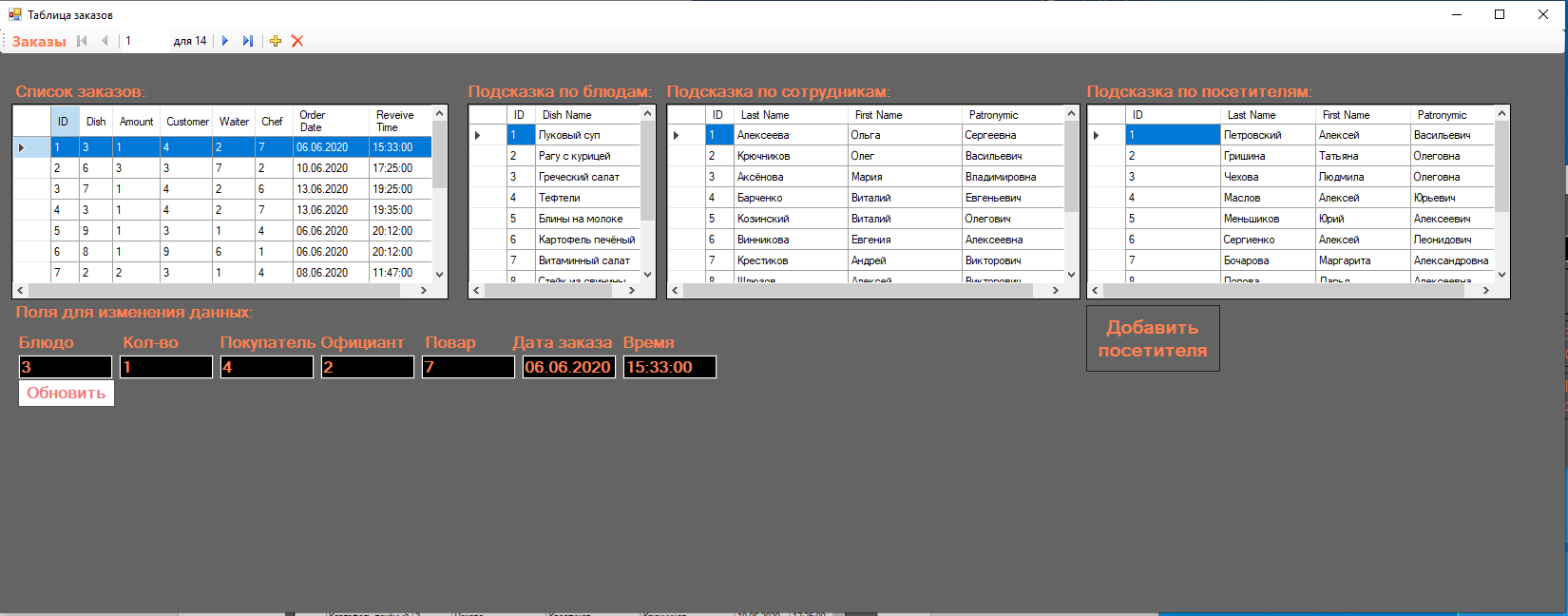
* 1. Форма «Просмотр заказов»

Позволяет просматривать совершённые заказы



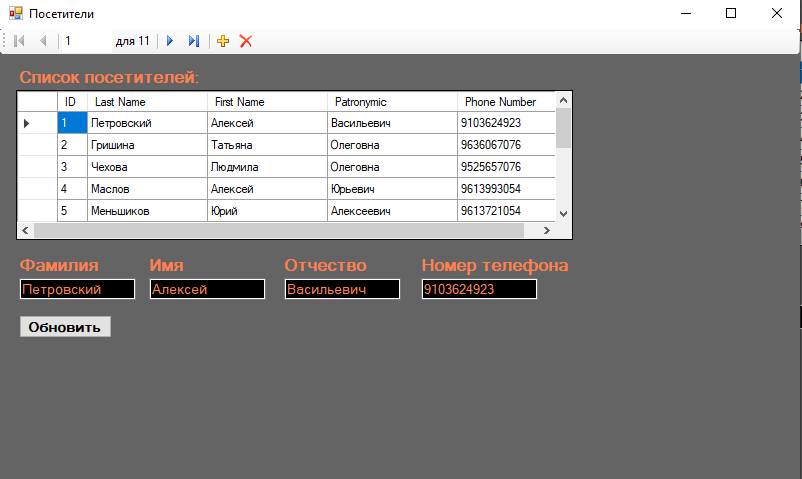
* 1. Форма «Таблица заказов»

Позволяет редактировать данные по заказам



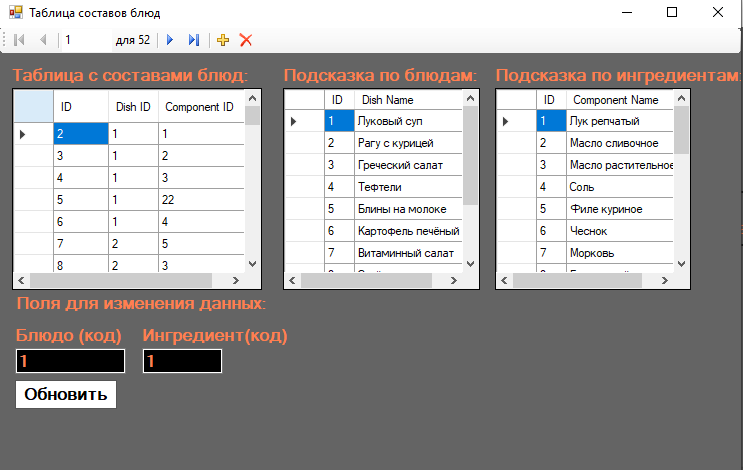
* 1. Форма «Посетители»

Позволяет редактировать данные о гостях ресторана



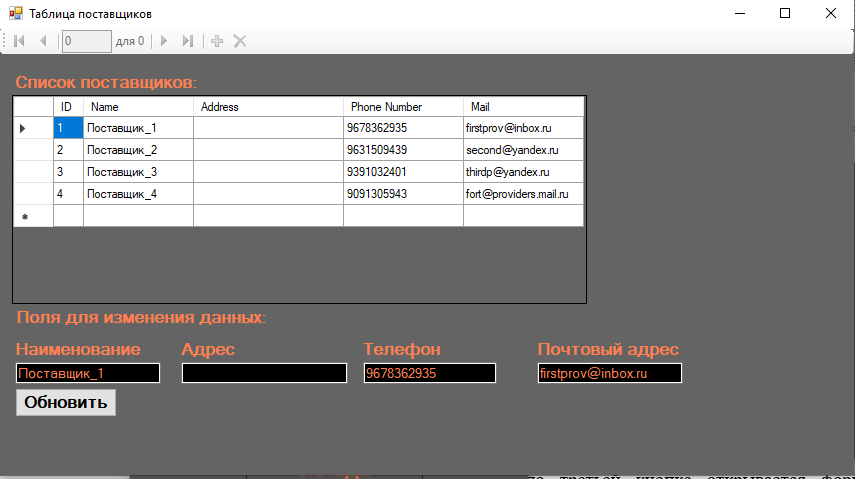
* 1. Форма «Таблица составов блюд»

