

Содержание

1 Анализ предметной области. ERD-диаграмма.....	4
2 Анализ предметной области. Диаграмма прецедентов	5
3 Анализ предметной области. Диаграмма деятельности.....	6
4 Анализ предметной области. Физическая модель базы данных	7
5 Анализ предметной области. Словарь данных	9
6 Импорт данных.....	11
7 Проектирование модулей программного решения.....	14
8 Пользовательский интерфейс приложения.....	18
9 Функционал программного решения.....	22

					ЮУМК.09.02.07.200.03.00.ПЗ				
Изм.	Лист	№ документа	Подп	Дата	Отчет по производственной практике ПП.02	Лит.		Лист	Листов
Разраб.		Габдрафиков Р.Р.						3	25
Пров.		Брюханова Я.О.							
Утв.		Брюханова Я.О.				ИС-401			

1 Анализ предметной области. ERD-диаграмма

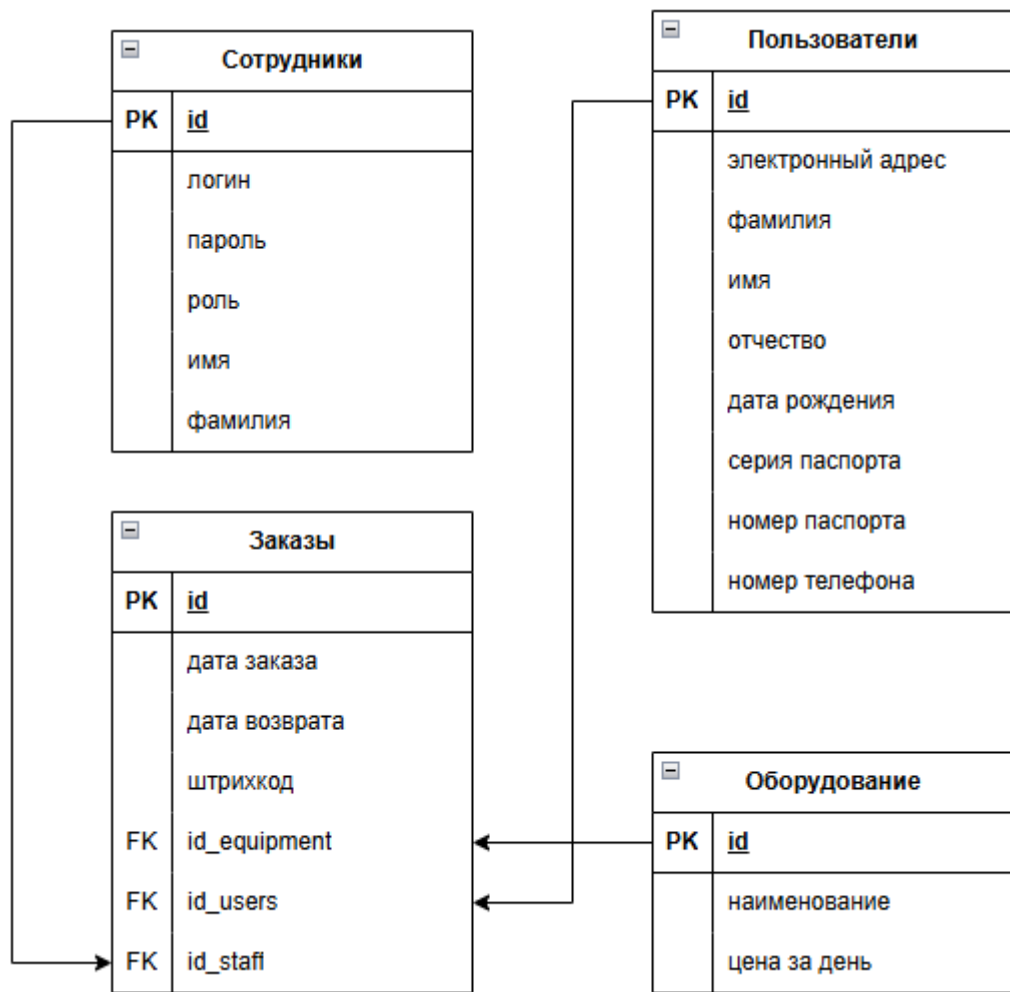


Рисунок 1 – Модель базы данных

В разработанной модели БД объектами являются таблицы:

- 1) «staff» (Сотрудники) – предназначена для хранения информации о сотрудниках.
- 2) «users» (Клиенты) – предназначена для хранения информации о клиентах.
- 3) «orders» (Заказы) – предназначена для хранения информации о заказах.
- 4) «equipment» (Услуги) – предназначена для хранения информации об услугах.

2 Анализ предметной области. Диаграмма прецедентов

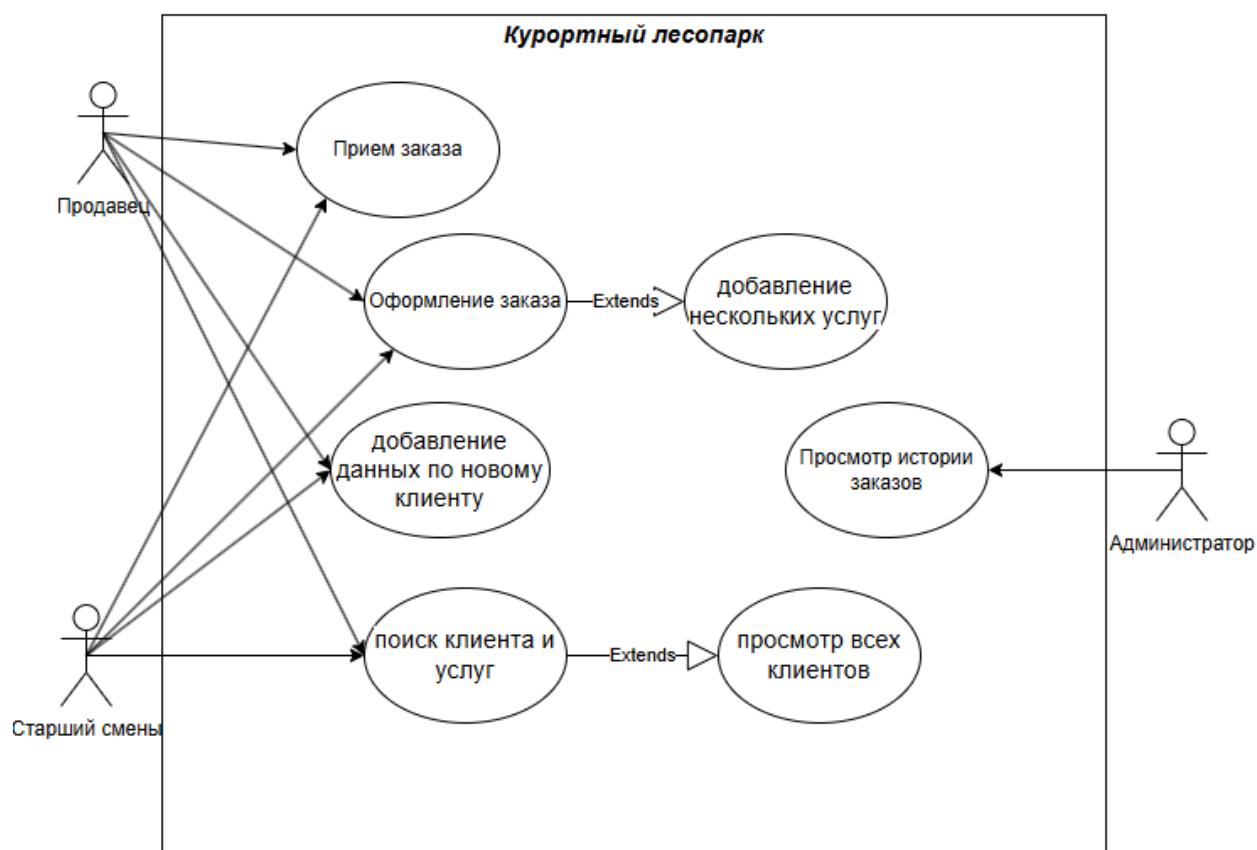


Рисунок 2 – Архитектура программного решения

С приложением могут работать следующие пользователи:

- 1) Администратор – имеет возможность просматривать историю заказов по точкам проката.
- 2) Старший смены – имеет возможность принимать заказы, оформлять заказы с добавлением нескольких услуг, добавлять, т.е. регистрировать нового клиента в системе и просматривать всех клиентов и услуг.
- 3) Продавец - имеет возможность принимать заказы, оформлять заказы с добавлением нескольких услуг, добавлять, т.е. регистрировать нового клиента в системе и просматривать всех клиентов и услуг.

