**Техническое задание**

**Для ресторана французской кухни**

**«Restaurant de Paris»**

Подготовили студентки 2 курса 2 группы

Мисюкевич Дарья Андреевна, Сидорова Алина Павловна, Буйко Алина Алексеевна

Минск, 2020г.

**Оглавление**

[Глава 1. Введение 3](#_Toc41514845)

[1.1. Общие сведения 3](#_Toc41514846)

[1.2. Назначение системы 3](#_Toc41514847)

[Глава 2. Требования к системе 4](#_Toc41514848)

[2.1. Функциональные требования 4](#_Toc41514849)

[2.1.1. Описание пользователей системы 4](#_Toc41514850)

[2.1.2. Описание вариантов использования 5](#_Toc41514851)

[2.2. Технические требования 8](#_Toc41514852)

[2.2.1. Архитектура системы 8](#_Toc41514853)

[2.2.2. Требования к пользовательскому интерфейсу 9](#_Toc41514854)

[2.2.3. Требования к надёжности 9](#_Toc41514855)

[Глава 3. Порядок приема системы 10](#_Toc41514856)

[3.1. Приемочные тесты 10](#_Toc41514857)

[3.2. Сроки исполнения 11](#_Toc41514858)

# **Глава 1. Введение**

## **1.1. Общие сведения**

Цель проекта – создание сервиса для сети ресторанов французской кухни.

## **1.2. Назначение системы**

Разработанная система “Restaurant de Paris” является веб-приложением, которым могут пользоваться как клиенты, так и администратор и шеф-повар

**Клиент может:**

* Войти в систему, введя имя и пароль.
* Создать заказ, выбрав блюда из меню.
* Редактировать заказ, добавив в него новые блюда.
* Посмотреть свой заказ.
* Оплатить заказ.

**Администратор** может:

* Войти в систему от имени администратора.
* Посмотреть список клиентов.
* Посмотреть историю заказов.
* Удалить заказ из истории.
* Удалить клиента из системы.

**Шеф-повар** может:

* Войти в систему от имени шеф-повара.
* Посмотреть меню.
* Добавить новые блюда в меню.
* Удалить блюда из меню.

