|  |  |  |
| --- | --- | --- |
| **КУРСОВОЙ ПРОЕКТ** | | |
| «**Разработка многопользовательской автоматизированной системы управления**  **организацией. Объект автоматизации – ресторан**» | | |
| по дисциплине «Архитектура программных систем» | | |
| Выполнил | | |
| студент гр. 3530904/80102 |  | Кузнецов С. А. |
|  | | |
| Руководитель |  | Андреев И. А. |
| «12» декабря 2020г. | | |

Оглавление

[Задание на курсовой проект 3](#_Toc58673229)

[Введение 4](#_Toc58673230)

[Анализ предметной деятельности 5](#_Toc58673231)

[Проектирование схемы базы данных 7](#_Toc58673232)

[Скрипты создания БД 8](#_Toc58673233)

[Описание таблиц 9](#_Toc58673234)

[Сложные запросы с их описаниями 10](#_Toc58673235)

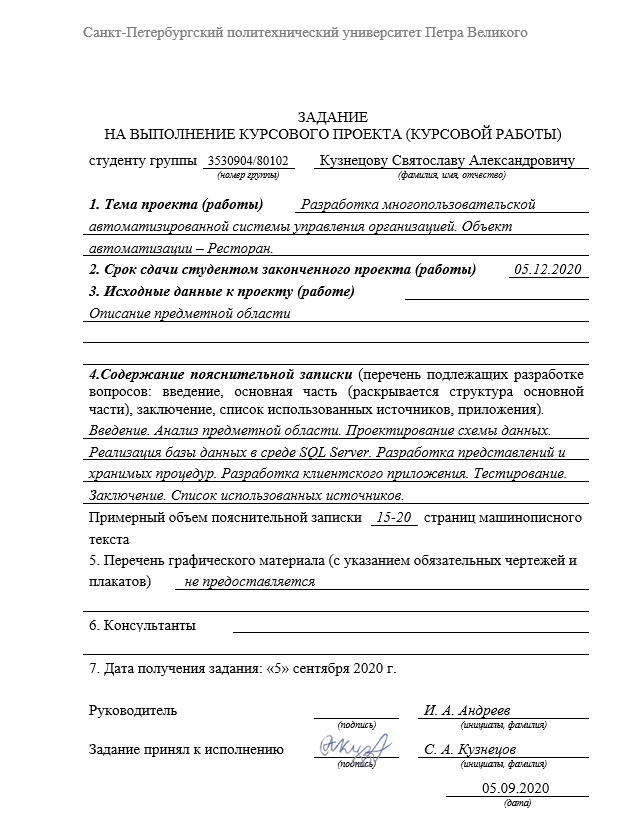
[Основные хранимые процедуры и триггеры 12](#_Toc58673236)

[Скриншоты приложения с описанием 16](#_Toc58673237)

[Заключение 30](#_Toc58673238)

[Список литературы 31](#_Toc58673239)

Задание на курсовой проект



Введение

Целью данного курсового проекта является разработка базы данных для ресторана, а также проектирование пользовательского интерфейса для удобства пользования системой различными группами пользователей.

Применение баз данных является одним из условий эффективной организации ресторана. Благодаря базам данных каждый повар и шеф-повар, а также клиенты, всегда имеют доступ к актуальной информации. Графический интерфейс упрощает доступ к информации и защищает базу данных от нежелательных изменений.

Анализ предметной деятельности

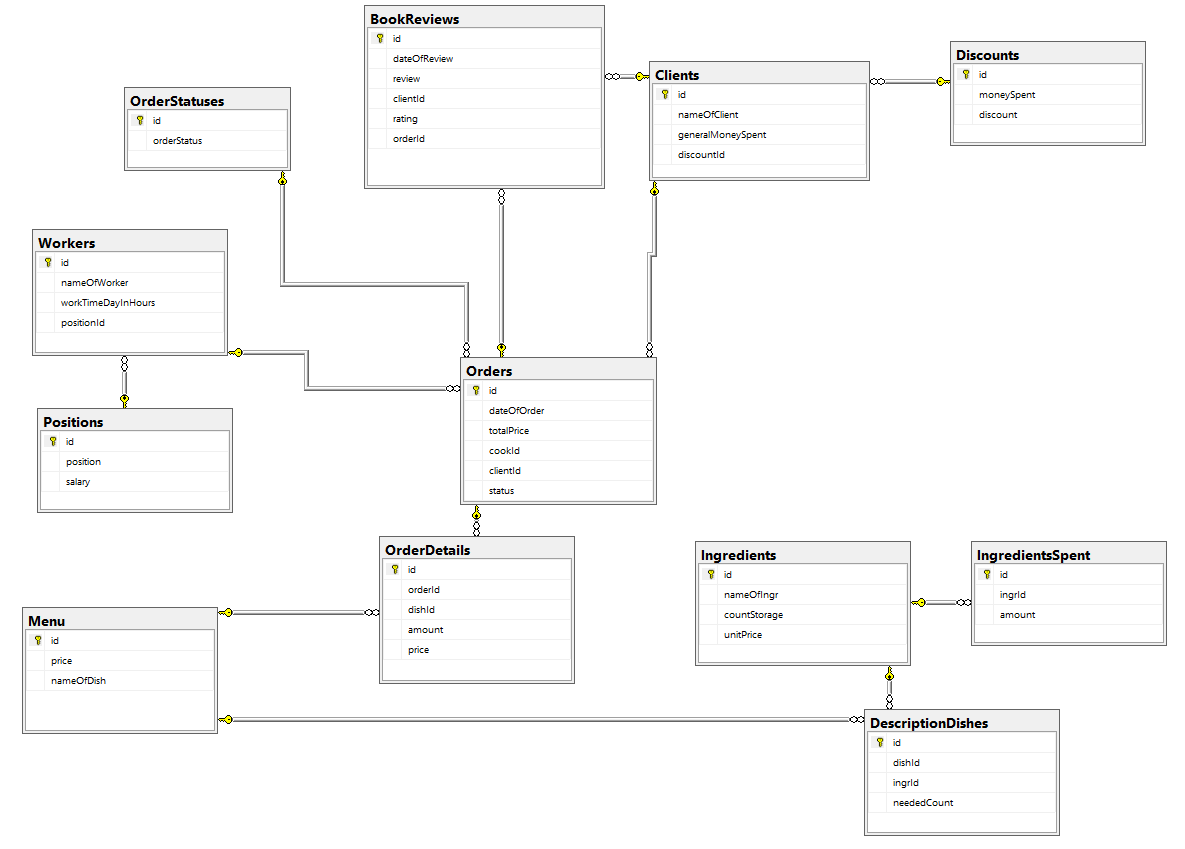
Группы пользователей разрабатываемой информационной системы (ИС)

|  |  |
| --- | --- |
| № пп | Наименование пользователя |
| 1 | Повар |
| 2 | Шеф-повар |
| 3 | Клиент |

Функции групп пользователей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № пп | Выполняемая функция | | Входные данные | Выходные данные | Функции, которые должны быть реализованы в ИС |
| ***Первый пользователь(группа)*** | | | | | |
| 1 | Посмотреть список заказов | | - | Список заказов | Страница с заказами, ожидающими выполнения |
| 2 | Приготовление блюда | | Id заказа | Информация о выполненном заказе | Изменение статуса заказа, количества потраченных ингредиентов и ингредиентов на складе |
| 3 | Просмотр требуемых ингредиентов для блюда | | Id блюда | Информация об ингредиентах блюда | Страница со списком блюд и ингредиентов |
| 4 | Добавление блюда в меню | | Название блюда, ингредиенты, цена | Информация о добавленном блюде | Страница с полями для создания блюда |
| 5 | Удаление блюда из меню | | Id блюда | Информация об удаленном блюде | Страница просмотра всех блюд |
| 6 | Закупки | | Id ингредиента, количество | Информация об успешном заказе ингредиента | Страница выбора ингредиента |
| ***Второй пользователь(группа)*** | | | | | |
| 1 | Просмотр списка работников | - | | Список работников с должностями и з\п | Страница просмотра списка работников |
| 2 | Добавление работника | ФИО работника, должность | | Информация о добавлении работника | Страница добавления работника |
| 3 | Удаление работника | Id работника | | Информация о удалении работника | Страница со списком работников |
| 4 | Подсчёт выручки | - | | Прибыль | Страница с затраченными, полученными деньгами и итоговой прибылью |
| 5 | Просмотр кол-ва проданных блюд в нужную дату | Дата | | Список блюд, кол-во | Страница выбора даты, страница со списком блюд |
| ***Третий пользователь(группа)*** | | | | | |
| 1 | Сделать заказ | Список блюд из меню, кол-во | | Информация о формировании заказа | Страница с выбором блюд |
| 2 | Просмотр меню | - | | Список блюд из меню | Страница со списком блюд |
| 3 | Просмотр книги отзывов и предложений | - | | Список отзывов | Страница со списком отзывов |
| 4 | Написать отзыв | Отзыв | | Информация об успешном написании отзыва | Страница с полями для отзыва |