3 Анализ предметной области. Диаграмма деятельности

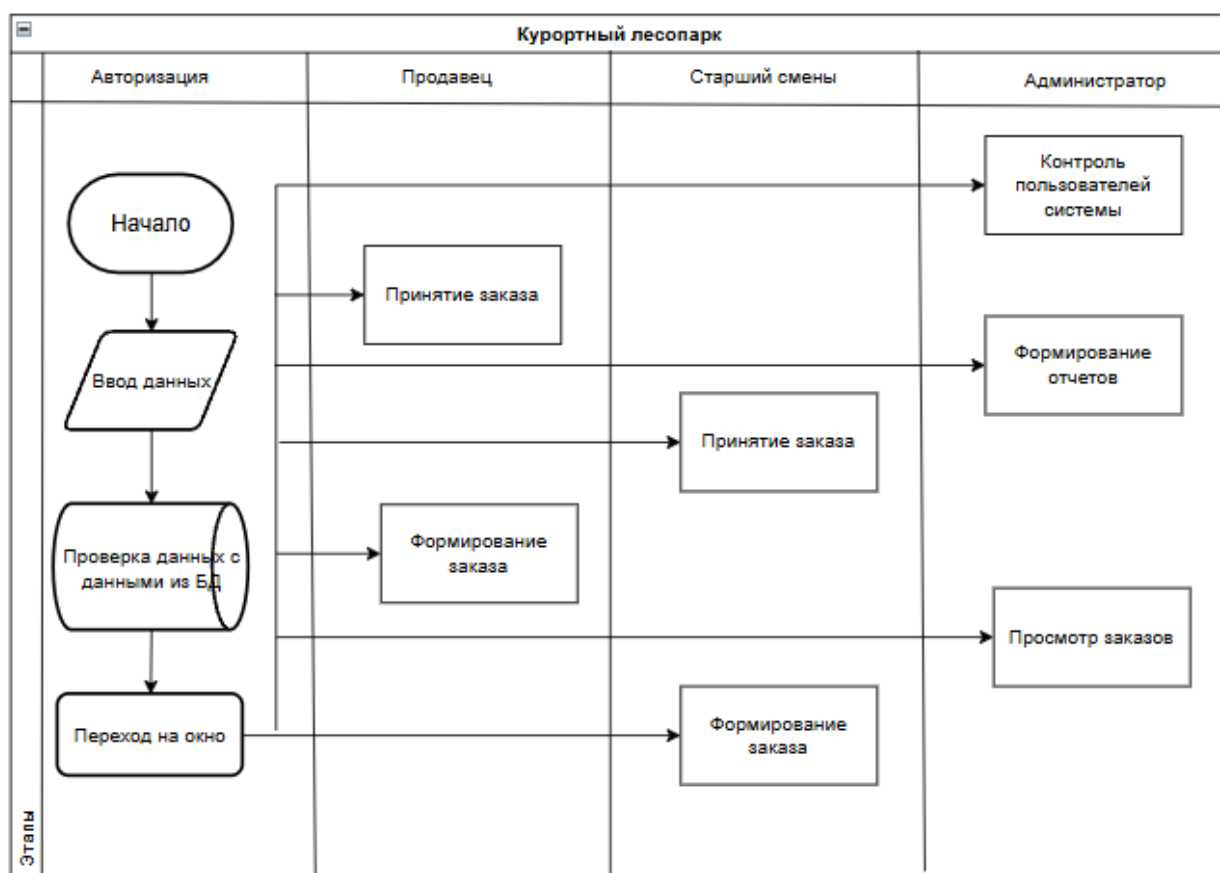


Рисунок 3 – Диаграмма деятельности

Диаграмма деятельности описывает процесс входа сотрудников в систему с учетом дифференцированного доступа. После успешного входа, в зависимости от уровня доступа, сотрудник получает доступ к одному из трех этапов.

4 Анализ предметной области. Физическая модель базы данных

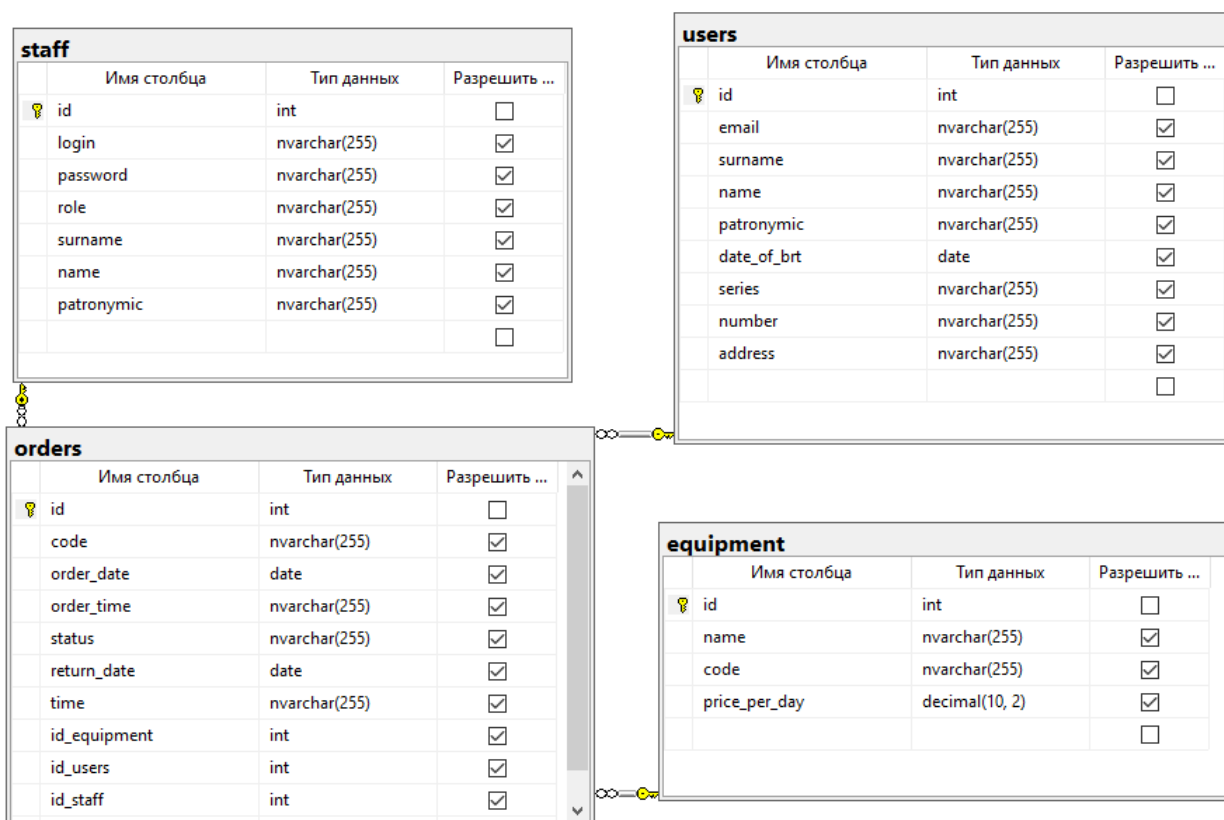


Рисунок 4 – Физическая модель БД

Сущность «orders» связана с сущностью «staff» типом связи «один-ко-многим», так как один сотрудник может регистрировать множество заказов.

