

Министерство науки и высшего образования Российской Федерации
Федерального государственного бюджетного образовательного учреждения высшего образования
Московский государственный технический университет имени Н. Э. Баумана
(Национальный исследовательский университет)
Московский техникум космического приборостроения

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ

по теме «Учет платы за отопление»

Специальность: 09.02.07_01 Информационные системы и
программирование – квалификация «Программист»

Группа ТИП-41

Проверил	Филиппов В.В.
	Осипова Н.М.
	Митрошенкова Е.А.
Разработал	Леонова Е.А.

СОДЕРЖАНИЕ

1 Анализ предметной области	3
2 Проектирование Базы Данных.....	6
3 Проектирование шаблона выходного документа	16
4 Проектирование интерфейса приложения.....	17
5 Тестирование разработанного приложения	26
6 Листинг программы	49
6.1 Листинг файла main.py	49
6.2 Листинг файла about.html.....	69
6.3 Листинг файла about_client.html.....	70
6.4 Листинг файла about_company.html.....	71
6.5 Листинг файла add_pokaz.html	72
6.6 Листинг файла base.html	73
6.7 Листинг файла change_info_client.html	74
6.8 Листинг файла change_info_company.html	76
6.9 Листинг файла change_pokaz.html	78
6.10 Листинг файла client.html.....	79
6.11 Листинг файла company.html.....	80
6.12 Листинг файла delete_client.html	81
6.13 Листинг файла info_client.html	82
6.14 Листинг файла itog.html	83
6.15 Листинг файла new_client.html	84
6.16 Листинг файла new_flat.html	86
6.17 Листинг файла new_schetchik.html.....	88
6.18 Листинг файла schetchik.html	89
6.19 Листинг файла style.css	90

1 Анализ предметной области

Учет платы за отопление – сайт, который помогает рассчитать стоимость оплаты за отопление. Плата взымается с каждой квартиры в зависимости от ее площади.

Входными данными являются: площадь квартиры, температура (которая поддерживалась в помещении) и тариф, согласно которому производится расчет стоимости услуг. Тариф руководствуется прайс-листом, а благодаря условиям учреждения предоставляется услуга (она производится благодаря разработанному сайту, который взаимодействует с БД). Там производятся все расчеты, работник учреждения обязан проводить проверку заполненного документа. Пользователь получает готовый чек для платы за отопление.

В отчете прилагаются рис.1.1 и рис.1.2 на которых отображены контекстная диаграмма и развернутая контекстная диаграмма соответственно. На них можно наглядно увидеть принцип работы сайта.



Рисунок 1.1 - Контекстная диаграмма



Рисунок 1.2 - Развёрнутая контекстная диаграмма

На сайте вводятся данные о температуре (на схеме Температура), после чего производится проверка: меньше или больше 18 градусов температура, согласно условиям учреждения. После по проверенной температуре, тарифу и площади квартиры производится расчет стоимости оплаты (согласно прайс-листву). Далее в готовый шаблон вводится стоимость оплаты, и информация о оплате и на выходе получается готовый к печати чек. Он проверяется работником учреждения, если все данные корректны, то пользователь получает готовый документ (чек) на оплату.

Ниже представлена схема DFD, она поможет лучше понять принцип работы:

Из блока «Квартира» берутся данные для блоков «Наличие счетчика» (учитывается информация о счетчиках), «Расчет стоимости» (площадь) и «Проверка температуры» (температура). После происходит проверка температуры и подбор тарифа (информация о нем берется из блока «Компания»), далее данные и информация о счетчиках обрабатываются и вводятся. Если всё корректно, то происходит занесение в БД и последующая

работа с ними. В случае, если данные не корректны происходит отказ в составлении чека.

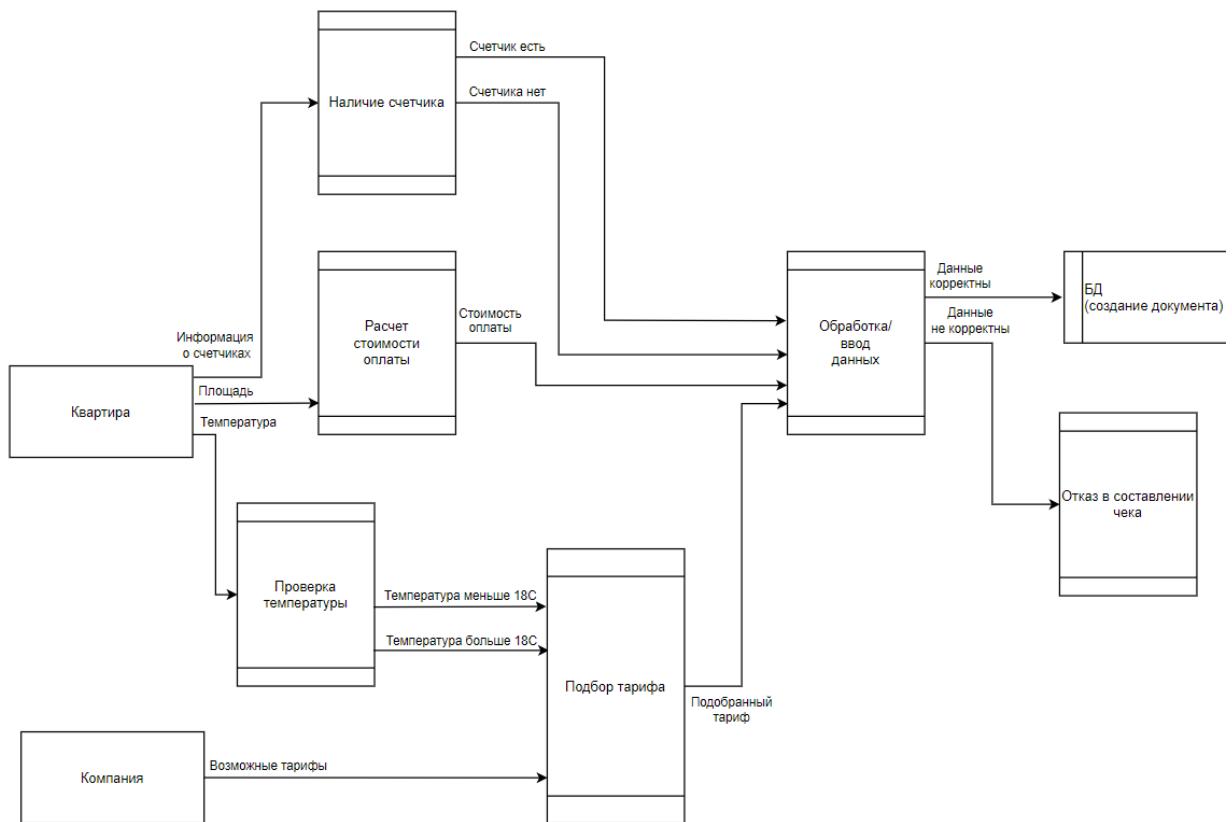


Рисунок 1.3 - Диаграмма DFD

Главной задачей сайта «Учет платы за отопление» является предоставление услуги составления платежного документа. Сайт должен:

- Иметь возможность создания документа расчета стоимости отопления. Документ должен заполняться автоматически
- Уметь работать с информацией о клиентах (добавлять, изменять, удалять)
- Уметь работать с информацией о компаниях (изменять)
- Уметь работать с показаниями на счетчиках (добавлять, изменять)
- Предоставлять информацию о клиенте, компаниях
- Заносить введенную информацию в БД

2 Проектирование Базы Данных

Проектирование базы данных – процесс создания необходимых схем (концептуальной и логической).

Для их создания необходимо вынести главные сущности:

- Company (информация о компаниях: название, тариф, реквизиты, ФИО руководителя)
- Sobstv (информация о собственниках: дата рождения, телефон, ФИО, серия и номер паспорта, пол)
- Flat (информация о квартирах: адрес, площадь, средняя температура)
- Lgot (информация о льготах: скидка)
- Schetchik (информация о счетчиках: дата установки, фирма счетчика)
- Pokaz (информация о показаниях счетчика: показания счетчика, месяц показаний)
- Pay (информация в чеке: дата оплаты, сумма оплаты)

Это можно рассмотреть на концептуальной схеме (рис.2.1).

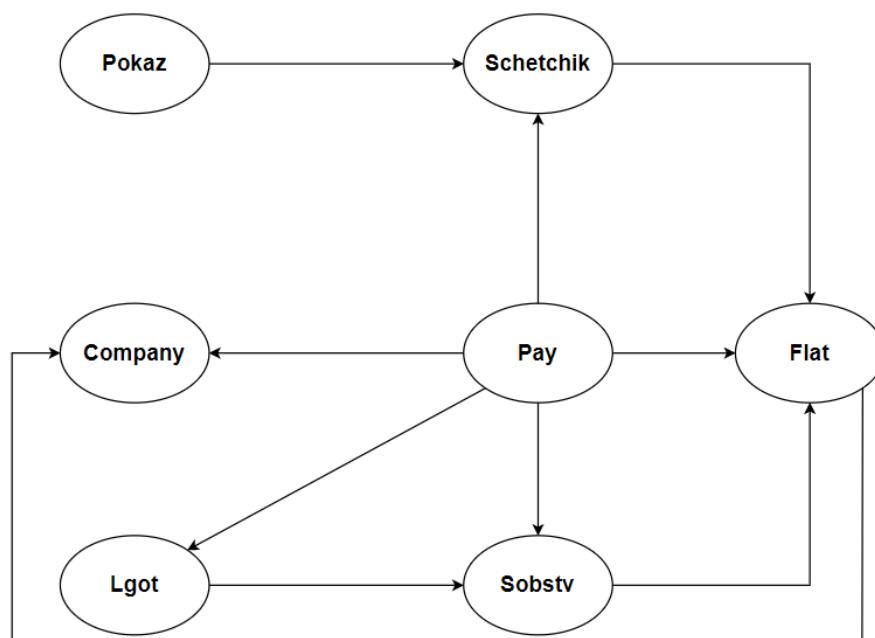


Рисунок 2.1 - Концептуальная схема

На рис.2.2 (Логическая схема) можно подробно рассмотреть связи между сущностями. Они связаны между собой связью один ко многим:

- От Company.ID_company к Pay.ID_company
- От Company.ID_company к Flat.ID_company
- От Sobstv.ID_sobstv к Lgot.ID_sobstv
- От Sobstv.ID_sobstv к Pay.ID_sobstv
- От Flat.ID_flat к Pay.ID_flat
- От Flat.ID_flat к Sobstv.ID_flat
- От Flat.ID_flat к Schetchik.ID_flat
- От Lgot.ID_lgot к Pay. ID_lfot
- От Schetchik.ID_schetchik к Pay.ID_schetchik
- От Schetchik.ID_schetchik к Pokaz.ID_schetchik

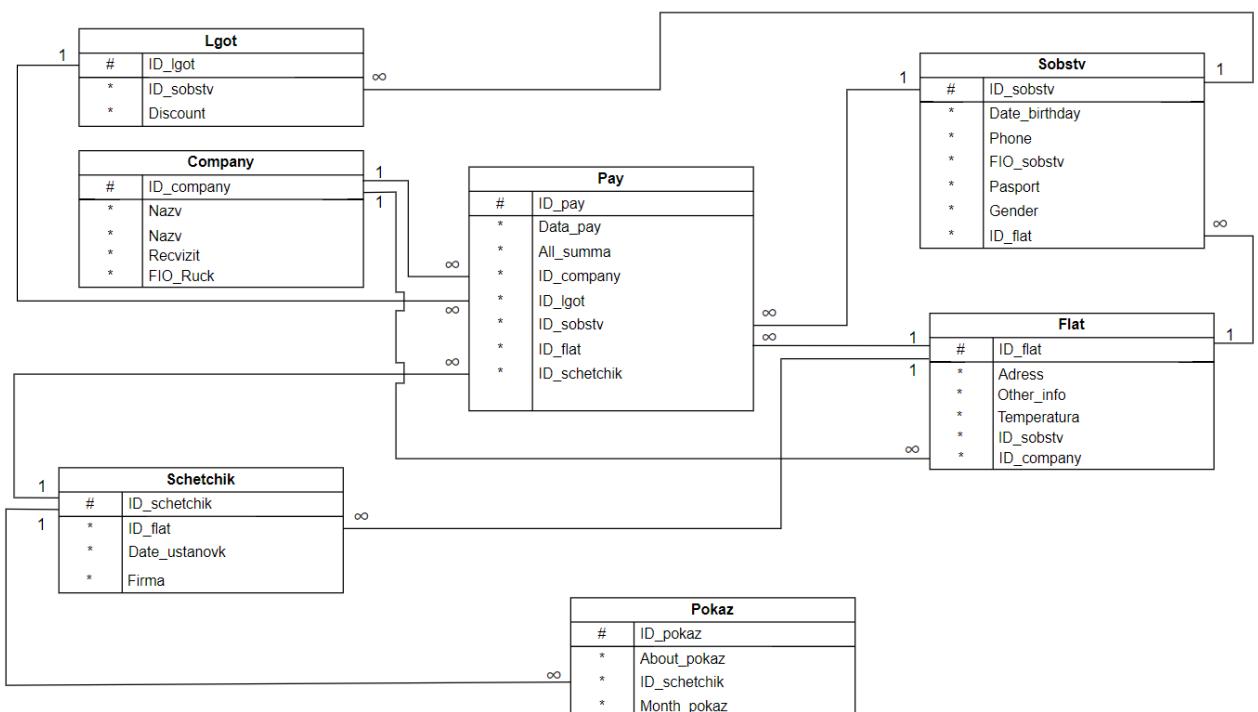


Рисунок 2.2 - Логическая схема

После Концептуальной и Логической схем необходимо создать саму БД в MySQL Server, с которой в последующем будет взаимодействовать сайт (в некоторых таблицах был добавлен столбец ID, не влияющий на работоспособность базы данных, но необходимый для работы сайта).

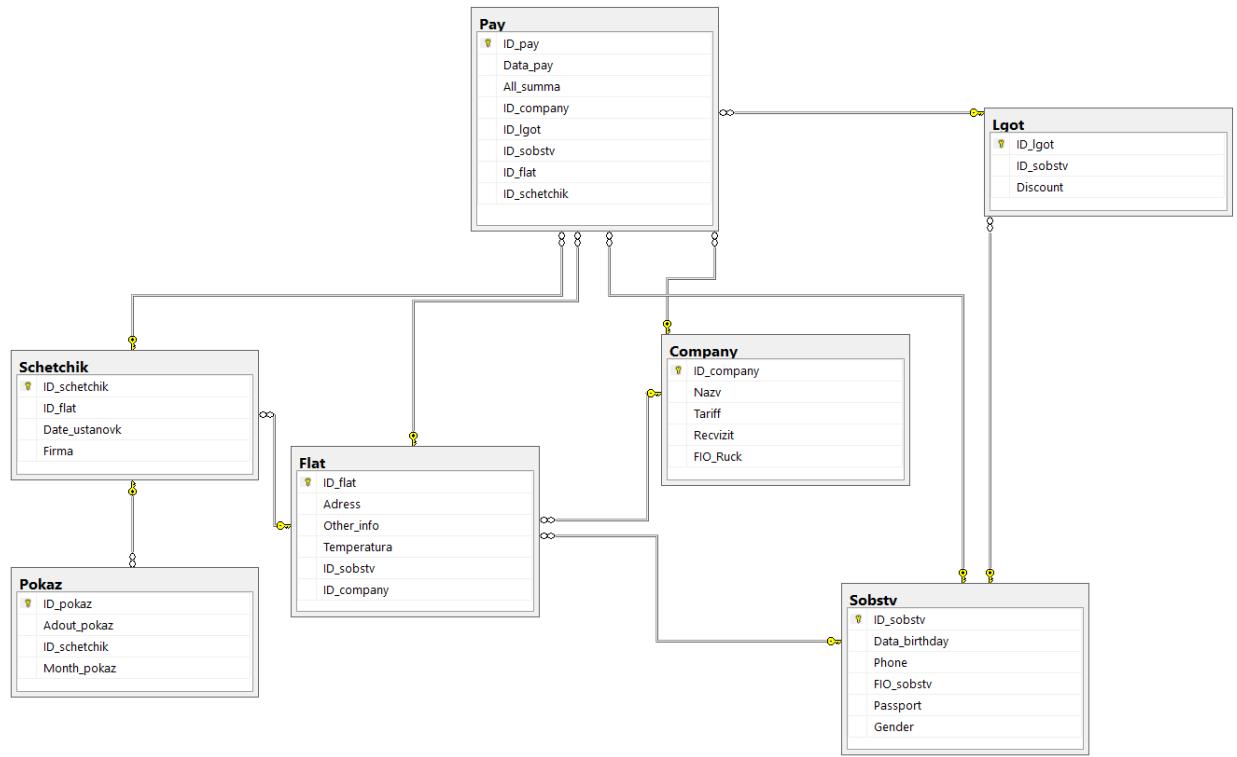


Рисунок 2.3 - Схема готовой базы данных в MySQL Server

На рис.2.3 можно увидеть взаимодействие таблиц в MySQL Server.

Для более полного понятия полей таблиц можно обратить внимание на описание полей в таблица 2.1 – таблица 2.7

Таблица 2.1 - Описание таблицы Sobstv

Имя поля	Тип поля	Описание поля
ID_sobstv	INT	ID собственника квартиры
Data_birthday	DATE	Дата рождения собственника
Phone	VARCHAR(12)	Номер телефона собственника
FIO_sobstv	VARCHAR(40)	ФИО собственника
Passport	VARCHAR(10)	Паспротные данные собственника (серия и номер паспорта)
Gender	VARCHAR(1)	Пол собственника

Таблица 2.2 - Описание таблицы Company

Имя поля	Тип поля	Описание поля
ID_company	INT	ID компании
Nazv	VARCHAR(20)	Название компании
Tariff	INT	Тариф компании (за 1 кв.м.)
Recvizit	VARCHAR(30)	Реквизиты компании (для оплаты)
FIO_Ruck	VARCHAR(40)	ФИО руководителя компании

Таблица 2.3 - Описание таблицы Flat

Имя поля	Тип поля	Описание поля
ID_flat	INT	ID квартиры
Adress	VARCHAR(40)	Адрес расположения квартиры
Other_info	VARCHAR(40)	Площадь квартиры
Temperatura	VARCHAR(3)	Средняя температура в квартире
ID_sobstv	INT	ID собственника (внешний ключ для связи с таблицей Sobstv)
ID_company	INT	ID компании (внешний ключ для связи с таблицей Company)

Таблица 2.4 - Описание таблицы Schetchik

Имя поля	Тип поля	Описание поля
ID_schetchik	INT	ID счетчика
ID_flat	INT	ID квартиры (внешний ключ для связи с таблицей Flat)
Date_ustanovk	DATE	Дата установки счетчика в квартире
Firma	VARCHAR(25)	Фирма, устанавливающая счетчик

Таблица 2.5 - Описание таблицы Lgot

Имя поля	Тип поля	Описание поля
ID_lgot	INT	ID льготы
ID_sobstv	INT	ID собственника (внешний ключ для связи с таблицей Sobstv)
Discount	VARCHAR(4)	Процент скидки (10%)

Таблица 2.6 - Описание таблицы Pokaz

Имя поля	Тип поля	Описание поля
ID_pokaz	INT	ID показаний счетчика
Adout_pokaz	VARCHAR(10)	Показания счетчика
ID_schetchik	INT	ID счетчика (внешний ключ для связи с таблицей Schetchik)
Month_pokaz	VARCHAR(8)	Месяц показаний счетчика

