**Описание предметной области**

|  |  |
| --- | --- |
| № пп | Наименование пользователя |
| 1 | Клиент |
| 2 | Клиннинг |
| 3 | Менеджер |
| 4 | Администратор |

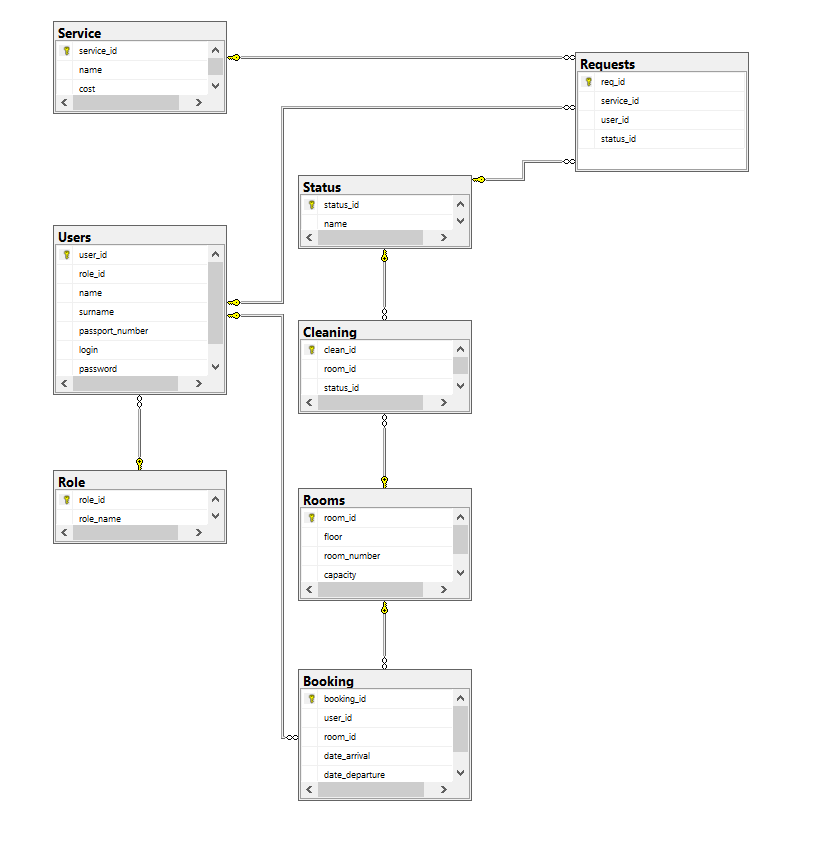
**Функции пользователей**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № пп | Выполняемая функция | Входные данные | Выходные данные | Функции, которые должны быть реализованы в ИС |
| **Клиент** | | | | |
| 1 | Посмотреть свою информацию | Паспортные данные | Информация о клиенте | Вывод информации о клиенте |
| 2 | Заказать услугу из списка | Id услуги, паспортные данные | Вывод сообщения о удачном заказе услуг | Добавить новую услугу в таблице Услуг |
| 3 | Заказать уборку номера | № комнаты, дата и время уборки | Вывод сообщения о удачном заказе клиннинга | Добавить заявку на уборку в таблице Уборка |
| 4 | Удалить заявку на уборку номера | № комнаты | Вывод сообщения об удачном удалении заявки | Удаление заявки из таблицы |
| **Клиннинг** | | | | |
| 1 | Просмотреть таблицу заявок на уборку |  | Вывод таблицы заявок на уборку | Вывод таблицы заявок на уборку |
| 2 | Редактировать таблицу заявок на уборку | Id заявки | Обновленная таблица заявок на уборку | Изменить таблицу заявок на уборку |
|  |  |  |  |  |
| **Менеджер** | | | | |
| 1 | Просмотр свободных номеров в отеле |  | Вывод таблицы свободных номеров | Вывод таблицы свободных номеров |
| 2 | Просмотр таблицы клиент-номер комнаты |  | Вывод таблицы с информацией о клиенте и его номер в отеле | Вывод представления клиент-комната |
| 3 | Просмотр таблицы заказов на услуги |  | Вывод таблицы с заказанными услугами | Вывод таблицы с заказанными услугами |
| 4 | Редактирование таблицы заказов на услуги | Id заявки, новый статус услуги | Обновленная таблица заказов услуг | Изменить таблицу заказов услуг |
| 5 | Добавление нового клиента | Информация о клиенте | Обновление таблицы о клиентах и номерах отеля | Добавление нового клиента в таблицу |
| 6 | Удалить клиента | Паспортные данные клиента | Обновленная таблица клиент-номер/ Обновленные список свободных номеров | Удалить клиента из таблицы |
| **Админ** | | | | |
| 1 | Добавление нового работника | Данные нового работника | Обновленная таблица работников | Добавление нового работница в таблицу |
| 2 | Просмотр списка работников и их должности |  | Таблица работников и их должность | Вывод таблицы работников |
| 3 | Уволить работника | Паспортные данные работника | Обновленная таблица работников | Удалить работника из таблицы |
| 4 | Просмотр таблицы заказов услуг |  | Вывод таблицы заказов услуг | Вывод таблицы заказов услуг |
| 5 | Просмотр таблицы Уборка |  | Вывод таблицы Уборка | Вывод таблицы Уборка |
| 6 | Просмотр списка услуг |  | Вывод всех доп. услуг | Вывод таблицы |
| 7 | Изменение каталога услуг | Новая услуга | Обновленная таблица услуг | Изменение услуги |
| 8 | Удалить услугу | Id услуги | Обновленная таблица услуг | Удаление услуги из таблицы |