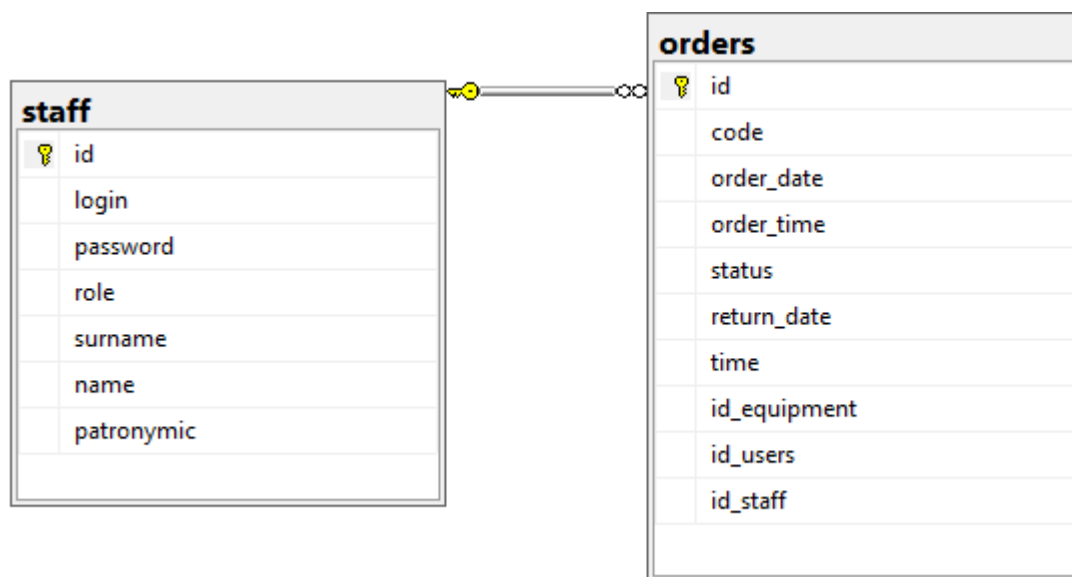


Рисунок 5 – Ассоциация «orders» и «staff»

Сущность «orders» связана с сущностью «users» типом связи «один-ко-многим», так как клиент заказывает множество заказов.

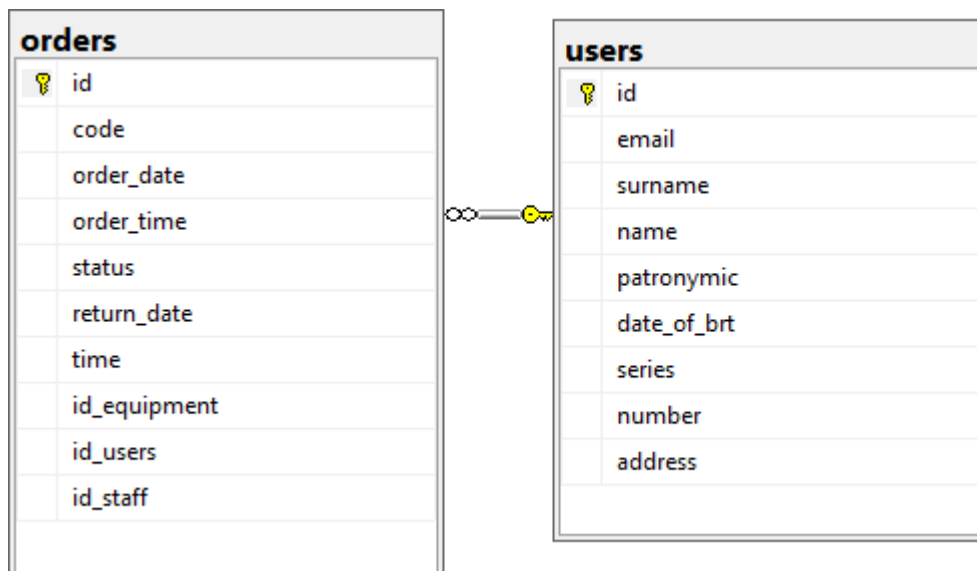


Рисунок 6 - Ассоциация «orders» и «users»

Сущность «orders» связана с сущностью «equipment» типом связи «один-ко-многим», так как в одном заказе может быть множество услуг.

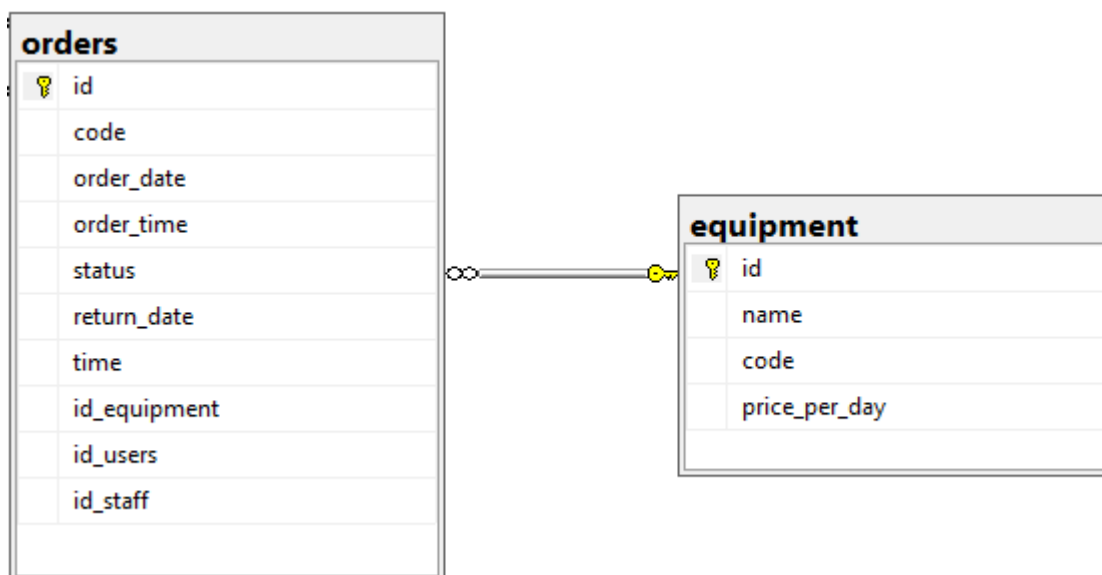


Рисунок 7 - Ассоциация «orders» и «equipment»

5 Анализ предметной области. Словарь данных

STAFF				
KEY	FIELD NAME	DATA TYPE/FIELD SIZE	REQUIRED	NOTES
PK	id	int	Y	
	login	nvarchar(255)	Y	
	password	nvarchar(255)	Y	
	role	nvarchar(255)	Y	роль сотрудника
	first_name	nvarchar(255)	Y	
	last_name	nvarchar(255)	Y	
USERS				
KEY	FIELD NAME	DATA TYPE/FIELD SIZE	REQUIRED	NOTES
PK	id	int	Y	
	email	nvarchar(255)	Y	
	surname	nvarchar(255)	Y	
	name	nvarchar(255)	Y	
	patronymic	nvarchar(255)	Y	
	date_of_brt	date	Y	
	series	nvarchar(255)	Y	
	number	nvarchar(255)	Y	
	number_phone	nvarchar(255)	Y	
	address	nvarchar(255)	Y	
EQUIPMENT				
KEY	FIELD NAME	DATA TYPE/FIELD SIZE	REQUIRED	NOTES
PK	id	int	Y	
	name	nvarchar(255)	Y	
	code	nvarchar(255)	Y	
	price_per_day	decimal(10,2)	Y	Стоимость проката инвентаря
ORDERS				
KEY	FIELD NAME	DATA TYPE/FIELD SIZE	REQUIRED	NOTES
PK	id	int	Y	
	code	nvarchar(255)	Y	
	order_date	date	Y	
	order_time	time(7)	Y	
	status	nvarchar(255)	Y	
	return_date	date	Y	
	time	nvarchar(255)	Y	
FK	id_equipment	int	Y	Foreign Key to оборудование
FK	id_users	int	Y	Foreign Key to пользователи
FK	id_staff	int	Y	Foreign Key to сотрудники

Рисунок 8 – Словарь данных

При проектировании словаря данных были задействованы основные типы данных, такие как «nvarchar» и «int».