Позволяет вносить изменения в состав блюд из меню ресторана



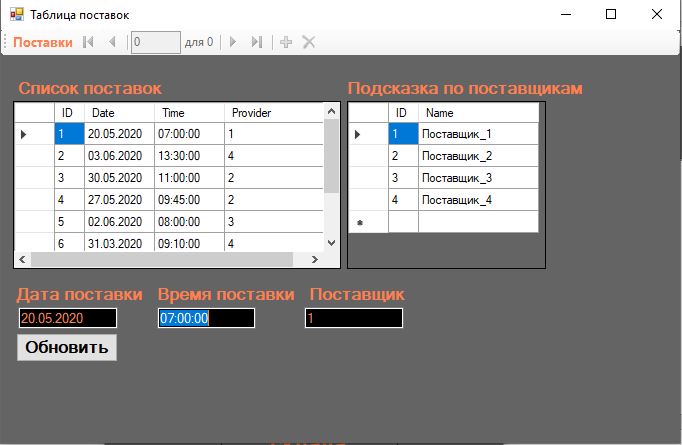
* 1. Форма «Таблица поставщиков»

Позволяет вносить изменения в данные о поставщиках



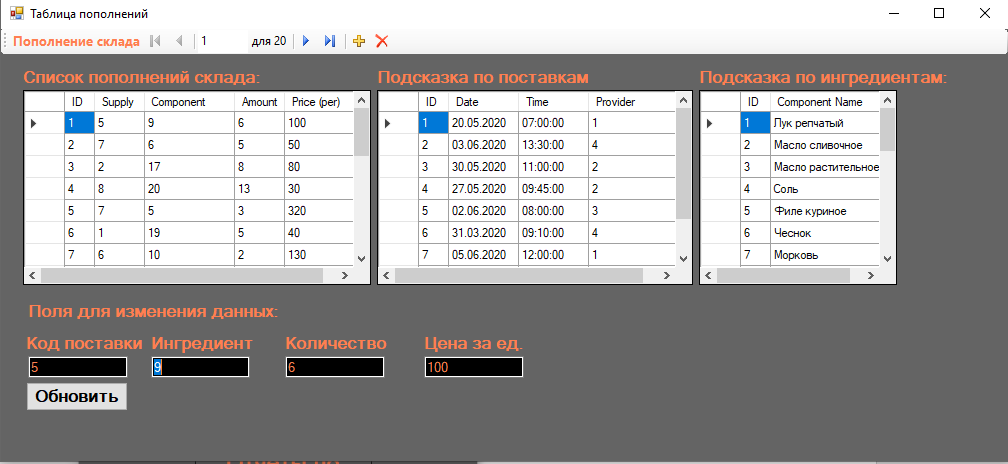
* 1. Форма «Таблица поставок»

Позволяет вносить изменения в данные о поставках



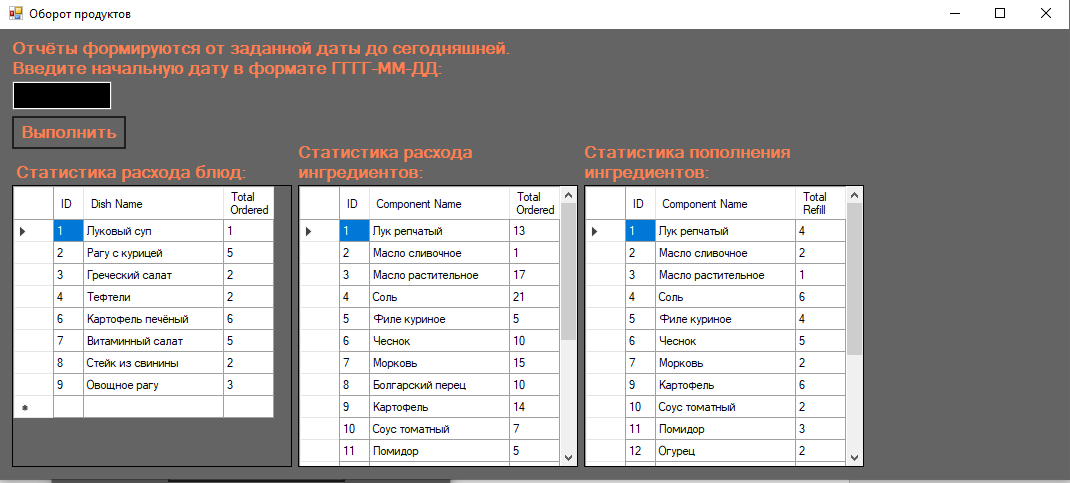
* 1. Форма «Таблица пополнений»

Позволяет редактировать данные о пополнениях склада



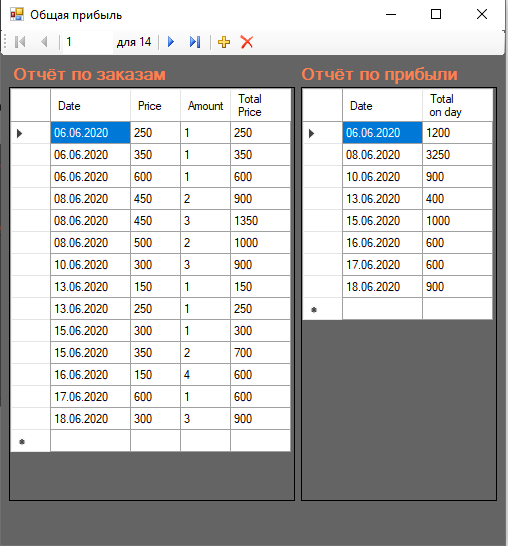
* 1. Форма «Оборот продуктов»

Позволяет получить отчёт по приходу и расходу ингредиентов



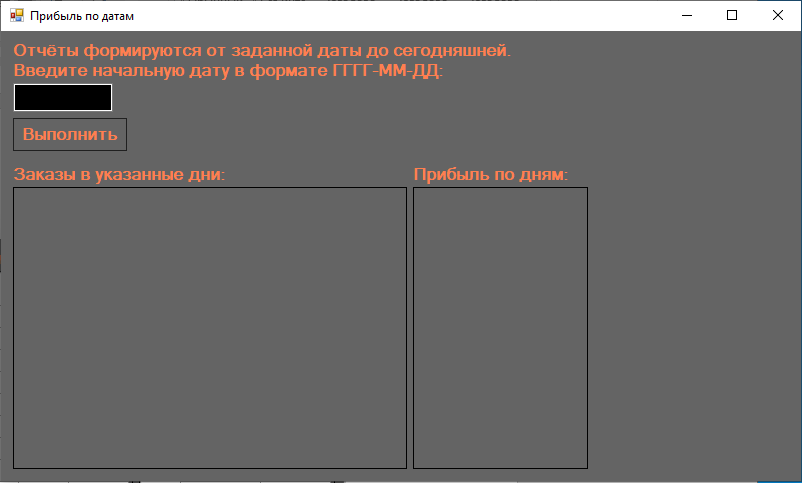
* 1. Форма «Общая прибыль»

Позволяет получить отчёт, в котором будет указана общая стоимость каждого заказа, а так же общая сумма заказов за день



* 1. Форма «Прибыль по датам»

Позволяет получить отчёт по прибыли ресторана за определённый период



# Запросы

* 1. Для формы «Просмотр заказов» реализован следующий запрос:

SELECT dbo.Dish.\_name, dbo.Orders.amount, C.last\_name AS Customer, A.last\_name AS Waiter, B.last\_name AS Chef, dbo.Orders.order\_date AS Date, dbo.Orders.receive\_time AS Time