**Хранимые данные**

|  |  |  |  |
| --- | --- | --- | --- |
| № пп | Хранимые данные | Пользователи, которым разрешен доступ | Ограничения по типу и значению |
| 1 | Заявки на услуги | Клиент, Менеджер, Администратор |  |
| 2 | Заявки на уборку | Клиент, Менеджер, Администратор |  |
| 3 | Информация о клиентах | Менеджер, Администратор |  |
| 4 | Информация о работниках | Администратор |  |
| 5 | Услуги | Администратор |  |

**Схема БД**



В таблице Role описаны должности сотрудников отеля и клиенты

В таблице Users собрана вся информация о сотрудниках и клиентах, их логин и пароль для входа в систему

В таблице Booking собрана вся информация про жильцов отеля

В таблице Requests собрана информация о статусах заказов доп.услуг

В таблице Status находятся все возможные статусы

В таблице Service информация про все возможные доп. услуги

В таблице Cleaning собрана информация для cleaning’ого сервиса

В таблице Rooms вся информация о комнатах

**Код создания таблиц**

Create Table Role

(role\_id int IDENTITY(1,1) PRIMARY KEY Check(role\_id Between 1 And 4) NOT NULL,

role\_name varchar(15) Not Null

)

Go

Create Table Users

(user\_id int IDENTITY(1,1) Primary KEY Not Null,

role\_id int FOREIGN KEY REFERENCES Role Not Null,

name varchar(15) Not Null,

surname varchar(15) Not Null,

passport\_number varchar(20) Not Null,

login varchar(20) Not Null,

password varchar(128) Not Null

)

Go

Create Table Rooms

(room\_id int Identity(1,1) Primary Key Not Null,

floor int Not Null,

room\_number int Not Null,

capacity int Not Null

)

Go

Create Table Booking

(booking\_id int Identity(1,1) Primary Key Not Null,

user\_id int Foreign Key References Users Not Null,

room\_id int Foreign Key References Rooms Not Null,

date\_arrival date Not Null,

date\_departure date Not Null

)

Go

Create Table Service

(service\_id int Identity(1,1) Primary Key Not Null,

name varchar(20) Not Null,

cost int Not Null

)

Go

Create Table Status

(status\_id int Identity(1,1) Primary Key Not Null,

name varchar(20) Not Null

)

Go

Create Table Requests

(req\_id int Identity(1,1) Primary Key Not Null,

service\_id int Foreign Key References Service Not Null,

user\_id int Foreign Key References Users Not Null,

status\_id int Foreign Key References Status Not Null

)

Go

Create Table Cleaning

(clean\_id int Identity(1,1) Primary Key Not Null,

room\_id int Foreign Key References Rooms Not Null,

status\_id int Foreign Key References Status Not Null

)