Помимо вышеперечисленных типов данных были использованы:

- «date» - использовался для хранения даты в одном столбце;
- «decimal(10,2)» – использовался для хранения цифрового типа данных;
- «time» - использовался для хранения данных о времени.

					ЮУМК.09.02.07.200.03.00.ПЗ	Лист
						10
Изм.	Лист.	№ докум.	Подпись	Дата		

6 Импорт данных

Импорт данных в таблицу сотрудники.

Для импорта осуществили разделение информации о ФИО пользователя на отдельные таблицы при помощи функции «Текст по столбцам». Выберем необходимый столбец, затем ориентир, по которому будет осуществляться разделение в нашем случае это пробел.

Перед разделение необходимо подготовить пустые поля для наших новых столбцов.

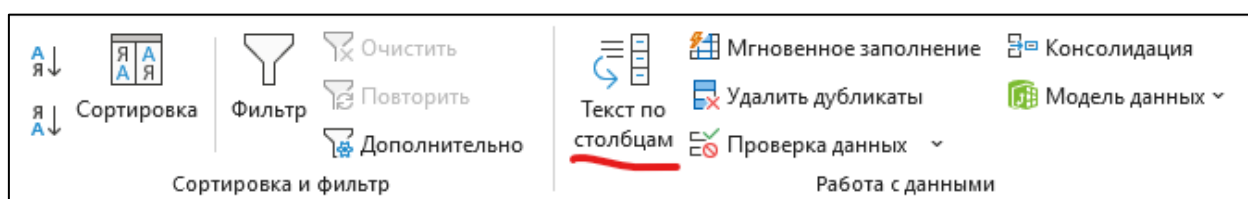


Рисунок 9 – Функция «Текст по столбцам»

ID	Login	Password	Role	Surname	Name	Patronymic
1	Ivanov@namecomp.ru	2L6KZG	Продавец	Иванов	Иван	Иванович
2	petrov@namecomp.ru	wWC67	Продавец	Петров	Петр	Петрович
3	fedorov@namecomp.ru	8ntwUp	Администратор	Федоров	Федор	Федорович
4	mirosov@namecomp.ru	YOytR	Старший смены	Миронов	Венеомин	Куприянович
5	shiryeu@namecomp.ru	RSbvHv	Старший смены	Ширяев	Ермолай	Венеоминович
6	ignatov@namecomp.ru	rwVDh9	Старший смены	Игнатов	Кассиан	Вансьевич
7	hohlov@namecomp.ru	LdNyos	Продавец	Хохлов	Владимир	Малсонович
8	strelkov@namecomp.ru	gynQMT	Продавец	Стрелков	Мстислав	Георгиевич
9	belyeva@namecomp.ru	AtnDjr	Продавец	Беллева	Лилия	Наумовна
10	smirnova@namecomp.ru	LIFRCZ	Продавец	Смирнова	Ульяна	Горлеевна

Рисунок 10 – Измененные данные сотрудников

В таблице пользователи было осуществлено разделение данных о ФИО, серии и номера паспорта и адреса проживания при помощи функции «Текст по столбцам».

1	lopezlis@hotmail.com	Абрамова	Александра	Мироновна	26.03.1999	1710	427875	410172, г. Санкт-Петербург, ул. Северная, 13, кв. 86
2	sethbishop@yahoo.com	Алексеев	Матвей	Викторович	02.08.1998	1452	339539	450375, г. Санкт-Петербург, ул. Клубная, 44, кв. 80
3	damaris61@okon.com	Борисова	Елена	Михайловна	23.05.1976	5086	666893	445685, г. Санкт-Петербург, ул. Зеленая, 7, кв. 47
4	yipraubaponou-5849@yopmail.com	Борисова	Ирина	Ивановна	14.10.1976	8755	921148	454311, г. Санкт-Петербург, ул. Новая, 19, кв. 78
5	boyd.koss@yahoo.com	Борисова	Тамара	Данииловна	29.05.1993	4658	621200	426083, г. Санкт-Петербург, ул. Механизаторов, 41, кв. 26
6	campbellkevin@gardner.com	Бочаров	Никита	Матвеевич	29.06.1997	1587	291249	125061, г. Санкт-Петербург, ул. Подгорная, 8, кв. 74
7	carsontamara@gmail.com	Васильева	Валерия	Дмитриевна	30.09.1999	1742	316556	614753, г. Санкт-Петербург, ул. Полевая, 35, кв. 39
8	grady.reilly@block.com	Васильева	Ксения	Константиновна	01.08.1999	4783	612567	660590, г. Санкт-Петербург, ул. Дачная, 37, кв. 70
9	anabelle07@schultz.info	Виноградов	Вячеслав	Дмитриевич	12.07.1976	2967	434531	410248, г. Санкт-Петербург, ул. Чкалова, 11, кв. 75
10	ziyeuddocrabri-4748@yopmail.com	Власов	Дмитрий	Александрович	17.08.1998	3455	719630	625283, г. Санкт-Петербург, ул. Победы, 46, кв. 7
11	jsteele@rojas-robinson.net	Воробьева	Василиса	Евгеньевна	13.01.1999	2460	169505	394060, г. Санкт-Петербург, ул. Фрунзе, 43, кв. 79
12	ron.treutel@quitzon.com	Герасимова	Дарья	Константиновна	13.10.1984	5493	684572	426629, г. Санкт-Петербург, ул. Весенняя, 32, кв. 46
13	connelly.makayla@yahoo.com	Голубев	Даниэль	Александрович	14.06.1999	3620	506034	450698, г. Санкт-Петербург, ул. Вокзальная, 14, кв. 37
14	deborah.christiansen@quigley.biz	Григорьев	Максим	Кириллович	26.05.1999	3560	491260	644133, г. Санкт-Петербург, ул. Гагарина, 28, кв. 69

Рисунок 11 - Измененные данные о клиентах

Таблица услуги остается без изменений.

ID	Наименование услуги	Код услуги	Стоимость, руб. за час
1	Аренда беседки	BSFBHV63	2000
2	Аренда лодки	DHBGFY563	700
3	Аренда шезлонга	HJBUJE21J	300
4	Инструктор по катанию на велосипеде	DJHGBS982	1000
5	Инструктор по катанию на лодке	638VVNQ3	1200
6	Инструктор по катанию на роликах	JHVSJF6	1000
7	Прокат велоколяски	OIJNB12	2000
8	Прокат велосипеда	8HFJHG443	1000
9	Прокат набора защитного оборудования	JUR8R	1000
10	Прокат роликов	JKFBJ09	1000
11	Прокат самоката	63748HF	700
12	Прокат шлема	87FDJKNJ	400
13	Прокат электросамоката	JFH7382	1200

Рисунок 12 – Данные об услугах

В таблице заказы время аренды было переведено в единую систему счисления при помощи функции «ЕСЛИ».

Были удалены названия единиц измерения при помощи функции «Найти и заменить».

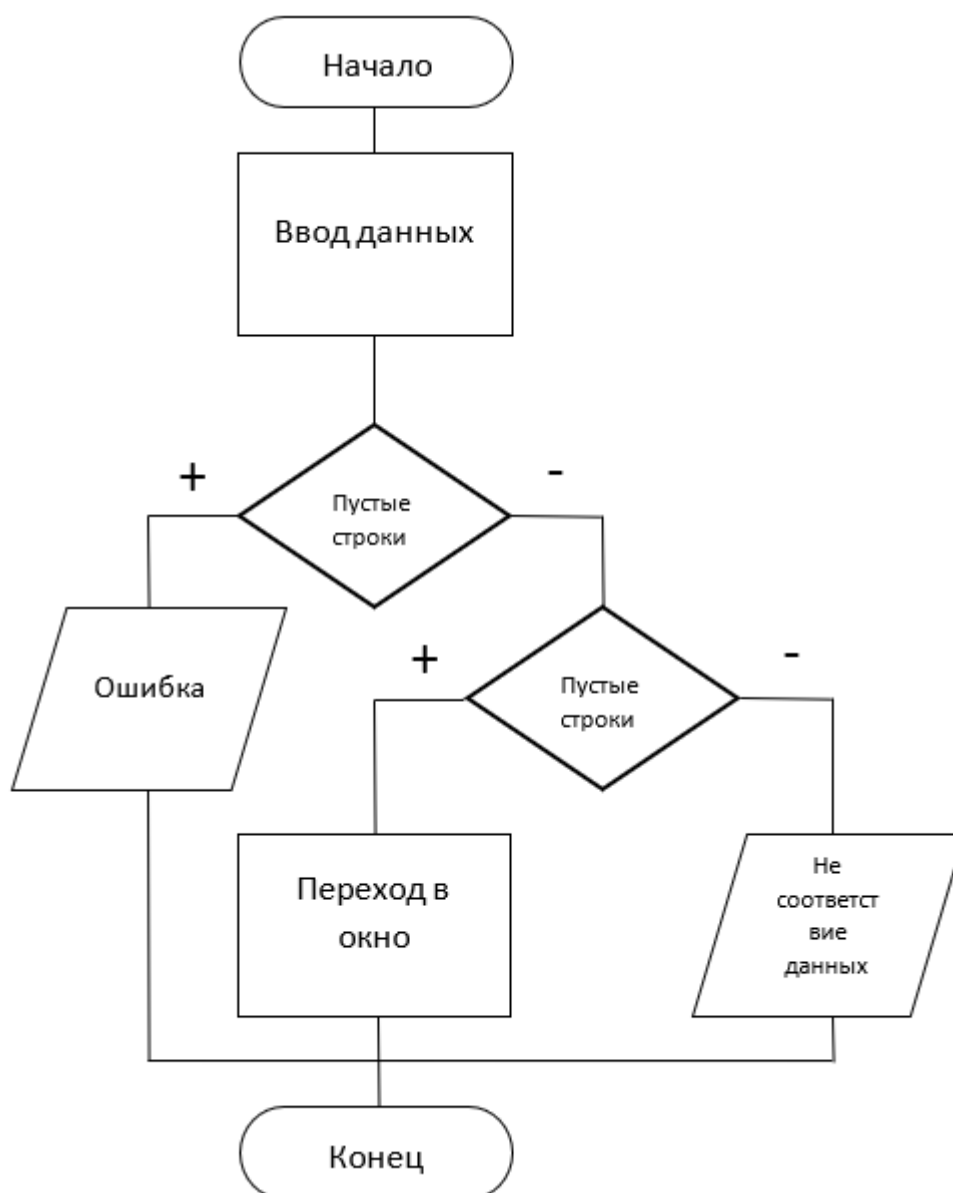
					ЮУМК.09.02.07.200.03.00.ПЗ	Лист
						12
Изм.	Лист.	№ докум.	Подпись	Дата		

ID	Код заказа	Дата создания	Время заказа	Статус	Дата закрытия	Время проката	Услуги	Код клиента	Сотрудники
1	1	12.03.2022	9:10	Новая		120	1	1	1
2	2	13.03.2022	10:10	В прокате		600	2	2	2
3	3	14.03.2022	11:10	В прокате		120	3	3	2
4	4	15.03.2022	12:10	Новая		600	4	4	4
5	5	16.03.2022	13:10	Закрыта	16.04.2022	320	5	5	5

Рисунок 13 – Измененные данные о заказах

Импорт в базу данных осуществлялся копированием данных из таблиц в excel и вставкой в пустые таблицы в БД.

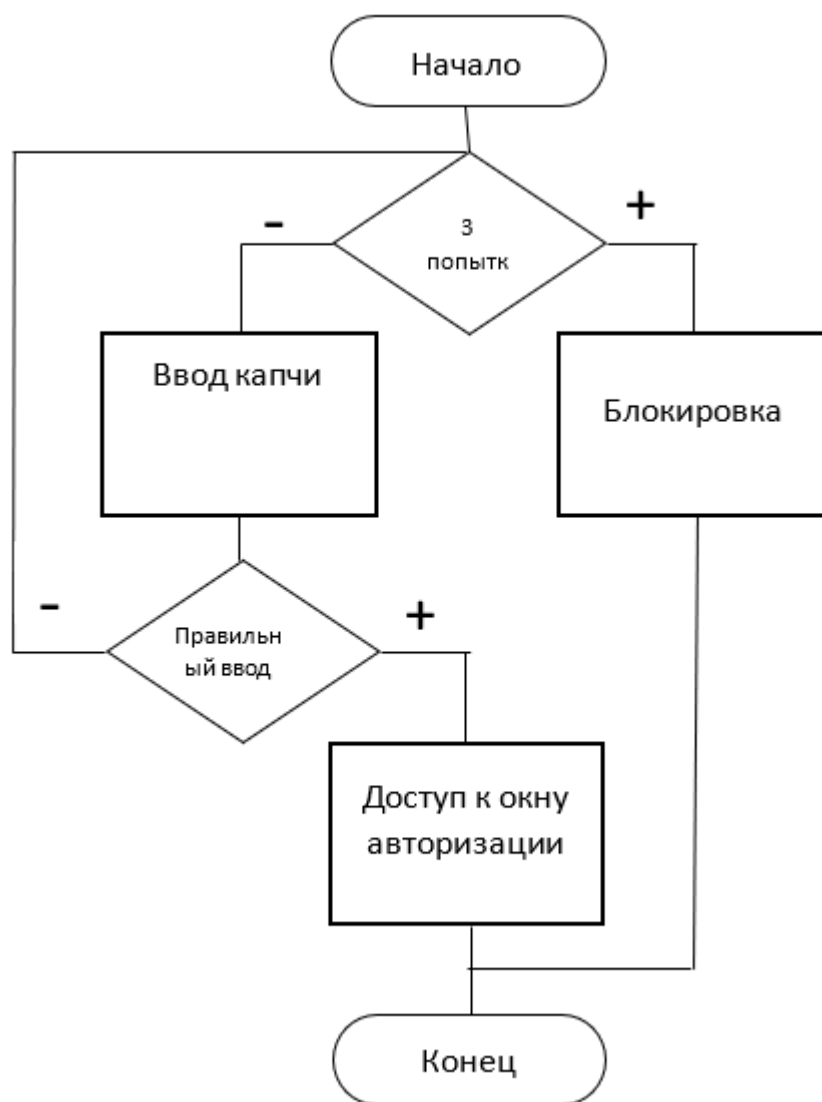
7 Проектирование модулей программного решения



Блок – схема 1 – Авторизация

Блок-схема 1 иллюстрирует процесс авторизации пользователя: при вводе данных система проверяет на наличие пустых полей. Если поля пусты, выводится сообщение об ошибке. В противном случае, система сверяет введенные данные с базой данных. При совпадении данных пользователь попадает в функциональное окно.

