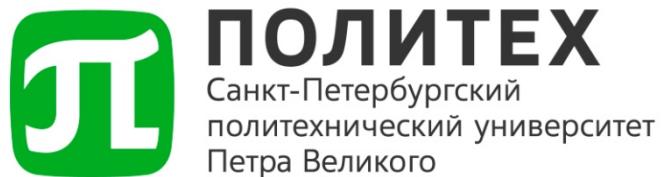


ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»
Высшая школа программной инженерии



КУРСОВАЯ РАБОТА

“Разработка многопользовательской системы управления организацией”
по дисциплине «Базы Данных»

Студент
3530202/90202

Потапова А. М.

Преподаватель

5

Гасанова А. А.

Санкт-Петербург
2021 г.

ЗАДАНИЕ
НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА (КУРСОВОЙ РАБОТЫ)
студенту
группы 3530202/90202 (номер группы) Потаповой Алине Михайловне (фамилия, имя, отчество)

1. Тема проекта (работы) Разработка многопользовательской
системы управления организацией.

Объект автоматизации – ресторан. Задание №9

2. Срок сдачи студентом законченного проекта (работы) 25.12.2021

3. Исходные данные к проекту (работе)

Описание предметной области

4. Содержание пояснительной записи (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 используемых источников

Примерный объем пояснительной
записки 15-20 страниц машинописного
текста

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

6. Консультанты _____

7. Дата получения задания: «11» сентября 2021 г.

Руководитель Гасанова И.А.
(подпись) (инициалы, фамилия)

Задание принял к исполнению Потапова А.М.
Алина Потапова М
(подпись) (инициалы, фамилия)

24.09.2021
(дата)

Оглавление

Анализ предметной области	4
Схема и описание базы данных.....	7
Запросы	11
Описание программы	21
Интерфейсы системы	24
Заключение.....	37
Список литературы.....	38
Приложение. Код создания таблиц.....	39

Анализ предметной области

Группы пользователей разрабатываемой информационной системы (ИС)

№ пп	Наименование пользователя
1	Клиент
2	Работник кухни
2	Администратор ресторана

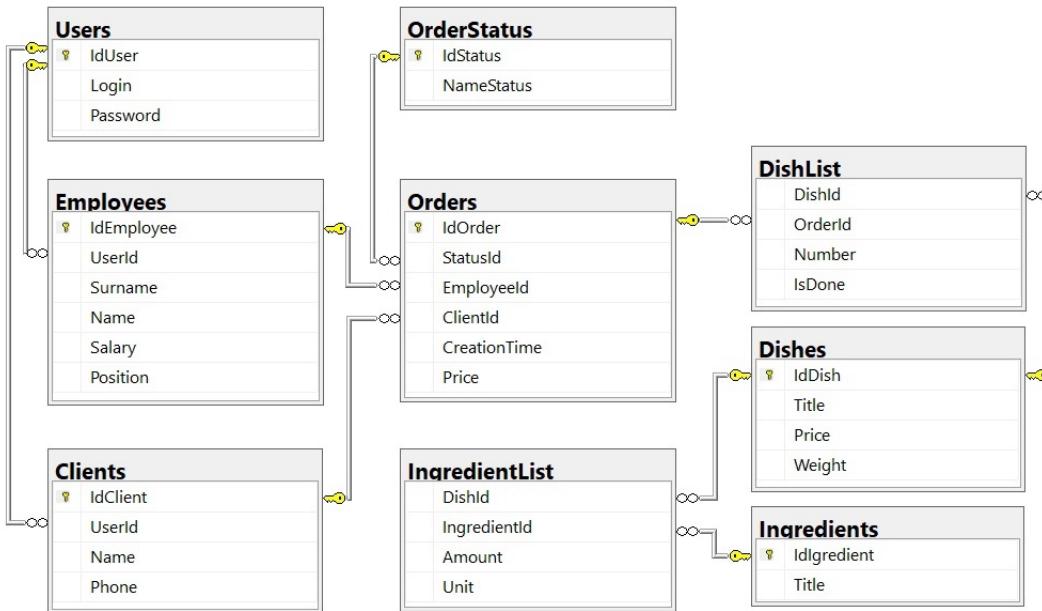
Функции групп пользователей

№ пп	Выполняемая функция	Входные данные	Выходные данные	Функции, которые должны быть реализованы в ИС
Покупатель				
1	Регистрация	Логин, пароль, имя, телефон	Таблица Users, Clients	Добавление нового клиента
2	Просмотр блюд в меню		Таблица Dishes	Вывод всех блюд в меню
3	Просмотр самых популярных блюд в меню		Таблица Dishes	Вывод списка блюд по популярности
4	Просмотр информации о блюде	ID блюда	Таблицы Dishes, IngredientList	Вывод информации о выбранном блюде
5	Создание заказа	ID выбранных (-ого) блюд(-а)	Таблицы Orders, DishList, Clients	Добавление нового заказа
6	Просмотр информации о заказе	ID заказа	Таблицы Orders, OrderStatus, DishList, Dishes	Вывод информации о заказе
7	Просмотр истории заказов		Таблицы Orders, DishList	Вывод всех созданных покупателем заказов
8	Отмена созданного заказа	ID заказа	Таблицы Orders, OrderStatus	Отмена созданного заказа
9	Редактирование персональной информации	Логин, пароль, имя, телефон	Таблицы Users, Clients	Изменение информации о клиенте

<i>Работник кухни</i>				
1	Просмотр всех блюд в меню		Таблица Dishes	Вывод всех блюд в меню
2	Просмотр заказов с определенным статусом	ID заказа, ID статуса заказа	Таблицы Orders , OrderStatus	Вывод списка заказов с выбранным статусом
3	Просмотр всех неприготовленных блюд		Таблицы Orders , DishList , Dishes	Вывод списка всех неприготовленных блюд (со значением флага <i>is_done</i> равным 0)
4	Просмотр всех приготовленных блюд		Таблицы Orders , DishList , Dishes	Вывод списка всех приготовленных блюд (со значением флага <i>is_done</i> равным 1)
6	Просмотр информации о заказе	ID заказа	Таблицы Orders , OrderStatus , DishList	Вывод информации о заказе
5	Просмотр состава (блюд) заказа	ID заказа	Таблицы Orders , DishList , Dishes , IngredientList , Ingredients	Вывод всех блюд в заказе
7	Отметка заказа принятым к исполнению	ID заказа, ID статуса	Таблицы Orders , OrderStatus	Изменение статуса заказа на «принят»
8	Отметка блюда приготовленным	ID блюда	Таблицы Dishes	Изменение значения флага <i>is_done</i> в 1
9	Отметка заказа приготовленным	ID заказа, ID статуса	Таблицы Orders , OrderStatus	Изменение статуса заказа на «приготовлен»
10	Отметка заказа переданным покупателю	ID заказа, ID статуса	Таблицы Orders , OrderStatus	Изменение статуса заказа на «выдан»
11	Редактирование персональной информации	Логин, пароль	Таблицы Users , Employees	Изменение логина и пароля сотрудника