Go

**Запросы**

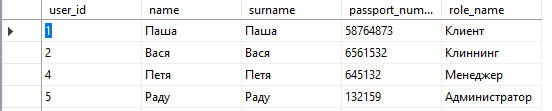
**Представления:**

* Создать представление, которое будет выводить имя и фамилию и должность

Create View Workers As

Select user\_id, name, surname, passport\_number, role\_name From Users, Role

Where Users.role\_id=Role.role\_id

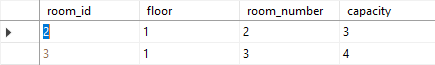


* Вывести информацию о комнатах, где вместимость больше 1 человека

Create View Rooms2Cap As

Select room\_id, floor, room\_number, capacity From Rooms

Where capacity>=2

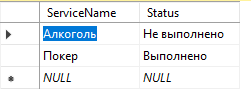


* Создать представление о услугах и их статусе

Create View ServList As

Select srv.name as ServiceName, st.name as Status From Service srv, Requests req, Status st

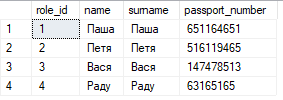
Where srv.service\_id=req.service\_id And req.status\_id=st.status\_id



**Вывод:**

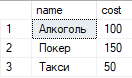
* Вывести role\_id, имя, фамилию и паспорт всех сотрудников и клиентов

Select role\_id, name, surname, passport\_number From Users



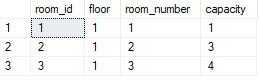
* Вывести название и цену доп. услуг

Select name, cost From Service



* Вывести всю информацию про комнаты

Select \* From Rooms



* Вывести дату прибытия и отбытия всех клиентов

Select date\_arrival, date\_departure From Booking

* Вывести логин и пароль всех пользователей

Select login, password From Users

* Вывести имя и фамилию клиентов

Select name, surname From Users

Where role\_id=1

* Вывести id комнаты, где уборка еще не производилась

Select room\_id From Cleaning

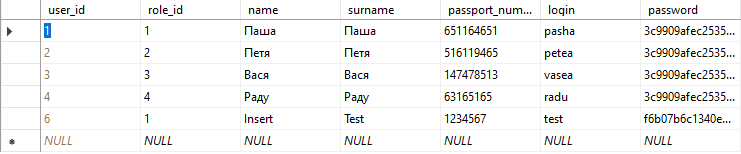
Where status\_id=2

**Добавление:**

* Добавить нового клиента

Insert Into Users(role\_id, name, surname, passport\_number, login, password)

Values('1','Insert', 'Test', '1234567', 'test', 'f6b07b6c1340e947b861def5f8b092d8ee710826dc56bd175bdc8f3a16b0b8acf853c64786a710dedf9d1524d61e32504e27d60de159af110bc3941490731578')



* Добавить новую доп. услугу

Insert Into Service(name, cost)

Values('Сигареты', '120')

* Добавить новый запрос на доп. услугу

Insert Into Requests(service\_id, user\_id, status\_id)

Values('4','1','2')

* Добавить новый запрос на уборку комнаты

Insert Into Cleaning(room\_id, status\_id)

Values('3','2')

* Добавить новый статус ожидания заявок

Insert Into Status(name)

Values('В процессе')

* Добавить новую комнату

Insert Into Rooms(floor, room\_number, capacity)

Values('5','125','6')

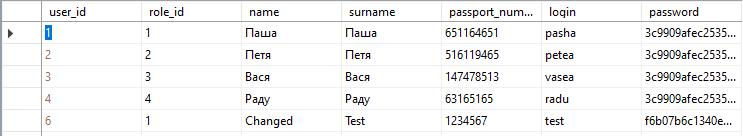
**Изменение:**

* Изменить имя у пользователя

Update Users

Set name='Changed'

Where passport\_number=1234567



* Изменить цену доп. услуги

Update Service

Set cost=200

Where service\_id=1

* Изменить статус заявки доп. услуги

Update Requests

Set status\_id=1

Where req\_id=1

* Изменить статус клиннинга

Update Cleaning

Set status\_id=1

Where clean\_id=1

* Изменить вместимость комнаты

Update Rooms

Set capacity=4

Where room\_id=1

* Изменить дату выезда клиента

Update Booking

Set date\_departure='16.10.2021'