Таблица 2.7 - Описание таблицы Pay

Имя поля	Тип поля	Описание поля
ID_pay	INT	ID квитанции оплаты (чека)
Data_pay	DATE	Дата оплаты
All_summa	VARCHAR(10)	Сумма оплаты
ID_company	INT	ID компании (внешний ключ для связи с таблицей Company)
ID_lgot	INT	ID льготы (внешний ключ для связи с таблицей Lgot)
ID_sobstv	INT	ID собственника (внешний ключ для связи с таблицей Sobstv)
ID_flat	INT	ID квартиры (внешний ключ для связи с таблицей Flat)

Продолжение таблицы 2.7 – Описание таблицы Pay

Имя поля	Тип поля	Описание поля
ID_schetchik	INT	ID счетчика (внешний ключ для связи с таблицей Schetchik)

Для создания БД в MySQL Server использовался sql-код: CREATE DATABASE Otoplenie;

Для построения таблиц в MySQL Server был использован sql-код:
USE Otoplenie;

```
CREATE TABLE Company(
ID_company INT NOT NULL PRIMARY KEY,
Nazv VARCHAR(20) NOT NULL,
Tariff INT NOT NULL,
Recvizit VARCHAR(30) NOT NULL,
FIO_Ruck VARCHAR(40) NOT NULL
);
```

```
CREATE TABLE Sobstv(
ID_sobstv INT NOT NULL PRIMARY KEY,
Data_birthday DATE NOT NULL,
Phone VARCHAR(12) NOT NULL,
FIO_sobstv VARCHAR(40) NOT NULL,
Passport VARCHAR(10) NOT NULL,
Gender VARCHAR(1) NOT NULL,
);
```

```
CREATE TABLE Flat(
ID_flat INT NOT NULL PRIMARY KEY,
Adress VARCHAR(40) NOT NULL,
Other_info VARCHAR(40) NOT NULL,
Temperatura VARCHAR(3) NOT NULL,
ID_sobstv INT NOT NULL FOREIGN KEY REFERENCES Sobstv(ID_sobstv),
ID_company INT NOT NULL FOREIGN KEY REFERENCES
Company(ID_company),
);
```

```
CREATE TABLE Lgot(
ID_lgot INT NOT NULL PRIMARY KEY,
ID_sobstv INT NOT NULL FOREIGN KEY REFERENCES Sobstv(ID_sobstv),
```

```
Discount VARCHAR(4) NOT NULL
);
```

```
CREATE TABLE Schetchik(
ID_schetchik INT NOT NULL PRIMARY KEY,
ID_flat INT NOT NULL FOREIGN KEY REFERENCES Flat(ID_flat),
Date_ustanovk DATE NOT NULL,
Firma VARCHAR(25) NOT NULL
);
```

```
CREATE TABLE Pokaz(
ID_pokaz INT NOT NULL PRIMARY KEY,
Adout_pokaz VARCHAR(10) NOT NULL,
ID_schetchik INT NOT NULL FOREIGN KEY REFERENCES
Schetchik(ID_schetchik),
Month_pokaz VARCHAR(8) NOT NULL
);
```

```
CREATE TABLE Pay(
ID_pay INT NOT NULL PRIMARY KEY,
Data_pay DATE NOT NULL,
All_summa VARCHAR(10) NOT NULL,
ID_company INT NOT NULL FOREIGN KEY REFERENCES
Company(ID_company),
ID_lgot INT NOT NULL FOREIGN KEY REFERENCES Lgot(ID_lgot),
ID_sobstv INT NOT NULL FOREIGN KEY REFERENCES Sobstv(ID_sobstv),
ID_flat INT NOT NULL FOREIGN KEY REFERENCES Flat(ID_flat),
ID_schetchik INT NOT NULL FOREIGN KEY REFERENCES
Schetchik(ID_schetchik)
);
```

В соответствии с sql-кодом при разработке сайта на Flask (веб-Фреймворк, написанный и используемый в языке Python) использовался код для создания форм, чтобы была возможность считывать данные:

```
class NewClient(FlaskForm):
    data_birthday = StringField('Дата рождения (по форме
ГГГГ.ММ.ДД):', validators=[DataRequired()])
    phone = StringField('Номер телефона (по форме +7*****):',
validators=[DataRequired()])
    fio_sobstv = StringField('ФИО Собственника:',
validators=[DataRequired()])
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',
validators=[DataRequired()])
    gender = StringField('Пол (м/ж):', validators=[DataRequired()])
```

```
    submit = SubmitField('Добавить')
class NewFlat(FlaskForm):
    gorod = StringField('Город:', validators=[DataRequired()])
    street = StringField('Улица:', validators=[DataRequired()])
    dom = StringField('Номер дома:', validators=[DataRequired()])
    id_flat = StringField('Номер квартиры:',
                          validators=[DataRequired()])
    other_info = StringField('Площадь квартиры (кв.м.):',
                            validators=[DataRequired()])
    temperatura = StringField('Средняя температура в квартире (от -100 до +100 градусов):',
                               validators=[DataRequired()])
    schetchik = StringField('Наличие счетчика (да/нет):',
                           validators=[DataRequired()])
    company = StringField('Обслуживающая компания:',
                          validators=[DataRequired()])
    submit = SubmitField('Добавить')
```

```
class NewSchetchik(FlaskForm):
    date_ustanovk = StringField('Дата установки счетчика (в формате ГГГГ.ММ.ДД):',
                                validators=[DataRequired()])
    firma = StringField('Фирма производства:',
                        validators=[DataRequired()])
    submit = SubmitField('Добавить')
```

```
class DeleteClient(FlaskForm):
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',
                          validators=[DataRequired()])
    submit = SubmitField('Удалить')
```

```
class AddPokaz(FlaskForm):
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',
                          validators=[DataRequired()])
    pokaz = StringField('Показания счетчика:',
                        validators=[DataRequired()])
    month_pokaz = StringField('Месяц:', validators=[DataRequired()])
    submit = SubmitField('Добавить')
```

```
class ChangePokaz(FlaskForm):
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',
                          validators=[DataRequired()])
```

```
pokaz = StringField('Новые показания счетчика:',  
validators=[DataRequired()])  
month_pokaz = StringField('Месяц:', validators=[DataRequired()])  
submit = SubmitField('Изменить')
```

```
class ChangeInfoCompany(FlaskForm):  
    nazv_old = StringField('Название:', validators=[DataRequired()])  
    tariff = StringField('Тариф (стоимость за 1 кв.м.):',  
validators=[DataRequired()])  
    recvizit = StringField('Реквизиты:', validators=[DataRequired()])  
    fio_ruck = StringField('ФИО Руководителя:',  
validators=[DataRequired()])  
    submit = SubmitField('Изменить')
```

```
class ChangeInfoClient(FlaskForm):  
    pasport_old = StringField('Серия и номер паспорта(ввод без  
пробела):', validators=[DataRequired()])  
    data_birthday = StringField('Дата рождения (по форме  
ГГГГ.ММ.ДД):', validators=[DataRequired()])  
    phone = StringField('Номер телефона (по форме +7*****):',  
validators=[DataRequired()])  
    fio_sobstv = StringField('ФИО Собственника:',  
validators=[DataRequired()])  
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',  
validators=[DataRequired()])  
    gender = StringField('Пол (м/ж):', validators=[DataRequired()])  
    submit = SubmitField('Изменить')
```

```
class Pay(FlaskForm):  
    data_pay = StringField('Дата:', validators=[DataRequired()])  
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',  
validators=[DataRequired()])  
    submit = SubmitField('Создать')
```

```
class AboutClient(FlaskForm):  
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',  
validators=[DataRequired()])  
    submit = SubmitField('Найти')
```

Завершающим этапом проектирования является создание SQL-запросов для вывода таблиц в интерфейс.

Запрос для таблицы компании:

```
SELECT * FROM Company
```

Запросы для таблицы клиенты:

```
SELECT FIO_sobstv, Data_birthday, Phone, Passport, Gender FROM  
Sobstv WHERE Sobstv.Passport=[номер паспорта]
```

```
SELECT ID_sobstv FROM Sobstv WHERE Sobstv.Passport=[номер  
паспорта]
```

```
SELECT Adress, Other_info FROM Flat WHERE  
Flat.ID_sobstv=[полученое ID из запроса выше]
```

```
SELECT ID_flat FROM Flat WHERE Flat.ID_sobstv=[полученое ID из  
запроса выше]
```

```
SELECT Date_ustanovk, Firma FROM Schetchik WHERE  
Schetchik.ID_flat=[полученое ID из запроса выше]
```

3 Проектирование шаблона выходного документа

Шаблон выходного документа (извещение и квитанция на оплату отопления) представляет из себя заранее созданный текст со специальными словами, что заключены в «{{ }}» и «} }», которые будут заменяться данными из БД. На рисунке 3.1 представлен шаблон выходного документа.

Квитанция на оплату Отопления	
Извещение	Получатель платежа: {{ nazv_pay }}, {{ fio_ruc }} р/с: {{ recvizit_pay }} Плательщик: {{ fio_pay }} Паспортные данные: {{ passport_pay }} Адрес: {{ adres_pay }} Телефон: {{ phone_pay }} Период оплаты: {{ month_pokaz_pay }} Площадь квартиры: {{ plosh_pay }} Средняя температура в квартире: {{ tempa_pay }} Счетчик в квартире: {{ shetchik_status_pay }} Дата установки счетчика: {{ ust_pay_data }} Фирма, устанавливающая счетчик: {{ firma_pay }} Показания счетчика: {{ about_pokaz_pay }} Скидка: {{ discount_pay }} Наименование платежа: Оплата Отопления Сумма: {{ itog_pay }} руб. В том числе скидка: -{{ itog_discount_pay }} руб. Итог к оплате: {{ opalata }} руб.
кассир	Плательщик: _____ (подпись) Дата: {{ data_pay }}
Квитанция	Получатель платежа: {{ nazv_pay }}, {{ fio_ruc }} р/с: {{ recvizit_pay }} Плательщик: {{ fio_pay }} Паспортные данные: {{ passport_pay }} Адрес: {{ adres_pay }} Телефон: {{ phone_pay }} Период оплаты: {{ month_pokaz_pay }} Площадь квартиры: {{ plosh_pay }} Средняя температура в квартире: {{ tempa_pay }} Счетчик в квартире: {{ shetchik_status_pay }} Дата установки счетчика: {{ ust_pay_data }} Фирма, устанавливающая счетчик: {{ firma_pay }} Показания счетчика: {{ about_pokaz_pay }} Скидка: {{ discount_pay }} Наименование платежа: Оплата Отопления Сумма: {{ itog_pay }} руб. В том числе скидка: -{{ itog_discount_pay }} руб. Итог к оплате: {{ opalata }} руб.
кассир	Плательщик: _____ (подпись) Дата: {{ data_pay }}

Рисунок 3.1 – Шаблон выходного документа

4 Проектирование интерфейса приложения

Пользовательский интерфейс – обеспечивает передачу информации между пользователем и программно-аппаратными компонентами компьютерной системы. В соответствии с этим определением мною был разработан интуитивно понятный интерфейс.