Проектирование схемы базы данных



Скрипты создания БД

Скрипт для создания базы данных:



Описание таблиц

Clients – таблица с данными о клиенте  
Discounts – таблица с типами скидок скидках  
BookReviews – таблица с данными об отзывах  
Orders – таблица с данными заказов

OrderStatuses – таблица с типами статусов заказов

OrderDetails – таблица с данными о каждом заказе

Workers – таблица с данными о работниках

Positions – таблица с типами должностей

Menu – таблица с данными о блюдах в меню

DescriptionDishes – таблица с данными о каждом блюде

Ingredients – таблица с ингредиентами

IngredientsSpent – таблица с данными о потраченных ингредиентах

Сложные запросы с их описаниями

/\*1\*/

/\*вывести все блюда, проданные в определенную дату\*/

select Menu.nameOfDish from Menu

join OrderDetails on OrderDetails.dishId=Menu.id

join (select \* from Orders where Orders.dateOfOrder = '20111111') as ord on ord.id=OrderDetails.orderId

/\*2\*/

/\*вывести скидку клиентов, которые сделали 3 или больше заказов\*/

select Clients.nameOfClient, Clients.id, Clients.generalMoneySpent, Discounts.discount, COUNT(Orders.clientId) as 'Количество заказов' from Clients

join Discounts on Clients.discountId=Discounts.id

join Orders on Orders.clientId=Clients.id

group by Clients.nameOfClient, Clients.id, Clients.generalMoneySpent, Discounts.discount

having COUNT(Orders.clientId) >= 3

/\*3\*/

/\*вывести все блюда, содержащие картошку\*/

select Menu.nameOfDish from Menu

join DescriptionDishes on DescriptionDishes.dishId=Menu.id

join Ingredients on Ingredients.id=DescriptionDishes.ingrId

where Ingredients.nameOfIngr='Картошка'

/\*4\*/

/\*вычисление выручки\*/

select (SUM(Orders.totalPrice) -

(

select SUM(Ingredients.unitPrice \* IngredientsSpent.amount) from Ingredients

join IngredientsSpent on Ingredients.id=IngredientsSpent.ingrId

)

) from Orders

/\*5\*/

/\*блюдо, которое ни разу не было заказано\*/

select m.nameOfDish, m.id from

(

select Menu.nameOfDish, Menu.id from Menu

left join OrderDetails on OrderDetails.dishId=Menu.id

left join Orders on Orders.id=OrderDetails.orderId

group by Menu.nameOfDish, Menu.id

having COUNT(OrderDetails.dishId)=0

) as m

/\*6\*/

/\*вывести все названия блюд у которах есть отзыв с оценкой 5\*/

select Menu.nameOfDish from Menu

join OrderDetails on OrderDetails.dishId=Menu.id

join Orders on OrderDetails.orderId=Orders.id

join BookReviews on BookReviews.orderId=Orders.id

where BookReviews.rating=5

/\*7\*/

/\*вывести всех клиентов, которые не оставляли отзывы\*/

select Clients.nameOfClient from Clients

except (select cl.nameOfClient from Clients as cl

join BookReviews on BookReviews.clientId=cl.id)

/\*8\*/

/\*по клиенту вывести список его заказов\*/

select Orders.dateOfOrder, Menu.nameOfDish, OrderDetails.amount, Orders.totalPrice, OrderStatuses.orderStatus from Orders

join OrderDetails on OrderDetails.orderId=Orders.id

join Menu on OrderDetails.dishId=Menu.id

join OrderStatuses on Orders.status=OrderStatuses.id

where Orders.clientId=1

group by Orders.dateOfOrder, Menu.nameOfDish, OrderDetails.amount, Orders.totalPrice, OrderStatuses.orderStatus

/\*9\*/

/\*вывести ингредиенты блюда\*/

select Ingredients.nameOfIngr from Ingredients

join DescriptionDishes on DescriptionDishes.ingrId=Ingredients.id

join Menu on Menu.id=DescriptionDishes.dishId

where Menu.nameOfDish='Борщ'

/\*10\*/

/\*вывести все выполненные заказы клиента без отзыва\*/

select Menu.nameOfDish, OrderDetails.amount, Orders.totalPrice, Orders.id, Orders.dateOfOrder from Orders

join OrderDetails on OrderDetails.orderId=Orders.id

join Menu on OrderDetails.dishId=Menu.id

join OrderStatuses on Orders.status=OrderStatuses.id

where Orders.clientId=1 and Orders.status=2

except

(

select Menu.nameOfDish, OrderDetails.amount, Orders.totalPrice, Orders.id, Orders.dateOfOrder from Orders

join OrderDetails on OrderDetails.orderId=Orders.id

join Menu on OrderDetails.dishId=Menu.id

join OrderStatuses on Orders.status=OrderStatuses.id

join BookReviews on BookReviews.orderId=Orders.id

where Orders.clientId=1 and Orders.status=2

)

/\*11\*/

/\*вывести все блюда, у которых был отзыв\*/

select Menu.nameOfDish from Menu

where exists

(

select \* from OrderDetails

join Orders on OrderDetails.orderId=Orders.id

join BookReviews on BookReviews.orderId=Orders.id

where Menu.id=OrderDetails.dishId

)

/\*12\*/

/\*вывести все заказы, выполненные без 3 самых больших скидок\*/

select \* from Orders

join Clients on Clients.id=Orders.clientId

where Clients.discountId not in

(

select top 2 Discounts.discount from Discounts order by Discounts.discount desc)

Основные хранимые процедуры и триггеры

1. Триггер на добавление клиента в таблицу с клиентами после регистрации..

ALTER TRIGGER [dbo].[addUser]

ON [dbo].[restoran]

AFTER Insert

AS

BEGIN

Insert into Clients (nameOfClient, generalMoneySpent, discountId) values

(

(select us.name from (select top 1 \* from Users order by id desc) as us),

0,

1

)

END

1. Процедура для добавления нового блюда в меню

ALTER PROCEDURE [dbo].[add\_dish]

-- Add the parameters for the stored procedure here

@price float,

@name varchar(50)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

insert into Menu (price, nameOfDish) values (@price, @name)

END

1. Процедура для добавления ингредиентов блюда

ALTER PROCEDURE [dbo].[add\_dish\_description]

-- Add the parameters for the stored procedure here

@amount int,

@ingr varchar(50)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

insert into DescriptionDishes (dishId, ingrId, neededCount) values

(

(select top 1 Menu.id from Menu order by id desc),

(select Ingredients.id from Ingredients where Ingredients.nameOfIngr=@ingr),

@amount

)

END

1. Процедура для получения ингредиентов блюда

ALTER PROCEDURE [dbo].[get\_ingredients\_of\_dish]

-- Add the parameters for the stored procedure here

@nameOfDish nvarchar(50)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

select Ingredients.nameOfIngr, DescriptionDishes.neededCount from Ingredients

join DescriptionDishes on DescriptionDishes.ingrId=Ingredients.id

join Menu on Menu.id=DescriptionDishes.dishId

where Menu.nameOfDish=@nameOfDish

END

1. Процедура для получения заказов клиента

ALTER PROCEDURE [dbo].[get\_orders\_of\_user]

-- Add the parameters for the stored procedure here

@id int

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

select Orders.id, Orders.dateOfOrder, Menu.nameOfDish, OrderDetails.amount, Orders.totalPrice, OrderStatuses.orderStatus from Orders

join OrderDetails on OrderDetails.orderId=Orders.id

join Menu on OrderDetails.dishId=Menu.id

join OrderStatuses on Orders.status=OrderStatuses.id

where Orders.clientId=@id

group by Orders.id, Orders.dateOfOrder, Menu.nameOfDish, OrderDetails.amount, Orders.totalPrice, OrderStatuses.orderStatus