Where booking\_id=1

**Удаление:**

* Удалить пользователя по паспорту

Delete From Users

Where passport\_number=1234567

* Удалить сервис

Delete From Service

Where service\_id=5

* Удалить статус

Delete From Status

Where status\_id=3

* Удалить запрос на доп. услугу

Delete From Requests

Where req\_id=6

* Удалить комнату

Delete From Rooms

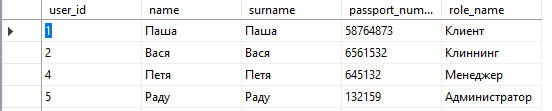
Where room\_id=15

**Сложные запросы:**

* Вывести имя, фамилию и статус пользователя

Select name, surname, passport\_number, role\_name From Users, Role

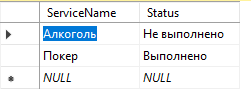
Where Users.role\_id=Role.role\_id



* Вывести услугу и ее статус

Select srv.name as ServiceName, st.name as Status From Service srv, Requests req, Status st

Where srv.service\_id=req.service\_id And req.status\_id=st.status\_id



* Вывести имя, фамилию, дату приезда и дата выезда клиента

Select u.name, u.surname, b.date\_arrival, b.date\_departure From Users u Join Booking b On

u.user\_id=b.user\_id



* Вывести этаж и количество на нем комнат

Select floor as Floor, Count(room\_number) as Count From Rooms

Group By floor

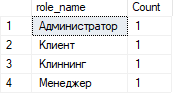


* Вывести статус и кол-во пользователей с таким статусом

Select r.role\_name, Count(u.user\_id) as Count From Users u Join Role r On

r.role\_id=u.role\_id

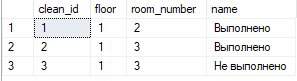
Group By r.role\_name



* Вывести clean\_id,информацию о комнате и информацию про уборку

Select cl.clean\_id, r.floor, r.room\_number, st.name From Cleaning cl, Rooms r, Status st

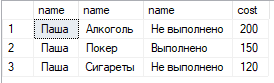
Where st.status\_id=cl.status\_id And cl.room\_id=r.room\_id



* Вывести имя клиента, заказанную услугу, статус услуги и ее цена

Select u.name, s.name, st.name, s.cost From Requests r, Service s, Status st, Users u

Where u.user\_id=r.user\_id And s.service\_id=r.service\_id And st.status\_id=r.status\_id



* Вывести имя клиента и информацию о его номере

Select u.name, r.floor, r.room\_number, r.capacity From Users u, Booking b, Rooms r

Where u.user\_id=b.user\_id And b.room\_id=r.room\_id



* Вывести имя и фамилию клиента, который имеет более 2 необслуженных услуг

Select u.name, u.surname, Count(r.req\_id) as Count From Users u, Requests r, Status st

Where u.user\_id=r.user\_id And r.status\_id=st.status\_id

Group By u.name, u.surname

Having Count(r.req\_id)>=2



* Вывести информацию о номере, где уборка еще не выполнена

Select r.floor, r.room\_number, st.name From Rooms r, Cleaning cl, Status st

Where r.room\_id=cl.room\_id And cl.status\_id=st.status\_id And st.status\_id=2



**ХП:**

* Создать процедуру, которая будет выводить статус пользователя по его паспортным данным

Go

Create Procedure userStatus(@passport As varchar(20)) As

Begin

Select r.role\_name From Users u Join Role r On

r.role\_id=u.role\_id

Where u.passport\_number=@passport

End;

Exec userStatus '63165165'



* Вывести сотрудников, указав их должности

Go

Create Procedure usersSt As

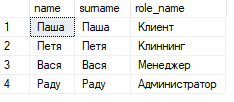
Begin

Select u.name, u.surname, r.role\_name From Users u Join Role r On

r.role\_id=u.role\_id

End;

Exec usersSt



* Добавит новую доп. услугу

Go

Create Procedure newService(@newName as varchar(20), @newCost as int) As

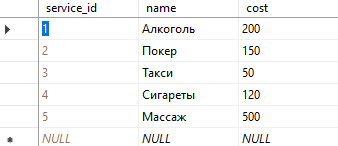
Begin

Insert Into Service(name, cost)

Values(@newName, @newCost)

End;

Exec newService 'Масcаж', '500'