<i>Администратор ресторана</i>				
1	Просмотр всех блюд в меню		Таблица Dishes	Вывод всех блюд в меню
2	Добавление нового блюда в меню	Название, стоимость, состав (список ингредиентов) блюда	Таблицы Dishes , IngredientList , Ingredients	Добавление нового блюда в меню
3	Добавление нового ингредиента	Название ингредиента	Таблица Ingredients	Добавление нового ингредиента
4	Редактирование информации об ингредиенте	ID ингредиента, название, количество	Таблица Ingredients	Изменение информации об ингредиенте
5	Редактирование информации о блюде	ID блюда, название, цена, состав (список ингредиентов)	Таблицы Dishes , IngredientList , Ingredients	Изменение информации о блюде
6	Редактирование информации о сотруднике	ID сотрудника	Таблица Employees	Изменение информации о сотруднике
7	Просмотр списка покупателей		Таблица Clients	Вывод всех покупателей
8	Просмотр всех сотрудников		Таблица Employees	Вывод всех сотрудников
9	Регистрация нового сотрудника	Имя, фамилия, логин, зарплата, должность, пароль	Таблица Users , Employees	Добавление нового сотрудника
10	Назначение сотрудника на исполнение заказа	ID заказа, ID сотрудника	Таблицы Employees , Users , Orders	Добавление связи между заказом и сотрудником
11	Удаление сотрудника	ID, имя и фамилия сотрудника	Таблица Users , Employees	Удаление сотрудника
12	Удаление ингредиента	ID ингредиента	Таблица Ingredients	Удаление ингредиента
13	Удаление блюда из меню	ID блюда	Таблица Dishes	Удаление блюда из меню
14	Редактирование персональной информации	Логин, пароль	Таблицы Users , Employees	Изменение логина и пароля сотрудника

Схема и описание базы данных



Синтаксис описания связей:

Номер. **Таблица 1** (поле в таблице 1) → **Таблица 2** (поле в таблице 2)
описание связи

1. **Orders** (StatusId) → **OrderStatus** (IdStatus)

Позволяет узнать статус заказа

2. **Orders** (EmployeeId) → **Employees** (IdEmployee)

Позволяет узнать информацию о сотруднике, который был назначен на исполнение заказа

3. **Orders** (ClientId) → **Clients** (IdClient)

Позволяет узнать информацию о клиенте, который совершил заказ

4. **DishList** (OrderId) → **Orders** (IdOrder)

Позволяет узнать информацию о списке блюд в заказе

5. **DishList** (DishId) → **Dishes** (IdDish)

Позволяет узнать информацию о блюде в заказе

6. **IngredientList** (DishId) → **Dishes** (IdDish)

Позволяет узнать информацию об ингредиентах блюда

7. **IngredientList** (IngredientId) → **Ingredients** (IdIngredient)

Позволяет узнать информацию о конкретном ингредиенте

Описание таблиц

1. Users – для хранения пользователей

	Column Name	Data Type	Allow Nulls
?	IdUser	int	<input type="checkbox"/>
	Login	varchar(64)	<input type="checkbox"/>
	Password	varchar(64)	<input type="checkbox"/>

IdUser – ID пользователя (PK, авто-инкремент)

Login – логин (NOT NULL, UNIQUE)

Password – пароль (NOT NULL)

2. Clients – для хранения клиентов

	Column Name	Data Type	Allow Nulls
?	IdClient	int	<input type="checkbox"/>
	UserId	int	<input type="checkbox"/>
	Name	varchar(64)	<input type="checkbox"/>
	Phone	varchar(16)	<input type="checkbox"/>

IdClient – ID клиента (PK, авто-инкремент)

UserId – ID пользователя (FK на Users)

Name – имя (NOT NULL)

Phone – номер телефона (NOT NULL, UNIQUE)

3. Employees – для хранения сотрудников

	Column Name	Data Type	Allow Nulls
?	IdEmployee	int	<input type="checkbox"/>
	UserId	int	<input type="checkbox"/>
	Surname	varchar(64)	<input type="checkbox"/>
	Name	varchar(64)	<input type="checkbox"/>
	Salary	money	<input type="checkbox"/>
	Position	varchar(64)	<input type="checkbox"/>

IdEmployee – ID сотрудника (PK, авто-инкремент)

UserId – ID пользователя (FK на Users)

Surname – фамилия (NOT NULL)

Name – имя (NOT NULL)

Salary – заработная плата (NOT NULL)

Position – должность (NOT NULL)

4. Orders – для хранения заказов

DESKTOP-HNHRFMH....ant - dbo.Orders			
	Column Name	Data Type	Allow Nulls
▼	IdOrder	int	<input type="checkbox"/>
	StatusId	int	<input type="checkbox"/>
	EmployeeId	int	<input checked="" type="checkbox"/>
	ClientId	int	<input type="checkbox"/>
	CreationTime	datetime	<input type="checkbox"/>
	Price	money	<input checked="" type="checkbox"/>

IdOrder – ID заказа (PK, авто-инкремент)

StatusId – ID статуса (FK на OrderStatus)

EmployeeId – ID сотрудника (FK на Employees)

ClientId – ID клиента (FK на Clients)

CreationTime – время создания заказа (DEFAULT ‘GETDATE()’)

Price – сумма заказа (NOT NULL)

5. OrderStatus – для хранения статусов заказа

DESKTOP-HNHRFMH....dbo.OrderStatus			
	Column Name	Data Type	Allow Nulls
▼	IdStatus	int	<input type="checkbox"/>
	NameStatus	varchar(32)	<input type="checkbox"/>

IdStatus – ID статуса (PK, авто-инкремент)

NameStatus – название статуса (NOT NULL, UNIQUE)

6. IngredientList – для хранения списка ингредиентов блюда

DESKTOP-HNHRFMH....bo.IngredientList			
	Column Name	Data Type	Allow Nulls
	DishId	int	<input type="checkbox"/>
	IngredientId	int	<input type="checkbox"/>
	Amount	int	<input type="checkbox"/>
	Unit	varchar(20)	<input type="checkbox"/>

DishId – ID блюда (FK на Dishes)

IngredientId – ID блюда (FK на Ingredients)

Amount – количество ингредиента (NOT NULL)

Unit – единица измерения (NOT NULL)

7. Ingredients – для хранения ингредиентов

DESKTOP-HNHRFMH....dbo.Ingredients			
	Column Name	Data Type	Allow Nulls
▼	IdIngredient	int	<input type="checkbox"/>
	Title	varchar(64)	<input type="checkbox"/>

IdIngredient – ID вагона (PK, авто-инкремент)

Title – название ингредиента (NOT NULL, UNIQUE)

8. Dishes – для хранения блюд

DESKTOP-HNHRFMH.kurs - dbo.Dishes			
	Column Name	Data Type	Allow Nulls
▼	IdDish	int	<input type="checkbox"/>
	Title	varchar(64)	<input type="checkbox"/>
	Price	money	<input type="checkbox"/>
	Weight	int	<input type="checkbox"/>

IdDish – ID блюда (PK, авто-инкремент)

Title – название блюда (NOT NULL, UNIQUE)

Price – стоимость блюда (NOT NULL)

9. DishList – для хранения списка блюд в заказе

DESKTOP-HNHRFMH.kurs - dbo.DishList			
	Column Name	Data Type	Allow Nulls
	DishId	int	<input type="checkbox"/>
	OrderId	int	<input type="checkbox"/>
	Number	int	<input type="checkbox"/>
	IsDone	bit	<input type="checkbox"/>

DishId – ID блюда (FK на Dishes)

OrderId – ID заказа (FK на Orders)

Number – количество блюд в заказе (NOT NULL)

IsDone – готово ли блюдо (DEFAULT 0)