FROM dbo.Orders INNER JOIN

dbo.Employee AS A ON A.empl\_id = dbo.Orders.waiter INNER JOIN

dbo.Employee AS B ON B.empl\_id = dbo.Orders.chef INNER JOIN

dbo.Customer AS C ON C.cust\_id = dbo.Orders.customer INNER JOIN

dbo.Dish ON dbo.Dish.dish\_id = dbo.Orders.dish

* 1. Для формы «Таблица заказов» реализованы следующие запросы:

SELECT dish\_id AS ID, \_name AS [Dish Name]

FROM dbo.Dish

SELECT cust\_id AS ID, last\_name AS [Last Name], first\_name AS [First Name], patronymic

FROM dbo.Customer

SELECT empl\_id, last\_name AS [Last Name], first\_name AS [First Name], patronymic, position, empl\_date AS [First Day], paycheck AS Salary, schedule, phone\_number AS [Phone Number], mailbox AS EMail

FROM dbo.Employee

* 1. Для формы «Таблица блюд» используется процедура, реализующая запрос:

proc [dbo].[Sostav] @dish\_id numeric(3) = NULL as select KRBD\_Restaraunt.dbo.Component.\_name as [Component], KRBD\_Restaraunt.dbo.Component.allergic as [Allergic] from KRBD\_Restaraunt.dbo.Component

join KRBD\_Restaraunt.dbo.Dish\_Structure on (Dish\_Structure.component = Component.comp\_id)

join KRBD\_Restaraunt.dbo.Dish on (Dish.dish\_id = Dish\_Structure.dish)

where (Dish.dish\_id = @dish\_id)

* 1. Для формы «Оборот продуктов» используются следующие процедуры:

proc [dbo].[Dish\_Consumption] @date as date = NULL as select Dish.dish\_id as ID, Dish.\_name as [Dish Name], sum(Orders.amount) as [Total Ordered]

from Dish

join Orders on (Dish.dish\_id = Orders.dish)

where (Orders.order\_date between @date AND GETDATE())

group by Dish.dish\_id, Dish.\_name

proc [dbo].[Component\_Consumption] @date as date = NULL as select Component.comp\_id as [ID], Component.\_name as [Component Name], sum(Orders.amount) as [Total Ordered]

from Component

join Dish\_Structure on (Component.comp\_id = Dish\_Structure.component)

join Dish on (Dish\_Structure.dish = Dish.dish\_id)

join Orders on (Dish.dish\_id = Orders.dish)

where (Orders.order\_date between @date AND GETDATE())

group by Component.comp\_id, Component.\_name

proc [dbo].[Component\_Refill] @date as date = NULL as select Component.comp\_id as [ID], Component.\_name as [Component Name], sum(Refill.amount) as [Total Refill] from Component

join Refill on (Refill.component = Component.comp\_id)

join Supply on (Supply.supply\_id = Refill.supply\_id)

where (Supply.supply\_date between @date and GETDATE())

group by comp\_id, Component.\_name

* 1. Для форм «Прибыль по дате» и «Общая прибыль» используются следующие процедуры:

proc [dbo].[Date\_Checks] @date1 as date = '2000-01-01' as select [Date], sum ([Total Price]) as [Total on day] from Orders\_checks where ([Date] between @date1 and GETDATE()) group by [Date]

proc [dbo].[Date\_orders] @date1 as date = '2000-01-01' as select Orders.order\_date as [Date], Dish.\_name as Dish, Dish.\_price as [Price], sum (Orders.amount) as Amount, Dish.\_price \* Orders.amount as [Total Price]

from Orders

join Dish on (Dish.dish\_id = orders.dish)

where (Orders.order\_date between @date1 and GETDATE())

group by Orders.order\_date, Dish.\_name, Dish.\_price, Orders.amount

# ПРИЛОЖЕНИЕ 3. Руководство пользователя

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия)

« \_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**Курсовой проект по курсу «Базы данных»**

**АС «Учёт и планирование меню ресторана»**

Руководство пользователя

(вид документа)

писчая бумага

(вид носителя)

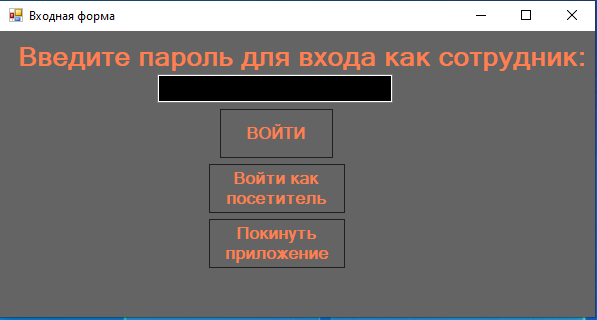
|  |  |
| --- | --- |
| Руководитель проекта: |  |
| Маслеников К. Ю. | (Подпись)  «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |
|  |  |
| Исполнитель проекта:  Забурунов Л. В., РТ5-41Б | (Подпись)  «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

Москва, 2020 г.

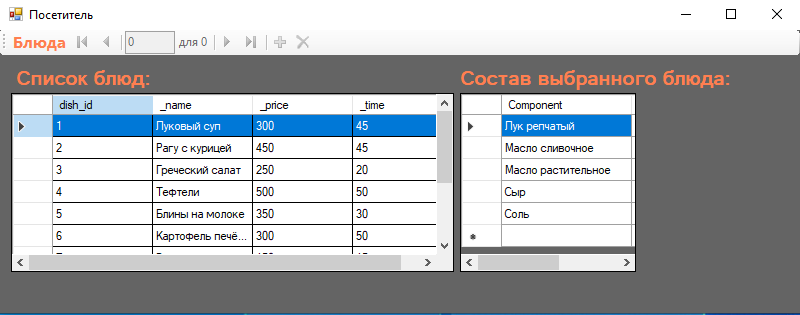
Для запуска приложения не требуется дополнительная установка или настройка; достаточно наличия отвечающего требования мощности компьютера.

Описание интерфейса программы:

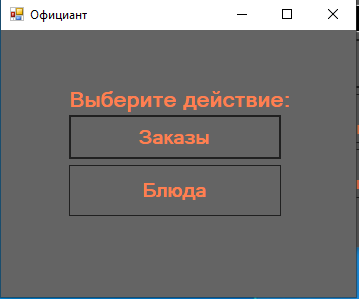
1. Входная форма позволяет авторизоваться в системе как сотрудник с помощью пароля, просмотреть меню как посетитель ресторана или же выйти из приложения. При вводе неверного пароля выводится соответствующее диалоговое окно.