На рисунке 4.1 представлен интерфейс главного меню приложения, которое умеет рассчитывать стоимость отопления, работать с данными клиентов, компаний и показаний на счетчиках.



Рисунок 4.1 – Интерфейс основного меню

На рисунке 4.2 можно увидеть интерфейс меню создания квитанции, при корректном вводе данных и обработке автоматически происходит переход на основное меню.

7.0.0.1:8048

Создание квитанции

СОЗДАНИЕ КВИТАНЦИИ

ДАТА ПО ФОРМАТУ ГГГГ.ММ.ДД:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПЕРИОД ОПЛАТЫ (УКАЖИТЕ МЕСЯЦ):

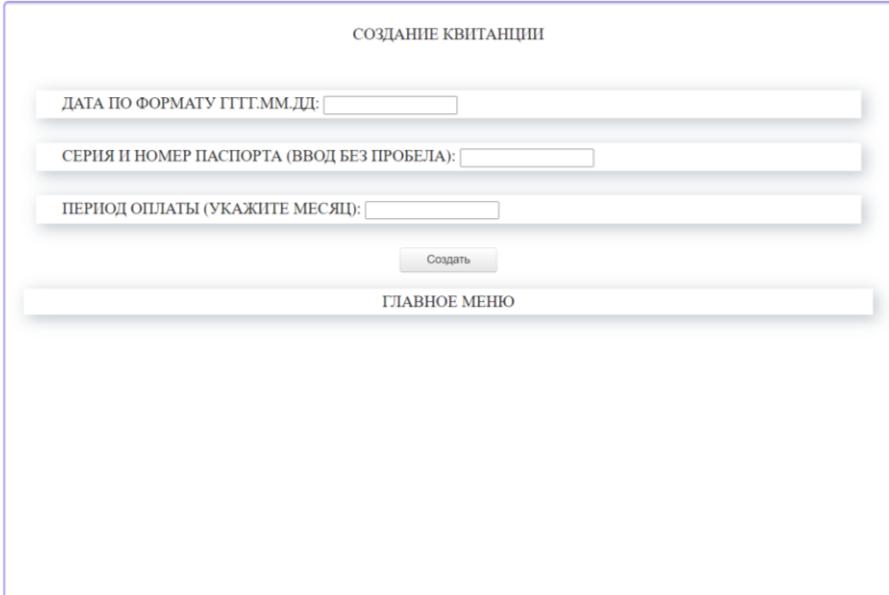


Рисунок 4.2 – Интерфейс меню создания квитанции

На рисунке 4.3 можно увидеть интерфейс меню работы с клиентами, далее необходимо выбрать действие для выполнения: добавление новых клиентов, изменение данных существующих клиентов, просмотр данных клиентов и удаление клиентов.

127.0.0.1:8046

Работа с клиентами (Добавление, изменение, просмотр, удаление информации)

РАБОТА С КЛИЕНТАМИ (ДОБАВЛЕНИЕ, ИЗМЕНЕНИЕ, ПРОСМОТР, УДАЛЕНИЕ ИНФОРМАЦИИ)

ДОБАВЛЕНИЕ НОВЫХ КЛИЕНТОВ

ИЗМЕНЕНИЕ ДАННЫХ СУЩЕСТВУЮЩИХ КЛИЕНТОВ

ПРОСМОТР ДАННЫХ КЛИЕНТА

УДАЛЕНИЕ КЛИЕНТА

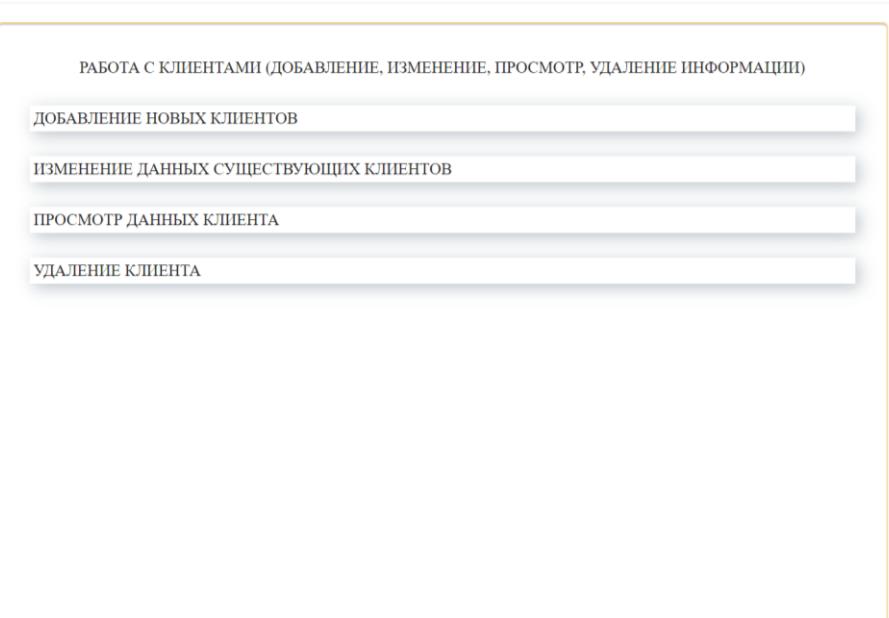


Рисунок 4.3 – Интерфейс меню работы с клиентами

На рисунке 4.4 можно увидеть интерфейс меню добавления клиента, при корректном вводе данных и обработке автоматически происходит переход в меню добавление квартиры.

Добавление клиента

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД):

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):

ФИО СОБСТВЕННИКА:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОЛ (М/Ж):

ГЛАВНОЕ МЕНЮ

Рисунок 4.4 – Интерфейс меню добавления клиента

На рисунке 4.5 можно увидеть интерфейс меню добавления квартиры, при корректном вводе данных и обработке автоматически происходит переход в меню добавление счетчика, если он имеется.

Добавление квартиры

НЕОБХОДИМО ВВЕСТИ ИНФОРМАЦИЮ О КВАРТИРЕ ДЛЯ РАНЕЕ ДОБАВЛЕННОГО СОБСТВЕННИКА

ГОРОД:

УЛИЦА:

НОМЕР ДОМА:

НОМЕР КВАРТИРЫ:

ПЛОЩАДЬ КВАРТИРЫ (КВ.М.):

СРЕДНЯЯ ТЕМПЕРАТУРА В КВАРТИРЕ (ОТ -100 ДО +100 ГРАДУСОВ):

НАЛИЧИЕ СЧЕТЧИКА (ДА/НЕТ):

ОБСЛУЖИВАЮЩАЯ КОМПАНИЯ:

Рисунок 4.5 – Интерфейс меню добавления квартиры

На рисунке 4.6 можно увидеть интерфейс меню добавления счетчика, при корректном вводе данных и обработке автоматически происходит переход в меню добавление показаний счетчика.

Добавление счетчика

ДАТА УСТАНОВКИ СЧЕТЧИКА (В ФОРМАТЕ ГГГГ.ММ.ДД):

ФИРМА ПРОИЗВОДСТВА:

Добавить

Рисунок 4.6 – Интерфейс меню добавления счетчика

На рисунке 4.7 можно увидеть интерфейс меню добавления показаний счетчика, при корректном вводе данных и обработке автоматически происходит переход в основное меню.

Добавление показаний счетчик

ДЛЯ ВНЕСЕНИЯ ПОКАЗАНИЙ СЧЕТЧИКА НЕОБХОДИМО ВВЕСТИ ПАСПОРТНЫЕ ДАННЫЕ СОБСТВЕННИКА!

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОКАЗАНИЯ СЧЕТЧИКА:

МЕСЯЦ:

НЕОБХОДИМО УКАЗЫВАТЬ ПОКАЗАНИЯ ЗА МЕСЯЦ, А НЕ ОБЩИЕ ПОКАЗАНИЯ! ИНАЧЕ СЧЕТ ЗА ОТОПЛЕНИЯ БУДЕТ НЕПРАВИЛЬНЫЙ!

Добавить

ГЛАВНОЕ МЕНЮ

Рисунок 4.7 – Интерфейс меню добавления показаний счетчика

На рисунках 4.8, 4.9, 4.10 показаны формы для изменения, просмотра, удаления данных клиента. При корректном вводе производится переход в основное меню (формы 4.8 и 4.10), переход на вкладку для просмотра данных (форма 4.9).

The screenshot shows a web-based application window titled 'Изменение информации о клиенте' (Change Client Information). At the top left is the URL '7.0.0.1:8046'. The main title is 'ИЗМЕНЕНИЕ ИНФОРМАЦИИ О КЛИЕНТЕ'. Below it is a section labeled 'ИЗМЕНЯЕМЫЕ ДАННЫЕ (ПАСПОРТНЫЕ ДАННЫЕ СОБСТВЕННИКА)':

- СЕРИЯ И НОМЕР ПАСПОРТА(ВВОД БЕЗ ПРОБЕЛА):

Below this is a section labeled 'НОВЫЕ ДАННЫЕ':

- ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД):
- НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):
- СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

At the bottom center is a 'Изменить' (Change) button. Below the button is a 'ГЛАВНОЕ МЕНЮ' (Main Menu) button.