**Триггеры:**

* Дата выезда не может быть раньше, чем дата въезда

Create Trigger checkDataST On Booking

After Insert, Update

As

Begin

Set Nocount On;

declare @data1 date

set @data1 = (Select date\_arrival From inserted)

declare @date2 date

set @date2 = (Select date\_departure From inserted)

if @data1>@date2

begin

rollback tran

raiserror('Data error',16,10)

end

end

Insert Into Booking(user\_id, room\_id, date\_arrival, date\_departure)

Values(1,3,'10.10.2021', '9.10.2021')



* Паспорт должен быть уникальным

CREATE TRIGGER IsPassport

ON Users

FOR INSERT, UPDATE

AS

BEGIN

IF EXISTS(SELECT passport\_number FROM Users WHERE passport\_number IN (SELECT passport\_number FROM Inserted) GROUP BY passport\_number HAVING count(\*) > 1)

BEGIN

RAISERROR('Есть такой паспорт!', 16, 1)

ROLLBACK TRAN

END

END



* Логин должен быть уникальным

CREATE TRIGGER IsLogin

ON Users

FOR INSERT, UPDATE

AS

BEGIN

IF EXISTS(SELECT login FROM Users WHERE login IN (SELECT login FROM Inserted) GROUP BY login HAVING count(\*) > 1)

BEGIN

RAISERROR('Есть такой логин!', 16, 1)

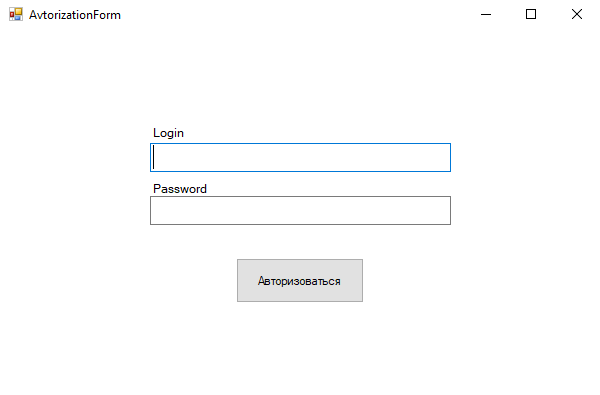
ROLLBACK TRAN

END

END

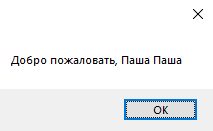
**Приложение:**

**Авторизация:**



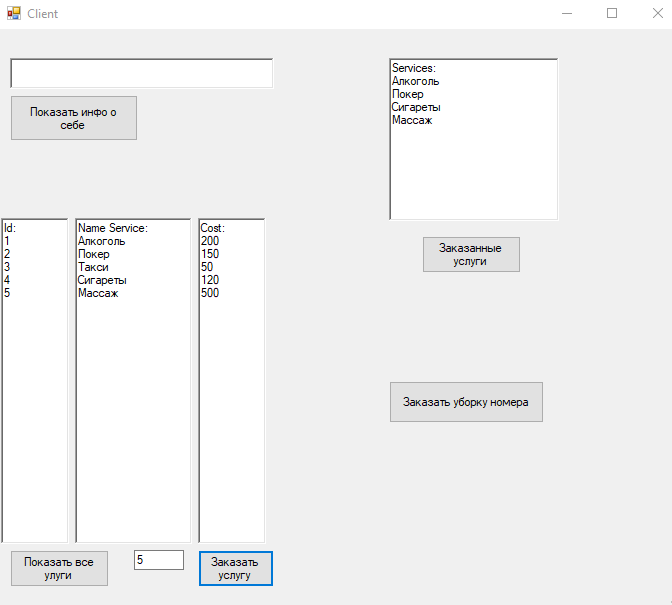
Авторизация включает только два поля: логин и пароль. Как мы уже видели в таблицах выше, пароли захэшированы. Тут же мы просто должны вбить настоящий пароль.

После успешной авторизации открывается окно приветствия:

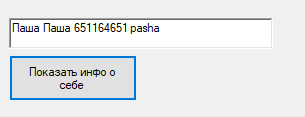


После этого уже открывается окно пользователя(для каждой роли оно разное).