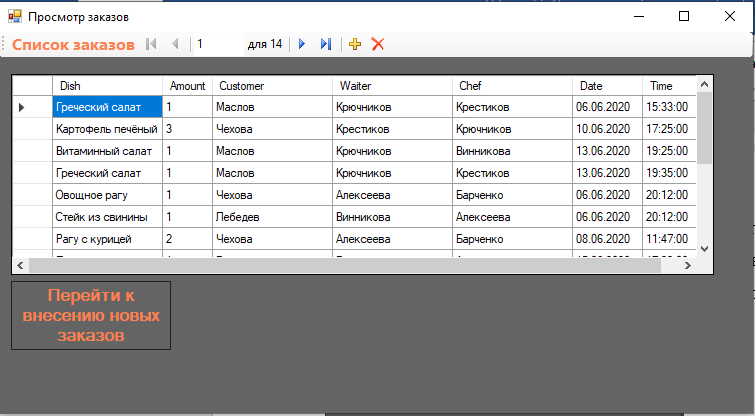
1. В случае входа как посетитель доступна единственная форма, в которой есть две таблицы: первая показывает основную информацию о блюде, а вторая выводит состав блюда, строка которого выбрана в первой таблице



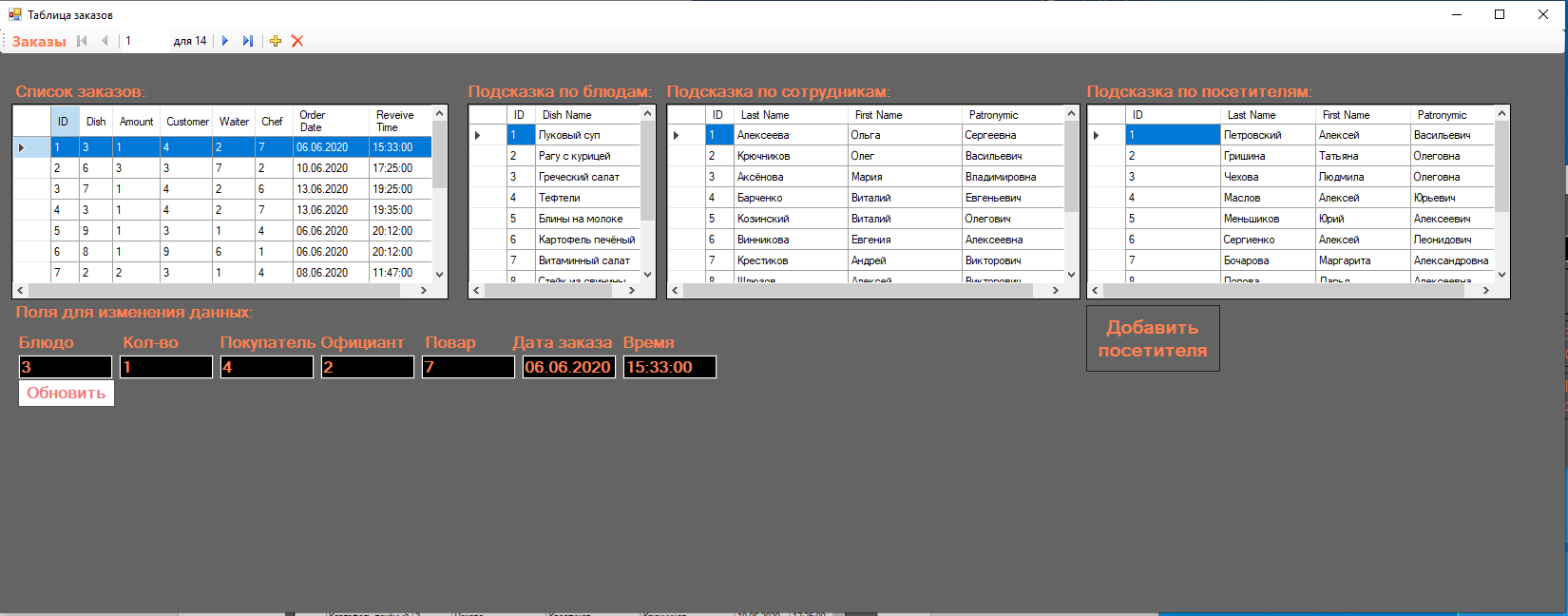
1. При авторизации официантом предлагает перейти по двум кнопкам к различным действиям: по первой кнопке возможно редактирование информации о заказах и посетителях, а по второй – просмотр меню



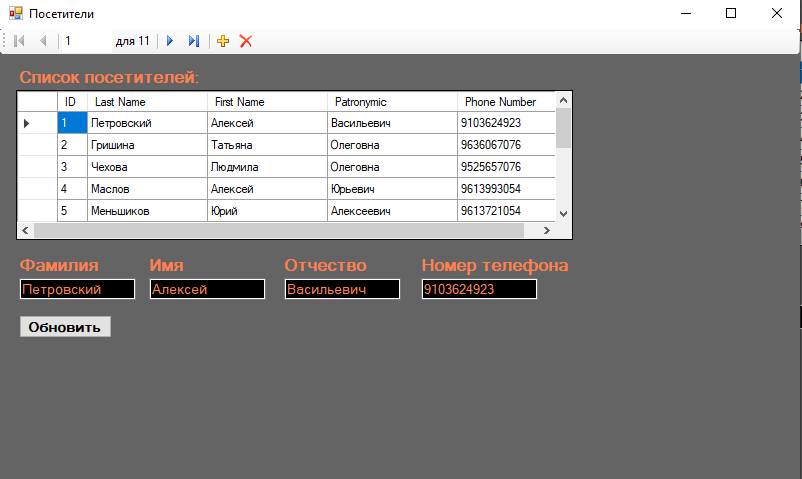
* 1. При переходе по первой кнопке открывается форма просмотра заказов, из которой по кнопке можно перейти к редактированию заказов



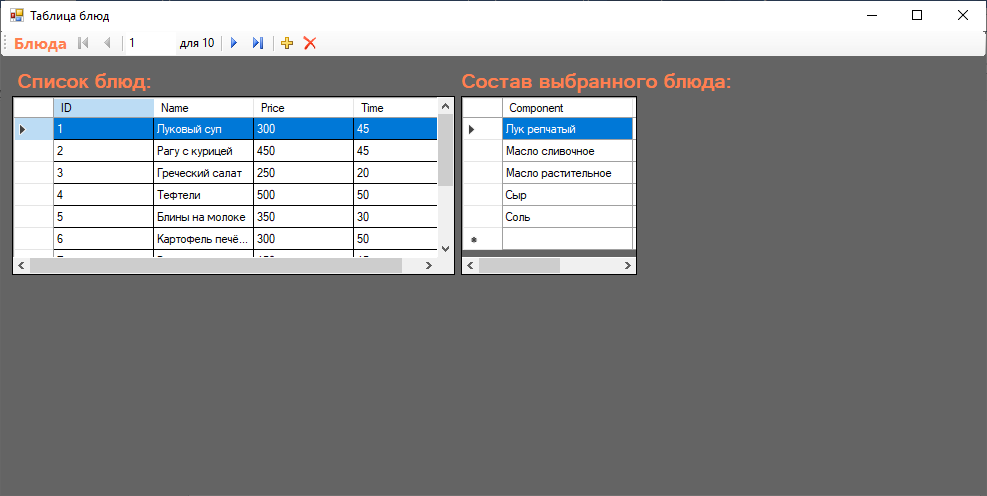
* + 1. В данной форме с помощью панели и кнопки в нижней части можно редактировать данные о заказах; с помощью кнопки в правой части формы можно перейти к редактированию данных о посетителях ресторана. Поскольку в базе данных хранятся значения-ключи, на форму выведены подсказки для того, чтобы пользователь смог вставить корректные ключи блюда, сотрудника и посетителя



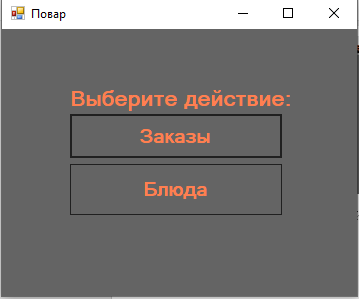
* + - 1. В данной форме с помощью панели внизу можно вносить изменения в данные о посетителях