Рисунок 4.8 – Интерфейс меню изменения информации о клиенте

The screenshot shows a web-based application window titled 'Данные клиента' (Client Data). At the top left is the URL '7.0.0.1:8046'. The main title is 'ДАННЫЕ КЛИЕНТА'. Below it is a search field:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

Below the search field is a 'Найти' (Find) button. At the bottom center is a 'ГЛАВНОЕ МЕНЮ' (Main Menu) button.

Рисунок 4.9 – Интерфейс меню просмотра данных клиента

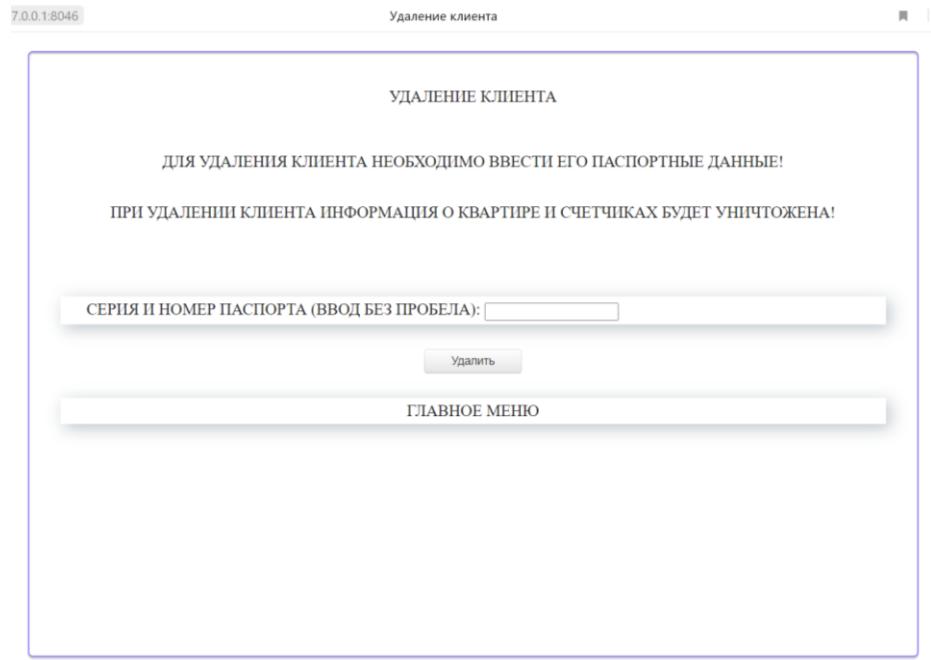


Рисунок 4.10 – Интерфейс меню удаления клиента

На рисунке 4.11 можно увидеть интерфейс меню работы с компаниями, далее необходимо выбрать: просмотр или изменение данных компаний.



Рисунок 4.11 – Интерфейс меню работы с компаниями

При выборе просмотр данных компаний можно увидеть все компании (рисунок 4.12), которые не удаляются, но есть возможность менять название, тариф, реквизиты и ФИО руководителя – рисунок 4.13.

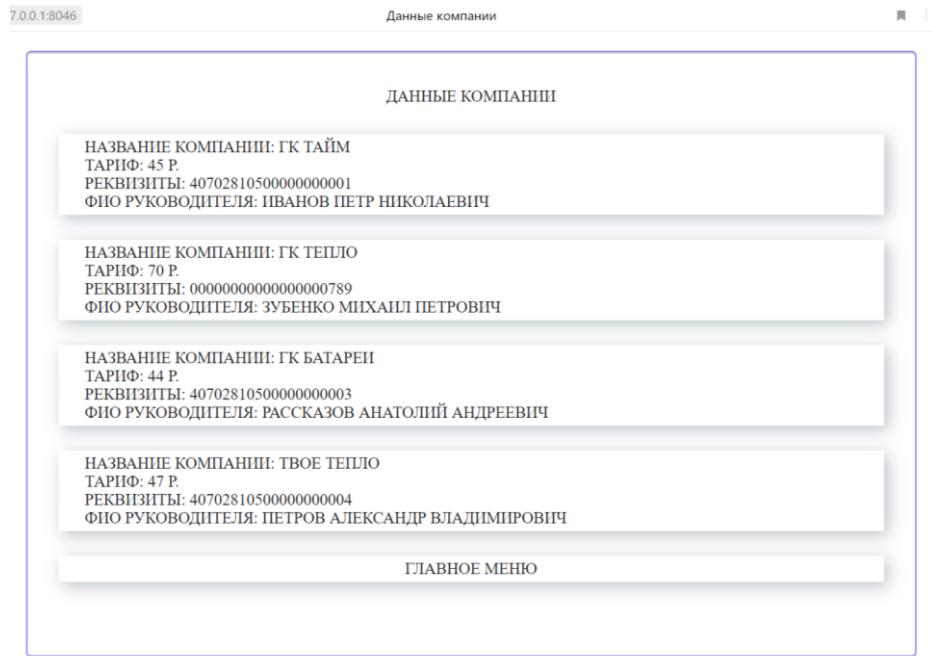


Рисунок 4.12 – Интерфейс меню просмотра данных компаний

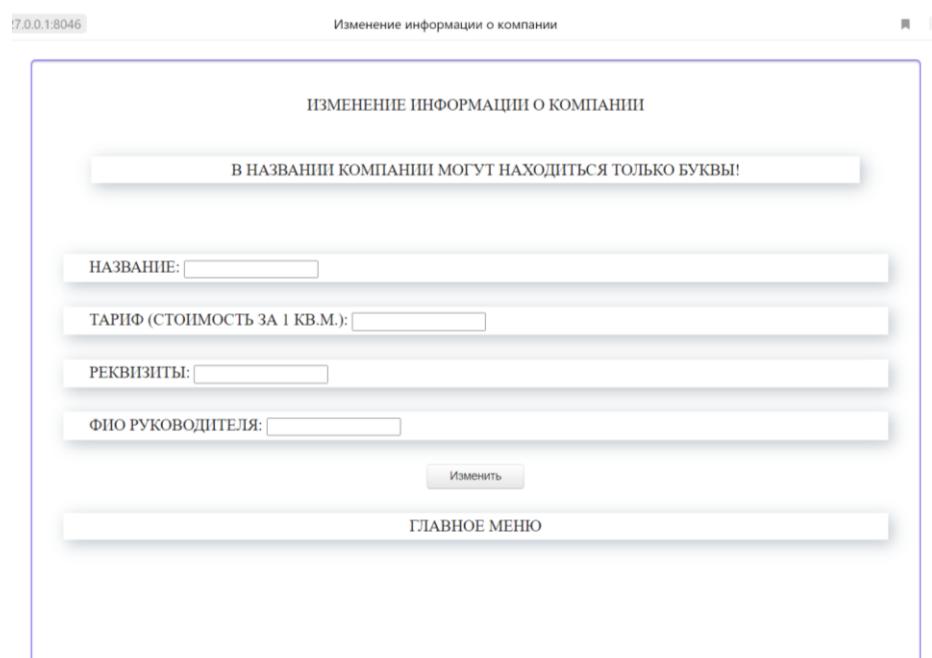


Рисунок 4.13 – Интерфейс меню изменения информации о компании

На рисунке 4.14 можно увидеть интерфейс меню работы с счетчиками, далее необходимо выбрать: добавление или изменение данных счетчика. При корректном заполнении форм добавления (рисунок 4.15), изменении (рисунок 4.16) показаний счетчика происходит переход в основное меню.



Рисунок 4.14 – Интерфейс меню работы с счетчиками

A screenshot of a web-based application window titled 'Добавление показаний счетчик'. The window has a light blue border and contains several input fields and instructions. At the top, it says 'ДОБАВЛЕНИЕ ПОКАЗАНИЙ СЧЕТЧИКА' and 'ДЛЯ ВНЕСЕНИЯ ПОКАЗАНИЙ СЧЕТЧИКА НЕОБХОДИМО ВВЕСТИ ПАСПОРТНЫЕ ДАННЫЕ СОБСТВЕННИКА!'. Below these are three input fields: 'СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):' (Passport series and number (input without spaces)), 'ПОКАЗАНИЯ СЧЕТЧИКА:' (Meter reading), and 'МЕСЯЦ:' (Month). A note below the fields states 'НЕОБХОДИМО УКАЗЫВАТЬ ПОКАЗАНИЯ ЗА МЕСЯЦ, А НЕ ОБЩИЕ ПОКАЗАНИЯ! ИНАЧЕ СЧЕТ ЗА ОТОПЛЕНИЯ БУДЕТ НЕПРАВИЛЬНЫЙ!'. At the bottom are a 'Добавить' (Add) button and a 'ГЛАВНОЕ МЕНЮ' (Main Menu) button.

Рисунок 4.15 – Интерфейс меню добавления показаний счетчика

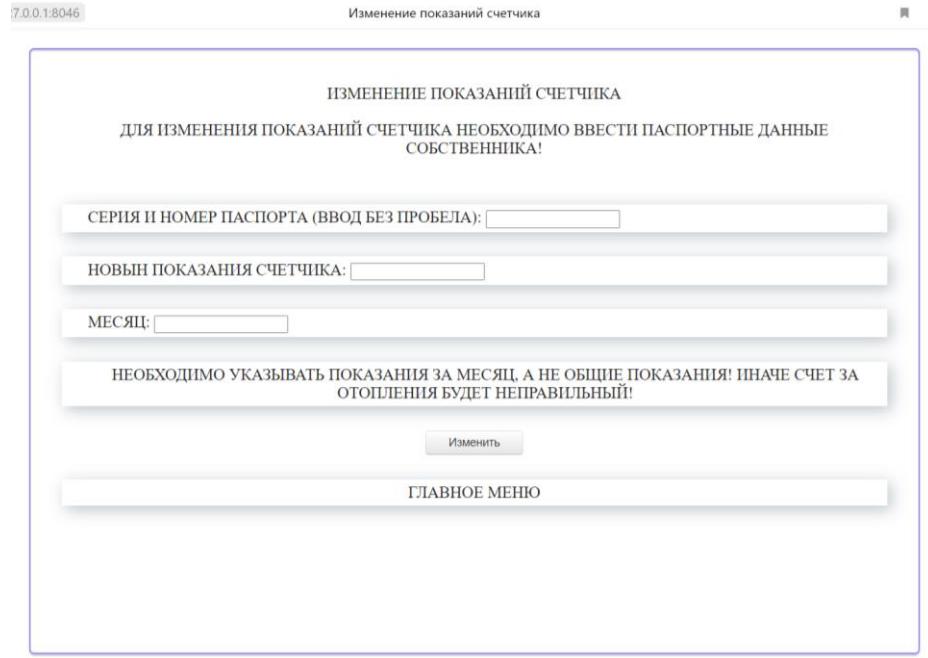


Рисунок 4.16 – Интерфейс меню изменения показаний счетчика

В приложение предусмотрен переход в основное меню, он происходит при нажатии на кнопку «ГЛАВНОЕ МЕНЮ».