# **Глава 2. Требования к системе**

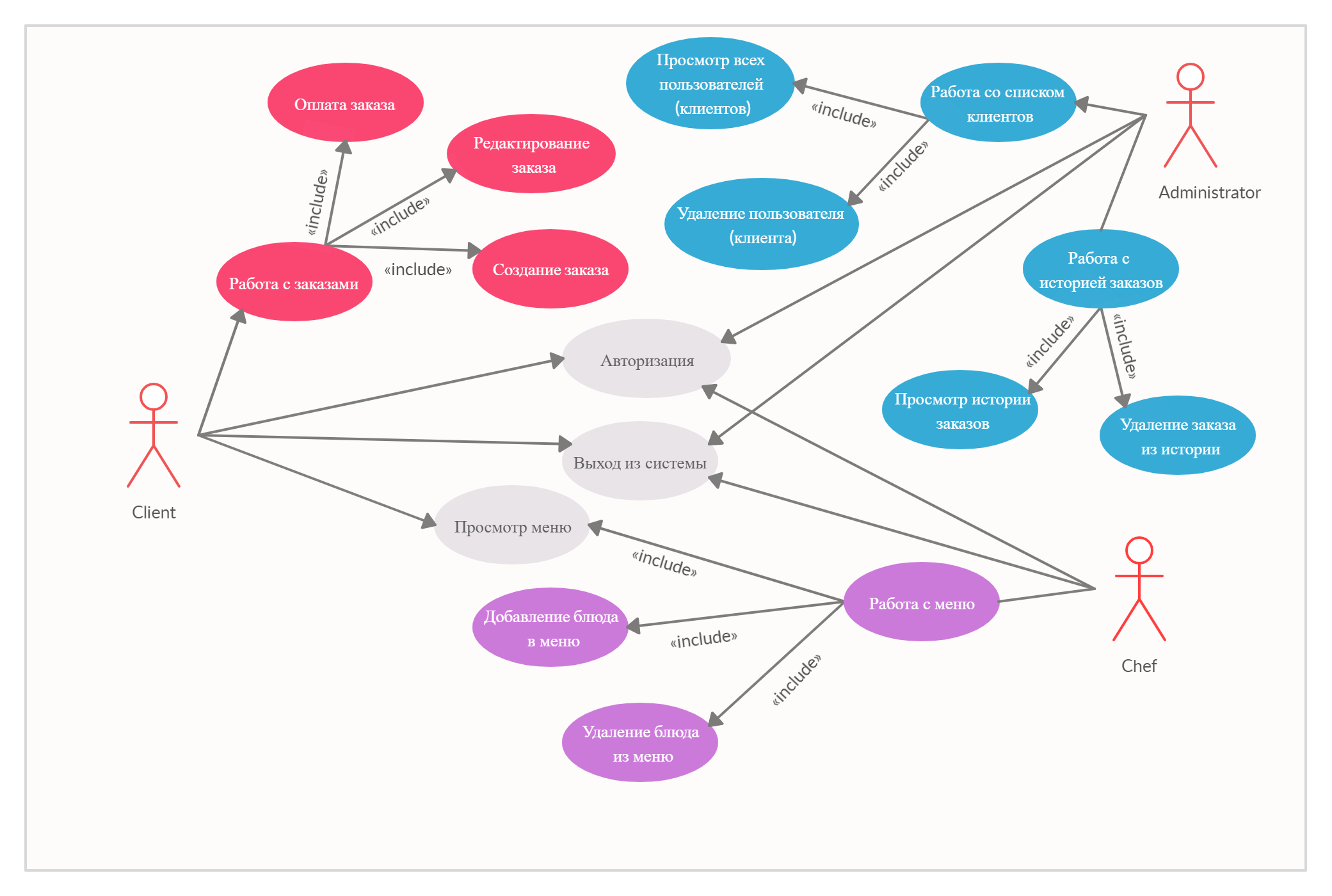
## **2.1. Функциональные требования**

### **2.1.1. Описание пользователей системы**

|  |  |  |
| --- | --- | --- |
| Номер | Актер | Описание |
| 1 | Клиент | Зарегистрированный пользователь, который может входить в систему, вводя имя и пароль. Может создать заказ, выбрав блюда из меню, отредактировать заказ, добавив в него новые блюда, посмотреть его и оплатить. |
| 2 | Администратор | Зарегистрированный пользователь, который может входить в систему от имени администратора. Может посмотреть список клиентов, историю заказов, удалить заказ из истории и клиента из системы. |
| 3 | Шеф-повар | Зарегистрированный пользователь, который может входить в систему от имени шеф-повара. Может посмотреть меню, добавить новые блюда в меню, удалить блюда из меню. |

### **2.1.2. Описание вариантов использования**

Диаграмма вариантов использования



**1. Авторизация**

**1. Actor:** Клиент, Администратор, Шеф-повар.  
**2. Goal in Context:** Авторизоваться в системе.  
**3. Trigger Event:** Необходимо Авторизоваться.  
**4. Pre-Condition:** Нет.  
**5. Success End Condition:** Клиент (Администратор, Шеф-повар) вошёл в систему.  
**6. Failed End Condition:** В авторизации отказано по какой-либо причин

**2. Просмотр истории заказов**

**1. Actor:** Администратор  
**2. Goal in Context:** Просмотреть историю заказов  
**3. Trigger Event:** Необходимо посмотреть историю заказов  
**4. Pre-Condition:** Пользователь(Администратор) должен быть авторизирован  
**5. Success End Condition:** Администратор посмотрел заказы  
**6. Failed End Condition:** Посмотреть заказы не удалось по какой-либо причине

### **3. Удаление заказа из истории**

**1. Actor:** Администратор  
**2. Goal in Context:** Удалить заказ из истории  
**3. Trigger Event:** Необходимо удалить заказ из истории  
**4. Pre-Condition:** Пользователь(Администратор) должен быть авторизирован  
**5. Success End Condition:** Администратор удалил заказ  
**6. Failed End Condition:** Удалить заказ не удалось по какой-либо причине

### **4. Просмотр всех пользователей (клиентов)**

**1. Actor:** Администратор  
**2. Goal in Context:** Посмотреть всех пользователей системы  
**3. Trigger Event:** Необходимо посмотреть список пользователей (клиентов)  
**4. Pre-Condition:** Пользователь(Администратор) должен быть авторизирован  
**5. Success End Condition:** Администратор посмотрел список пользователей (клиентов)  
**6. Failed End Condition:** Посмотреть пользователей не удалось по какой-либо причине