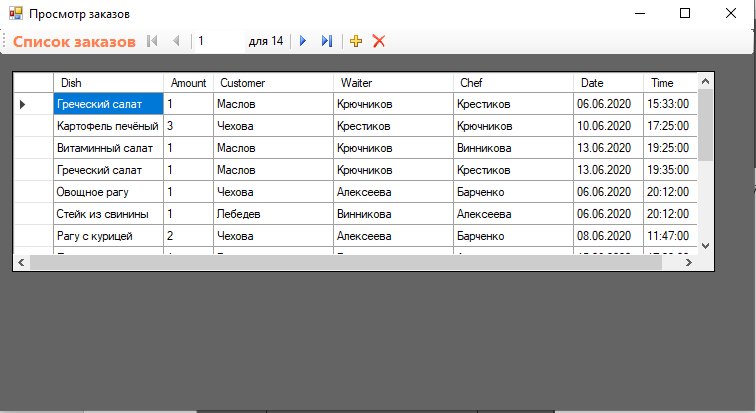
* 1. При переходе по второй кнопке можно просмотреть меню аналогично при входе как посетитель



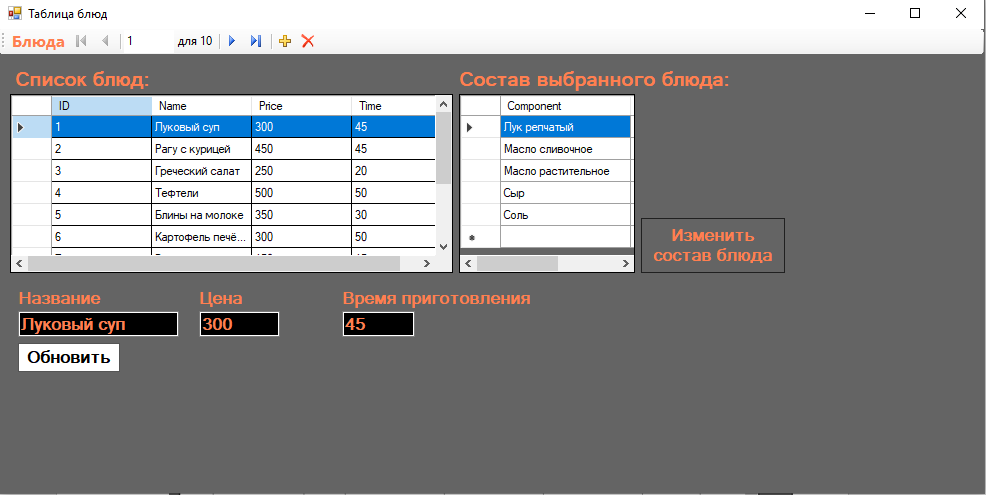
1. При авторизации поваром открывается подобная официанту форма переходов



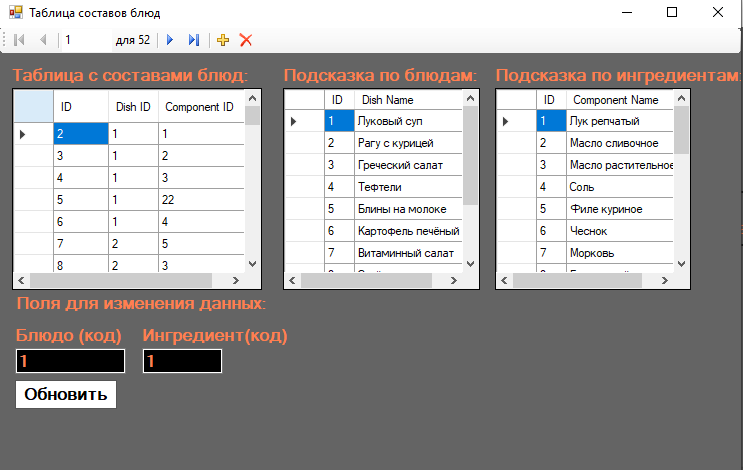
* 1. При переходе по первой кнопке открывается форма для просмотра заказов



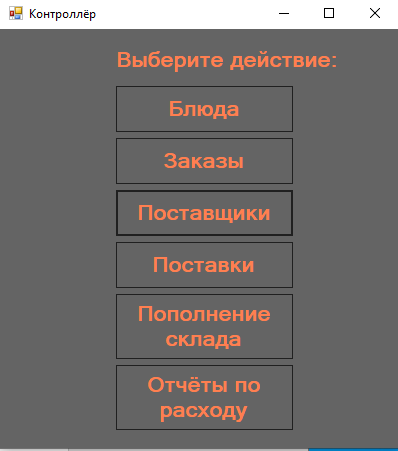
* 1. При переходе по второй кнопке открывается форма для редактирования меню; в правой части формы доступна кнопка открытия формы, отвечающей за составы блюд в меню



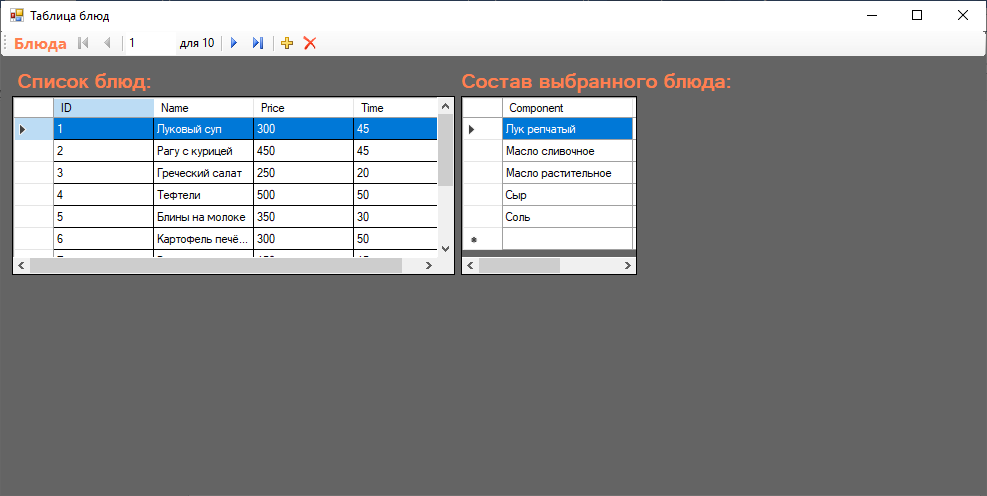
* + 1. На данной форме можно внести изменения в состав блюд; присутствует аналогичная пункту 3.1.1 подсказка



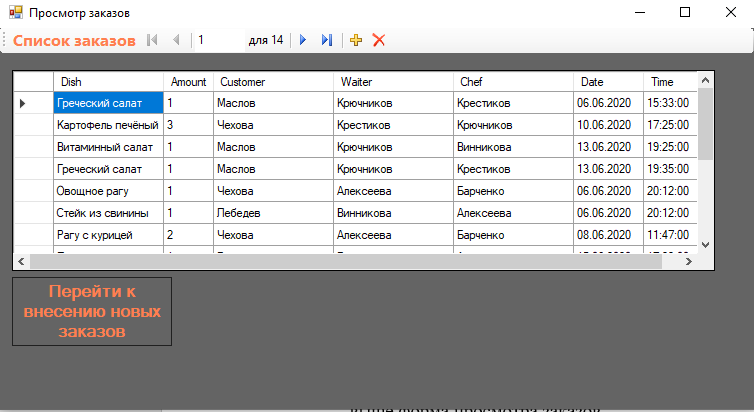
1. При авторизации контролёром поставок программа встречает знакомой формой с выбором действий, однако здесь набор действий расширяется.