5 Тестирование разработанного приложения

Работоспособность программы – это состояние программы, при котором она выполняет необходимые требования, установленные при анализе предметной области.

Чтобы программа (приложение) работало корректно были предусмотрены наличие ошибок при вводе данных пользователем. На рисунках 5.20, 5.27, 5.41, 5.42 продемонстрированы корректно введенные данные.

На рисунках 5.30, 5.31, 5.32, 5.33, 5.37 продемонстрированы изменения и вывод данных (изменённых в том числе).

На рисунке 5.43 продемонстрирован выходной документ. На остальных рисунках - ошибки.

The screenshot shows a web-based application window titled 'Добавление клиента' (Client Addition). At the top left is the URL '7.0.0.1:8046'. The main title of the form is 'ДОБАВЛЕНИЕ КЛИЕНТА'. The form contains several input fields:

- 'ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД)': Input field containing '07 02 20001'.
- 'НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):' Input field with placeholder '+7*****'.
- 'ФИО СОБСТВЕННИКА:' Input field containing 'Максимова Ирина Влади'. To its right is a yellow exclamation mark icon with the text 'Вы пропустили это поле.' (You have skipped this field).
- 'СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):' Input field containing '2583691470'.
- 'ПОЛ (М/Ж):' Input field containing 'ж'.

At the bottom of the form is a 'Добавить' (Add) button and a 'ГЛАВНОЕ МЕНЮ' (Main Menu) button.

Рисунок 5.1 – Ошибка: пустое поле

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА
НЕВЕРНОЕ ЧИСЛО СИМВОЛОВ В ДАТЕ!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД):

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):

ФИО СОБСТВЕННИКА:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОЛ (М/Ж):

ГЛАВНОЕ МЕНЮ

This screenshot shows a client addition form with a validation error. The error message 'НЕВЕРНОЕ ЧИСЛО СИМВОЛОВ В ДАТЕ!' (Incorrect number of symbols in the date) is displayed above the date input field. The date '07.02.2001' is entered in the format 'ГГГГ.ММ.ДД'. Below the date input is a numeric phone number '+7321654987007'. The form also includes fields for name ('Максимова Ирина Влади'), passport series and number ('2583691470'), and gender ('ж'). A 'Добавить' (Add) button is at the bottom.

Рисунок 5.2 – Ошибка: неверное число символов в дате

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА
НЕВЕРНЫЙ ФОРМАТ, ВВЕДИТЕ ПО ФОРМАТУ ГГГГ.ММ.ДД

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД):

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):

ФИО СОБСТВЕННИКА:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОЛ (М/Ж):

ГЛАВНОЕ МЕНЮ

This screenshot shows a client addition form with a validation error. The error message 'НЕВЕРНЫЙ ФОРМАТ, ВВЕДИТЕ ПО ФОРМАТУ ГГГГ.ММ.ДД' (Incorrect format, enter in the format ГГГГ.ММ.ДД) is displayed above the date input field. The date '07.02.2001' is entered in the format 'ГГГГ.ММ.ДД'. Below the date input is a numeric phone number '+7321654987007'. The form also includes fields for name ('Максимова Ирина Влади'), passport series and number ('2583691470'), and gender ('ж'). A 'Добавить' (Add) button is at the bottom.

Рисунок 5.3 – Ошибка: неверный формат ввода даты

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА
ГОД УКАЗАН НЕВЕРНО!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД):

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):

ФИО СОБСТВЕННИКА:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОЛ (М/Ж):

ГЛАВНОЕ МЕНЮ

This screenshot shows a client addition form with a validation error. The error message 'ГОД УКАЗАН НЕВЕРНО!' (The year entered is incorrect) is displayed above the date input field. The date '2022.04.08' is entered in the format 'ГГГГ.ММ.ДД'. Below the date input is a phone number input field containing '+7321654987007'. The 'Добавить' (Add) button is visible at the bottom left of the form area.

Рисунок 5.4 – Ошибка: год указан неверно

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА
КОЛИЧЕСТВО ДНЕЙ НЕ ДОПУСТИМО В МЕСЯЦЕ!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД):

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):

ФИО СОБСТВЕННИКА:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОЛ (М/Ж):

ГЛАВНОЕ МЕНЮ

This screenshot shows a client addition form with a validation error. The error message 'КОЛИЧЕСТВО ДНЕЙ НЕ ДОПУСТИМО В МЕСЯЦЕ!' (The number of days is not allowed in the month) is displayed above the date input field. The date '2004.04.79' is entered in the format 'ГГГГ.ММ.ДД'. Below the date input is a phone number input field containing '+7321654987007'. The 'Добавить' (Add) button is visible at the bottom left of the form area.

Рисунок 5.5 – Ошибка: количество дней не допустимое в месяце

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА
ВВЕДИТЕ ЧИСЛО ДНЕЙ БОЛЬШЕ 0!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД): 2004.04.00

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****): +7321654987007

ФИО СОБСТВЕННИКА: Максимова Ирина Влади

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА): 2583691470

ПОЛ (М/Ж): Ж

Добавить

ГЛАВНОЕ МЕНЮ

This screenshot shows a client addition form with a red error message: 'ВВЕДИТЕ ЧИСЛО ДНЕЙ БОЛЬШЕ 0!' (Enter a number of days greater than 0!). The birth date field contains '2004.04.00'. Other fields include a phone number, name, passport details, and gender.

Рисунок 5.6 – Ошибка: число дней 0

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА
ВВЕДИТЕ МЕСЯЦ БОЛЬШЕ 0!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД): 2004.00.07

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****): +7321654987007

ФИО СОБСТВЕННИКА: Максимова Ирина Влади

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА): 2583691470

ПОЛ (М/Ж): Ж

Добавить

ГЛАВНОЕ МЕНЮ

This screenshot shows a client addition form with a red error message: 'ВВЕДИТЕ МЕСЯЦ БОЛЬШЕ 0!' (Enter a month greater than 0!). The birth date field contains '2004.00.07'. Other fields include a phone number, name, passport details, and gender.

Рисунок 5.7– Ошибка: указание 0 месяца

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА
НЕВЕРНО ВВЕДЕН МЕСЯЦ. ВВЕДИТЕ ПО ФОРМАТУ ГГГГ.ММ.ДД

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД): 2004.13.07

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****): +7321654987007

ФИО СОБСТВЕННИКА: Максимова Ирина Влади

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА): 2583691470

ПОЛ (М/Ж): ж

Добавить

ГЛАВНОЕ МЕНЮ

Рисунок 5.8 – Ошибка: ввода месяца

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА
ВВЕДИТЕ ДАТУ ЧЕРЕЗ ТОЧКУ! (ГГГГ.ММ.ДД)

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД): 2004/12/07

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****): +7321654987007

ФИО СОБСТВЕННИКА: Максимова Ирина Влади

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА): 2583691470

ПОЛ (М/Ж): ж

Добавить

ГЛАВНОЕ МЕНЮ

Рисунок 5.9 – Ошибка: ввода даты

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА
В ПОЛЕ ДАТА ПРИСУТСТВУЮТ ПОСТОРОННИЕ СИМВОЛЫ

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД):

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):

ФИО СОБСТВЕННИКА:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОЛ (М/Ж):

ГЛАВНОЕ МЕНЮ

This screenshot shows a client addition form with a validation error. The error message 'В ПОЛЕ ДАТА ПРИСУТСТВУЮТ ПОСТОРОННИЕ СИМВОЛЫ' (There are foreign symbols in the DATE field) is displayed above the date input field. The date '2004 MM.07' is entered into the field.

Рисунок 5.10 – Ошибка: посторонние символы

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА
НЕВЕРНОЕ ЧИСЛО СИМВОЛОВ В НОМЕРЕ ТЕЛЕФОНА!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД):

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):

ФИО СОБСТВЕННИКА:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОЛ (М/Ж):

ГЛАВНОЕ МЕНЮ

This screenshot shows a client addition form with a validation error. The error message 'НЕВЕРНОЕ ЧИСЛО СИМВОЛОВ В НОМЕРЕ ТЕЛЕФОНА!' (Incorrect number of symbols in the phone number) is displayed above the phone number input field. The phone number '+7321654987007' is entered into the field.

Рисунок 5.11– Ошибка: неверное число символов номера телефона

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА

НОМЕР ТЕЛЕФОНА НАЧИНАЕТСЯ С НЕ С +7!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД): 2004.05.07

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****): 89244716523

ФИО СОБСТВЕННИКА: Максимова Ирина Влади

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА): 2583691470

ПОЛ (М/Ж): ж

Добавить

ГЛАВНОЕ МЕНЮ

This screenshot shows a client addition form with several input fields. The 'Номер телефона' field contains the value '89244716523', which is missing the leading '+7'. A red error message 'НОМЕР ТЕЛЕФОНА НАЧИНАЕТСЯ С НЕ С +7!' is displayed above the form. The rest of the fields contain valid data: date of birth '2004.05.07', passport series '2583691470', and gender 'ж' (female).

Рисунок 5.12 – Ошибка: номер телефона начинается не с +7

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА

В НОМЕРЕ ТЕЛЕФОНА ПРИСУТСТВУЮТ ПОСТОРОННИЕ СИМВОЛЫ!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД): 2004.05.07

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****): +79244716523

ФИО СОБСТВЕННИКА: Максимова Ирина Влади

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА): 2583691470

ПОЛ (М/Ж): ж

Добавить

ГЛАВНОЕ МЕНЮ

This screenshot shows a client addition form with several input fields. The 'Номер телефона' field contains the value '+79244716523', which includes an unnecessary '+' sign. A red error message 'В НОМЕРЕ ТЕЛЕФОНА ПРИСУТСТВУЮТ ПОСТОРОННИЕ СИМВОЛЫ!' is displayed above the form. The rest of the fields contain valid data: date of birth '2004.05.07', passport series '2583691470', and gender 'ж' (female).

Рисунок 5.13 – Ошибка: присутствуют посторонние символы

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА

В ПОЛЕ ФИО СОБСТВЕННИКА ПРИСУТСТВУЮТ ПОСТОРОННИЕ СИМВОЛЫ!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД):

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):

ФИО СОБСТВЕННИКА:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОЛ (М/Ж):

ГЛАВНОЕ МЕНЮ

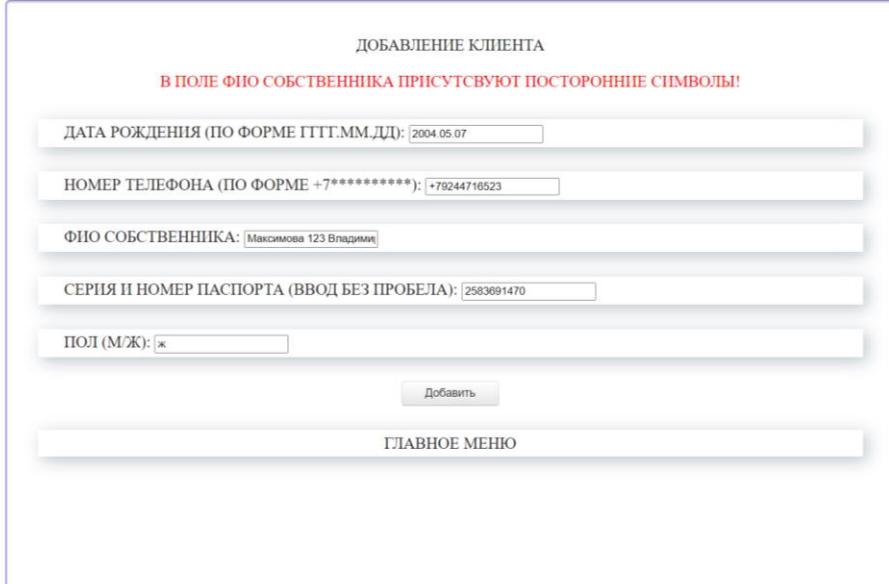


Рисунок 5.14 – Ошибка: присутствуют посторонние символы

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА

В ПОЛЕ ФИО СОБСТВЕННИКА УКАЖИТЕ ФИО!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД):

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):

ФИО СОБСТВЕННИКА:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОЛ (М/Ж):

ГЛАВНОЕ МЕНЮ

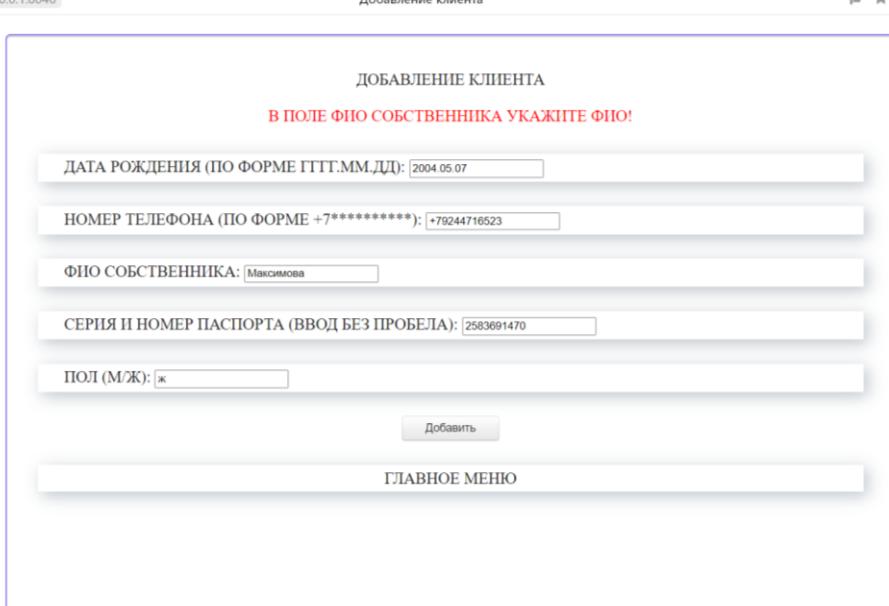


Рисунок 5.15 – Ошибка: некорректно указано ФИО

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА

НЕВЕРНОЕ ЧИСЛО СИМВОЛОВ В СЕРИИ И НОМЕРЕ ПАСПОРТА!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД):

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):

ФИО СОБСТВЕННИКА:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОЛ (М/Ж):

ГЛАВНОЕ МЕНЮ

Рисунок 5.16 – Ошибка: неверное число символов в серии и номере паспорта

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА

В ПОЛЕ ПАСПОРТ ПРИСУТСТВУЮТ ПОСТОРОННИЕ СИМВОЛЫ (УБЕРИТЕ ".")

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД):

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****):

ФИО СОБСТВЕННИКА:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОЛ (М/Ж):

ГЛАВНОЕ МЕНЮ

Рисунок 5.17 – Ошибка: присутствуют посторонние символы

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА
В ПОЛЕ ПАСПОРТ ПРИСУТСТВУЮТ ПОСТОРОННИЕ СИМВОЛЫ!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД): 2004.05.07

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****): +79244716523

ФИО СОБСТВЕННИКА: Максимова Ирина Влади

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА): 2583691a7a

ПОЛ (М/Ж): ж

Добавить

ГЛАВНОЕ МЕНЮ

Рисунок 5.18 – Ошибка: присутствуют посторонние символы

7.0.0.1:8046 Добавление клиента

ДОБАВЛЕНИЕ КЛИЕНТА
ПОЛ НЕОБХОДИМО УКАЗАТЬ ОДНОЙ БУКВОЙ: М/Ж!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД): 2004.05.07

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****): +79244716523

ФИО СОБСТВЕННИКА: Максимова Ирина Влади

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА): 2583691479

ПОЛ (М/Ж): д

Добавить

ГЛАВНОЕ МЕНЮ

Рисунок 5.19 – Ошибка: пол указан неверно

7.0.0.1:8046 Добавление клиента

Добавление клиента
ПОЛ НЕОБХОДИМО УКАЗАТЬ ОДНОЙ БУКВОЙ: М/Ж!

ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД): 2004.05.07

НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****): +79244716523

ФИО СОБСТВЕННИКА: Максимова Ирина Влади

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА): 2583691479

ПОЛ (М/Ж): ж

Добавить

ГЛАВНОЕ МЕНЮ

Рисунок 5.20 – Все данные введены корректно

7.0.0.1:8046 Добавление квартиры

Добавление квартиры
В ПОЛЕ ПЛОЩАДЬ КВАРТИРЫ НЕ МОЖЕТ БЫТЬ 0 ПЕРВЫМ!

НЕОБХОДИМО ВВЕСТИ ИНФОРМАЦИЮ О КВАРТИРЕ ДЛЯ РАНЕЕ ДОБАВЛЕННОГО СОБСТВЕННИКА

ГОРОД: Москва

УЛИЦА: Строителей

НОМЕР ДОМА: 45

НОМЕР КВАРТИРЫ: 147

ПЛОЩАДЬ КВАРТИРЫ (КВ.М.): 0

СРЕДНЯЯ ТЕМПЕРАТУРА В КВАРТИРЕ (ОТ -100 ДО +100 ГРАДУСОВ): +7

НАЛИЧИЕ СЧЕТЧИКА (ДА/НЕТ): да

ОБСЛУЖИВАЮЩАЯ КОМПАНИЯ: гк

Добавить

Рисунок 5.21 – Ошибка: некорректный ввод площади квартиры

ДОБАВЛЕНИЕ КВАРТИРЫ
В ПОЛЕ ПЛОЩАДЬ КВАРТИРЫ НЕ МОЖЕТ БЫТЬ "." ПЕРВЫМ!

НЕОБХОДИМО ВВЕСТИ ИНФОРМАЦИЮ О КВАРТИРЕ ДЛЯ РАНЕЕ ДОБАВЛЕННОГО СОБСТВЕНИКА

ГОРОД:

УЛИЦА:

НОМЕР ДОМА:

НОМЕР КВАРТИРЫ:

ПЛОЩАДЬ КВАРТИРЫ (КВ.М.):

СРЕДНЯЯ ТЕМПЕРАТУРА В КВАРТИРЕ (ОТ -100 ДО +100 ГРАДУСОВ):

НАЛИЧИЕ СЧЕТЧИКА (ДА/НЕТ):

ОБСЛУЖИВАЮЩАЯ КОМПАНИЯ:

Рисунок 5.22 – Ошибка: некорректный ввод площади квартиры

ДОБАВЛЕНИЕ КВАРТИРЫ
В ПОЛЕ ТЕМПЕРАТУРА НЕОБХОДИМО УКАЗАТЬ +/- ПЕРЕД ЗНАЧЕНИЕМ ТЕМПЕРАТУРЫ!