### **5. Удаление пользователя (клиента)**

**1. Actor:** Администратор  
**2. Goal in Context:** удалить пользователя (Клиента)  
**3. Trigger Event:** Необходимо удалить пользователя (Клиента)  
**4. Pre-Condition:** Необходимо быть администратором  
**5. Success End Condition:** Администратор удалил пользователя (Клиента)  
**6. Failed End Condition:** Удаление пользователя (Клиента) не удалось по какой-либо причине

### **6. Создание заказа**

**1. Actor:** Клиент  
**2. Goal in Context:** Создать заказ  
**3. Trigger Event:** Необходимо создать заказ  
**4. Pre-Condition:** Необходимо быть авторизованным Клиентом  
**5. Success End Condition:** Клиент создал заказ.  
**6. Failed End Condition:** Создать заказ не удалось по какой-либо причине.

### **7. Редактирование заказа**

**1. Actor:** Клиент  
**2. Goal in Context:** Редактировать заказ  
**3. Trigger Event:** Необходимо отредактировать заказ  
**4. Pre-Condition:** Необходимо быть авторизованным Клиентом, создавшим заказ  
**5. Success End Condition:** Клиент отредактировал заказ  
**6. Failed End Condition:** Отредактировать заказ не удалось по какой-либо причине

### **8. Оплата заказа**

**1. Actor:** Клиент  
**2. Goal in Context:** Оплатить заказ  
**3. Trigger Event:** Необходимо оплатить заказ  
**4. Pre-Condition:** Необходимо быть авторизованным Клиентом, создавшим заказ  
**5. Success End Condition:** Клиент оплатил заказ  
**6. Failed End Condition:** Оплатить заказ не удалось по какой-либо причине

### **9. Просмотр меню**

**1. Actor:** Шеф-Повар, Клиент  
**2. Goal in Context:** Просмотреть список блюд в меню  
**3. Trigger Event:** Необходимо посмотреть список блюд в меню  
**4. Pre-Condition:** Пользователь( Шеф-повар, Клиент) должен быть авторизирован  
**5. Success End Condition:** Клиент, Шеф-повар просмотрел меню  
**6. Failed End Condition:** Просмотреть меню не удалось по какой-либо причине

### **10. Добавление блюда в меню**

**1. Actor:** Шеф-повар  
**2. Goal in Context:** Добавить блюдо в меню  
**3. Trigger Event:** Необходимо добавить блюдо в меню  
**4. Pre-Condition:** Шеф-повар должен быть авторизирован  
**5. Success End Condition:** Шеф-повар добавил блюдо в меню  
**6. Failed End Condition:** добавление блюда в меню не удалось по какой-либо причине

### **11. Удаление блюда из меню**

**1. Actor:** Шеф-повар  
**2. Goal in Context:** Удалить блюдо из меню  
**3. Trigger Event:** Необходимо удалить блюдо из меню  
**4. Pre-Condition:** Шеф-повар должен быть авторизирован  
**5. Success End Condition:** Шеф-повар удалил блюдо из меню  
**6. Failed End Condition:** Удаление блюда из меню не удалось по какой-либо причине

### **12. Выход из системы**

**1. Actor:** Клиент, Шеф-повар, Администратор  
**2. Goal in Context:** Выйти из системы и сохранить данные  
**3. Trigger Event:** Необходимо выйти из системы  
**4. Pre-Condition:** Клиенту необходимо оплатить заказ  
**5. Success End Condition:** Пользователь вышел из системы  
**6. Failed End Condition:** Выйти из системы не удалось по какой-либо причине