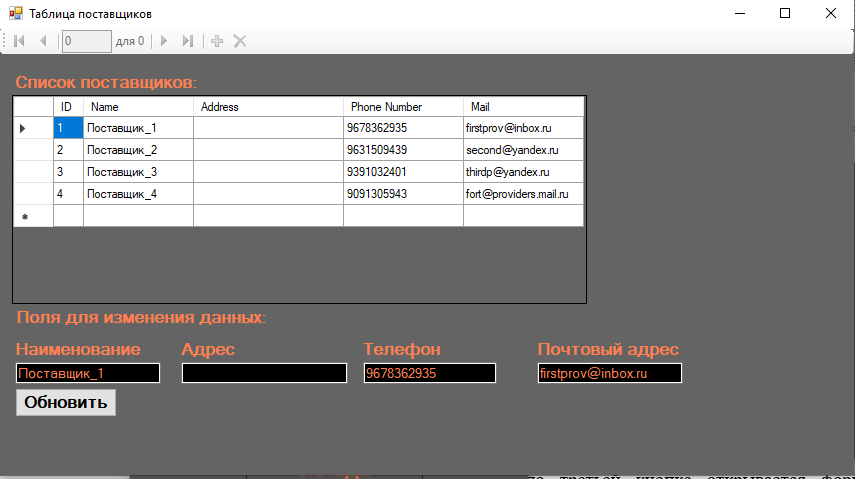
* 1. При переходе по первой кнопке открывается описанная выше форма просмотра меню



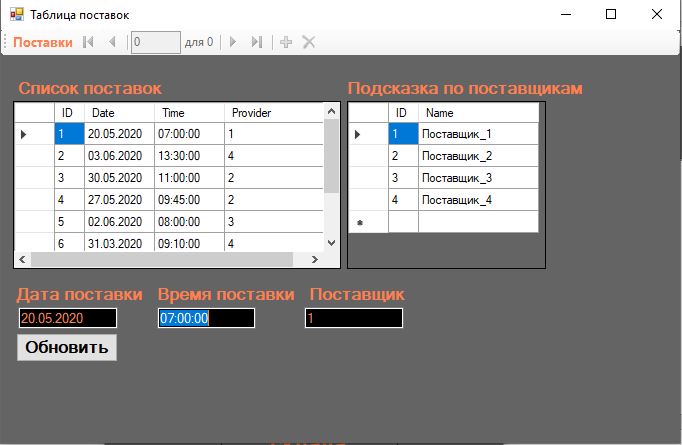
* 1. При переходе по второй кнопке открывается описанная выше форма просмотра заказов



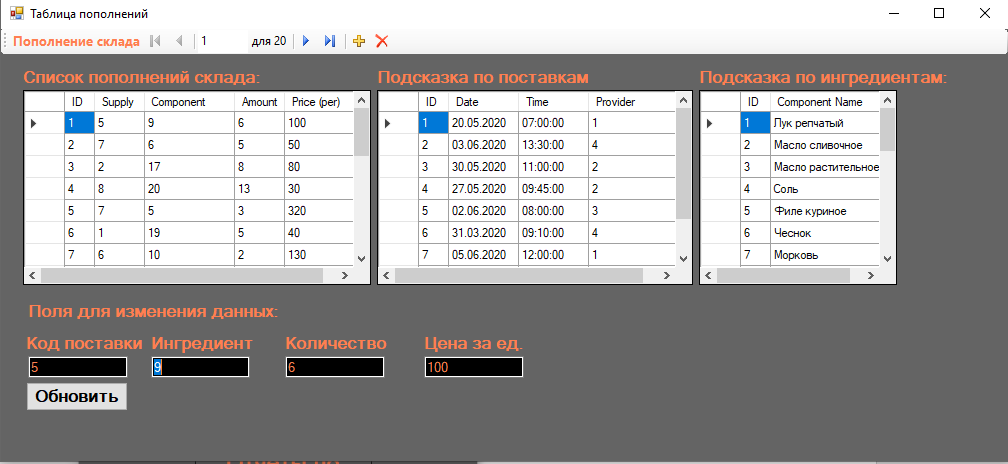
* 1. При переходе по третьей кнопке открывается форма редактирования данных о поставщиках



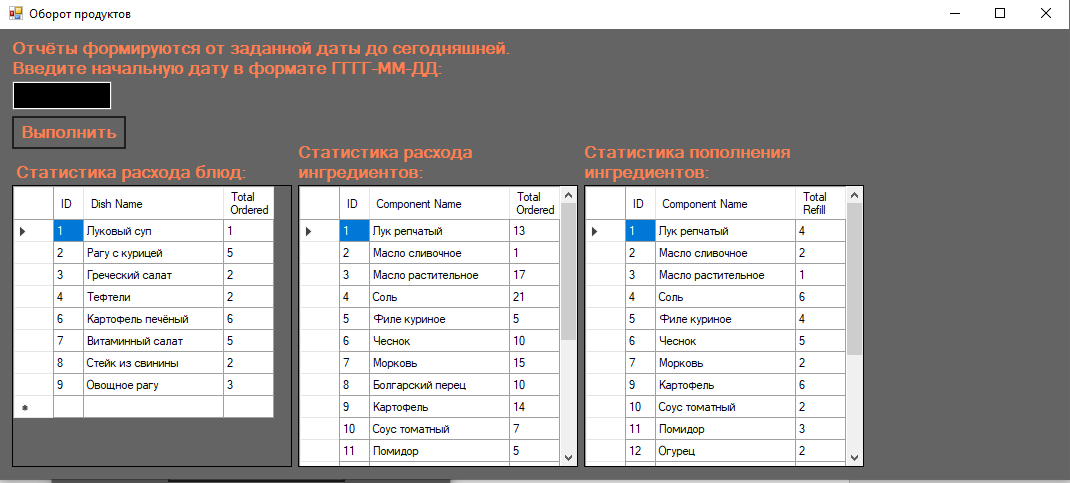
* 1. При переходе по четвёртой кнопке открывается форма редактирования данных о поставках



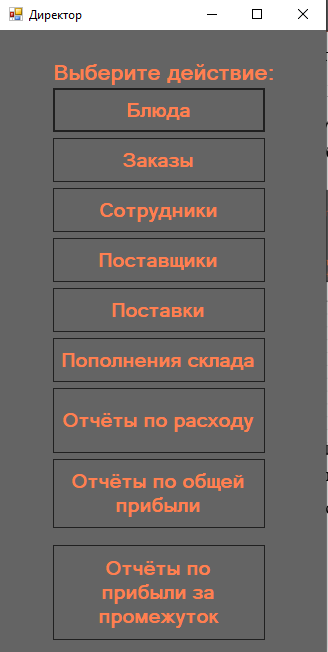
* 1. При переходе по пятой кнопке открывается форма редактирования данных о пополнениях склада



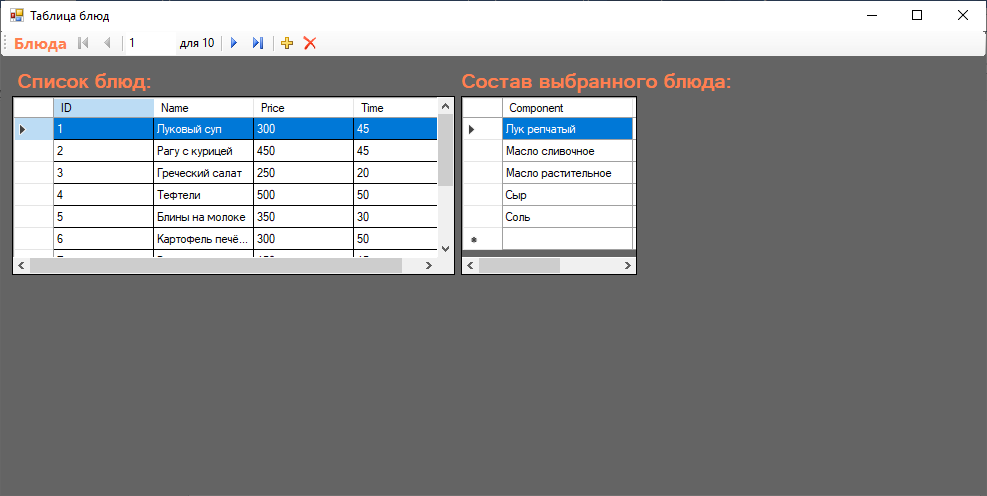
* 1. При переходе по пятой кнопке открывается форма с отчётом по приходу и расходу ингредиентов. Отчёт формируется от заданной даты до текущей; если необходимо просмотреть выборку за всё время, то необходимо очистить поле и нажать кнопку выполнения



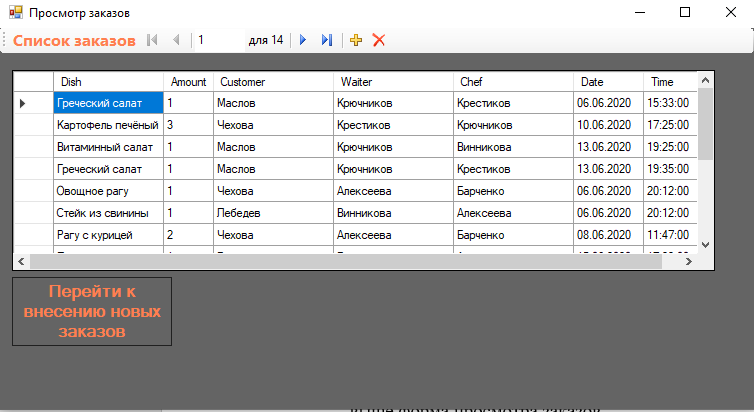
1. При авторизации директором приветственная форма выглядит следующим образом



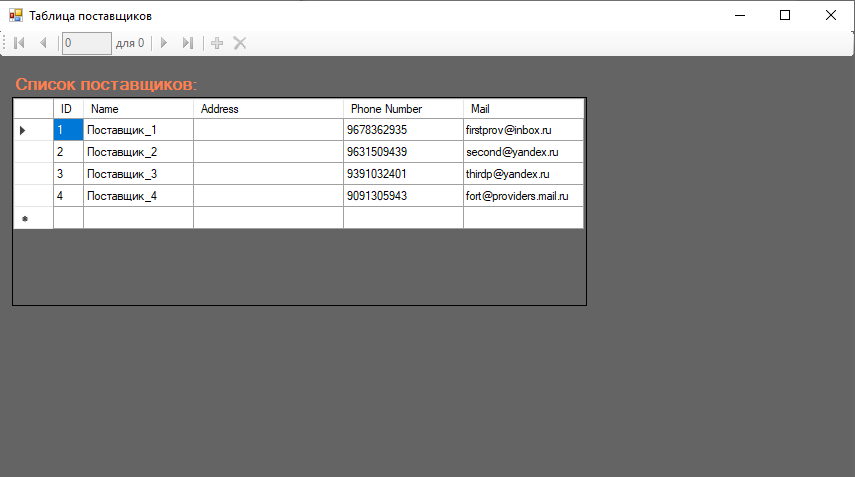
* 1. При переходе по первой кнопке открывается описанная выше форма просмотра меню



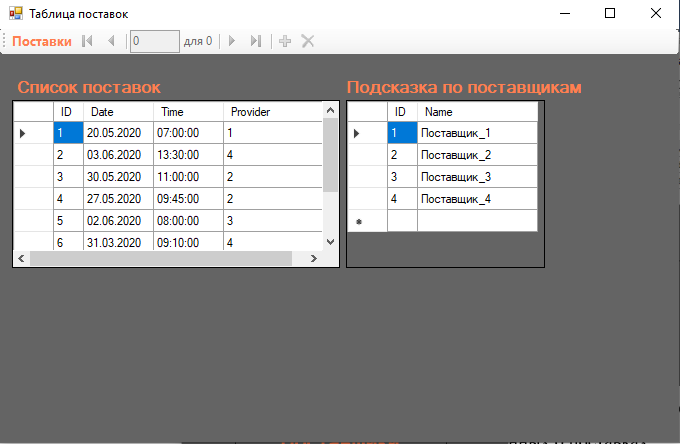
* 1. При переходе по второй кнопке открывается описанная выше форма просмотра заказов



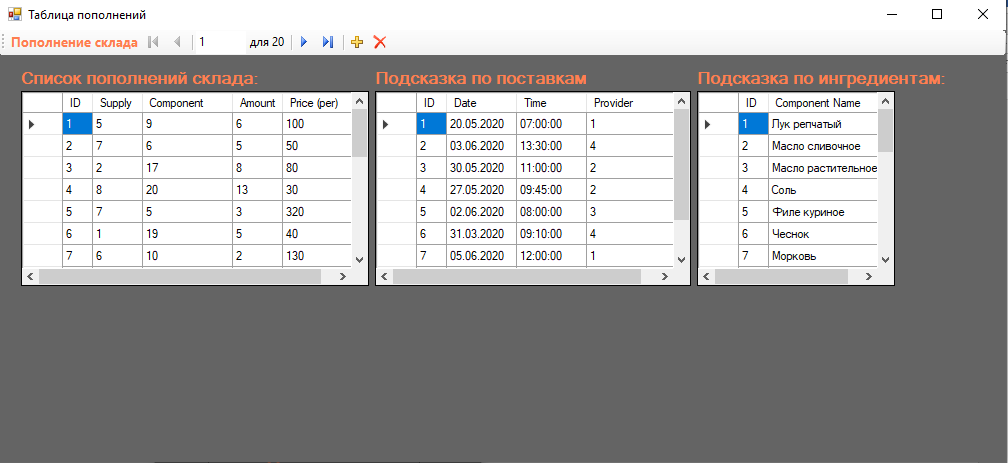
* 1. При переходе по третьей кнопке открывается форма просмотра данных о поставщиках



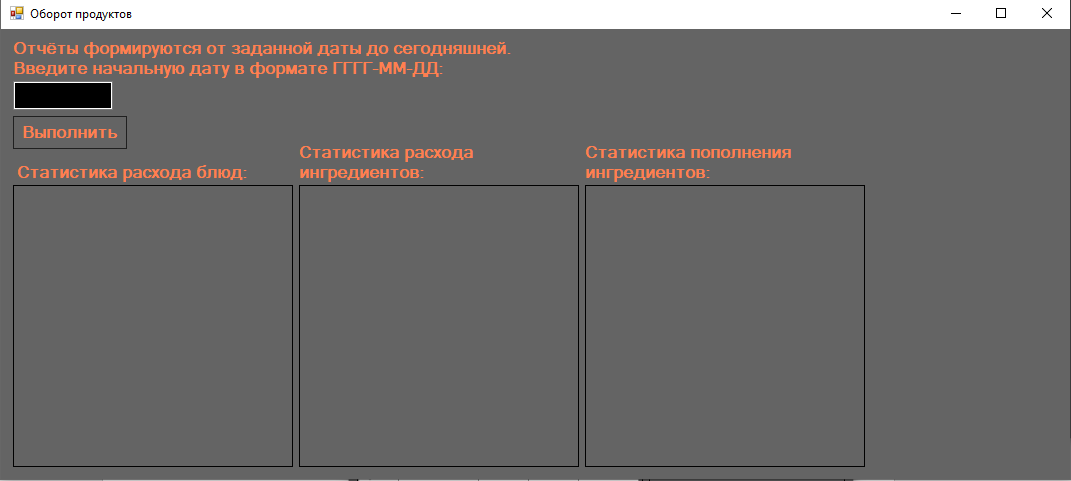
* 1. При переходе по четвёртой кнопке открывается форма просмотра данных о поставках



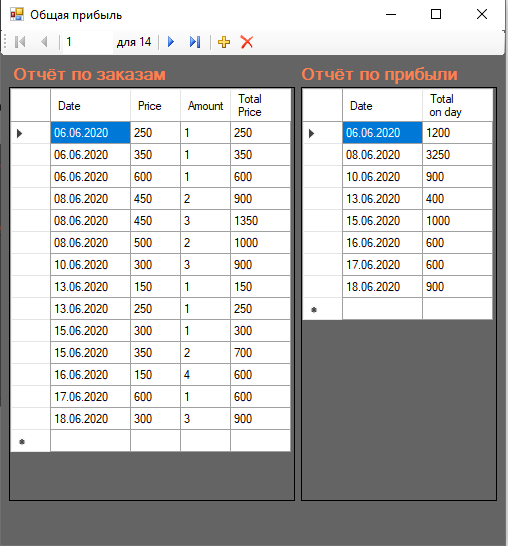
* 1. При переходе по пятой кнопке открывается форма редактирования данных о пополнениях склада



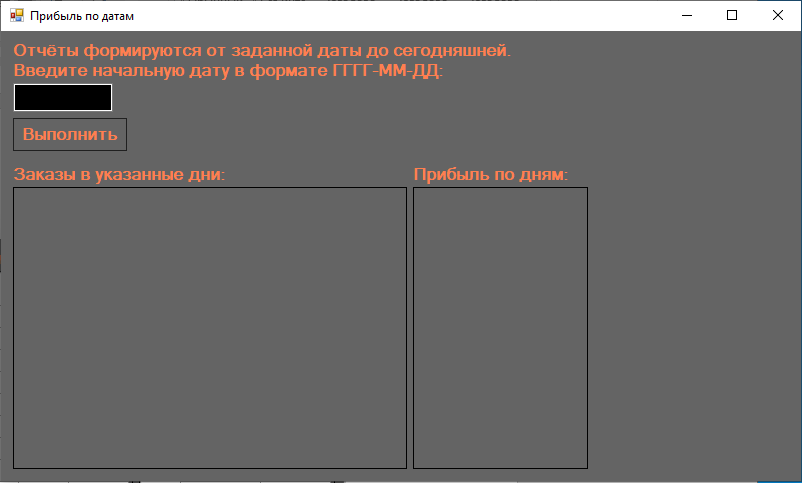
* 1. При переходе по шестой кнопке открывается знакомая форма с отчётом по приходу и расходу ингредиентов.



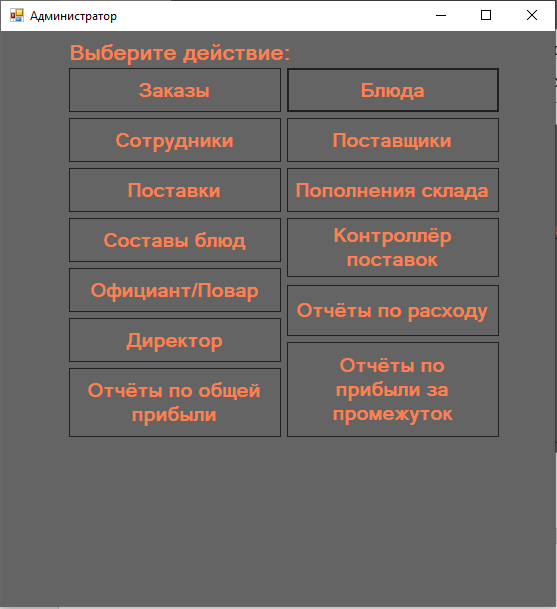
* 1. При нажатии на следующую кнопку открывается форма с отчётом по прибыли: левая таблица показывает итоговую сумму для заказов, правая считает общую сумму за день.



* 1. Последняя кнопка открывает форму с отчётом о прибыли в заданный интервал дат (от заданной до текущей).



1. При авторизации администратом приложение открывает форму, из которой можно открыть доступ ко всем описанным выше формам (где возможно – в режиме редактирования), а так же к стартовым формам каждой из категорий пользователей.



* 1. Единственным отличием является переход по кнопке «Официант/Повар». Связано оно с тем, что для официантов и поваров используются одни и те же формы, но приложение скрывает ненужные элементы в зависимости от категории доступа. В случае администратора на данной форме будут доступны все элементы; таким образом, получается совмещение пунктов 3.1 и 4.2.

# ПРИЛОЖЕНИЕ 4. Испытания

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия)

« \_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**Курсовой проект по курсу «Базы данных»**

**АС «Учёт и планирование меню ресторана»**

Программа и методика испытаний

(вид документа)

писчая бумага

(вид носителя)

|  |  |
| --- | --- |
| Руководитель проекта: |  |
| Маслеников К. Ю. | (Подпись)  «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |
|  |  |
| Исполнитель проекта:  Забурунов Л. В., РТ5-41Б | (Подпись)  «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

Москва, 2020 г.

# Объект проводимых испытаний

Объектом испытаний является разработанная АС.

# Цель проводимых испытаний

Целью испытаний является установление соответствия системы техническим требования.

# Состав и порядок проводимых испытаний

1. Для входной формы проверена работа всех трёх кнопок: кнопка войти реагирует диалоговым окном в случае ввода пароля, не соответствующего ни одной категории доступа, либо соответствующей категории доступа формой выбора дальнейших действий; кнопка входа для посетителей открывает меню; кнопка выхода останавливает работу приложения.
2. Все формы, использующие данные из БД («Просмотр заказов», «Таблица заказов», «Посетители», «Таблица блюд», «Таблица составов блюд», «Таблица поставщиков», «Таблица поставок», «Таблица пополнений», «Таблица сотрудников», «Оборот продуктов», «Общая прибыль», «Прибыль по датам», «Посетитель»), отображают получаемые данные корректно: все значения соответствуют источнику данных (проверка производилась в SQL Server Management Studio).
3. Все формы, которые позволяют пользователю изменять данные в БД («Таблица заказов», «Посетители», «Таблица блюд», «Таблица составов блюд», «Таблица поставщиков», «Таблица поставок», «Таблица пополнений», «Таблица сотрудников»), работают корректно: значения, заносимые в панель элементов, передаются в исходную запись, а кнопка обновления сохраняет результат в источнике данных.
4. Формы, на которых реализована отчётность («Оборот продуктов», «Общая прибыль», «Прибыль по датам»), обрабатывают даты (в случае корректного ввода) без ошибок и выводят корректный результат запроса (проверка производилась в SQL Server Management Studio).
5. Формы выбора действий для каждой категории доступа («Официант/Повар», «Контролёр», «Директор», «Администратор») при нажатии на кнопку открывают форму, указанную на самой кнопке.