Запросы

Простые запросы

1. Вывод всех блюд в меню

```
SELECT * FROM Dishes
```

2. Вывод всех сотрудников

```
SELECT * FROM Employees
```

3. Вывод всех ингредиентов

```
SELECT * FROM Ingredients
```

4. Изменение значения флага, отвечающего за готовность блюда

```
DECLARE
```

```
@tempOrderId int,  
@tempDoneDishId int,  
@tempIsDone bit
```

```
UPDATE DishList SET DishList.IsDone = @tempIsDone  
WHERE DishList.OrderId = @tempOrderId AND DishList.DishId =  
@tempDoneDishId
```

5. Изменение информации об ингредиенте

```
DECLARE
```

```
@tempIngredientId int,  
@tempIngredientTitle varchar(64)
```

```
UPDATE Ingredients SET Title = @tempIngredientTitle  
WHERE IdIngredient = @tempIngredientId
```

6. Изменение информации о блюде

```
DECLARE
```

```
@tempDishId int,  
@tempDishTitle varchar(64),  
@tempDishPrice money,  
@tempDishWeight int
```

```
UPDATE Dishes  
SET Title = @tempDishTitle, Price = @tempDishPrice, Weight =  
@tempDishWeight  
WHERE IdDish = @tempDishId
```

7. Изменение информации об ингредиенте блюда

```
DECLARE  
    @tempDishId int,  
    @amount int,  
    @unit varchar(20)  
  
UPDATE IngredientList  
SET Amount = @amount, Unit = @unit  
WHERE DishId = @tempDishId
```

8. Изменение информации о сотруднике

```
DECLARE  
    @tempEmoloyeeId int,  
    @tempEmoloyeeSurname varchar(64),  
    @tempEmoloyeeName varchar(64),  
    @tempEmoloyeeSalary money,  
    @tempEmoloyeePosition varchar(64)  
  
UPDATE Employees  
SET Surname = @tempEmoloyeeSurname, Name = @tempEmoloyeeName, Salary  
= @tempEmoloyeeSalary, Position = @tempEmoloyeePosition  
WHERE IdEmployee = @tempEmoloyeeId
```

9. Удаление сотрудника

```
DECLARE  
    @surname varchar(64),  
    @name varchar(64)  
DELETE FROM Employees  
WHERE Employees.Surname = @surname AND Employees.Name = @name
```

10. Удаление ингредиента

```
DECLARE  
    @delIngredientTitle varchar(64)  
DELETE FROM Ingredients  
WHERE Ingredients.Title = @delIngredientTitle
```

11. Удаление блюда из меню

```
DECLARE  
    @delDishTitle varchar(64)  
DELETE FROM Dishes  
WHERE Dishes.Title = @delDishTitle
```

Сложные запросы

1. Вывод всех блюд в заказе

— Клиент

```
SELECT Dishes.Title, DishList.Number, Dishes.Price * DishList.Number AS 'Total  
Price' FROM Dishes  
JOIN DishList ON DishList.DishId = Dishes.IdDish  
JOIN Orders ON Orders.IdOrder = DishList.OrderId AND Orders.IdOrder =  
@order_id  
JOIN OrderStatus ON Orders.StatusId = OrderStatus.IdStatus
```

— Сотрудник

```
SELECT Dishlist.DishId, Dishes.Title, DishList.Number, DishList.IsDone  
FROM Dishes  
JOIN DishList ON DishList.DishId = Dishes.IdDish  
JOIN Orders ON Orders.IdOrder = DishList.OrderId AND Orders.IdOrder =  
@order_id  
JOIN OrderStatus ON Orders.StatusId = OrderStatus.IdStatus
```

2. Вывод информации о блюде

— Клиент

```
SELECT Ingredients.Title FROM Ingredients  
JOIN IngredientList ON IngredientList.IngredientId = Ingredients.IdIngredient  
JOIN Dishes ON Dishes.IdDish = IngredientList.DishId AND Dishes.IdDish =  
@dish_id
```

— Сотрудник

```
SELECT Ingredients.Title, IngredientList.Amount, IngredientList.Unit FROM  
Ingredients  
JOIN IngredientList ON IngredientList.IngredientId = Ingredients.IdIngredient  
JOIN Dishes ON Dishes.IdDish = IngredientList.DishId AND Dishes.IdDish =  
@dish_id
```

3. Вывод информации о заказе

— Клиент

```
SELECT OrderStatus.NameStatus, FORMAT(Orders.CreationTime, 'dd.MM.yyyy  
HH:mm') AS 'Creation Time', Orders.Price FROM Orders  
JOIN OrderStatus ON Orders.StatusId = OrderStatus.IdStatus AND Orders.IdOrder =  
@order_id  
JOIN Employees ON Orders.EmployeeId = Employees.IdEmployee  
JOIN Clients ON Orders.ClientId = Clients.IdClient
```

— Сотрудник

```
SELECT OrderStatus.NameStatus, Employees.Surname AS  
'Employee', Clients.Name AS  
'Client', Clients.Phone, FORMAT(Orders.CreationTime, 'dd.MM.yyyy HH:mm') AS  
'Creation Time', Orders.Price FROM Orders  
JOIN OrderStatus ON Orders.StatusId = OrderStatus.IdStatus AND Orders.IdOrder  
= @order_id  
JOIN Employees ON Orders.EmployeeId = Employees.IdEmployee  
JOIN Clients ON Orders.ClientId = Clients.IdClient
```

4. Вывод всех созданных пользователем заказов

```
SELECT Orders.IdOrder, OrderStatus.NameStatus, FORMAT(Orders.CreationTime,  
'dd.MM.yyyy HH:mm') AS 'Creation Time', Orders.Price  
FROM Orders  
JOIN OrderStatus ON OrderStatus.IdStatus = Orders.StatusId  
JOIN Clients ON Clients.IdClient = Orders.ClientId AND Clients.IdClient =  
@client_id
```

5. Вывод списка заказов с определенным статусом

```
ALTER PROCEDURE OrderListByStatus  
@status varchar(64)  
AS  
BEGIN  
SELECT Orders.IdOrder, FORMAT(Orders.CreationTime, 'dd.MM.yyyy HH:mm')  
AS 'Creation Time', Clients.Name AS 'Client', Clients.Phone, Orders.Price  
FROM Orders  
JOIN OrderStatus ON Orders.StatusId = OrderStatus.IdStatus AND NameStatus =  
@status  
JOIN Employees ON Orders.EmployeeId = Employees.IdEmployee  
JOIN Clients ON Orders.ClientId = Clients.IdClient  
ORDER BY Orders.CreationTime DESC  
END;
```

6. Вывод всех неприготовленных блюд

```
SELECT Dishes.Title, DishList.Number FROM Dishes  
JOIN DishList ON DishList.IsDone = 0 AND DishList.DishId = Dishes.IdDish
```

7. Вывод всех приготовленных блюд

```
SELECT Dishes.Title, DishList.Number FROM Dishes  
JOIN DishList ON DishList.IsDone = 1 AND DishList.DishId = Dishes.IdDish
```

8. Добавление связи между сотрудником и заказом

```
DECLARE  
@orderId int,
```

```
@employeeName varchar(64),  
@employeeSurname varchar(64)
```

```
UPDATE Orders SET EmployeeId = (SELECT Employees.IdEmployee  
FROM Employees WHERE Name = @employeeName AND Surname =  
@employeeSurname)  
WHERE IdOrder = @orderId
```