END

1. Процедура для получения общей выручки

ALTER PROCEDURE [dbo].[get\_profit]

-- Add the parameters for the stored procedure here

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

select SUM(Orders.totalPrice),

(

select SUM(Ingredients.unitPrice \* IngredientsSpent.amount) from Ingredients

join IngredientsSpent on Ingredients.id=IngredientsSpent.ingrId

),

(

select

(

SUM(Orders.totalPrice) -

(

select SUM(Ingredients.unitPrice \* IngredientsSpent.amount) from Ingredients

join IngredientsSpent on Ingredients.id=IngredientsSpent.ingrId

)

) from Orders

) from Orders

END

1. Процедура для получения заказов без отзыва

ALTER PROCEDURE [dbo].[get\_user\_orders\_without\_review]

-- Add the parameters for the stored procedure here

@id int

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

select Menu.nameOfDish, OrderDetails.amount, Orders.totalPrice, Orders.id, Orders.dateOfOrder from Orders

join OrderDetails on OrderDetails.orderId=Orders.id

join Menu on OrderDetails.dishId=Menu.id

join OrderStatuses on Orders.status=OrderStatuses.id

where Orders.clientId=@id and Orders.status=2

except

(

select Menu.nameOfDish, OrderDetails.amount, Orders.totalPrice, Orders.id, Orders.dateOfOrder from Orders

join OrderDetails on OrderDetails.orderId=Orders.id

join Menu on OrderDetails.dishId=Menu.id

join OrderStatuses on Orders.status=OrderStatuses.id

join BookReviews on BookReviews.orderId=Orders.id

where Orders.clientId=@id and Orders.status=2

)

END

1. Процедура для удаления блюда из меню

ALTER PROCEDURE [dbo].[remove\_dish]

-- Add the parameters for the stored procedure here

@name varchar(50)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

delete DescriptionDishes from

(select DescriptionDishes.id from DescriptionDishes

join Menu on Menu.id=DescriptionDishes.dishId

where Menu.nameOfDish=@name) as d

where d.id=DescriptionDishes.id

delete Menu from

(select \* from Menu

where Menu.nameOfDish=@name) as m

where m.id=Menu.id

END

1. Процедура для изменения количества ингредиентов на складе, обновления статуса заказа и изменения количества потраченных ингредиентов

ALTER PROCEDURE [dbo].[remove\_ingredients\_from\_storage]

-- Add the parameters for the stored procedure here

@amount int,

@name varchar(50),

@orderId int

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

update Ingredients set Ingredients.countStorage-=@amount where Ingredients.id=

(select Ingredients.id from Ingredients where Ingredients.nameOfIngr=@name)

update Orders set Orders.status=2 where Orders.id=@orderId

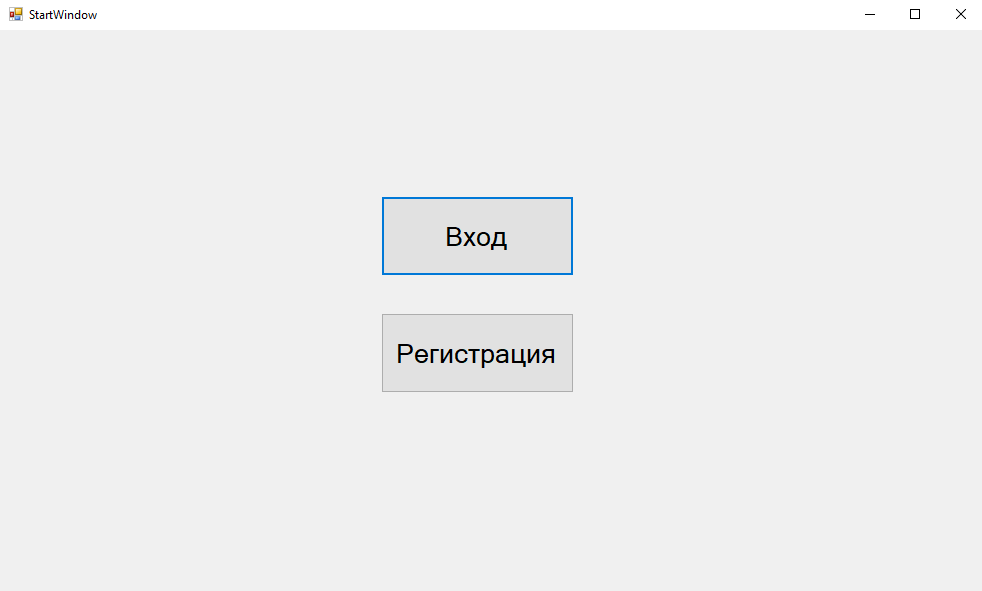
update IngredientsSpent set IngredientsSpent.amount+=@amount where IngredientsSpent.ingrId=

(select Ingredients.id from Ingredients where Ingredients.nameOfIngr=@name)

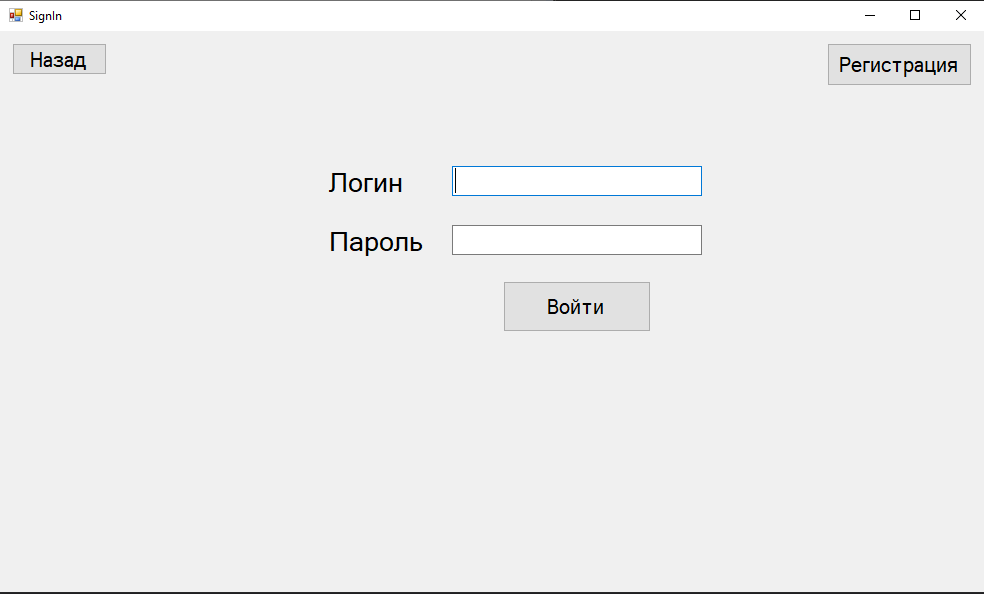
END

Скриншоты приложения с описанием

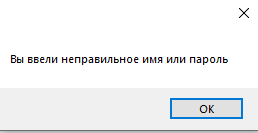
Стартовое окно



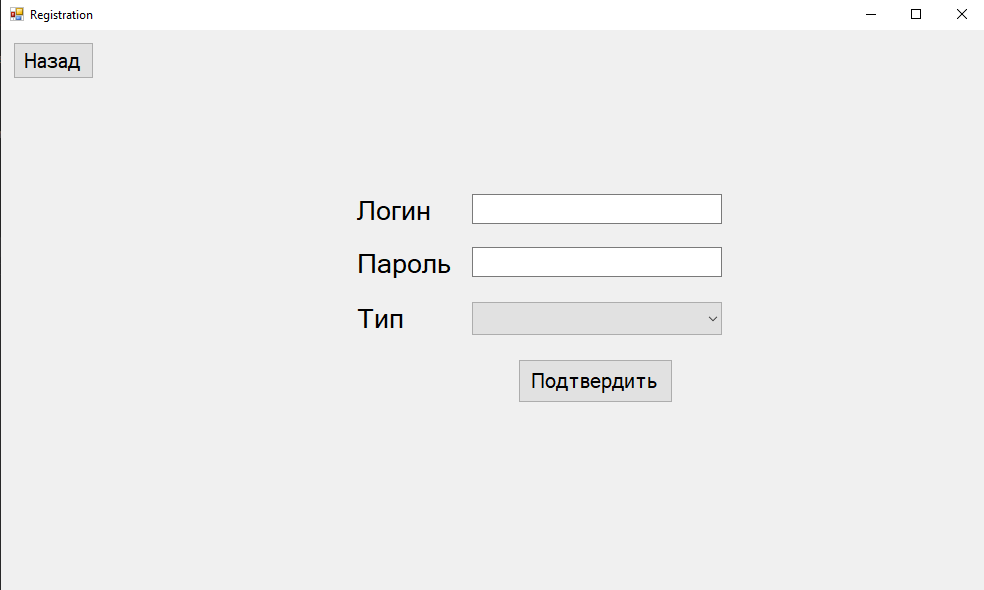
Окно авторизации



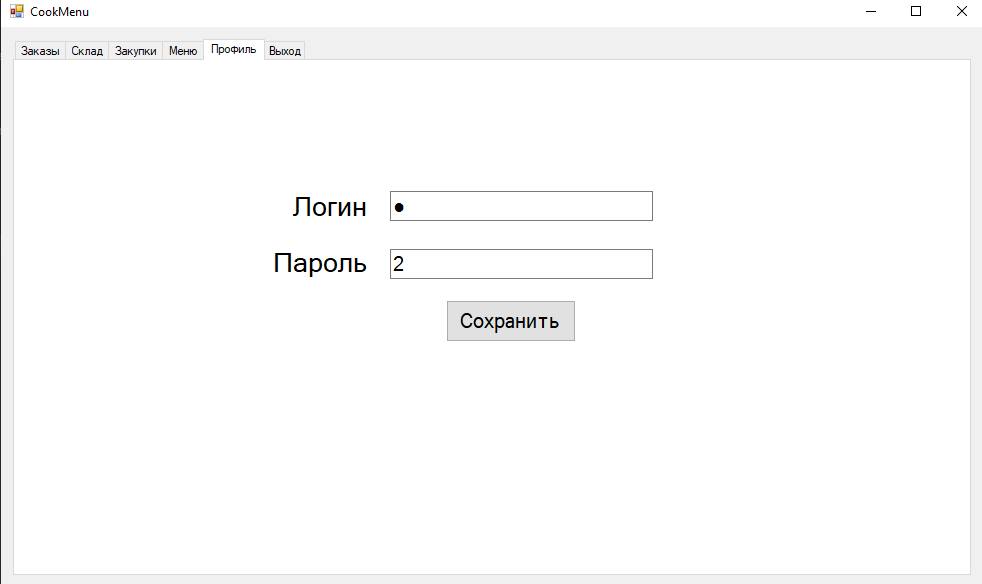
Окно ошибки при неправильно введенных данных