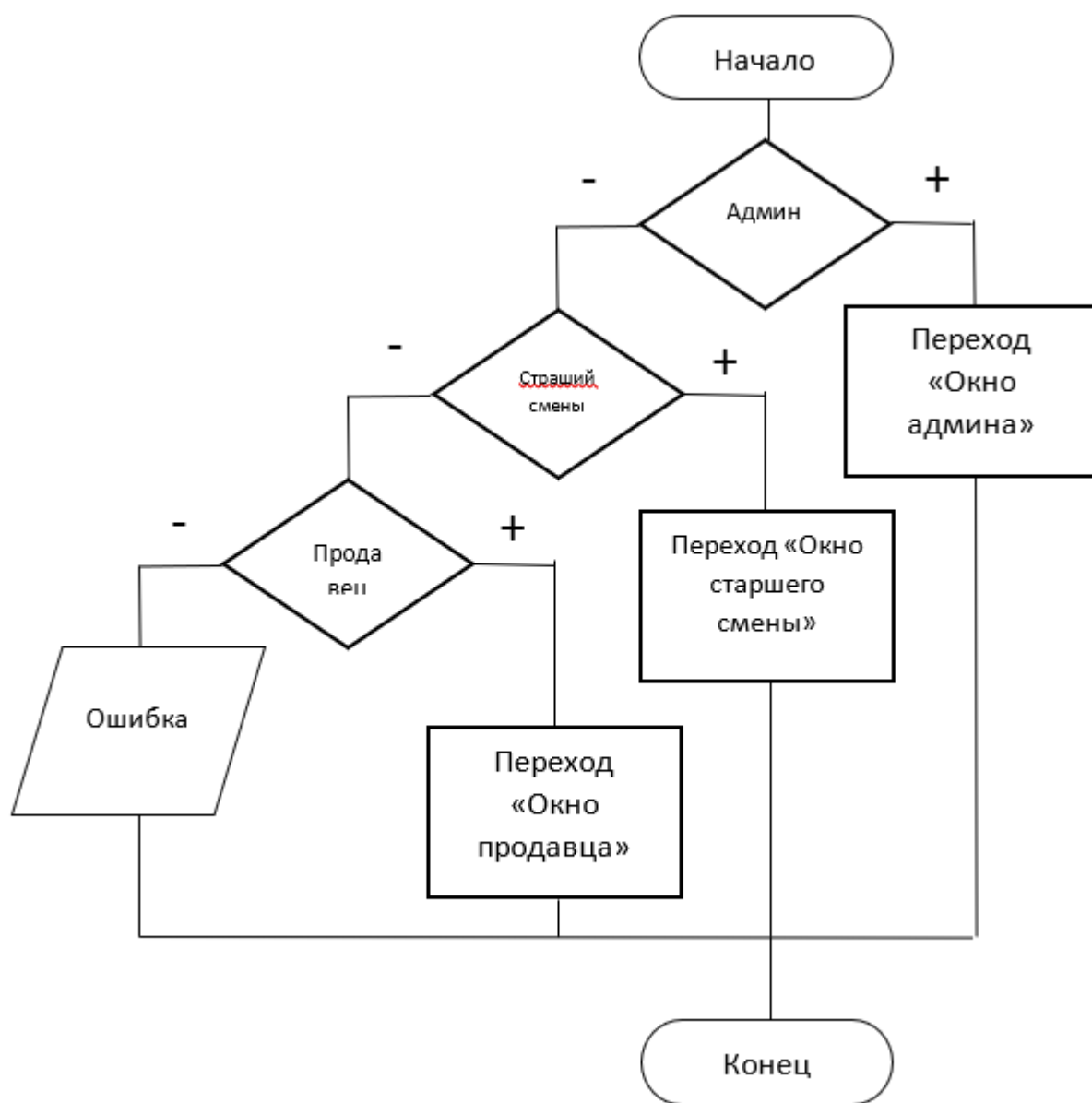
В случае несовпадения выводится сообщение об ошибке, и доступ в функциональное окно не предоставляется.



Блок – схема 2 – Капча

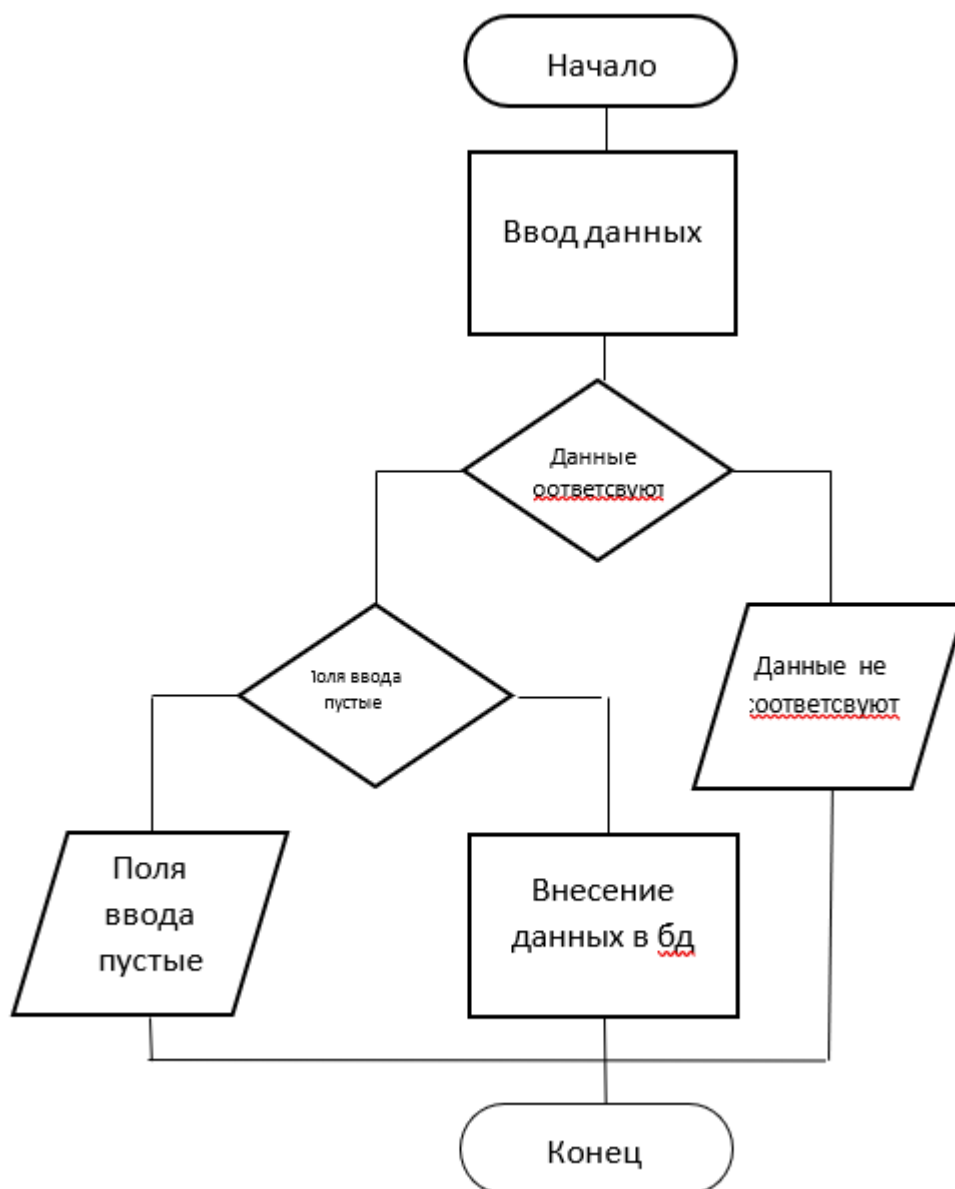
Блок-схема 2 описывает механизм отображения капчи после неудачной попытки авторизации (неверный логин или пароль). Пользователю предоставляется три попытки для ввода капчи. После трех неудачных попыток доступ к системе блокируется на определённый период.

Успешный ввод капчи позволяет пользователю повторить попытку ввода логина и пароля.



Блок – схема 3 – Разграничение прав доступа

Блок-схема 3 отображает механизм контроля доступа после успешной авторизации. Система проверяет роль пользователя, сопоставляя его логин и пароль с ролями, определёнными в системе. Если роль найдена, пользователь получает доступ к соответствующему меню и функциям, исключая доступ к функциям других ролей. В случае ошибки или отсутствия роли, пользователю выводится сообщение об ошибке, и доступ к функциональным возможностям не предоставляется.



Блок – схема 4 – Оформление заказа

Блок-схема 4 иллюстрирует процесс оформления заказа. Система проверяет введенные пользователем данные на полноту и соответствие требованиям. При обнаружении ошибок или пустых полей выводится сообщение об ошибке. В случае корректного ввода данных заказ оформляется успешно.