9. Изменение значения статуса заказа

```
DECLARE  
@orderId int,  
@statusName varchar(64)
```

```
UPDATE Orders SET Orders.StatusId = (SELECT IdStatus FROM  
OrderStatus WHERE NameStatus = @statusName)  
WHERE Orders.IdOrder = @orderId
```

10. Вывод всех заказов

```
SELECT Orders.IdOrder, FORMAT(Orders.CreationTime, 'dd.MM.yyyy  
HH:mm') AS 'Creation Time', Clients.Name AS 'Client', Clients.Phone,  
Orders.Price, OrderStatus.NameStatus  
FROM Orders  
JOIN OrderStatus ON Orders.StatusId = OrderStatus.IdStatus  
JOIN Employees ON Orders.EmployeeId = Employees.IdEmployee  
JOIN Clients ON Orders.ClientId = Clients.IdClient  
ORDER BY Orders.CreationTime DESC
```

11. Вывод списка покупателей

```
CREATE VIEW Clients_List  
AS  
(SELECT Clients.Name, Clients.Phone, COUNT(Orders.IdOrder) AS OrderCount  
FROM Clients  
JOIN Orders ON Orders.ClientId = Clients.IdClient  
GROUP BY Clients.name, Clients.Phone)  
UNION ALL  
(SELECT Clients.Name, Clients.Phone, 0 AS OrderCount FROM Clients  
WHERE NOT EXISTS (SELECT * FROM Orders WHERE Orders.ClientId =  
Clients.IdClient))
```

Представления

1. Вывод всех блюд по убыванию их популярности.

```
ALTER VIEW TopDishes
AS
(SELECT TOP 100 Dishes.IdDish, Dishes.Title, Dishes.Weight, Dishes.Price FROM
Dishes JOIN DishList ON DishList.DishId = Dishes.IdDish GROUP BY
Dishes.IdDish, Dishes.Title, Dishes.Weight, Dishes.Price HAVING
SUM(DishList.Number) > 0
ORDER BY SUM(DishList.Number) DESC)
UNION ALL
(SELECT TOP 100 Dishes.IdDish, Dishes.Title, Dishes.Weight, Dishes.Price
FROM Dishes WHERE NOT EXISTS (SELECT * FROM DishList WHERE
DishList.DishId = Dishes.IdDish) ORDER BY Dishes.Title DESC)
```

2. Вывод всех клиентов по убыванию количества заказов.

```
ALTER VIEW TopClients
AS
(SELECT TOP 100 Clients.Name, Clients.Phone, COUNT(Orders.IdOrder) AS
OrderCount FROM Clients JOIN Orders ON Orders.ClientId = Clients.IdClient
GROUP BY Clients.name,
ORDER BY OrderCount DESC)
UNION ALL
(SELECT TOP 100 Clients.Name, Clients.Phone, 0 AS OrderCount FROM Clients
WHERE NOT EXISTS (SELECT * FROM Orders WHERE Orders.ClientId =
Clients.IdClient))
```

Хранимые процедуры

1. Регистрация сотрудника

```
CREATE PROCEDURE RegEmployee
@login varchar(64),
@password varchar(64),
@surname varchar(64),
@name varchar(64),
@salary money,
@position varchar(64)
AS
BEGIN TRY
BEGIN TRANSACTION
INSERT INTO Users(login, password)
VALUES (@login, @password)
INSERT INTO Employees(UserId, Surname, Name, Salary, Position)
VALUES (SCOPE_IDENTITY(), @surname, @name, @salary, @position)
```

```
    SELECT IdEmployee FROM Employees WHERE IdEmployee =
SCOPE_IDENTITY()
COMMIT TRANSACTION
END TRY
BEGIN CATCH
ROLLBACK TRANSACTION;
END CATCH
```

2. Регистрация клиента

```
CREATE PROCEDURE RegClient
    @login varchar(64),
    @password varchar(64),
    @name varchar(64),
    @phone varchar(15)
AS
BEGIN TRY
BEGIN TRANSACTION
    INSERT INTO Users(login, password)
    VALUES (@login, @password)
    INSERT INTO Clients(UserId, Name, Phone)
    VALUES (SCOPE_IDENTITY(), @name, @phone)
    SELECT IdClient FROM Clients WHERE IdClient = SCOPE_IDENTITY()
COMMIT TRANSACTION
END TRY
BEGIN CATCH
ROLLBACK TRANSACTION;
END CATCH
```

3. Добавление ингредиента

```
CREATE PROCEDURE AddIngredient
    @title varchar(64)
AS
BEGIN TRY
BEGIN TRANSACTION
    INSERT INTO Ingredients(Title)
    VALUES (@title)
    SELECT IdIngredient FROM Ingredients WHERE IdIngredient =
SCOPE_IDENTITY()
COMMIT TRANSACTION
END TRY
BEGIN CATCH
ROLLBACK TRANSACTION;
END CATCH
```

4. Добавление блюда с ингредиентами

```
CREATE PROCEDURE AddDish
    @title varchar(64),
    @price money,
    @weight int
AS
BEGIN TRY
BEGIN TRANSACTION
    INSERT INTO Dishes>Title, Weight, Price)
    VALUES (@title, @weight, @price)
    SELECT IdDish FROM Dishes WHERE IdDish = SCOPE_IDENTITY()
COMMIT TRANSACTION
END TRY
BEGIN CATCH
ROLLBACK TRANSACTION;
END CATCH

CREATE PROCEDURE AddDishIngredient
    @dishTitle varchar(64),
    @ingredientTitle varchar(64),
    @amount int,
    @unit varchar(20)
AS
BEGIN
    INSERT INTO IngredientList(DishId, IngredientId, Amount, Unit)
    VALUES ((SELECT IdDish FROM Dishes WHERE Title = @dishTitle),
    (SELECT IdIngredient FROM Ingredients WHERE Title = @ingredientTitle) ,
    @amount, @unit)
END;
```

5. Добавление заказа

```
CREATE PROCEDURE AddOrder
    @clientId int
AS
BEGIN
    INSERT INTO Orders(ClientId, StatusId, Price)
    VALUES (@clientId, (SELECT IdStatus FROM OrderStatus
    WHERE NameStatus = 'Обрабатывается'),
    (SELECT SUM(Dishes.Price * DishList.Number) FROM Dishes, DishList
    WHERE DishList.DishId = Dishes.IdDish AND DishList.OrderId IS NULL))
    UPDATE DishList SET DishList.OrderId = SCOPE_IDENTITY()
    WHERE DishList.OrderId IS NULL
END;
```

```
CREATE PROCEDURE AddDishInDishList
    @dishTitle varchar(64),
    @number int
```

```
AS
BEGIN
    INSERT INTO DishList(DishId, Number, IsDone)
    VALUES ((SELECT IdDish FROM Dishes WHERE Title =
    @dishTitle), @number, '0')
END;
```

6. Вход в систему

```
CREATE PROCEDURE log_in
    @login varchar(64),
    @password varchar(64)
AS
BEGIN
    (SELECT IdClient AS Id, 0 AS Role FROM Clients
    JOIN Users ON Clients.UserId = Users.IdUser WHERE Users.Login = @login
    AND Users.Password = @password)
    UNION ALL
    (SELECT IdEmployee AS Id, 1 AS Role FROM Employees
    JOIN Users ON Employees.UserId = Users.IdUser WHERE Users.Login =
    @login AND Users.Password = @password AND Employees.Position != 'Администратор')
    UNION ALL
    (SELECT IdEmployee AS Id, 2 AS Role FROM Employees
    JOIN Users ON Employees.UserId = Users.IdUser WHERE Users.Login =
    @login AND Users.Password = @password AND Employees.Position =
    'Администратор')
END;
```