Окно регистрации

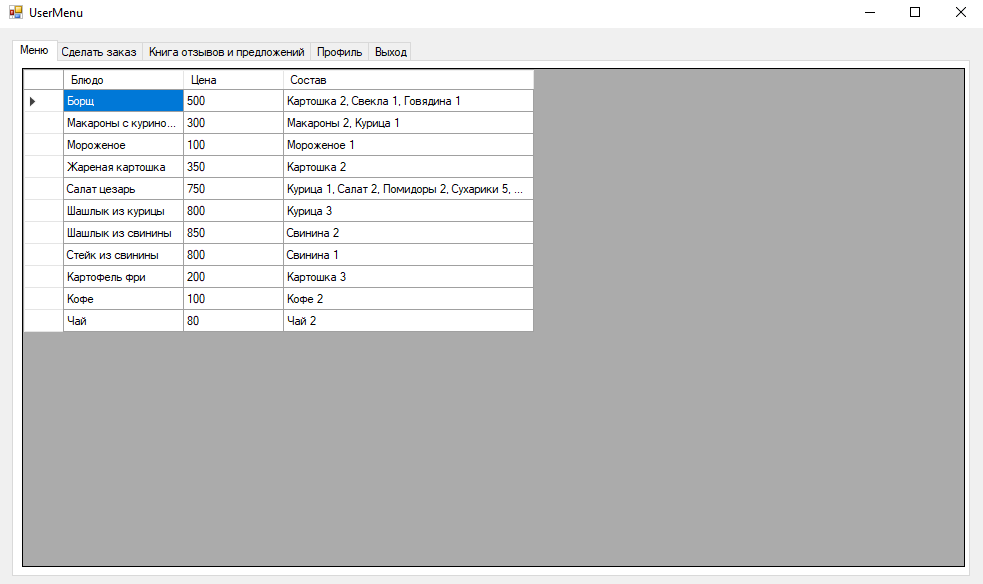


Окно редактирования профиля пользователя

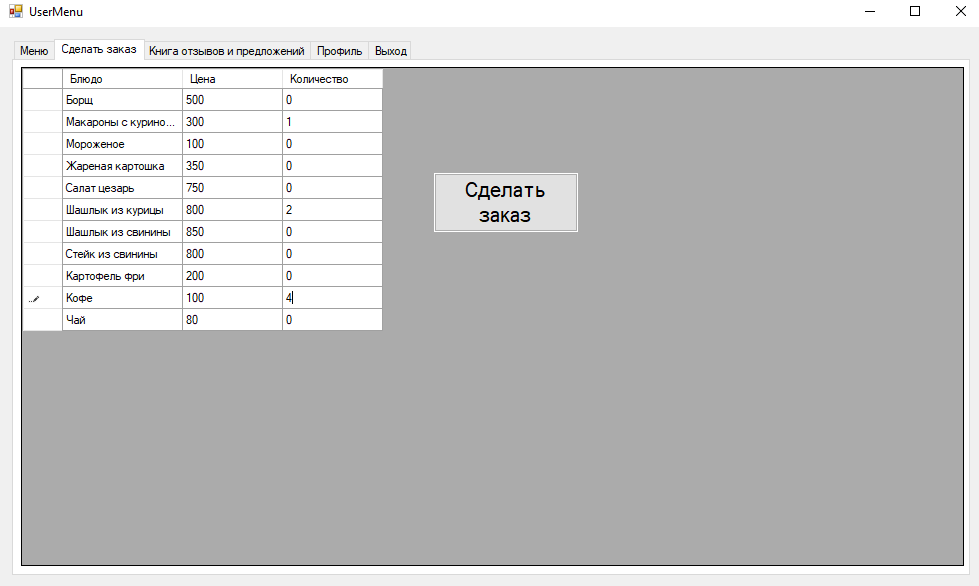


**Клиент**

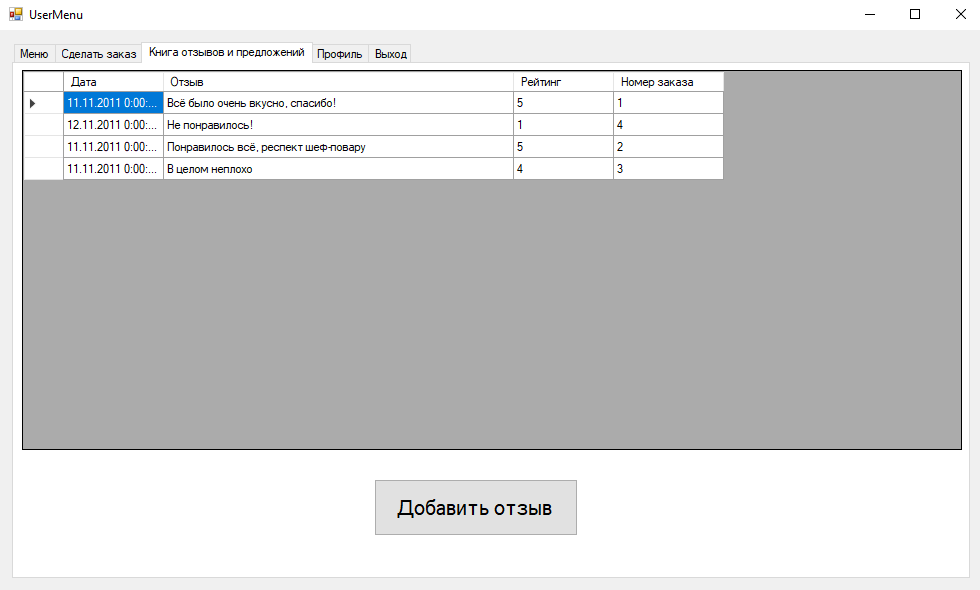
Окно просмотра меню



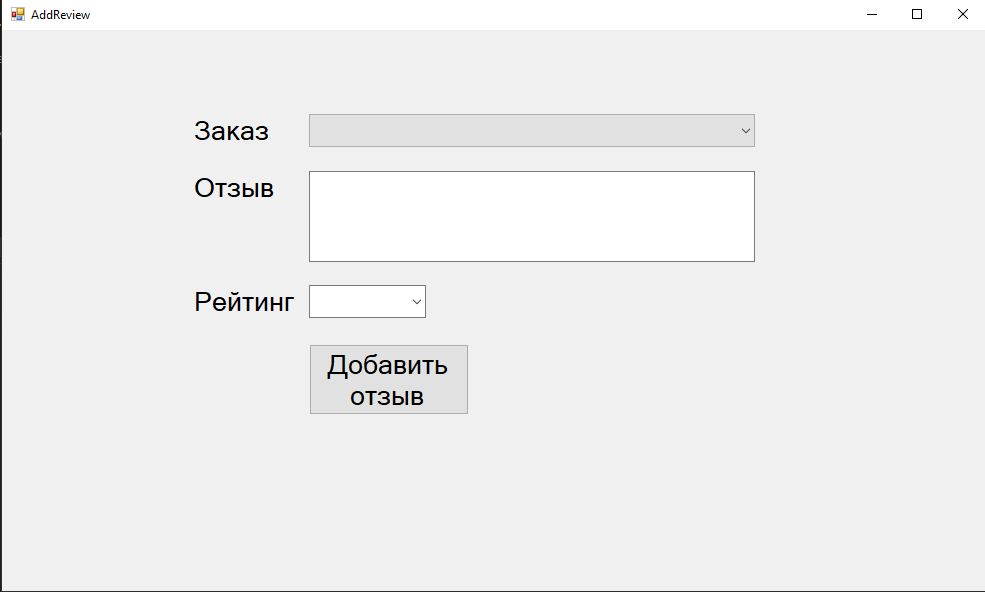
Окно выбора заказа



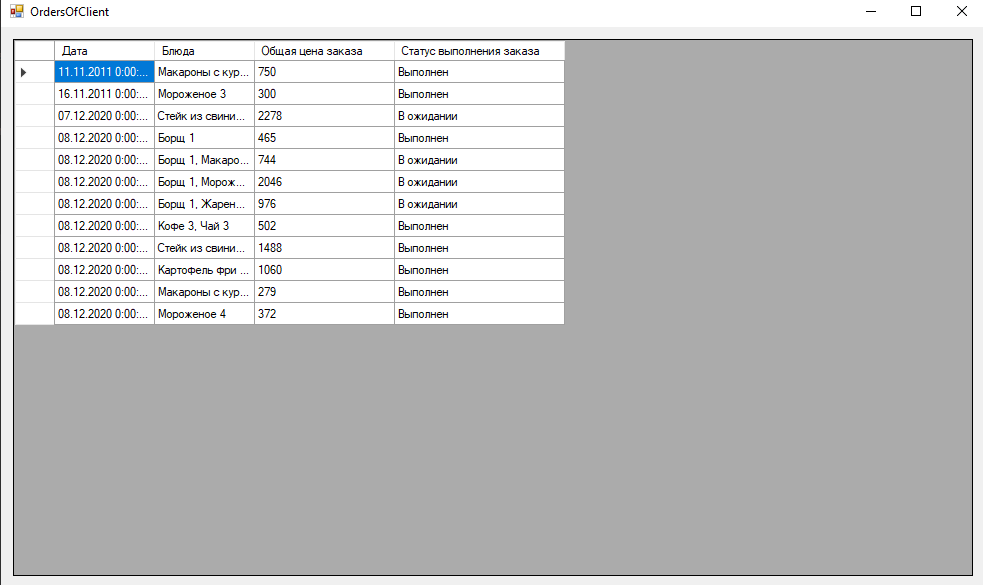
Окно просмотра книги отзывов и предложений