8 Пользовательский интерфейс приложения

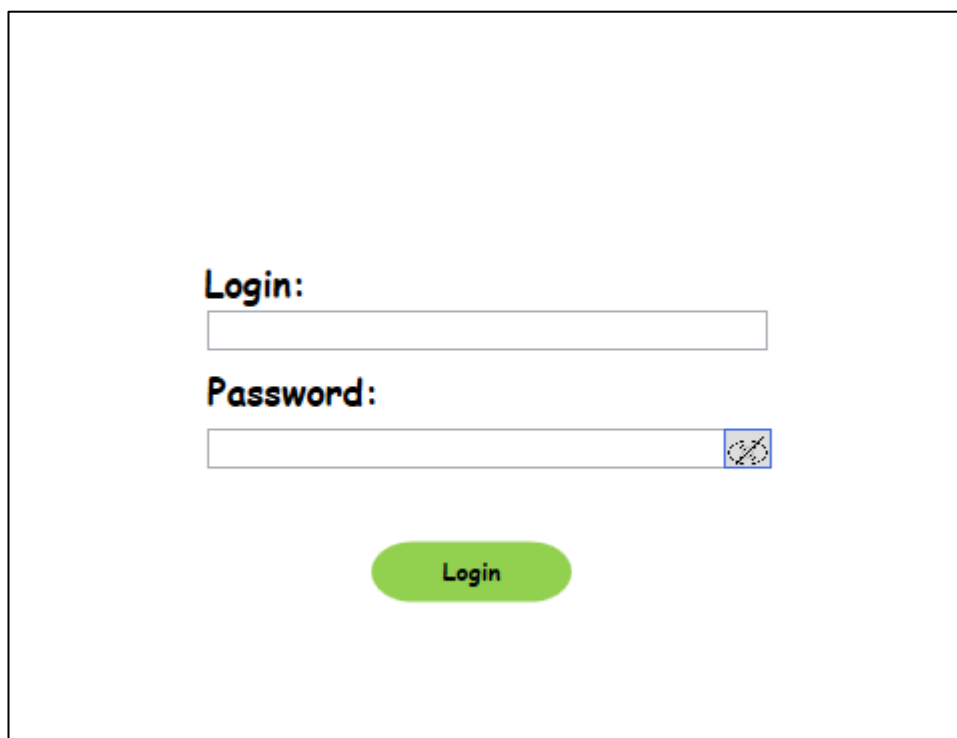


Рисунок 14 – Окно авторизации

На окне авторизации имеются текстовые поля для ввода логина и пароля, кнопки. Пароль скрыт маской ввода, если пользователь желает увидеть пароль, ему необходимо поставить галочку и пароль станет видимым. Кнопка «Login» отвечает за переход пользователя в меню. В случае если пароль будет введен неверно, то откроется капча для проверки пользователя/



Рисунок 15 – Окно капчи

На окне капчи расположено место с генерацией картинки и сочетания букв, текстовое поле для ввода капчи, кнопка «Проверить» для проверки капчи на правильность ввода и в случае правильности пользователя пересылает в окно авторизации и ему дается еще одна попытка для ввода логина или пароля. В случае если пользователь ввел капчу неверно, то ему дается еще 2 попытки, в общей сумме 3 попытки на ввод. Кнопка «Обновить» отвечает за создание нового сочетания картинки и слова.

Выход		00:00:07					
Код заказа	Дата проката	Время проката	Статус	Дата возврата	Время(минуты)	Услуги	К
1	3/12/2022 12:00:00 AM	9:10	Новая		120	Аренда беседки	A ^
2	3/13/2022 12:00:00 AM	10:10	В прокате		600	Аренда лодки	A
3	3/14/2022 12:00:00 AM	11:10	В прокате		120	Аренда шезлонга	Бс
4	3/15/2022 12:00:00 AM	12:10	Новая		600	Инструктор по катанию на велосипеде	Бс
5	3/16/2022 12:00:00 AM	13:10	Закрыта	4/16/2022 12:00:00 AM	320	Инструктор по катанию на лодке	Бс
6	3/17/2022 12:00:00 AM	14:10	Новая		480	Инструктор по катанию на роликах	Бс
7	3/18/2022 12:00:00 AM	15:10	Новая		240	Прокат велоколяски	Вс
8	3/19/2022 12:00:00 AM	16:10	В прокате		360	Прокат велосипеда	Вс
9	3/20/2022 12:00:00 AM	10:00	В прокате		720	Прокат набора защитного оборудования	Вс
10	3/21/2022 12:00:00 AM	11:00	Новая		120	Прокат роликов	Вс
11	3/22/2022 12:00:00 AM	12:00	Закрыта	3/22/2022 12:00:00 AM	600	Прокат самоката	Вс
12	3/23/2022 12:00:00 AM	13:00	В прокате		120	Прокат шлема	Гс
13	3/24/2022 12:00:00 AM	14:00	В прокате		600	Прокат электросамоката	Гс
14	3/25/2022 12:00:00 AM	15:00	В прокате		320	Аренда беседки	Гс
15	3/26/2022 12:00:00 AM	16:00	Закрыта	4/26/2022 12:00:00 AM	480	Аренда лодки	Гс
16	3/27/2022 12:00:00 AM	17:00	Новая		240	Аренда шезлонга	Дс
17	3/28/2022 12:00:00 AM	18:00	В прокате		360	Инструктор по катанию на велосипеде	Зс
18	3/29/2022 12:00:00 AM	19:00	В прокате		720	Инструктор по катанию на лодке	Зс

Рисунок 16 – Окно администратора

В окне администратора представлена таблица, в которой располагается информация о заказах и кнопка «Выход», отвечающая за переход пользователя в окно «Авторизации», а также таймер.

Выход

00:00:04

Код:

Дата заказа:

Выбор даты

15

Время заказа:

Статус:

Дата возврата:

Выбор даты

15

Время:

Услуги:

Клиент:

Регистрация

Сотрудник:

сделать заказ

Рисунок 17 – Окно продавца и старшего смены

В окне продавца и старшего смены располагается кнопка «Регистрация», отвечающая за добавление данных о клиентах (Рисунок 18). Также присутствуют поля для добавления нового заказа, кнопка «Выход» и таймер.

Назад

Фамилия:

Имя:

Отчество

Логин:

Дата рождения

Серия:

Номер:

Адрес:

Выбор даты 15

Сохранить

Рисунок 18 – Окно регистрации клиента

В окне регистрации клиента располагаются текстовые поля для ввода информации о клиенте и кнопка, которая осуществляет добавление данных и переход в окно с добавлением заказа.

					ЮУМК.09.02.07.200.03.00.ПЗ	Лист
						21
Изм.	Лист.	№ докум.	Подпись	Дата		

9 Функционал программного решения

```
private void Save_Click(object sender, RoutedEventArgs e)
{
    if (login_tb.Text != null && surname_tb.Text != null && name_tb.Text != null && patronymic_tb.Text != null &&
        dr_dp.SelectedDate != null && series_tb.Text != null && number_tb.Text != null && address_tb.Text != null)
    {
        db = new Entities.brEntities();
        Entities.users users = new Entities.users
        {
            email = login_tb.Text,
            surname = surname_tb.Text,
            name = name_tb.Text,
            patronymic = patronymic_tb.Text,
            date_of_brt = Convert.ToDateTime(dr_dp.SelectedDate),
            series = series_tb.Text,
            number = number_tb.Text,
            address = address_tb.Text,
        };
        db.users.Add(users);
        db.SaveChanges();
        Staff staff = new Staff();
        staff.Show();
        MessageBox.Show("Данные добавлены");
    }
    else { MessageBox.Show("Заполните данные о клиенте", "", MessageBoxButton.OK, MessageBoxImage.Warning); }
}
```

Рисунок 19 – Добавление нового клиента

Данный код обрабатывает событие нажатия кнопки "Сохранить". Он проверяет, заполнены ли все поля формы. Если все поля заполнены, данные сохраняются в базу данных. После успешного сохранения открывается форма «Staff» и выводится сообщение об успешном добавлении. Если поля не заполнены, выводится предупреждение.