7. Изменение информации о клиенте

```
CREATE PROCEDURE changeClientInfo
    @clientId int,
    @login varchar(64),
    @password varchar(64),
    @name varchar(64),
    @phone varchar(16)
AS
BEGIN TRY
    BEGIN TRANSACTION
        UPDATE Users SET Login = @login, Password = @password
        WHERE IdUser = (SELECT UserId From Clients WHERE IdClient = @clientId)
        UPDATE Clients SET Name = @name, Phone = @phone
        WHERE IdClient = @clientId
    COMMIT TRANSACTION
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
END CATCH
```

8. Изменение информации о сотруднике

```
CREATE PROCEDURE changeEmployeeUser
    @employeeId int,
    @login varchar(64),
    @password varchar(64)
AS
BEGIN
    UPDATE Users SET Login = @login, Password = @password
    WHERE IdUser = (SELECT UserId From Employees WHERE IdEmployee =
@employeeId)
END;
```

Описание программы

1. Среда разработки: Visual Studio Code
2. Языки программирования: JavaScript, HTML, CSS
3. Движок: Node.js
4. Фреймворки:
 - Tediouss
 - Express
 - Body-parser
5. СУБД: Microsoft SQL Server

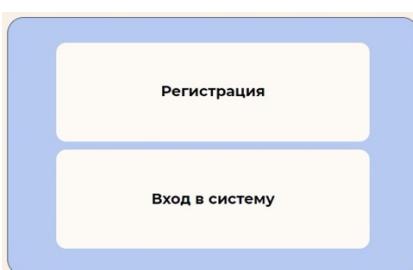
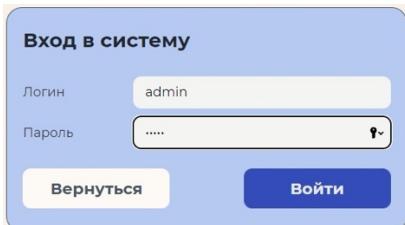
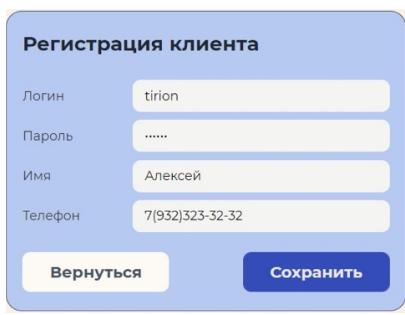
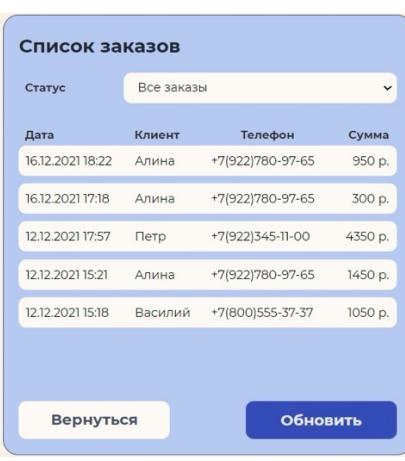
Предлагаю вам подробно ознакомиться со всеми применяемыми мною инструментами по ходу проектирования ПО.

Инструмент	Описание
Среда разработки	Visual Studio Code — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений.
Языки программирования	JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Обычно используется как встраиваемый язык для программного доступа к объектам приложений. HTML — стандартизованный язык разметки документов для просмотра веб-

	<p>страниц в браузере. В моем случае веб-браузер получает HTML документ от сервера по протоколам с локального диска, далее интерпретирует код в интерфейс, который будет отображаться на экране монитора.</p> <p>CSS — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.</p>
Движок	<p>Node.js — среда выполнения кода на JavaScript, которая построена на основе движка JavaScript Chrome V8, который позволяет транслировать вызовы на языке JavaScript в машинный код. Node.js прежде всего предназначен для создания серверных приложений на языке JavaScript</p>
Фреймворки	<p>Tedious — это фреймворк Node.js, который обеспечивает реализацию протокола TDS, который используется для взаимодействия с экземплярами Microsoft SQL Server.</p> <p>Express — это минималистичный и гибкий веб-фреймворк для приложений Node.js, предоставляющий обширный набор функций для мобильных и веб-приложений.</p> <p>Body-parser — это пакет, который позволяет разбирать тела запросов, хранящиеся в req.body, выступая в роли ПО промежуточного слоя для серверов, основанных на Express. Он обрабатывает</p>

	запросы до их попадания в соответствующие обработчики.
СУБД	Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft.

Интерфейсы системы

<p>Окно выбора входа или регистрации</p> 	<p>Вход в систему</p>  <p>Регистрация клиента</p> 
	<p>Главная страница администратора</p> 
<p>Раздел «Заказы»</p> 	<p>Страница заказов</p> 

Ввод заказов по статусу

Список заказов

Статус	Принят		
Дата	Клиент	Телефон	Сумма
12.12.2021 17:57	Петр	+7(922)345-11-00	4350 р.
12.12.2021 15:18	Василий	+7(800)555-37-37	1050 р.

Вернуться **Обновить**

Детали заказа

Детали заказа

Статус	Обрабатывается	
Информация		
Сумма заказа: 300 р.	Дата: 16.12.2021 17:18	
Клиент: Алина	Телефон: +7(922)780-97-65	
Сотрудник: не назначен		
Блюдо	Кол-во	Готово
Яичница	2	false

Вернуться **Назначить сотрудника**

Назначение сотрудника на исполнение заказа

Выбор сотрудника

Сотрудник	Семенов Максим
-----------	----------------

Вернуться **Назначить**

Сотрудник назначен на исполнения заказа

Детали заказа

Статус	Обрабатывается	
Информация		
Сумма заказа: 300 р.	Дата: 16.12.2021 17:18	
Клиент: Алина	Телефон: +7(922)780-97-65	
Сотрудник: Семенов		
Блюдо	Кол-во	Готово
Яичница	2	false

Вернуться

Раздел «Список клиентов»

Список клиентов

Выход списка клиентов по количеству заказов

Список клиентов

Имя	Телефон	Заказы
Алина	+7(922)780-97-65	3
Петр	+7(922)345-11-00	1
Василий	+7(800)555-37-37	1
Павел	+7(922)53-69-89	0

Вернуться

Раздел «Список блюд»

Список блюд

Выход списка блюд в меню

Список блюд

Название	Вес	Цена
Омлет	140 гр.	100 р.
Паста Карбонара	250 гр.	390 р.
Рамен	400 гр.	290 р.
Тосты	120 гр.	120 р.
Фруктовый салат	150 гр.	150 р.
Чизкейк	100 гр.	180 р.
Шоколадный брауни	150 гр.	250 р.

Вернуться

Обновить

Поиск блюда по названию

Список блюд

Название	Вес	Цена
Паста Карбонара	250 гр.	390 р.

Вернуться

Обновить

Выход состава блюда

Состав блюда

Блюдо	
Паста Карбонара	
Ингредиент	Кол-во
Масло	10 мл.
Чеснок	1 зуб.
Спагетти	100 гр.
Бекон	75 гр.
Сливки	75 мл.

Изменить **Добавить**

Вернуться **Удалить**

Изменение информации о блюде

Блюдо

Название	Паста Карбонара
Стоимость	390
Вес	250

Вернуться **Сохранить**

Добавление ингредиента блюда

Добавление ингредиента

Название	Не выбрано
Количество	
Е.И.	

Вернуться **Добавить**

Добавление блюда

Добавление блюда

Название	
Стоимость	
Вес	

Вернуться **Добавить**

Список сотрудников

Список сотрудников

Фамилия	Имя	ЗП	Должность
Айдаров	Ренат	36000	Работник кухни
Гукова	Виктория	52000	Администратор
Семенов	Максим	11000	Работник кухни

[Вернуться](#)

[Удалить](#)

Раздел «Список сотрудников»

[Список сотрудников](#)

Редактирование информации о сотруднике

Информация о сотруднике

Фамилия	Айдаров
Имя	Ренат
Зарплата	36000
Должность	Работник кухни

[Вернуться](#)

[Сохранить](#)

Раздел «Список ингредиентов»

[Список ингредиентов](#)

Вывод списка ингредиентов

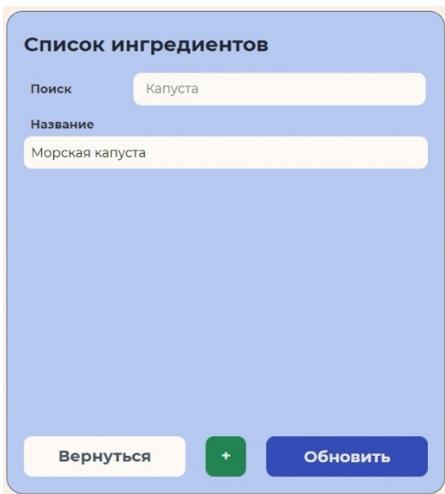
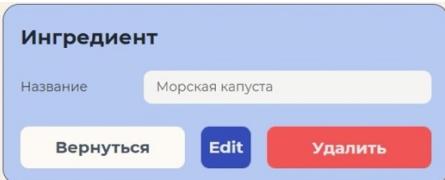
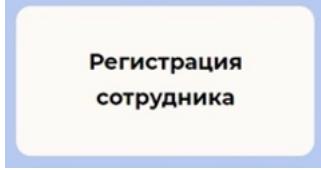
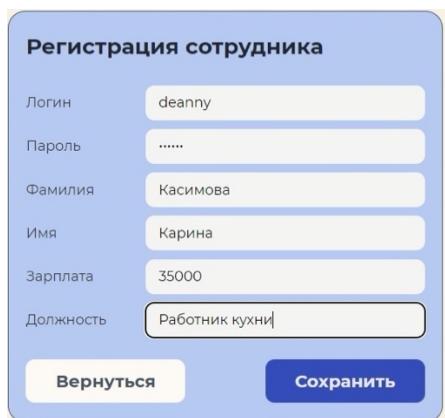
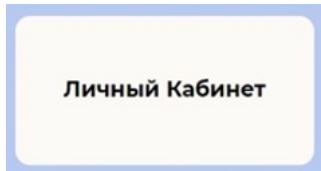
Список ингредиентов

Поиск	<input type="text"/>
Название	
Апельсин	
Бекон	
Виноград	
Груша	
Масло	
Морская капуста	
Огурец	

[Вернуться](#)



[Обновить](#)

	<p><i>Поиск ингредиента по названию</i></p>  <p>Список ингредиентов</p> <p>Поиск: Капуста</p> <p>Название: Морская капуста</p> <p>Вернуться + Обновить</p>												
	<p><i>Редактирование информации об ингредиенте</i></p>  <p>Ингредиент</p> <p>Название: Морская капуста</p> <p>Вернуться Edit Удалить</p>												
<p><i>Раздел «Регистрация сотрудника»</i></p>  <p>Регистрация сотрудника</p>	<p><i>Регистрация нового сотрудника</i></p>  <p>Регистрация сотрудника</p> <table border="1"> <tbody> <tr> <td>Логин</td> <td>deanny</td> </tr> <tr> <td>Пароль</td> <td>.....</td> </tr> <tr> <td>Фамилия</td> <td>Касимова</td> </tr> <tr> <td>Имя</td> <td>Карина</td> </tr> <tr> <td>Зарплата</td> <td>35000</td> </tr> <tr> <td>Должность</td> <td>Работник кухни</td> </tr> </tbody> </table> <p>Вернуться Сохранить</p>	Логин	deanny	Пароль	Фамилия	Касимова	Имя	Карина	Зарплата	35000	Должность	Работник кухни
Логин	deanny												
Пароль												
Фамилия	Касимова												
Имя	Карина												
Зарплата	35000												
Должность	Работник кухни												
<p><i>Раздел «Регистрация сотрудника»</i></p>  <p>Личный Кабинет</p>	<p><i>Личный кабинет (редактирование логина и пароля)</i></p>  <p>Личный Кабинет</p> <table border="1"> <tbody> <tr> <td>Логин</td> <td>admin</td> </tr> <tr> <td>Пароль</td> <td>Новый пароль</td> </tr> <tr> <td>Фамилия</td> <td>Гукова</td> </tr> <tr> <td>Имя</td> <td>Виктория</td> </tr> <tr> <td>Зарплата</td> <td>52000</td> </tr> <tr> <td>Должность</td> <td>Администратор</td> </tr> </tbody> </table> <p>Вернуться Сохранить</p>	Логин	admin	Пароль	Новый пароль	Фамилия	Гукова	Имя	Виктория	Зарплата	52000	Должность	Администратор
Логин	admin												
Пароль	Новый пароль												
Фамилия	Гукова												
Имя	Виктория												
Зарплата	52000												
Должность	Администратор												

Главная страница работника кухни

Вывод заказов, на которые назначен/был назначен сотрудник

Список заказов

Дата	Клиент	Телефон	Сумма
16.12.2021 17:18	Алина	+7(922)780-97-65	300 р.
12.12.2021 15:18	Василий	+7(800)555-37-37	1050 р.

Детали заказа

Детали заказа

Статус	Обрабатывается	
Информация		
Сумма заказа: 300 р.	Дата: 16.12.2021 17:18	
Клиент: Алина	Телефон: +7(922)780-97-65	
Блюдо Кол-во Готово		
Яичница	2	false

Раздел «Заказы»

Изменение значения статуса заказа

Детали заказа

Статус	Принят	
Информация		
Сумма заказа: 300 р.	Дата: 16.12.2021 17:18	
Клиент: Алина	Телефон: +7(922)780-97-65	
Блюдо Кол-во Готово		
Яичница	2	false

Выход списка блюд

Список блюд

Поиск

Название

Омлет
Паста Карбонара
Рамен
Тосты
Фруктовый салат
Чизкейк

Вернуться **Обновить**

Поиск блюда по названию

Список блюд

Поиск Паста

Название

Паста Карбонара

Вернуться **Обновить**

Выход состава блюда

Состав блюда

Ингредиент	Кол-во
Масло	10 мл.
Чеснок	1 зуб.
Спагетти	100 гр.
Бекон	75 гр.
Сливки	75 мл.
Пармезан	20 гр.
Соль	3 гр.

Вернуться

Раздел «Список блюд»

Список блюд

Выход списка блюд в заказах

Состав блюд в заказах

Статус	Все блюда
Название	Кол-во
Фруктовый салат	3
Яичница	4
Яичница	2

Вернуться **Обновить**

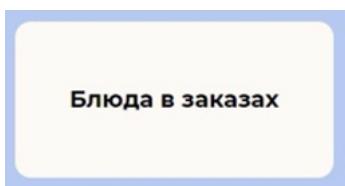
Выход блюд в заказах с определенным статусом

Состав блюд в заказах

Статус	Не готовы
Название	Кол-во
Яичница	4
Яичница	2

Вернуться **Обновить**

Раздел «Блюда в заказах»

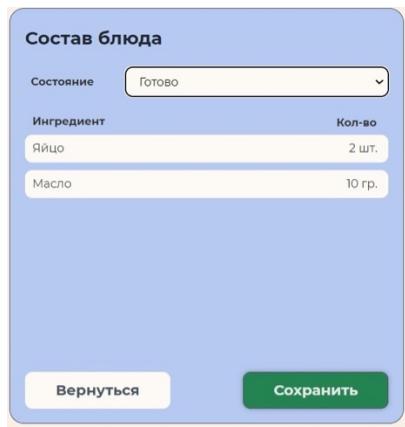
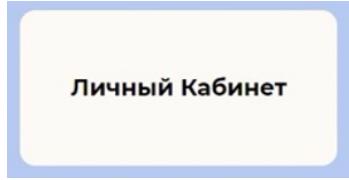
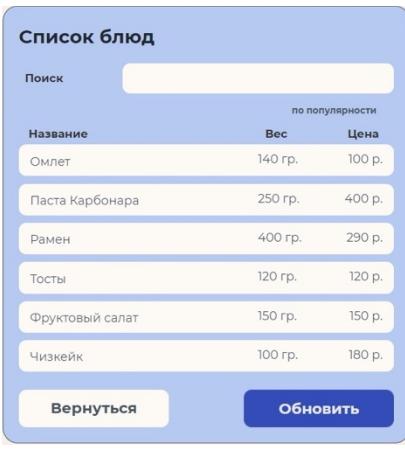


Выход состава блюда

Состав блюда

Состояние	Не готово
Ингредиент	Кол-во
Яйцо	2 шт.
Масло	10 гр.

Вернуться **Сохранить**

	<p><i>Изменение значения флага, отвечающего за готовность блюда</i></p> 																					
<p><i>Раздел «Личный кабинет»</i></p> 	<p><i>Личный кабинет (редактирование логина и пароля)</i></p> 																					
	<p><i>Главная страница работника кухни</i></p> 																					
<p><i>Раздел «Меню»</i></p> 	<p><i>Выход списка всех блюд в меню</i></p>  <table border="1"> <thead> <tr> <th>Название</th> <th>Вес</th> <th>Цена</th> </tr> </thead> <tbody> <tr> <td>Омлет</td> <td>140 гр.</td> <td>100 р.</td> </tr> <tr> <td>Паста Карбонара</td> <td>250 гр.</td> <td>400 р.</td> </tr> <tr> <td>Рамен</td> <td>400 гр.</td> <td>290 р.</td> </tr> <tr> <td>Тосты</td> <td>120 гр.</td> <td>120 р.</td> </tr> <tr> <td>Фруктовый салат</td> <td>150 гр.</td> <td>150 р.</td> </tr> <tr> <td>Чизкейк</td> <td>100 гр.</td> <td>180 р.</td> </tr> </tbody> </table>	Название	Вес	Цена	Омлет	140 гр.	100 р.	Паста Карбонара	250 гр.	400 р.	Рамен	400 гр.	290 р.	Тосты	120 гр.	120 р.	Фруктовый салат	150 гр.	150 р.	Чизкейк	100 гр.	180 р.
Название	Вес	Цена																				
Омлет	140 гр.	100 р.																				
Паста Карбонара	250 гр.	400 р.																				
Рамен	400 гр.	290 р.																				
Тосты	120 гр.	120 р.																				
Фруктовый салат	150 гр.	150 р.																				
Чизкейк	100 гр.	180 р.																				

Вывод списка всех блюд в меню по популярности

Список блюд		
Поиск		
Название	Вес	Цена
Яичница	150 гр.	150 р.
Фруктовый салат	150 гр.	150 р.
Паста Карбонара	250 гр.	400 р.
Шоколадный брауни	150 гр.	250 р.
Чизкейк	100 гр.	180 р.
Тосты	120 гр.	120 р.

Вывод информации о выбранном блюде

Информация о блюде	
Цена	Вес
250 р.	400 гр.
Ингредиент	
Масло	
Чеснок	
Спагетти	
Бекон	
Сливки	
Пармезан	

[Вернуться](#)

Создание заказа

Создание заказа

Информация о заказе

Сумма заказа: 0 р.
Дата: 17.12.2021 16:29

Состав заказа

Блюдо	Кол-во	Стоимость

[Отменить](#) [Создать](#)

Раздел «Создание заказа»

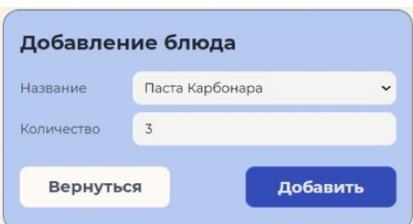
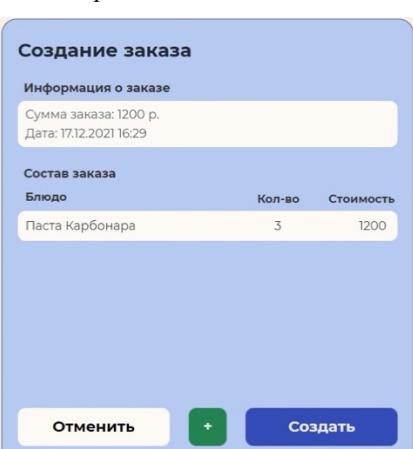
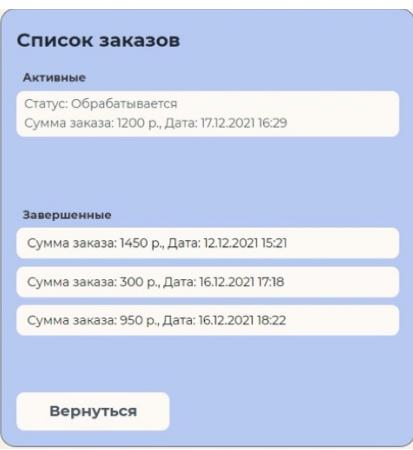
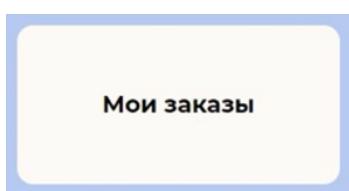
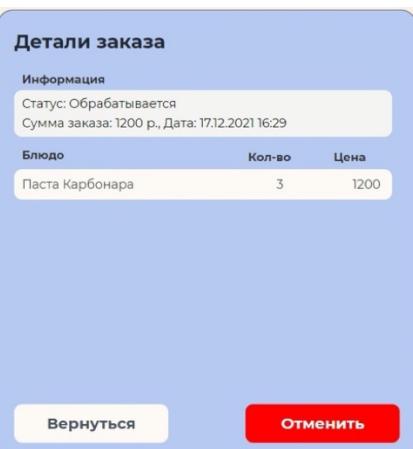
Создание заказа

**Вы уверены, что хотите отменить
запрос?**

Вы уверены, что хотите отменить заказ?