Окно добавления отзыва

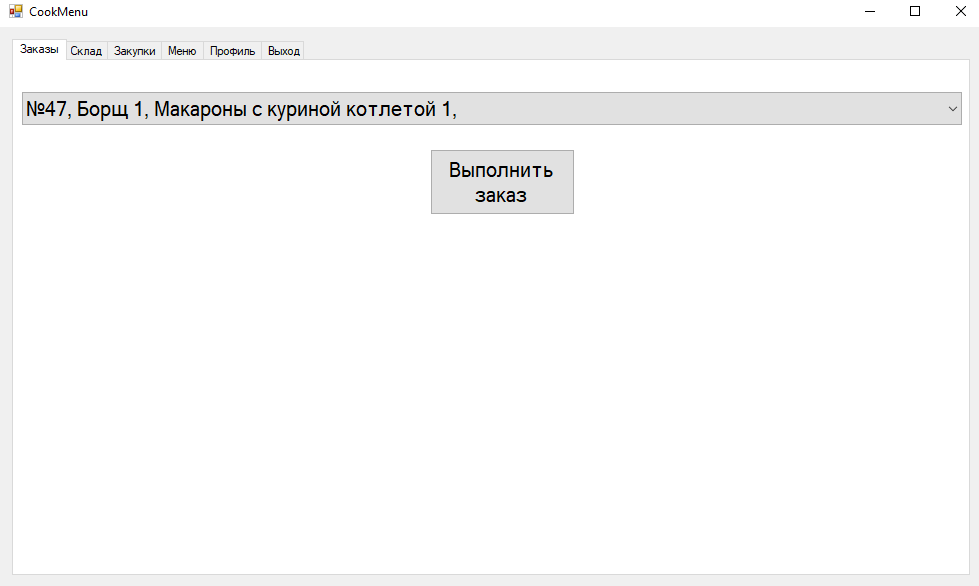


Окно просмотра собственных заказов клиента

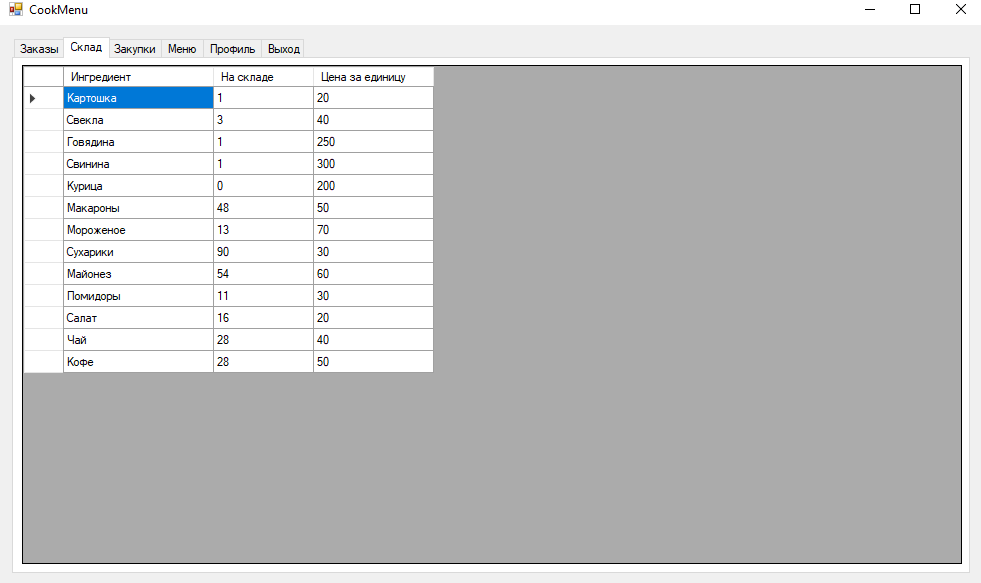


**Повар**

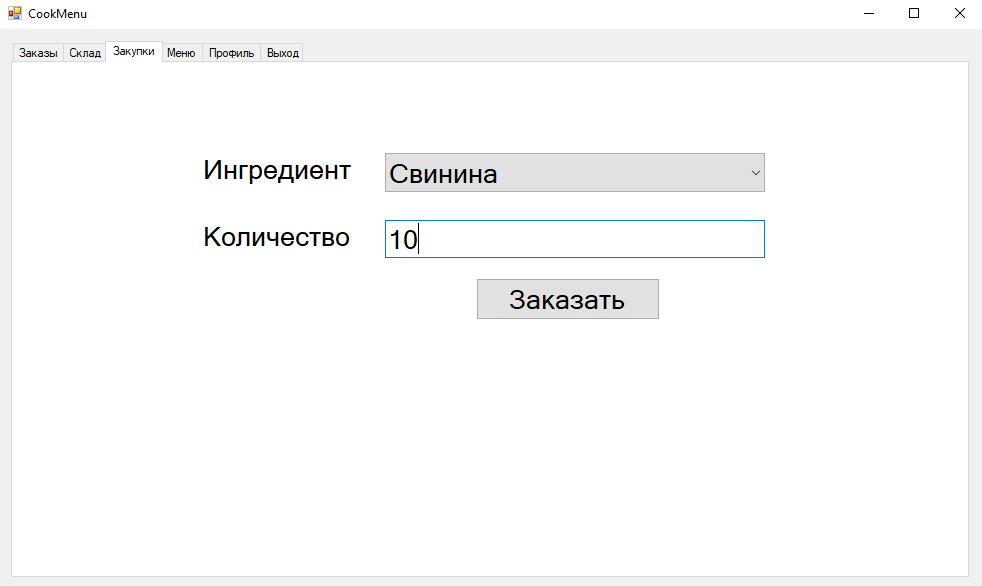
Окно выбора заказа для приготовления



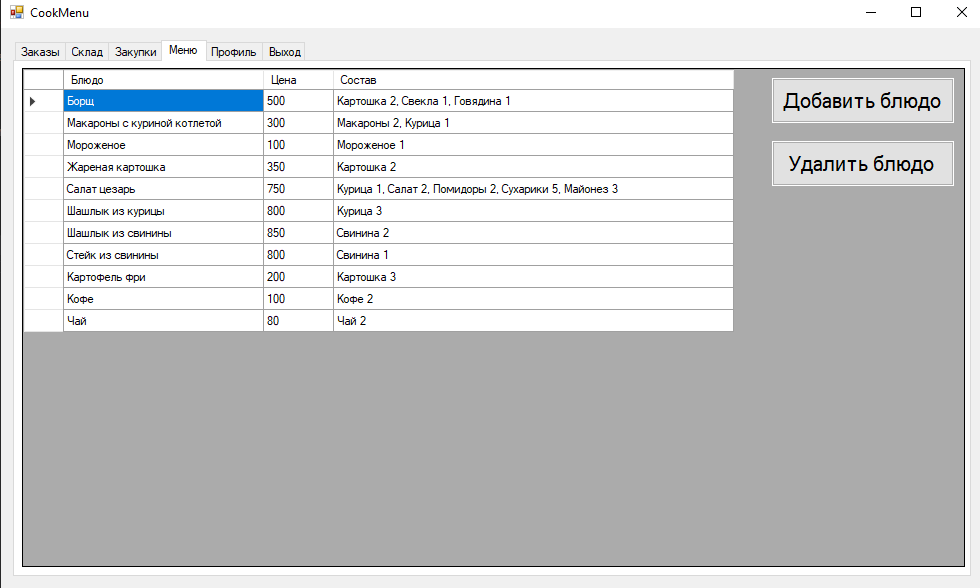
Окно просмотра ингредиентов на складе



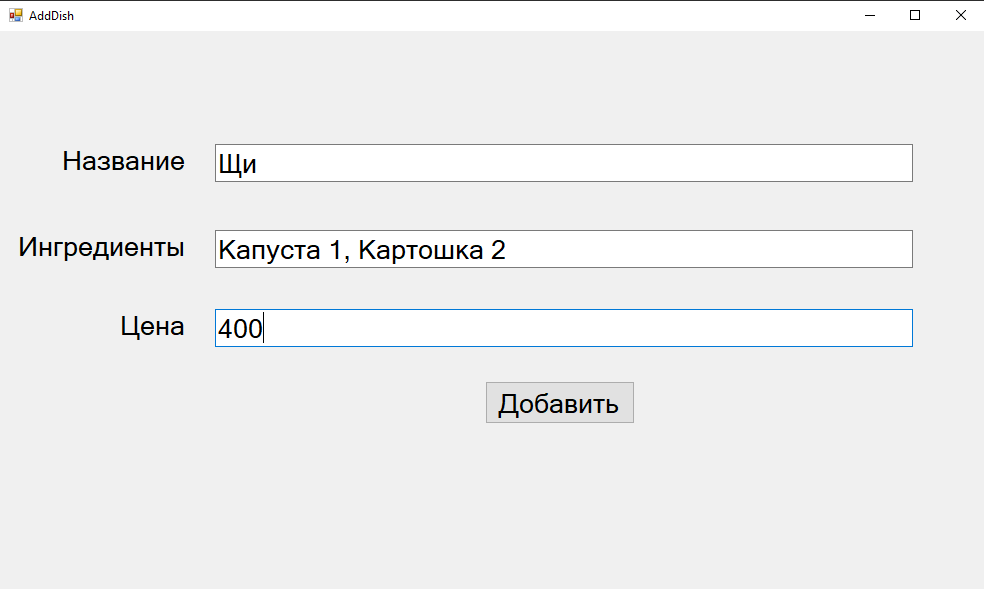
Окно закупок



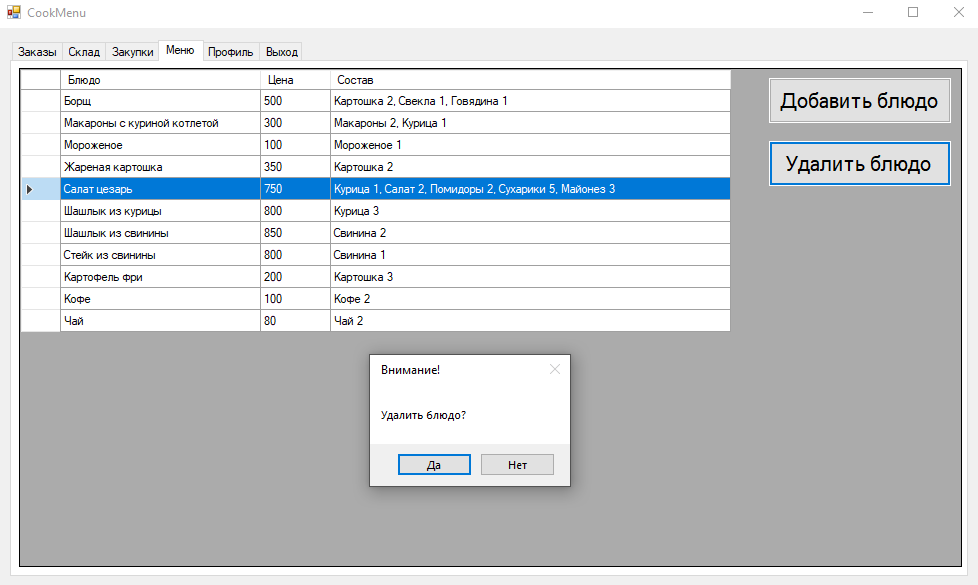
Окно просмотра составов блюд



Окно добавления нового блюда в меню

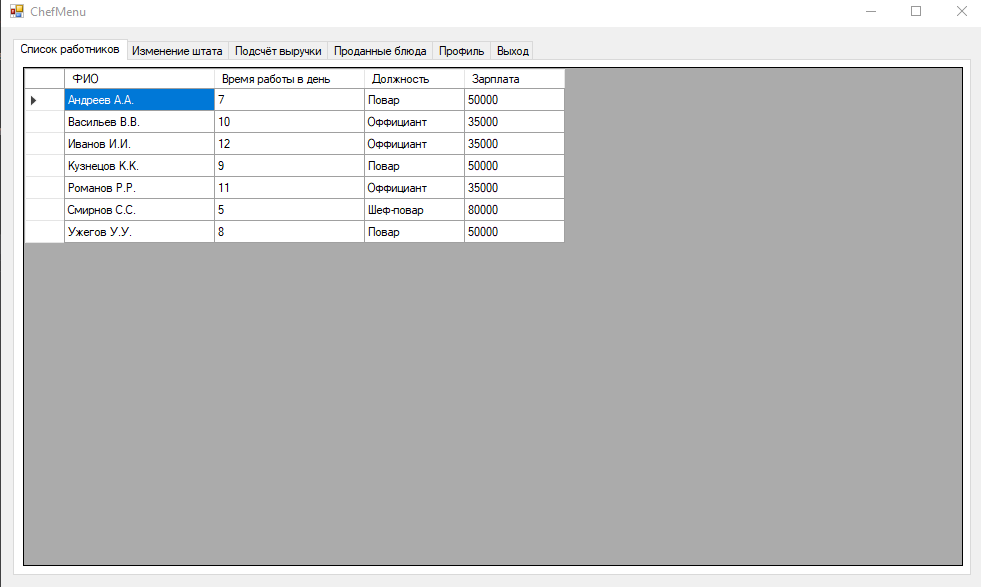


Удаление блюда из меню

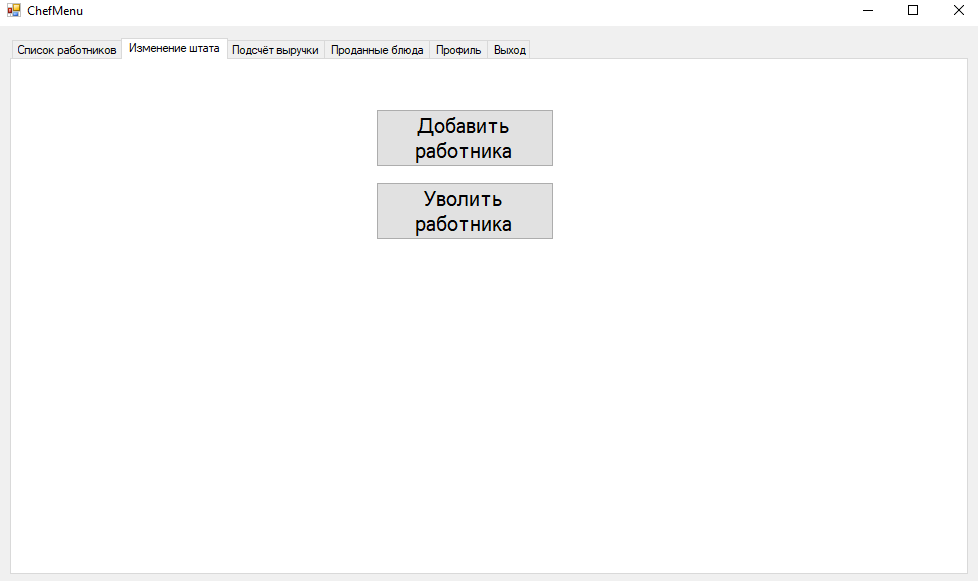


**Шеф-повар**

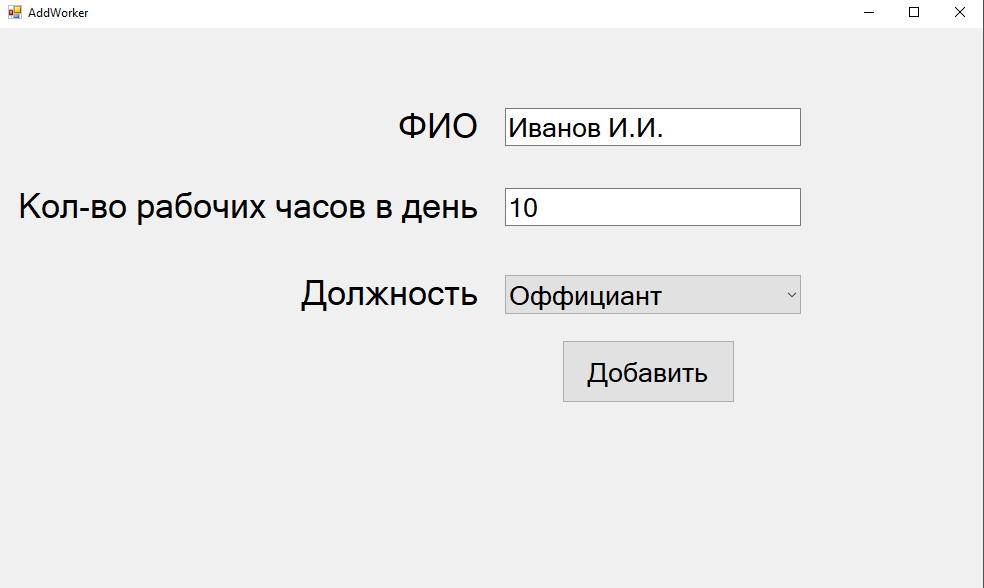
Окно просмотра списка работников



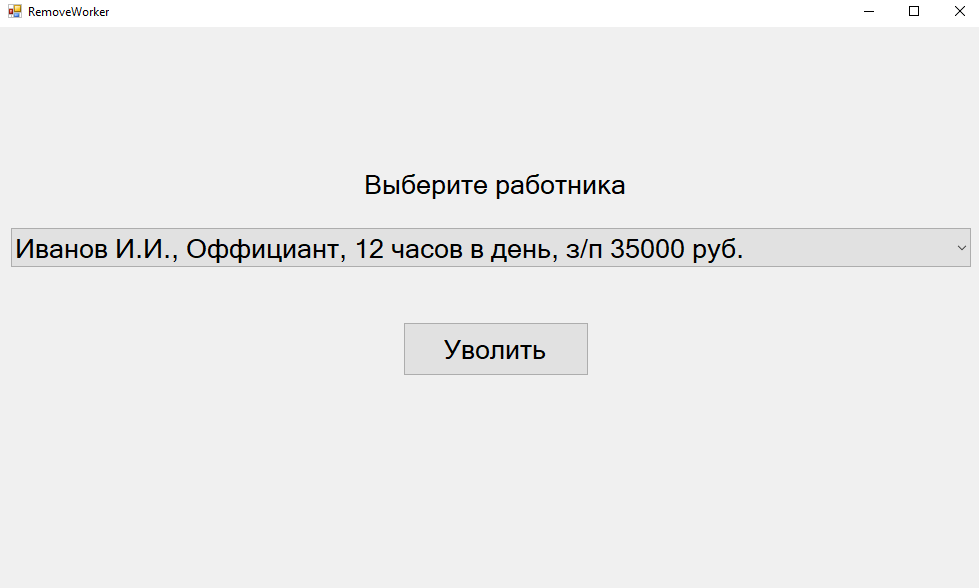
Окно выбора действий с работниками



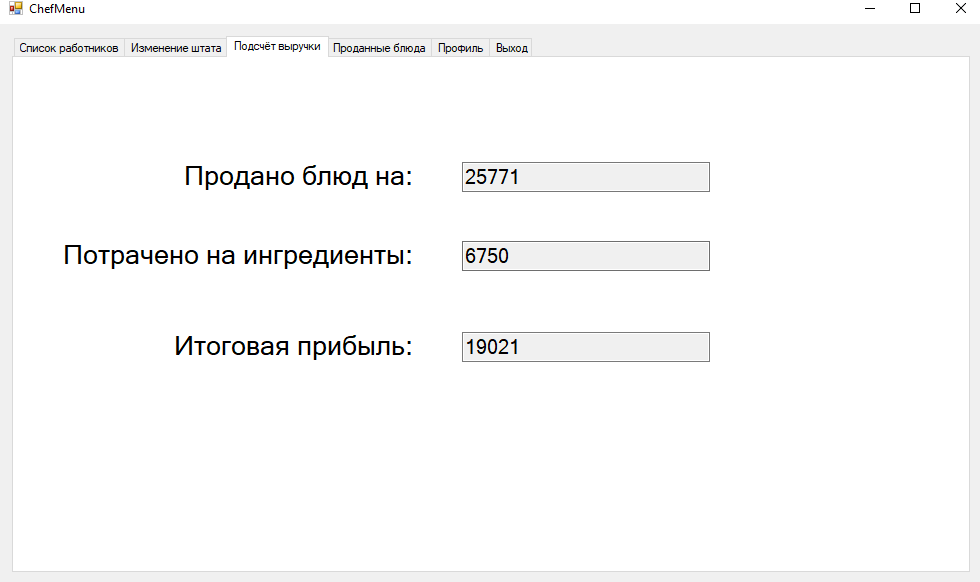
Окно добавления работника



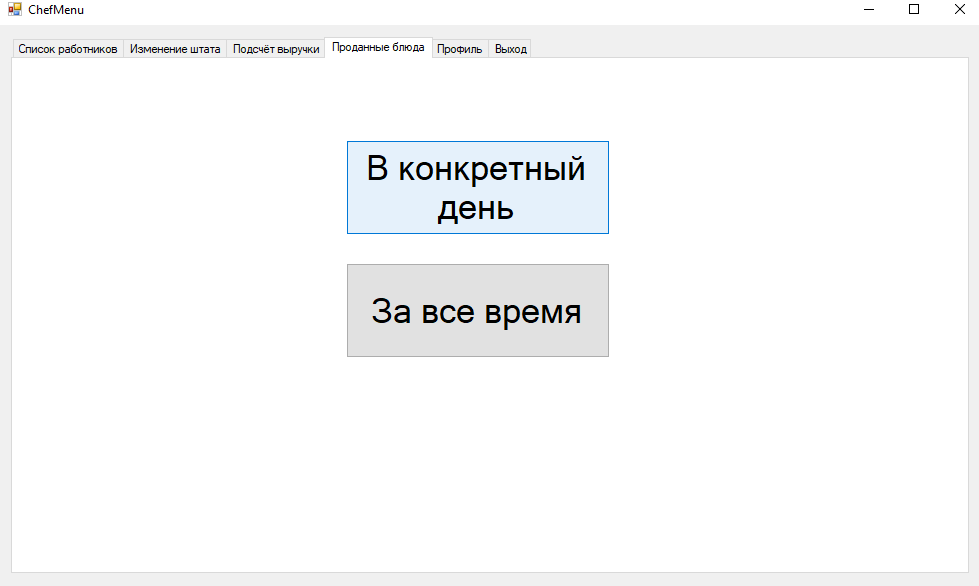
Окно удаления работника



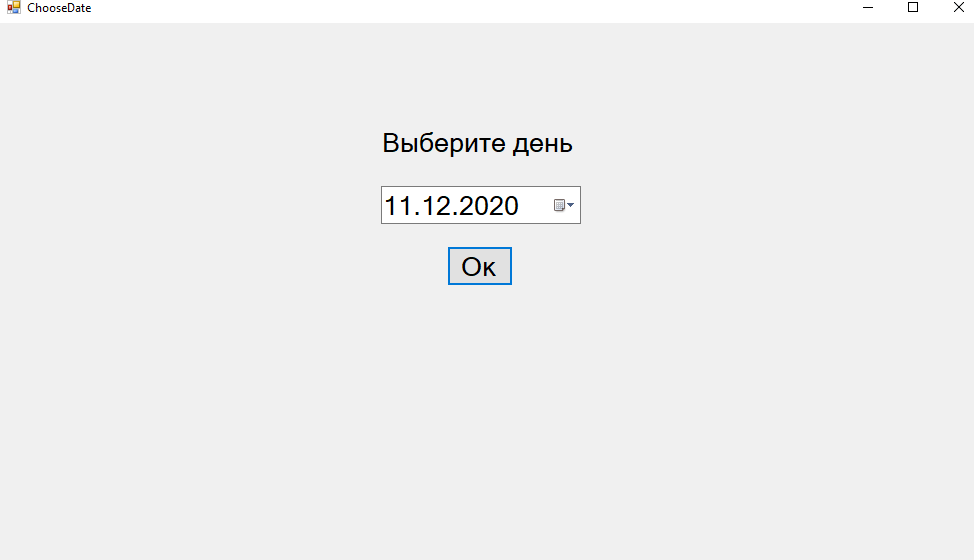
Окно просмотра выручки



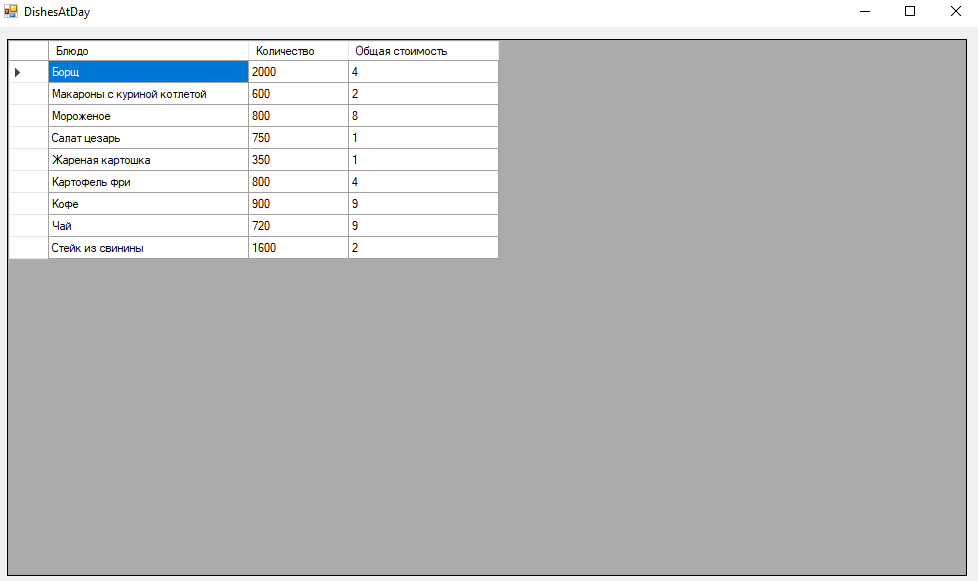
Окно выбора типа просмотра проданных блюд



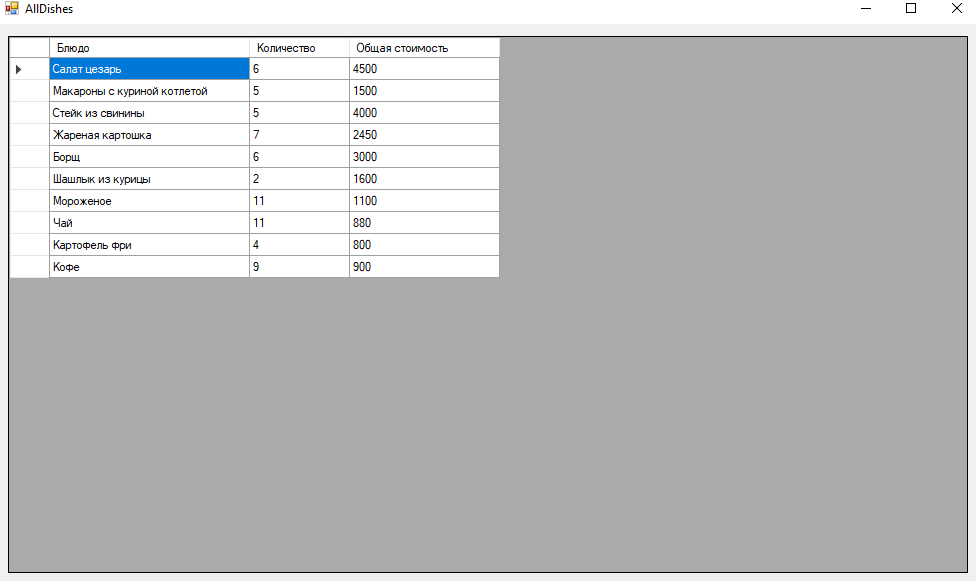
Окно выбора даты для просмотра проданных блюд



Окно проданных блюд в конкретную дату



Окно просмотра проданных блюд за все время



Заключение

В результате курсового проектирования была разработана база данных для ресторана. Также был разработан визуальный интерфейс для управления ей.

Список литературы

1. Документация по с#. <https://docs.microsoft.com/ru-ru/dotnet/csharp/>
2. Документация по SQL. <https://docs.microsoft.com/ru-ru/sql/sql-server/?view=sql-server-ver15>
3. Документация по Windows Forms. <https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/?view=netframeworkdesktop-4.8>