					ЮУМК.09.02.07.200.03.00.ПЗ	Лист
						22
Изм.	Лист.	№ докум.	Подпись	Дата		

```

private void save_bt_Click(object sender, RoutedEventArgs e)
{
    if (code_tb.Text != null && order_dp.SelectedDate != null &&
        order_time.Text != null && status_cb.Text != null &&
        return_dp.SelectedDate != null && time_tb.Text != null &&
        equipmnets_cb.Text != null && clients_cb.Text != null && StaFF_cb.Text != null)
    {
        db = new Entities.brEntities();
        Entities.orders orders = new Entities.orders()
        {
            code = code_tb.Text,
            order_date = Convert.ToDateTime(order_dp.SelectedDate),
            order_time = order_time.Text,
            status = status_cb.Text,
            return_date = Convert.ToDateTime(return_dp.SelectedDate),
            time = time_tb.Text,
            id_equipment = Convert.ToInt32(equipmnets_cb.SelectedIndex + 1),
            id_users = Convert.ToInt32(clients_cb.SelectedIndex + 1),
            id_staff = Convert.ToInt32(StaFF_cb.SelectedIndex + 1),
        };
        db.orders.Add(orders);
        db.SaveChanges();
        MessageBox.Show("Заказ добавлен!!! ", "", MessageBoxButton.OK, MessageBoxImage.Information);
    }
    else { MessageBox.Show("Заполните данные ", "", MessageBoxButton.OK, MessageBoxImage.Warning); }
}

```

Рисунок 20 – Добавление заказа

Данный код обрабатывает нажатие кнопки «Сохранить». Он проверяет, заполнены ли все поля формы. Если все поля заполнены, данные о заказе сохраняются в базу данных. После сохранения выводится сообщение об успехе. Если поля не заполнены, выводится предупреждающее сообщение.

					ЮУМК.09.02.07.200.03.00.ПЗ	Лист
						23
Изм.	Лист.	№ докум.	Подпись	Дата		

```

Ссылка: 1
private void Button_Click(object sender, RoutedEventArgs e)
{
    db = new Entities.brEntities();
    string email = emailTextBox.Text;
    string password = passwordBox.Password;
    if (email == "" && password == "")
    {
        MessageBox.Show("Введите свои данные");
    }
    else
    {
        if (db.staff.Any(x => x.login != email && x.password != password))
        {
            if (db.staff.Any(x => x.login == email && x.password == password))
            {
                if (db.staff.Any(x => x.role == "Администратор" && x.login == email))
                {
                    Admin admin = new Admin();
                    admin.Show();
                    admin.WindowState = this.WindowState;
                    this.Close();
                }
                else if (db.staff.Any(x => x.role == "Старший смены" && x.login == email))
                {
                    Staff staff = new Staff();
                    this.Close();
                    staff.Show();
                }
                else if (db.staff.Any(x => x.role == "Продавец" && x.login == email))
                {
                    Staff staff = new Staff();
                    staff.Show();
                    staff.WindowState = this.WindowState;
                    this.Close();
                }
            }
        }
        else
        {
            MessageBox.Show("Неверный логин или пароль");
            fail++;
            if (fail == 3)
            {
                Captcha captcha = new Captcha();
                captcha.Show();
                this.Close();
                fail = 0;
            }
        }
    }
}
}

```

Рисунок 21 – Разграничение прав доступа

Данный код обрабатывает вход пользователя. Он проверяет поля логин и пароль. Если оба пусты, выводится сообщение об ошибке. В противном случае, он проверяет наличие пользователя с введенными данными в базе

					ЮУМК.09.02.07.200.03.00.ПЗ	Лист
						24
Изм.	Лист.	№ докум.	Подпись	Дата		

данных. Если пользователь найден, открывается форма «Admin», «Staff» (в зависимости от роли пользователя), а текущая форма закрывается. Если данные неверны, выводится сообщение об ошибке. После трех неудачных попыток открывается форма «Captcha».

```
private void GenerateCaptcha()
{
    // Генерация случайной капчи
    Random random = new Random();
    generatedCaptcha = "";
    int power = 4;
    for (int i = 0; i < power; i++)
    {
        generatedCaptcha += (char)('A' + random.Next(0, 32));
    }

    CaptchaTextBlock.Text = generatedCaptcha;

    // Генерация случайного текста для капчи

    // Отображение текста капчи
    Label captchaLabel = new Label();
    captchaLabel.Content = captchaText;
    captchaLabel.FontSize = 20;
    captchaLabel.Margin = new Thickness(10);
    captchaLabel.Foreground = Brushes.Black;

    // Добавление шума и линий к капче
    DrawingVisual visual = new DrawingVisual();
    using (DrawingContext context = visual.RenderOpen())
    {
        for (int i = 0; i < 100; i++)
        {
            context.DrawEllipse(Brushes.Black, null, new Point(random.Next(0, 200), random.Next(0, 50)), 1, 1);
            context.DrawEllipse(Brushes.Black, null, new Point(random.Next(0, 200), random.Next(0, 50)), 2, 2);
        }
        context.DrawLine(new Pen(Brushes.Black, 2), new Point(0, random.Next(0, 30)), new Point(200, random.Next(0, 50)));
        context.DrawLine(new Pen(Brushes.Black, 2), new Point(0, random.Next(0, 30)), new Point(200, random.Next(0, 50)));
    }

    VisualBrush visualBrush = new VisualBrush(visual);
    CaptchaTextBlock.Background = visualBrush;
}
```

Рисунок 22 – Генерация капчи

Этот код генерирует и отображает случайную капчу. Сначала генерируется 4-символьная строка из кириллических букв. Затем создается визуальный элемент, на котором рисуются случайные точки и линии, создавая эффект шума. Наконец, этот визуальный элемент устанавливается в качестве фона, где отображается сгенерированная текстовая капча.

					ЮУМК.09.02.07.200.03.00.ПЗ	Лист
						25
Изм.	Лист.	№ докум.	Подпись	Дата		

```

// Ссылка 1
private void Check_Click(object sender, RoutedEventArgs e)
{
    // Проверка введенной пользователем капчи
    if (InputTextBox.Text == generatedCaptcha)
    {
        MessageBox.Show("Капча введена правильно!");
        MainWindow mainWindow = new MainWindow();
        mainWindow.Show();
        this.Close();
        // Переход на нужное окно
    }
    else
    {
        attempts++;
        if (attempts == 3)
        {
            MessageBox.Show("Слишком много неверных попыток. Попробуйте снова через 10 секунд.");
            InputTextBox.IsEnabled = false;
            Refresh.IsEnabled = false;
            Check.IsEnabled = false;
            System.Threading.Thread.Sleep(10000);
            GenerateCaptcha();
            InputTextBox.IsEnabled = true;
            Refresh.IsEnabled = true;
            Check.IsEnabled = true;
            attempts = 0;
        }
        else
        {
            MessageBox.Show(" Попробуйте снова.");
            GenerateCaptcha();
        }
    }
}
}

```

Рисунок 23 – Проверка введенной капчи

Этот код представляет собой обработчик события нажатия кнопки «Check», который проверяет введенный пользователем текст капчи.

Проверка капчи: если текст совпадает с сгенерированной капчей, выводится сообщение о правильности ввода, создается и открывается новое окно, а текущее окно закрывается.

Неверный ввод: если введенная капча неверна, увеличивается счётчик попыток. Если количество попыток достигает 3, пользователю показывается сообщение о превышении лимита попыток. Поля ввода и кнопки отключаются на 10 секунд, после чего капча обновляется, и элементы управления снова становятся доступными. Если попыток меньше трёх, выводится сообщение о необходимости повторной попытки, и капча генерируется заново. Таким

образом, код реализует механизм проверки капчи с ограничением на количество попыток.