Hot

Да

	<p><i>Добавление блюда в заказ</i></p>  <p>Добавление блюда</p> <p>Название: Паста Карбонара Количество: 3</p> <p>Вернуться Добавить</p>						
	<p><i>Завершение создания заказа</i></p>  <p>Создание заказа</p> <p>Информация о заказе Сумма заказа: 1200 р., Дата: 17.12.2021 16:29</p> <p>Состав заказа</p> <table border="1"> <thead> <tr> <th>Блюдо</th> <th>Кол-во</th> <th>Стоимость</th> </tr> </thead> <tbody> <tr> <td>Паста Карбонара</td> <td>3</td> <td>1200</td> </tr> </tbody> </table> <p>Отменить + Создать</p>	Блюдо	Кол-во	Стоимость	Паста Карбонара	3	1200
Блюдо	Кол-во	Стоимость					
Паста Карбонара	3	1200					
	<p><i>Выход списка активных и завершенных заказов</i></p>  <p>Список заказов</p> <p>Активные Статус: Обрабатывается Сумма заказа: 1200 р., Дата: 17.12.2021 16:29</p> <p>Завершенные Сумма заказа: 1450 р., Дата: 12.12.2021 15:21 Сумма заказа: 300 р., Дата: 16.12.2021 17:18 Сумма заказа: 950 р., Дата: 16.12.2021 18:22</p> <p>Вернуться</p>						
<p><i>Раздел «Мои заказы»</i></p>  <p>Мои заказы</p>	<p><i>Выход деталей заказа (отмена заказа)</i></p>  <p>Детали заказа</p> <p>Информация Статус: Обрабатывается Сумма заказа: 1200 р., Дата: 17.12.2021 16:29</p> <p>Блюдо Кол-во Цена</p> <table border="1"> <thead> <tr> <th>Блюдо</th> <th>Кол-во</th> <th>Цена</th> </tr> </thead> <tbody> <tr> <td>Паста Карбонара</td> <td>3</td> <td>1200</td> </tr> </tbody> </table> <p>Вернуться Отменить</p>	Блюдо	Кол-во	Цена	Паста Карбонара	3	1200
Блюдо	Кол-во	Цена					
Паста Карбонара	3	1200					

Раздел «Личный кабинет»

Личный Кабинет

Личный кабинет (редактирование персональных данных)

Личный Кабинет

Логин ramapp

Пароль Новый пароль

Имя Алина

Телефон 7(922)780-97-65

Вернуться

Сохранить

Заключение

В результате выполнения курсовой работы мною была реализована многопользовательская система для работы ресторана. По ходу проектирования приложения мною были получены:

- новые теоретические знания,
- практический опыт,
- опыт работы с MSSQL и Node.js

Также хочется упомянуть о сложностях, с которыми я столкнулась по ходу проектирования. Основная трудность состояла в том, что я ранее не имела дела с Node.js, поэтому в начале разработки системы пришлось потратить немало времени на изучение функционала фреймворков этой платформы. Но мне удалось преодолеть все сложности и создать систему с удобным интерфейсом, которая отвечает всем моим требованиям.

Список литературы

1. Документация MS SQL [Электронный ресурс], URL:
<https://docs.microsoft.com/ru-ru/sql/?view=sql-server-ver15>
2. Документация языка программирования HTML [Электронный ресурс], URL: <https://devdocs.io/html/>
3. Подключение к SQL с помощью Node.js [Электронный ресурс], URL:
<https://docs.microsoft.com/ru-ru/sql/connect/node-js/step-3-proof-of-concept-connecting-to-sql-using-node-js?view=sql-server-ver15>
4. Документация Node.js [Электронный ресурс], URL: <https://nodejs.org/en/>
5. Дейт, К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильяме", 2005.

Приложение. Код создания таблиц

1. **Users** – для хранения пользователей

```
CREATE TABLE Users
(IdUser int IDENTITY(1,1) PRIMARY KEY,
Login varchar(64) NOT NULL,
Password varchar(64) NOT NULL,
CONSTRAINT AK_Login UNIQUE(Login))
GO
```

2. **Clients** – для хранения клиентов

```
CREATE TABLE Clients
(IdClient int IDENTITY(1,1) PRIMARY KEY,
UserId int NOT NULL FOREIGN KEY REFERENCES Users
ON DELETE CASCADE ON UPDATE CASCADE,
Name varchar(64) NOT NULL,
Phone varchar(16) NOT NULL,
CONSTRAINT AK_Phone UNIQUE(Phone))
GO
```

3. **Employees** – для хранения сотрудников

```
CREATE TABLE Employees
(IdEmployee int IDENTITY(1,1) PRIMARY KEY,
UserId int NOT NULL FOREIGN KEY REFERENCES Users
ON DELETE CASCADE ON UPDATE CASCADE,
Surname varchar(64) NOT NULL,
Name varchar(64) NOT NULL,
Salary money NOT NULL,
Position varchar(64) NOT NULL)
GO
```

4. **Dishes** – для хранения блюд

```
CREATE TABLE Dishes
(IdDish int IDENTITY(1,1) PRIMARY KEY,
Title varchar(64) NOT NULL,
CONSTRAINT AK_DishTitle UNIQUE>Title),
Price money NOT NULL,
Weight int NOT NULL)
GO
```

5. **OrderStatus** – для хранения статусов заказа

```
CREATE TABLE OrderStatus
(IdStatus int IDENTITY(1,1) PRIMARY KEY,
NameStatus varchar(32) NOT NULL,
```

```
CONSTRAINT AK_NameStatus UNIQUE(NameStatus))  
GO
```

6. **Ingredients** – для хранения ингредиентов

```
CREATE TABLE Ingredients  
(IdIngredient int IDENTITY(1,1) PRIMARY KEY,  
Title varchar(64) NOT NULL,  
CONSTRAINT AK_Title UNIQUE>Title))  
GO
```

7. **Orders** – для хранения заказов

```
CREATE TABLE Orders  
(IdOrder int IDENTITY(1,1) PRIMARY KEY,  
StatusId int NOT NULL FOREIGN KEY REFERENCES OrderStatus  
ON DELETE CASCADE ON UPDATE CASCADE,  
EmployeeId int FOREIGN KEY REFERENCES Employees  
ON DELETE SET NULL ON UPDATE CASCADE,  
ClientId int NOT NULL FOREIGN KEY REFERENCES Clients  
ON DELETE NO ACTION ON UPDATE NO ACTION,  
CreationTime datetime DEFAULT GETDATE(),  
Price money)  
GO
```

8. **DishList** – для хранения списка блюд в заказе

```
CREATE TABLE DishList  
(DishId int FOREIGN KEY REFERENCES Dishes  
ON DELETE SET NULL ON UPDATE CASCADE,  
OrderId int NOT NULL FOREIGN KEY REFERENCES Orders  
ON DELETE CASCADE ON UPDATE CASCADE,  
Number int NOT NULL,  
IsDone bit DEFAULT 0)  
GO
```

9. **IngredientList** – для хранения списка ингредиентов блюда

```
CREATE TABLE IngredientList  
(DishId int NOT NULL FOREIGN KEY REFERENCES Dishes  
ON DELETE CASCADE ON UPDATE CASCADE,  
IngredientId int FOREIGN KEY REFERENCES Ingredients  
ON DELETE SET NULL ON UPDATE CASCADE,  
Amount int NOT NULL,  
Unit varchar(20) NOT NULL)  
GO
```