­

## **2.2. Технические требования**

### **2.2.1. Архитектура системы**

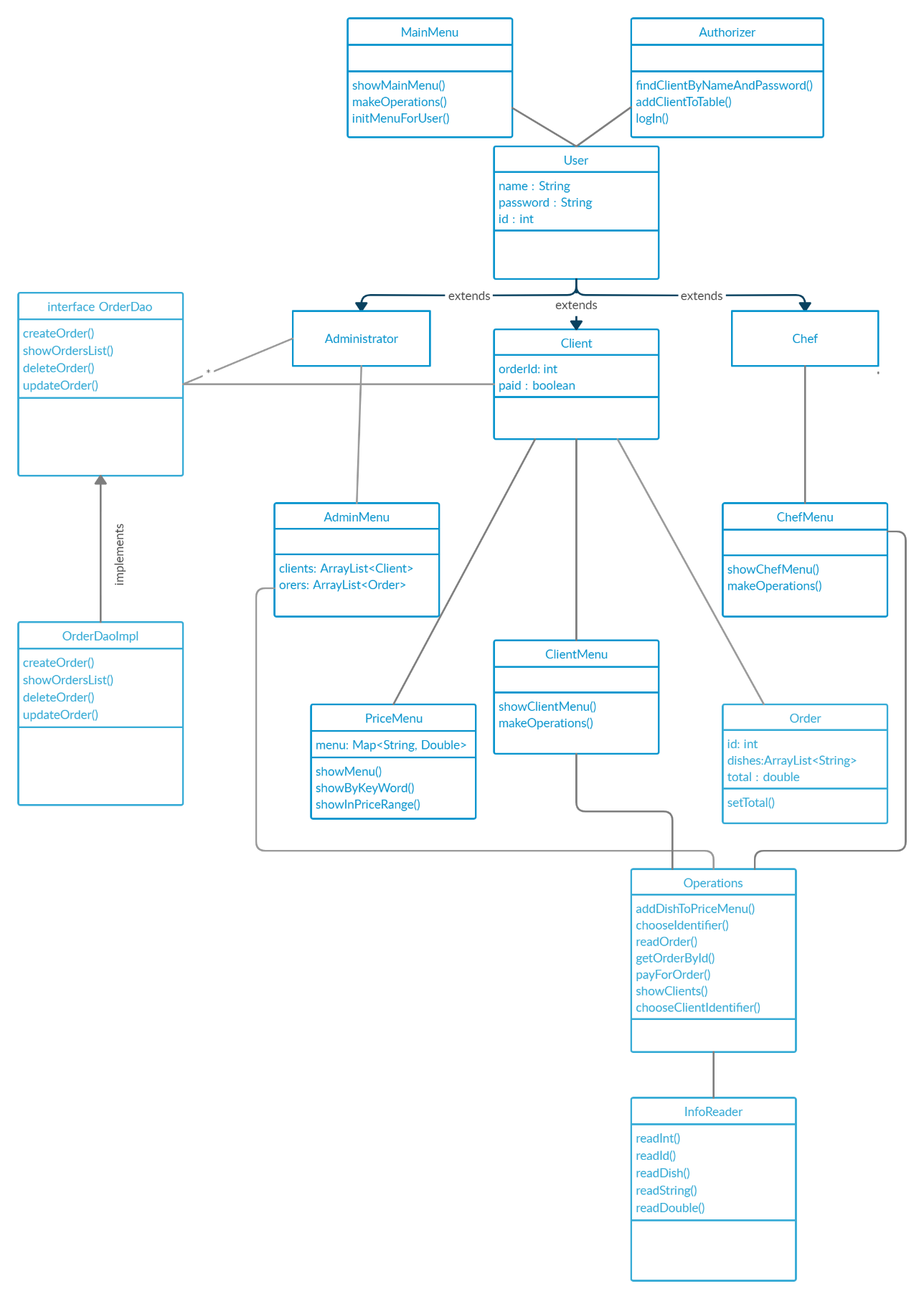
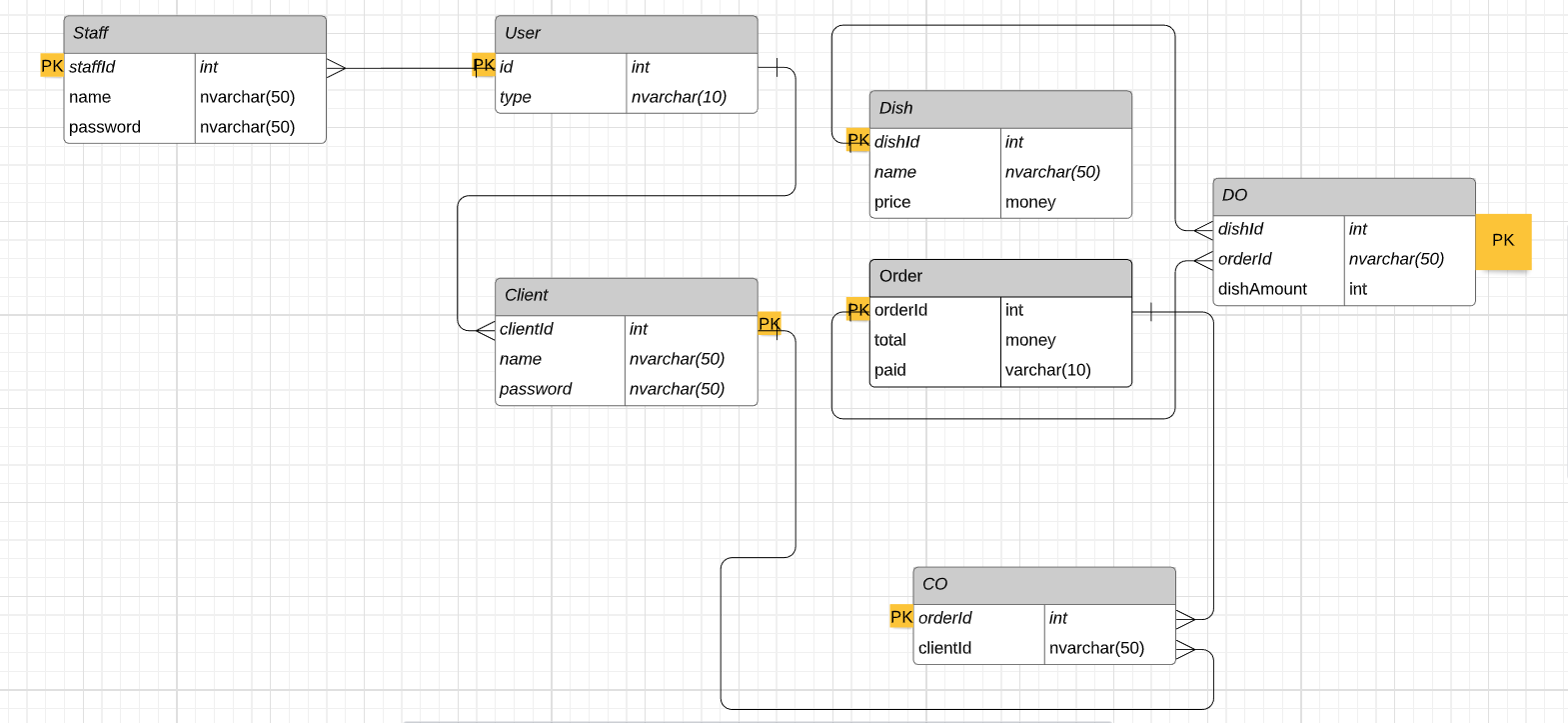
Диаграмма классов: 