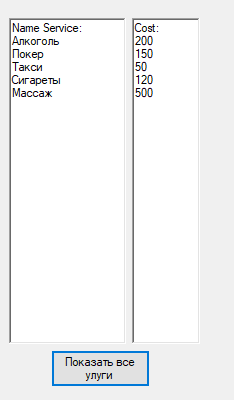
**Клиент:**



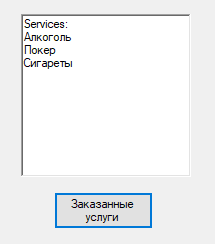
Тут клиент может посмотреть свою информацию:



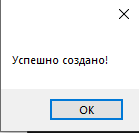
Посмотреть список всех доступных услуг:

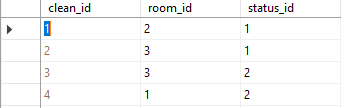


Посмотреть заказанные ранее услуги:

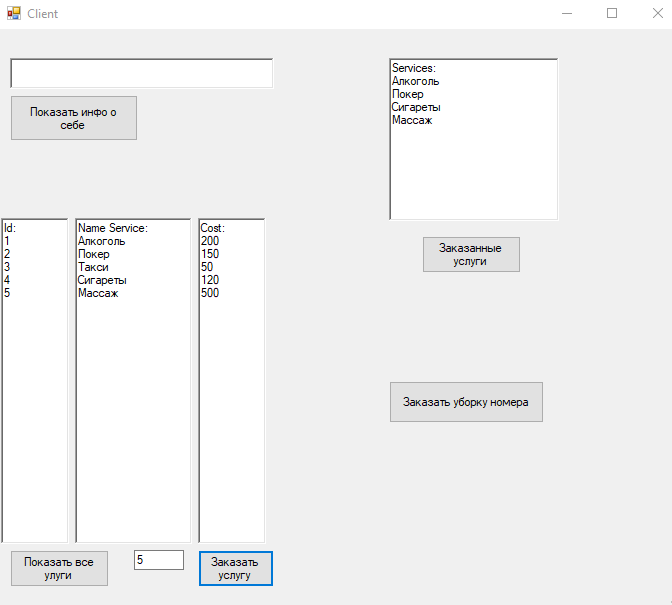


Заказать уборку номера:



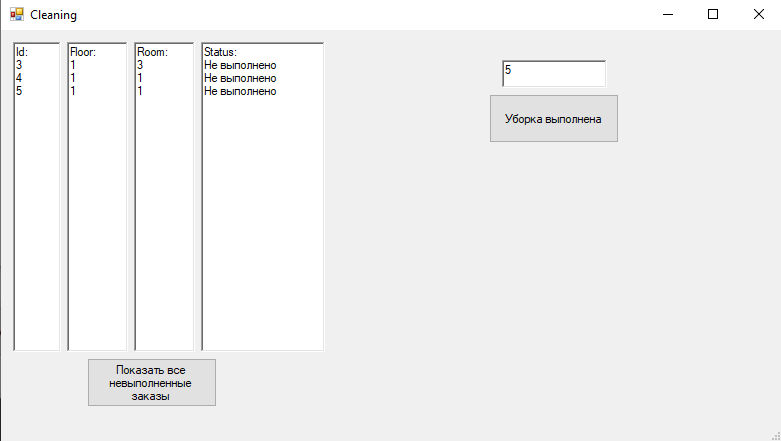


Заказать доп. услугу:

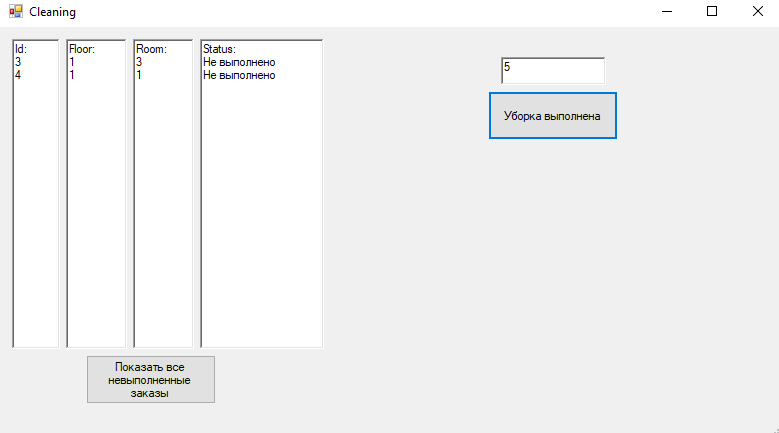


**Клиннинг:**

Выводит список всех неубранных еще комнат:

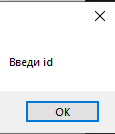


Также можно ввести id уборки из левой таблицы и нажать кнопку «Уборка выполнена»:

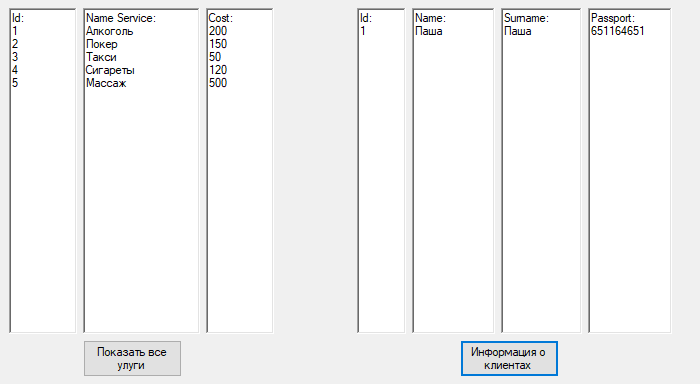


Она уходит их левых таблиц, но не удаляется. У нее меняется только статус.

Если поле id будет пустое, то вывод ошибки:

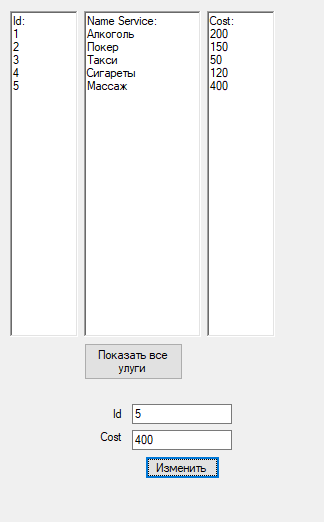


**Менеджер:**

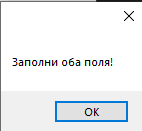


Может смотреть все услуги и всех клиентов.

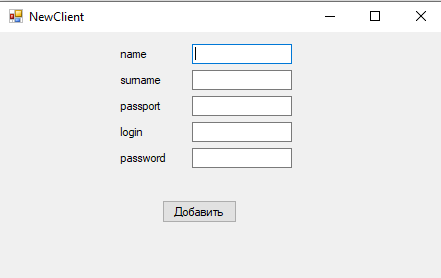
Может менять цену услуги, введя id услуги и новую стоимость:



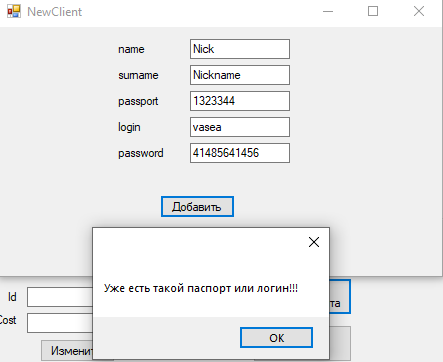
Если какое-либо поле не будет заполнено – ошибка:

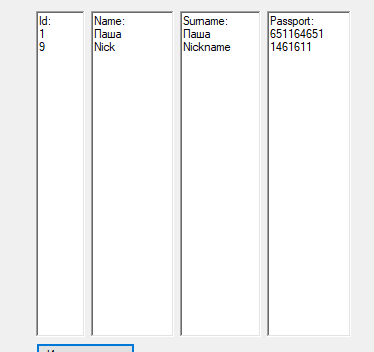


Можно добавлять новых клиентов:

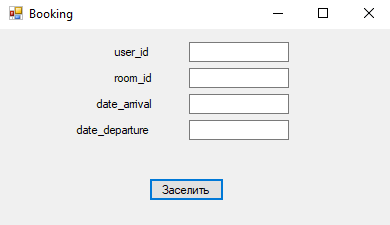


Тут работает обработчик ошибок, т.е. идет проверка логина и паспорта, они должны быть уникальными, иначе:

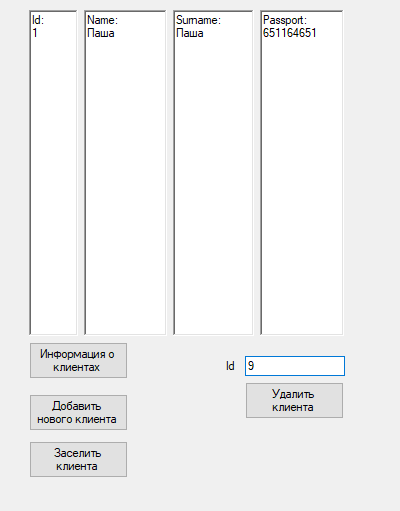




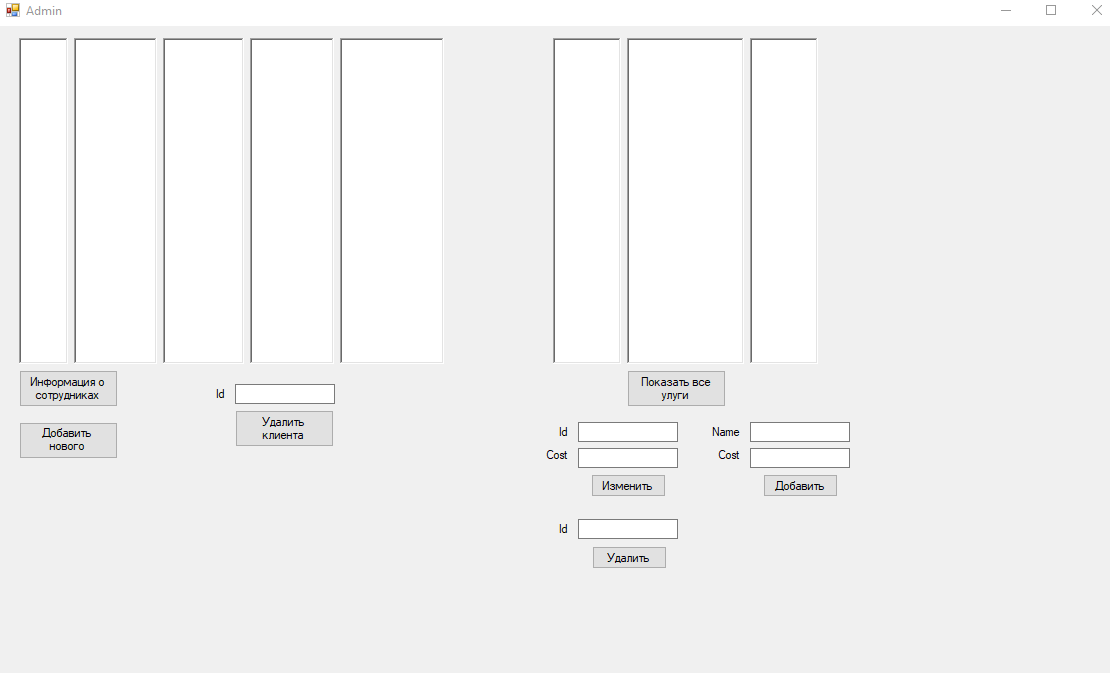
Можем заселить клиента:



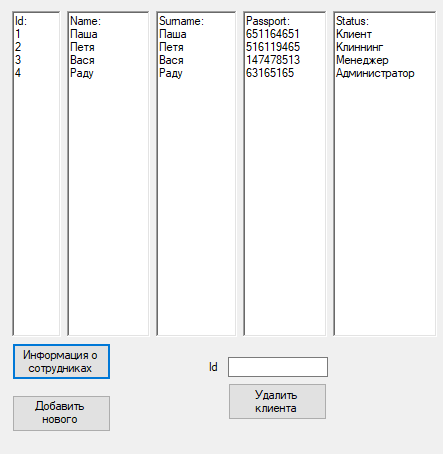
Удалить клиента:



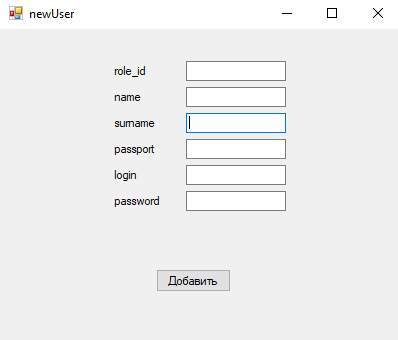
**Администратор:**



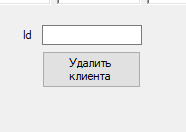
Посмотреть информацию о сотрудниках:



Добавить нового:



Уволить:



Показать/добавить/изменить/удалить услуги:

