

КУРСОВОЙ ПРОЕКТ

**Разработка прототипа приложения типа клиент-сервер для управления
Ресторана**
по дисциплине «Базы данных»

Выполнил

студент гр. 3530904/90101



Казимиров Н.А.

Руководитель
к.т.н.

Андреев И.А.

«4» декабря 2021 г.

Оглавление

Задание	3
Введение	4
Анализ предметной области	5
Схема Базы Данных.....	9
Реализация	10
Десять сложных запросов	13
Хранимые процедуры и триггеры.....	15
Клиентское приложение.....	16
Форма входа	16
Меню Администратора.....	17
Меню Повара.....	20
Меню Посетителя	222
Меню Закупщика	25
Вывод.....	28
Список литературы	29

Задание НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА (КУРСОВОЙ РАБОТЫ)

Санкт-Петербургский политехнический университет Петра Великого

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА (КУРСОВОЙ РАБОТЫ)

студенту

группы

3530904/90001

(номер группы)

Казимиров Никита Андреевич

(фамилия, имя, отчество)

1. Тема проекта (работы)

Разработка многопользовательской

автоматизированной системы управления организацией. Объект

автоматизации - ресторан

2. Срок сдачи студентом законченного проекта (работы)

04.12.2021

3. Исходные данные к проекту (работе)

Описание предметной области

4. Содержание пояснительной записки (перечень подлежащих разработке вопросов: введение, основная часть (раскрывается структура основной части), заключение, список использованных источников, приложения).

Введение. Анализ предметной области. Проектирование схемы данных.

Реализация базы данных в среде SQL Server. Разработка представлений и

хранимых процедур. Разработка клиентского приложения. Тестирование.

Заключение. Список использованных источников.

Примерный объем пояснительной записки 15-20 страниц машинописного текста

5. Перечень графического материала (с указанием обязательных чертежей и плакатов) не предоставляется

6. Консультанты

7. Дата получения задания: «4» сентября 2021 г.

Руководитель

(подпись)

(инициалы, фамилия)

Задание принял к исполнению

(подпись)

(инициалы, фамилия)

(дата)

Введение

Необходимо реализовать многопользовательскую автоматизированную систему управления работы ресторана. В системе должно быть 4 типа пользователей: Администратор, Посетитель, Повар, Закупщик.

Применение баз данных является необходимым условием эффективной организации работы ресторана. Благодаря базам данных каждый Администратор, посетитель, повар и закупщик всегда имеют доступ к актуальной информации. А благодаря графическому интерфейсу написанному на языке C# происходит удобный и безопасный доступ к информации, содержащийся в базе данных.

Анализ предметной области

Группы пользователей разрабатываемой информационной системы (ИС)

№ пп	Наименование пользователя
1	Администратор
2	Посетитель
3	Повар
4	Поставщик

Функции групп пользователей

№ пп	Выполняемая функция	Входные данные	Выходные данные	Функции, которые должны быть реализованы в ИС
Первый пользователь(группа)				
1	Добавление персонала	ФИО, дата рождения, должность, зарплата, номер телефона	Список персонала	Добавление нового официанта
2	Увольнение персонала	ФИО сотрудника, его ID		Увольнение нерадивых официантов
3	Бронирование столиков	ФИО, список свободных столиков, дата брони, номер телефона		Бронирование столика на имя клиента, внесение номера стола
4	Ответ на отзывы посетителей	Список отзывов		Ответ на отзывы покупателей
Второй пользователь(группа)				
1	Просмотр свободных столов и бронь столика	ФИО, выбор свободного столика, дата, номер телефона	Номер забронированного столика и дата брони	Бронирование столика на определенное время
2	Отмена брони	Номер забронированного столика и дата брони		Отмена бронирования
3	Добавление заказа	Список блюд	Наименование, количество, цена	Заказ блюда и напитков
4	Просмотр меню		Список блюд и напитков	Наименования блюд и напитков, их цена
5	Добавление отзывов о ресторане	Список отзывов		Оценка ресторана, обслуживания и качества блюд
Третий пользователь(группа)				
1	Редактирование состава меню	Таблица меню		Редактирование ингредиентов для блюд

2	Просмотр заказов		Список заказов	Получение информации о заказах и их статусе
3	Добавление/удаление блюд			
Четвертый пользователь(группа)				
1	Закупка продуктов	Список ингредиентов на складе		Получение данных о продуктах со склада и последующая закупка недостающих товаров
2	Отслеживание товара на складе			Контроль и управление за количеством продуктов

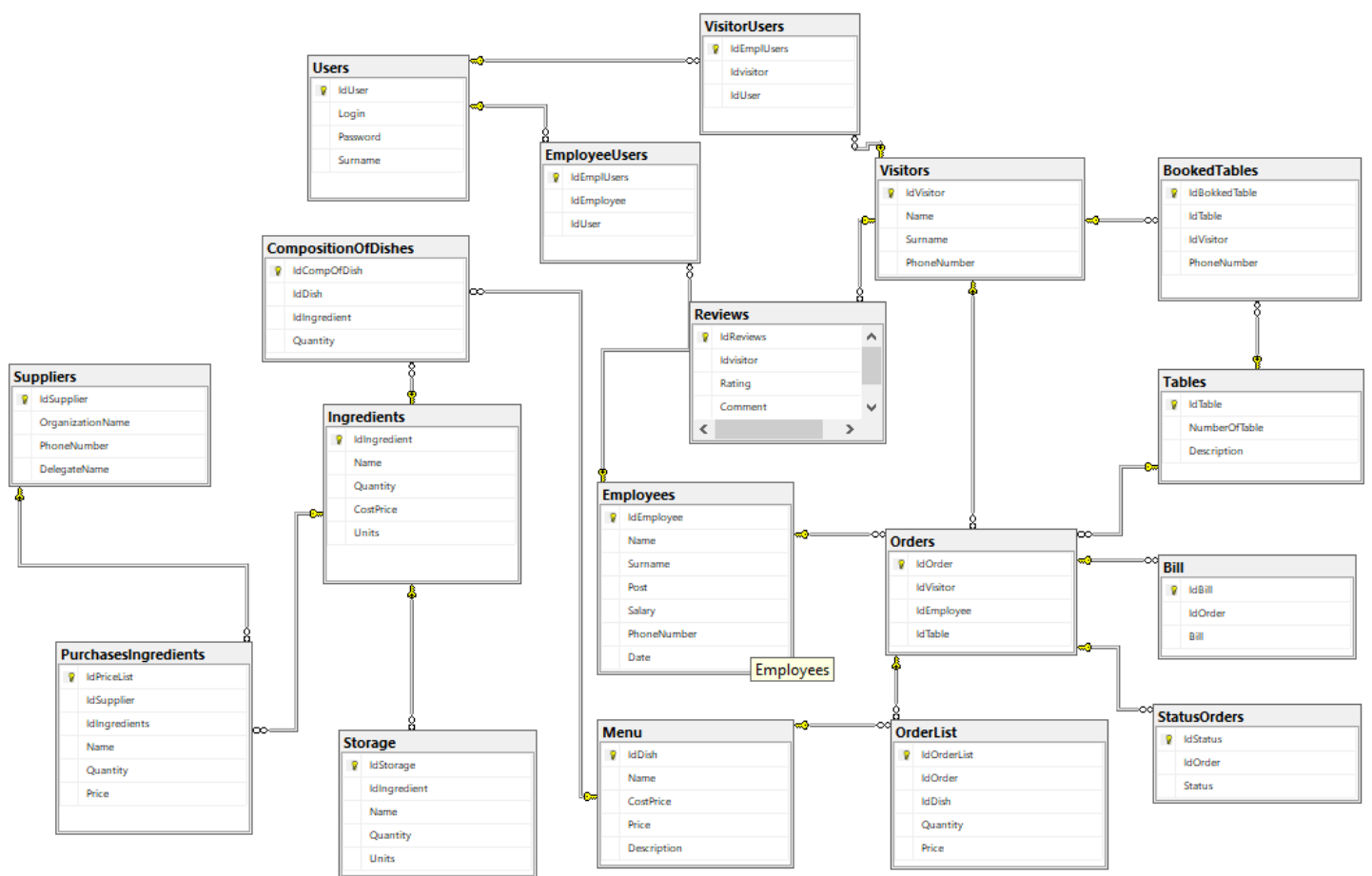
Хранимые данные

№ пп	Хранимые данные	Пользователи, которым разрешен доступ	Ограничения по типу и значению
1	Список сотрудников	Администратор	IDсотрудника (int) Фамилия (varchar 20) Имя (varchar 15) Отчество (varchar 20) Должность (varchar 20) Зарплата (int) Номер телефона (char 11) Дата приема на работу
2	Список посетителей		IDпосетителя (int) Фамилия (varchar 20) Имя (varchar 15) Отчество (varchar 20) Номер телефона (char 11)
3	Список забронированных столиков	Администратор, повар	IDзабронированногоСтолика (int) IDклиента(int) Дата бронирования Номер телефона посетителя (char 11)
4	Список свободных столиков	Администратор, посетитель	IDстола (int) Описание(varchar 100)
5	Заказ	Администратор, посетитель, повар	IDзаказа (int) IDпосетителя (int) IDсотрудника (int) IDстола (int) Счет (int) IDстатус(int)

6	Состав заказа	Повар, администратор	IDзаказа (int) ID блюда или напитка (int) Количество(int) Цена (int)
7	Меню ресторана	Администратор, посетитель, повар	ID блюд или напитка (int) Название блюда (varchar 50) Себестоимость (int) Цена блюда (int) Описание блюда (varchar(100))
8	Список отзывов	Администратор, посетитель, повар	IDотзыва (int) Оценка (int) Комментарий (varchar(100)) Ответ (varchar(100)) IDадминистратора (int)
9	Состав блюда	Повар	ID блюда или напитка (int) IDингредиента (int) Количество (int)
10	Таблица ингредиентов	Администратор, повар	IDингредиента (int) Название (varchar(50)) Количество (int) Цена (int) Единица измерения(varchar(10))
11	Склад	Повар, администратор, поставщик	IDсклада (int) IDингредиента (int) Название (varchar(50)) Количество (int) Единица измерения(varchar(10))
12	Таблица поставщиков	Поставщик, администратор	IDпоставщика (int) Название организации (varchar(50)) Номер телефона (char 11) Имя представителя (varchar(30))
13	Статус заказов	Повар, администратор	IDстатус (int) Статус заказа (varchar(30))
14	Закупка ингредиентов	Поставщик	IDпрайсЛист (int) IDпоставщика (int) IDингредиента (int) IDсклада (int) Название (varchar(30)) Количество (int) Цена (int)
№ пп	Хранимые данные	Пользователи, которым разрешен доступ	Ограничения по типу и значению

1	Список сотрудников	Администратор	IDсотрудника (int) Фамилия (varchar 20) Имя (varchar 15) Отчество (varchar 20) Должность (varchar 20) Зарплата (int) Номер телефона (char 11) Дата приема на работу
2	Список посетителей		IDпосетителя (int) Фамилия (varchar 20) Имя (varchar 15) Отчество (varchar 20) Номер телефона (char 11)
3	Список забронированных столиков	Администратор, повар	IDзабронированногоСтолика (int) IDклиента(int) Дата бронирования Номер телефона посетителя (char 11)
4	Список свободных столиков	Администратор, посетитель	IDстола (int) Описание(varchar 100)

Схема Базы Данных



Реализация

Система реализована при помощи MS SQL Server. Ниже показаны созданные таблицы:

```
USE master
GO

CREATE DATABASE Kursach ON
( NAME = Kursovaya_Polina,
  FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\
Kursach.mdf',
  SIZE = 3072KB,
  MAXSIZE = UNLIMITED,
  FILEGROWTH = 1024KB )

LOG ON
( NAME= Kursach_log,
  FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\
Kursach.ldf',
  SIZE = 1024KB,
  MAXSIZE = 2048GB,
  FILEGROWTH = 10% )

GO

USE Kursach
GO

CREATE TABLE Employees(
  IdEmployee int IDENTITY (1,1) PRIMARY KEY,
  Name varchar(30) NOT NULL,
  Surname varchar(50) NOT NULL,
  Post varchar(30) NOT NULL,
  Salary int NOT NULL,
  PhoneNumber char(11) NOT NULL,
  Date date NOT NULL
)
GO

CREATE TABLE Visitors(
  IdVisitor int IDENTITY (1,1) PRIMARY KEY,
  Name varchar(30) NOT NULL,
  Surname varchar(50) NOT NULL,
  PhoneNumber char(11) NOT NULL,
)
GO

CREATE TABLE Tables(
  IdTable int IDENTITY (1,1) PRIMARY KEY,
  NumberOfTable int NOT NULL,
  Description varchar(100) NOT NULL
)
GO

CREATE TABLE BookedTables(
  IdBokkedTable int IDENTITY (1,1) PRIMARY KEY,
  IdTable int NOT NULL FOREIGN KEY REFERENCES Tables(IdTable) ON DELETE CASCADE,
  IdVisitor int NOT NULL FOREIGN KEY REFERENCES Visitors(IdVisitor) ON DELETE CASCADE,
  PhoneNumber char(11) NOT NULL,
  BookingDate dateTime NOT NULL
)
GO
```

```

CREATE TABLE Orders(
    IdOrder int IDENTITY (1,1) PRIMARY KEY,
    IdVisitor int NOT NULL FOREIGN KEY REFERENCES Visitors(IdVisitor) ON DELETE CASCADE,
    IdEmployee int NOT NULL FOREIGN KEY REFERENCES Employees(IdEmployee) ON DELETE CASCADE,
    IdTable int NOT NULL FOREIGN KEY REFERENCES Tables(IdTable) ON DELETE CASCADE,
)
GO

CREATE TABLE StatusOrders(
    IdStatus int IDENTITY (1,1) PRIMARY KEY,
    IdOrder int NOT NULL FOREIGN KEY REFERENCES Orders(IdOrder) ON DELETE CASCADE,
    Status varchar(100) NOT NULL,
)
GO

CREATE TABLE Bill(
    IdBill int IDENTITY (1,1) PRIMARY KEY,
    IdOrder int NOT NULL FOREIGN KEY REFERENCES Orders(IdOrder) ON DELETE CASCADE,
    Bill int NOT NULL,
)
GO

CREATE TABLE Menu(
    IdDish int IDENTITY (1,1) PRIMARY KEY,
    Name varchar(50) NOT NULL,
    CostPrice int NOT NULL,
    Price int NOT NULL,
    Description varchar(100) NOT NULL
)
GO

CREATE TABLE OrderList(
    IdOrderList int IDENTITY (1,1) PRIMARY KEY,
    IdOrder int NOT NULL FOREIGN KEY REFERENCES Orders(IdOrder) ON DELETE CASCADE,
    IdDish int NOT NULL FOREIGN KEY REFERENCES Menu(IdDish) ON DELETE CASCADE,
    Quantity int NOT NULL,
    Price int NOT NULL,
)
GO

CREATE TABLE Ingredients(
    IdIngredient int IDENTITY (1,1) PRIMARY KEY,
    Name varchar(30) NOT NULL,
    Quantity int NOT NULL,
    CostPrice int NOT NULL,
    Units varchar(15) NOT NULL,
)
GO

CREATE TABLE CompositionOfDishes(
    IdCompOfDish int IDENTITY (1,1) PRIMARY KEY,
    IdDish int NOT NULL FOREIGN KEY REFERENCES Menu(IdDish) ON DELETE CASCADE,
    IdIngredient int NOT NULL FOREIGN KEY REFERENCES Ingredients(IdIngredient) ON DELETE
CASCADE,
    Quantity int NOT NULL,
)
GO

CREATE TABLE Storage(
    IdStorage int IDENTITY (1,1) PRIMARY KEY,
    IdIngredient int NOT NULL FOREIGN KEY REFERENCES Ingredients(IdIngredient) ON DELETE
CASCADE,
    Name varchar(30) NOT NULL,
    Quantity int NOT NULL,
    Units varchar(15) NOT NULL,
)

```

```

)
GO

CREATE TABLE Suppliers(
    IdSupplier int IDENTITY (1,1) PRIMARY KEY,
    OrganizationName varchar(50) NOT NULL,
    PhoneNumber char(12) NOT NULL,
    DelegateName varchar(30) NOT NULL
)
GO

CREATE TABLE PurchasesIngredients(
    IdPricelist int IDENTITY (1,1) PRIMARY KEY,
    IdSupplier int NOT NULL REFERENCES Suppliers(IdSupplier),
    IdIngredients int NOT NULL REFERENCES Ingredients(IdIngredient),
    IdStorage int NOT NULL REFERENCES Storage(IdStorage),
    Name varchar(30) NOT NULL,
    Quantity int NOT NULL,
    Price int NOT NULL,
)
GO

CREATE TABLE Reviews(
    IdReviews int IDENTITY (1,1) PRIMARY KEY,
    Idvisitor int NOT NULL FOREIGN KEY REFERENCES Visitors(IdVisitor) ON DELETE CASCADE,
    Rating int NOT NULL,
    Comment varchar(100) NOT NULL,
    IdEmployee int NOT NULL FOREIGN KEY REFERENCES Employees(IdEmployee) ON DELETE CASCADE,
    Answer varchar(100) NOT NULL,
)
GO

CREATE TABLE Users(
    IdUser int IDENTITY (1,1) PRIMARY KEY,
    Login varchar(30) NOT NULL,
    Password varchar(15) NOT NULL,
    Surname varchar(50) NOT NULL
)
GO

CREATE TABLE VisitorUsers(
    IdEmplUsers int IDENTITY (1,1) PRIMARY KEY,
    Idvisitor int NOT NULL FOREIGN KEY REFERENCES Visitors(IdVisitor) ON DELETE CASCADE,
    IdUser int NOT NULL FOREIGN KEY REFERENCES Users(IdUser) ON DELETE CASCADE,
)
GO

CREATE TABLE EmployeeUsers(
    IdEmplUsers int IDENTITY (1,1) PRIMARY KEY,
    IdEmployee int NOT NULL FOREIGN KEY REFERENCES Employees(IdEmployee) ON DELETE CASCADE,
    IdUser int NOT NULL FOREIGN KEY REFERENCES Users(IdUser) ON DELETE CASCADE,
)
GO

```

Десять сложных запросов

/* Все возможные сотрудники и покупатели */

```
SELECT Employees.Name, Employees.Surname
FROM Employees
UNION
SELECT Visitors.Name, Visitors.Surname
FROM Visitors
```

/* Поставщики, которые представлены в Закупке Ингридиентов */

```
SELECT * FROM Suppliers
WHERE IdSupplier IN
(SELECT idSupplier FROM PurchasesIngredients)
```

/* Блюдо, которе никто не заказывал */

```
SELECT Name FROM Menu
WHERE NOT EXISTS
(SELECT * FROM OrderList WHERE OrderList.IdDish = Menu.IdDish)
```

/* Состав Заказа */

```
SELECT OrderList.IdOrder, Menu.Name, Menu.Price, Menu.CostPrice, OrderList.Quantity,
Orders.IdTable, Orders.Bill
FROM OrderList
JOIN Menu ON OrderList.IdDish=Menu.IdDish
JOIN Orders ON OrderList.IdOrder=Orders.IdOrder
```

/* Необходимые ингридиенты для приготовления определённого блюда */

```
SELECT Menu.Name, Menu.Price, Ingredients.Name, CompositionOfDishes.Quantity
FROM Menu INNER
JOIN CompositionOfDishes ON CompositionOfDishes.IdDish = Menu.IdDish
INNER JOIN Ingredients ON CompositionOfDishes.IdIngredient =Ingredients.IdIngredient
where CompositionOfDishes.IdDish = 1
```

/* Что необходимо заказать */

```
SELECT Storage.Name FROM Ingredients
JOIN Storage ON Ingredients.IdIngredient = Storage.IdIngredient
WHERE Storage.Quantity - Ingredients.Quantity < '0'
```

/* Ингридиенты которые использовались в блюдах */

```
SELECT Name From Ingredients
WHERE EXISTS
(SELECT * FROM CompositionOfDishes WHERE Ingredients.IdIngredient =
CompositionOfDishes.IdIngredient)
```

/* Ингридиент которого нет на скалде но есть в закупке */

```
SELECT PurchasesIngredients.Name FROM PurchasesIngredients
EXCEPT
SELECT Storage.Name FROM Storage
```

/* Сотрудник с большим количесвтом заказов */

```
SELECT Count(*) AS Number_Of_orders, Name, Surname, Post
FROM Orders INNER JOIN Employees ON Orders.IdEmployee=Employees.IdEmployee
GROUP BY Orders.IdEmployee, Employees.Name, Employees.Surname, Employees.Post
HAVING Count(Orders.IdOrder)>Max(1)
```

/* Общая выручка */

```
SELECT
Sum(OrderList.Price * OrderList.Quantity)
```

```

    AS Price, Count(*) AS Количество_блюд
FROM Orders INNER JOIN OrderList
ON Orders.IdOrder=OrderList.IdOrder

```

```

/* Показывает у каждого сотрудника информацию о заказе и его статус */
SELECT Employees.IdEmployee, Employees.Name, Orders.IdOrder, Bill.Bill, Orders.IdTable,
Visitors.Name, StatusOrders.Status FROM Orders
    JOIN Employees ON Orders.IdEmployee=Employees.IdEmployee
    JOIN Visitors ON Orders.IdVisitor=Visitors.IdVisitor
    JOIN Bill ON Orders.IdOrder=Bill.IdOrder
    JOIN StatusOrders ON Orders.IdOrder=StatusOrders.IdStatus

```

```

/* Самое популярное блюдо */
SELECT Count(*) AS Quantity, Name, Menu.Price
FROM OrderList INNER JOIN Menu ON OrderList.IdDish=Menu.IdDish
INNER JOIN Orders ON OrderList.IdOrder=Orders.IdOrder
    GROUP BY Menu.Name, Menu.Price
    HAVING Count(OrderList.Quantity)>Max(Quantity)

```

Хранимые процедуры и триггеры

```
/* Выборка всех ингредиентов */
CREATE PROCEDURE all_Ingredients
AS
    SELECT Name, Quantity, CostPrice FROM Ingredients
GO

EXEC all_Ingredients

/*Занести в таблицу Orders нового заказа*/
CREATE PROCEDURE insert_new_order (@id_Visitor AS INT, @id_Empoye AS INT, @id_Table AS INT,
@id_Status AS INT, @bill AS INT, @status AS VARCHAR(30))
AS    INSERT INTO Orders (IdVisitor,IdEmployee, IdTable, IdStatus, Bill, Status) VALUES
(@id_Visitor, @id_Empoye, @id_Table, @id_Status, @bill, @status);
GO*/

DECLARE @visitor int = 2, @emp int= 3, @table int= 2, @status_ int = 2, @bill_ int = 1200,
@stat varchar(30) = 'Завершен';
EXEC insert_new_order @visitor, @emp, @table, @status_, @bill_, @stat;

/* Блюдо, которое еще не заказывали */
CREATE PROCEDURE Dish_that_not_ordered
AS
    SELECT Name FROM Menu
    WHERE NOT EXISTS
    (SELECT * FROM OrderList WHERE OrderList.IdDish = Menu.IdDish)
GO

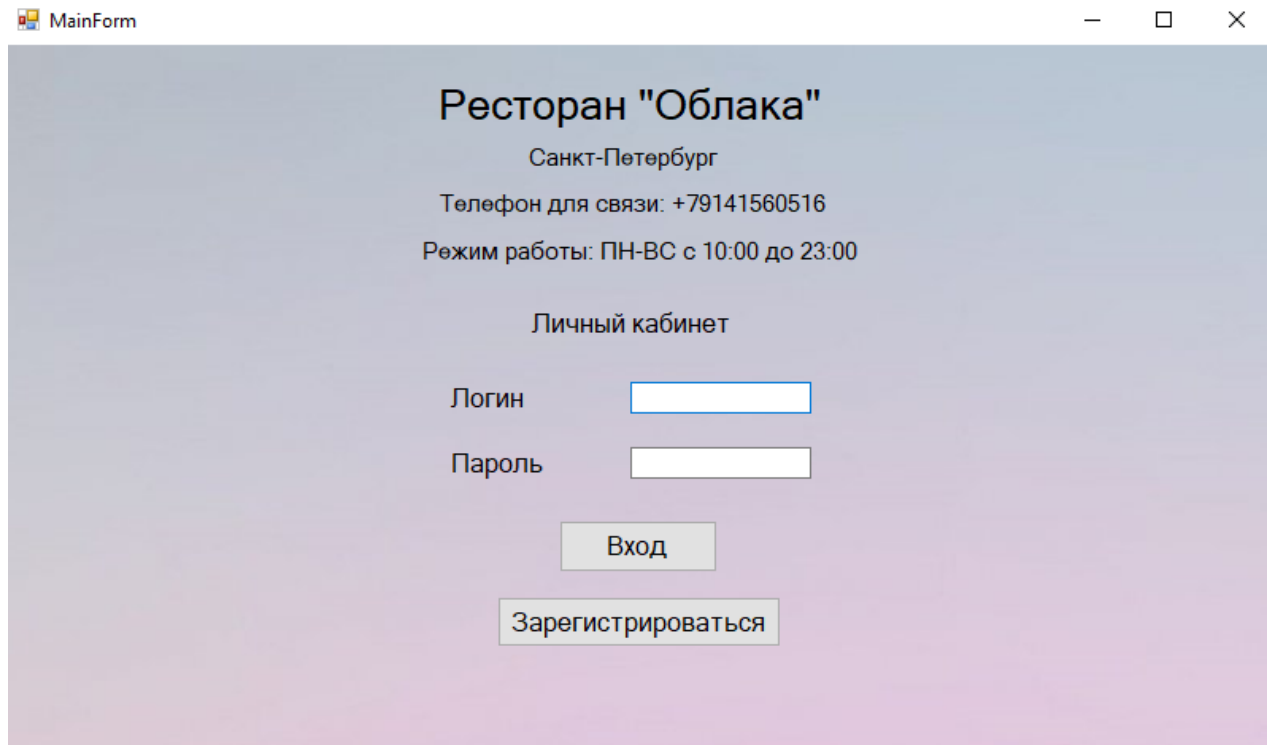
EXEC Dish_that_not_ordered

/* ТРИГГЕР на чек в заказе */
CREATE TRIGGER orders_insert_trigger
ON Orders
AFTER INSERT, UPDATE
AS
UPDATE Orders SET Bill = Bill + Bill* 0.3
WHERE IdOrder = (SELECT IdOrder FROM INSERTED)
GO
```

Клиентское приложение

Для упрощения взаимодействия с сервером на языке C# было создано клиентское приложение, реализующее описанный функционал.

Форма входа



The screenshot shows a Windows application window titled 'MainForm'. The background is a light blue-to-purple gradient. The text is centered and reads:

Ресторан "Облака"
Санкт-Петербург
Телефон для связи: +79141560516
Режим работы: ПН-ВС с 10:00 до 23:00

Below this information is the section 'Личный кабинет' (Personal Account). It contains two input fields: 'Логин' (Login) and 'Пароль' (Password). Below the password field are two buttons: 'Вход' (Login) and 'Зарегистрироваться' (Register).

Регистрация:

Registration

Назад

Имя

Фамилия

Номер телефона

Логин

Пароль

Зарегистрироваться

Меню Администратора

Меню администратора

Назад

Добавить
сотрудника

Уволить сотрудника

Статистика

Посмотреть
отзывы

Добавление Сотрудника:

AddEmployee

Назад

Имя

Фамилия

Должность

Зарплата

Номер телефона

Дата приема на работу

Логин

Пароль

Добавить

Удаление Сотрудника:

DeleteEmployee

Назад

Id Сотрудника

Имя

Фамилия

Уволить

Показать всех сотрудников

	IdEmployee	Name	Surname	Post
▶	1	Никита	Казимиров	Администрат
	2	Полина	Фомина	Повар-стаже
	4	Игорь	Вребицкий	Повар
	7	Андрей	Гальченко	Закупщик
*				

Просмотр статистики:

Statistic

Назад

Показать

Общая выручка
Лучший сотрудник
Самое популярное блюдо

Просмотр отзывов посетителей:

ShowReview

Назад

	IdVisitor	Rating	Comment	IdReviews
▶	1	5	Шикарный р	1
	3	5	Вкусно и быс	2
*				

Меню Повара

Меню повара

Назад

Добавить блюдо в меню

Убрать блюдо из меню

Изменить состав блюда

Посмотреть статус заказов

Добавление блюда в меню:

AddDish

Наименование блюда

Себестоимость

Цена

Описание блюда

Добавить

Показать меню

Назад

	IdDish	Name	CostPrice	Price	Description
▶	1	Борщ	250	350	Вкусный нав
	2	Солянка	380	350	Традиционн
	3	Жареная кар	150	200	Жареная кар
	4	Салат Цезар	290	350	Лекгий и дот
	5	Овощное раг	150	240	Блюдо из ту
*					

Удаление блюда из меню:

DeleteDish

Назад

Наименование блюда

Удалить

Показать меню

	IdDish	Name	CostPrice	Price	Description
▶	1	Борщ	250	350	Вкусный нав
	2	Солянка	380	350	Традиционн
	3	Жареная кар	150	200	Жареная кар
	4	Салат Цезар	290	350	Легкий и дот
	5	Овощное раг	150	240	Блюдо из ту
*					

Изменение состава блюд:

AddIngredientOfDish

Наименование блюда

Название ингредиента

Количество

Добавить ингредиент

Удалить ингредиент

Назад

Показать меню

Показать ингредиент

	IdDish	Name	CostPrice	Price
▶	1	Борщ	250	350
	2	Солянка	380	350
	3	Жареная кар	150	200
	4	Салат Цезар	290	350
	5	Овощное раг	150	240
*				

	IdIngredient	Name	Quantity	CostPrice	Units
▶	1	Картошка	15	15	кг
	2	Курица	15	120	кг
	3	Морковка	20	12	кг
	4	Лист салата	10	9	кг
	5	Свекла	15	17	кг
	7	Яйцо	10	8	шт
*					

Меню Посетителя

Menu_visitor

Назад

Сделать заказ

Добавить блюдо к заказу

Посмотреть свои заказы

Оставить отзыв

Сделать заказ:

AddOrder

Назад

Для начала выберите повара и номер стола

Выберите повара Полина

Номер стола 6

Выбрать Показать поваров

Выберите желаемые блюда

Наименование блюда

Количество

Цена

Ваш номер заказа

Добавить

Показать меню

AddOrder

Назад

Для начала выберите повара и номер стола

Выберите повара

Полина

Выбрать

Показать поваров

Номер стола

6

Выберите желаемые блюда

Наименование блюда

Борщ

Добавить

Количество

2

Показать меню

Цена

350

Ваш номер заказа

39

Добавить блюда к заказу:

NewDish

Назад

Ваш номер заказа

37

Добавить

Наименование блюда

Салат Цезарь

Показать меню

Количество

1

	IdDish	Name	Price	Description
►	1	Борщ	350	Вкусный нав
	2	Солянка	350	Традиционн
	3	Жареная кар	200	Жареная кар
	4	Салат Цезар	350	Легкий и дот
	5	Овощное раг	240	Блюдо из ту
*				

Посмотреть свои заказы:

ShowOrders

Назад

	IdOrder	Name	Quantity	Price	Status
▶	29	Солянка	2	350	завершён
	29	Жареная к	1	200	завершён
	31	Борщ	1	350	завершён
	31	Овощное	3	240	завершён
	31	Салат Цез	1	350	завершён
	31	Овощное	2	240	завершён
	36	Солянка	2	350	завершён
	36	Борщ	1	350	завершён
*					

Оставить отзыв:

Review

Назад

Оцените ресторан и добавь комментарий

Оценка

Комментарий

1
2
3
4
5

ресторан по Цена/Качество

Добавить
отзыв

Меню Закупщика

MenuPurchaser

Назад

Добавить новый ингредиент

Удалить ингредиент

Заказать ингредиент

Информация

Добавить новый ингредиент:

AddNewIngr

Назад

Наименование ингредиента Капуста

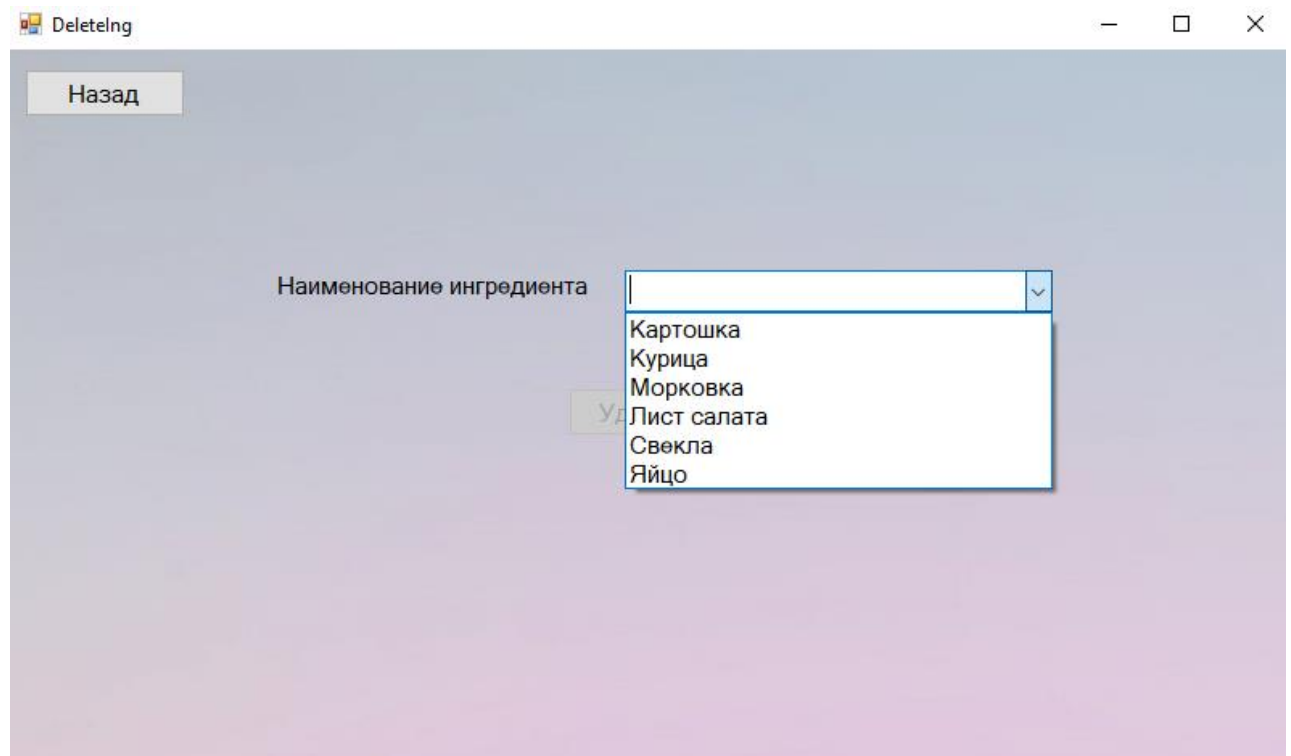
Количество 5

Себестоимость 13

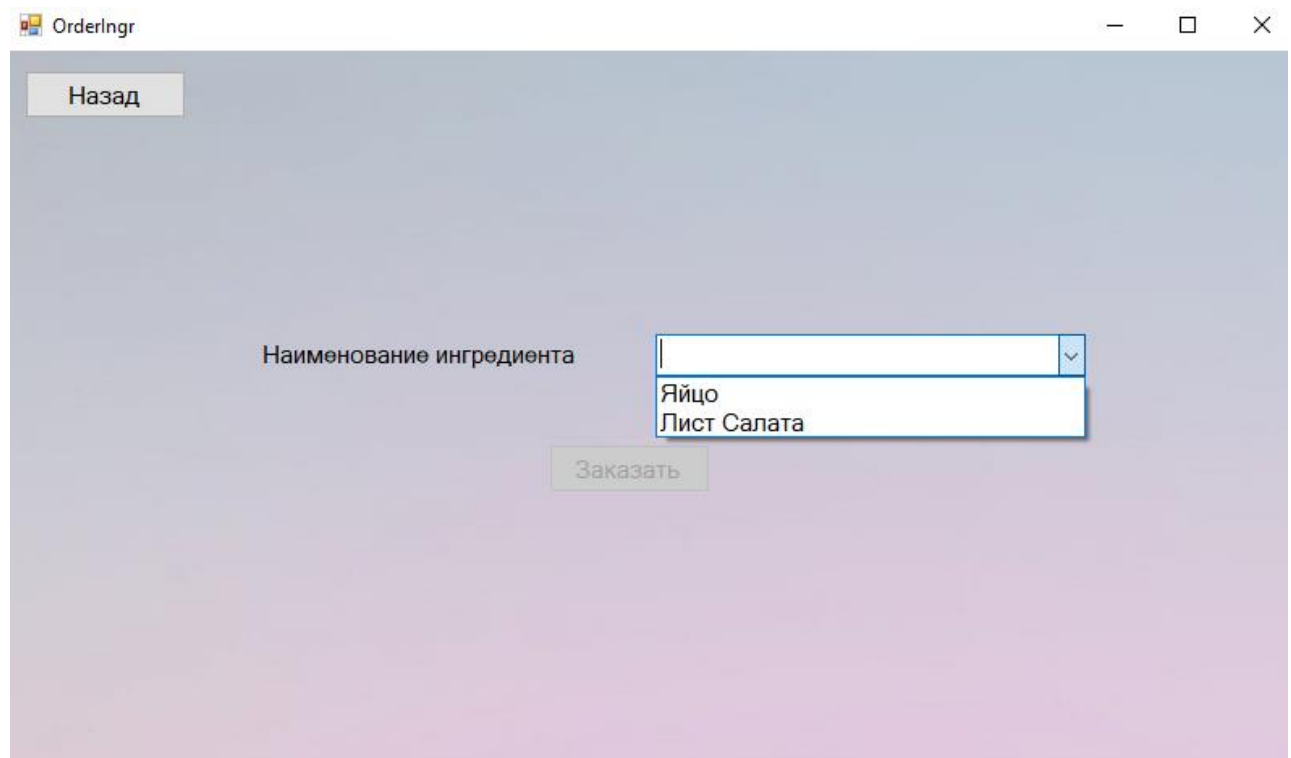
Ед. измерения кг

Добавить

Удалить ингредиент:



Заказать ингредиент:



Показать информация:

Information

Назад

Поставщики
Склад
Ингредиенты
Прайс-лист

Information

Назад

Склад

	IdIngredient	Name	Quantity	Units
▶	7	Яйцо	0	шт
	1	Картошка	100	кг
	2	Курица	50	кг
	3	Морковка	50	кг
	4	Лист Салата	0	кг
	5	Свекла	50	кг
*				

Вывод

В ходе курсовой работы была реализована система для работы ресторана. Для её реализации была использована MSSQLSMS. Клиентская часть выполнена с помощью платформы Visual Studio 2019 на языке C#.

Список литературы

1. Руководство по MS SQL Server <https://metanit.com/sql/sqlserver/>
2. Справочник по Transact-SQL <https://docs.microsoft.com/ru-ru/sql/tsql/language-reference?view=sql-server-ver15>