Схема базы данных:



### 

### **2.2.2. Требования к пользовательскому интерфейсу**

Интерфейс должен быть интуитивно понятным и должен обеспечивать быструю навигацию всеми разделами системы. Сервис должен быть адаптирован под различные устройства (ПК, ноутбуки, планшеты, смартфоны).

**2.2.3. Требования к надежности**

Сервис должен обеспечивать защиту личных данных пользователей.

**Глава 3. Порядок приема системы**

## **3.1. Приемочные тесты**

1) Тестируемый компонент – регистрация:

Предусловие: под вводимым логином нет зарегистрированных пользователей.

Действия:

1. Открыть форму регистрации.

2. Ввести данные для регистрации (имя, пароль и повтор пароля) в поля и подтвердить регистрацию.

3. Перейти на страницу авторизации и авторизоваться под только что созданным аккаунтом (ввести имя и пароль).

4. Открываем страницу профиля

Ожидаемый результат: последовательность действий выполнялась без ошибок, удалось войти в систему под именем и паролем, указанным при регистрации.

2) Тестируемый компонент –меню:

Предусловие: имеются логины и пароли для двух аккаунтов: пользователя и шеф-повара заведения.

Действия:

1. Авторизоваться со страницы шеф-повара.

2. Перейти в секцию с меню и добавить туда новое блюдо.

3. Выйти из системы.

4. Войти в систему с аккаунта пользователя.

4. Открыть секцию с меню.

Необходимый результат: последовательность действий выполнялась без ошибок, при открытии секции с меню отображается добавленное блюдо.

## 

**3.2. Сроки исполнения**

|  |  |
| --- | --- |
| Задача | Срок исполнения |
| Разработка архитектуры проекта | 1 месяц |
| Проектирование базы данных | 3 недели |
| Разработка дизайна сайта | 2 недели |
| Создание макетов страниц сайта | 2 недели |
| Верстка страниц сайта | 1 месяц |
| Тестирование сайта | 1 неделя |
| Презентация проекта | 5 дней |