НЕОБХОДИМО ВВЕСТИ ИНФОРМАЦИЮ О КВАРТИРЕ ДЛЯ РАНЕЕ ДОБАВЛЕННОГО СОБСТВЕНИКА

ГОРОД:

УЛИЦА:

НОМЕР ДОМА:

НОМЕР КВАРТИРЫ:

ПЛОЩАДЬ КВАРТИРЫ (КВ.М.):

СРЕДНЯЯ ТЕМПЕРАТУРА В КВАРТИРЕ (ОТ -100 ДО +100 ГРАДУСОВ):

НАЛИЧИЕ СЧЕТЧИКА (ДА/НЕТ):

ОБСЛУЖИВАЮЩАЯ КОМПАНИЯ:

Рисунок 5.23 – Ошибка: некорректный ввод температуры

7.0.0.1:8046 Добавление квартиры

ДОБАВЛЕНИЕ КВАРТИРЫ
В ПОЛЕ НАЛИЧИЕ СЧЕТЧИКА НЕОБХОДИМО НАПИСАТЬ ДА/НЕТ!

НЕОБХОДИМО ВВЕСТИ ИНФОРМАЦИЮ О КВАРТИРЕ ДЛЯ РАНЕЕ ДОБАВЛЕННОГО СОБСТВЕННИКА

ГОРОД:

УЛИЦА:

НОМЕР ДОМА:

НОМЕР КВАРТИРЫ:

ПЛОЩАДЬ КВАРТИРЫ (КВ.М.):

СРЕДНЯЯ ТЕМПЕРАТУРА В КВАРТИРЕ (ОТ -100 ДО +100 ГРАДУСОВ):

НАЛИЧИЕ СЧЕТЧИКА (ДА/НЕТ):

ОБСЛУЖИВАЮЩАЯ КОМПАНИЯ:

Рисунок 5.24 – Ошибка: некорректный ввод показаний счетчика

7.0.0.1:8046 Добавление квартиры

ДОБАВЛЕНИЕ КВАРТИРЫ
ОШИБКА В НАЗВАНИИ ОБСЛУЖИВАЮЩЕЙ КОМПАНИИ!

НЕОБХОДИМО ВВЕСТИ ИНФОРМАЦИЮ О КВАРТИРЕ ДЛЯ РАНЕЕ ДОБАВЛЕННОГО СОБСТВЕННИКА

ГОРОД:

УЛИЦА:

НОМЕР ДОМА:

НОМЕР КВАРТИРЫ:

ПЛОЩАДЬ КВАРТИРЫ (КВ.М.):

СРЕДНЯЯ ТЕМПЕРАТУРА В КВАРТИРЕ (ОТ -100 ДО +100 ГРАДУСОВ):

НАЛИЧИЕ СЧЕТЧИКА (ДА/НЕТ):

ОБСЛУЖИВАЮЩАЯ КОМПАНИЯ:

Рисунок 5.25 – Ошибка: некорректный ввод названия обслуживающей
компании

7.0.0.1:8046 Добавление квартиры

ДОБАВЛЕНИЕ КВАРТИРЫ
ОШИБКА В НАЗВАНИИ ОБСЛУЖИВАЮЩЕЙ КОМПАНИИ!

НЕОБХОДИМО ВВЕСТИ ИНФОРМАЦИЮ О КВАРТИРЕ ДЛЯ РАНЕЕ ДОБАВЛЕННОГО СОБСТВЕНИКА

ГОРОД:

УЛИЦА:

НОМЕР ДОМА:

НОМЕР КВАРТИРЫ:

ПЛОЩАДЬ КВАРТИРЫ (КВ.М.):

СРЕДНЯЯ ТЕМПЕРАТУРА В КВАРТИРЕ (ОТ -100 ДО +100 ГРАДУСОВ):

НАЛИЧИЕ СЧЕТЧИКА (ДА/НЕТ):

ОБСЛУЖИВАЮЩАЯ КОМПАНИЯ:

Рисунок 5.26 – Ошибка: некорректный ввод названия обслуживающей компании

7.0.0.1:8046 Добавление счетчика

ДОБАВЛЕНИЕ СЧЕТЧИКА

ДАТА УСТАНОВКИ СЧЕТЧИКА (В ФОРМАТЕ ГГГГ.ММ.ДД):

ФИРМА ПРОИЗВОДСТВА:

Рисунок 5.27 – Все данные введены корректно

7.0.0.1:8046 Добавление показаний счетчик

ДОБАВЛЕНИЕ ПОКАЗАНИЙ СЧЕТЧИК
ДЛЯ ВНЕСЕНИЯ ПОКАЗАНИЙ СЧЕТЧИКА НЕОБХОДИМО ВВЕСТИ ПАСПОРТНЫЕ ДАННЫЕ СОБСТВЕННИКА!
МЕСЯЦ ПОКАЗАНИЙ УКАЗАН НЕВЕРНО!

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОКАЗАНИЯ СЧЕТЧИКА:

МЕСЯЦ:

НЕОБХОДИМО УКАЗЫВАТЬ ПОКАЗАНИЯ ЗА МЕСЯЦ, А НЕ ОБЩИЕ ПОКАЗАНИЯ! ИНАЧЕ СЧЕТ ЗА ОТОПЛЕНИЯ БУДЕТ НЕПРАВИЛЬНЫЙ!

ГЛАВНОЕ МЕНЮ

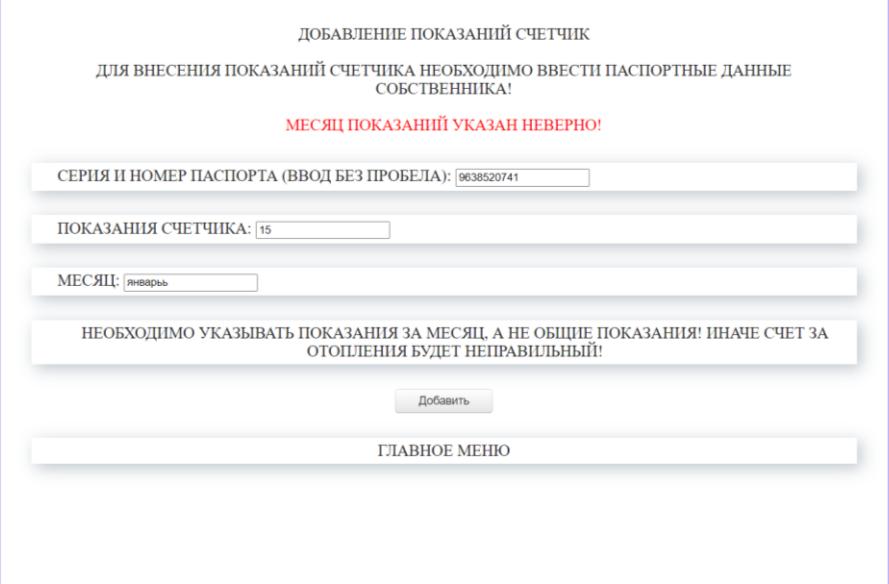


Рисунок 5.28 – Ошибка: месяц показаний указан неверное

7.0.0.1:8046 Добавление показаний счетчик

ДОБАВЛЕНИЕ ПОКАЗАНИЙ СЧЕТЧИК
ДЛЯ ВНЕСЕНИЯ ПОКАЗАНИЙ СЧЕТЧИКА НЕОБХОДИМО ВВЕСТИ ПАСПОРТНЫЕ ДАННЫЕ СОБСТВЕННИКА!
ИНФОРМАЦИЯ О ДАННОМ СОБСТВЕННИКЕ НЕ ИМЕЕТСЯ!

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОКАЗАНИЯ СЧЕТЧИКА:

МЕСЯЦ:

НЕОБХОДИМО УКАЗЫВАТЬ ПОКАЗАНИЯ ЗА МЕСЯЦ, А НЕ ОБЩИЕ ПОКАЗАНИЯ! ИНАЧЕ СЧЕТ ЗА ОТОПЛЕНИЯ БУДЕТ НЕПРАВИЛЬНЫЙ!

ГЛАВНОЕ МЕНЮ

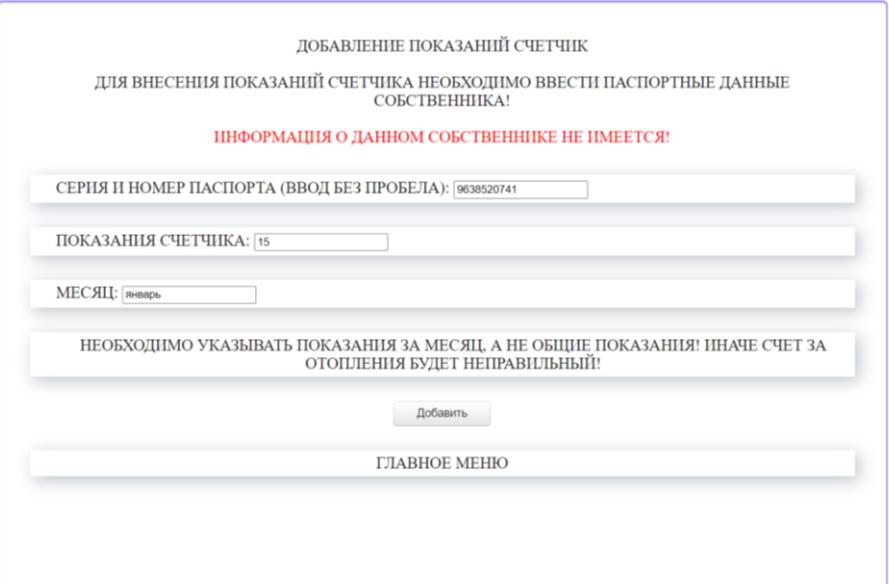


Рисунок 5.29 – Ошибка: пользователь не существует в БД

7.0.0.1:8048 Максимова Ирина Владимировна

МАКСИМОВА ИРИНА ВЛАДИМИРОВНА
ФИО: МАКСИМОВА ИРИНА ВЛАДИМИРОВНА
ДАТА РОЖДЕНИЯ: 2004-05-07
НОМЕР ТЕЛЕФОНА: +79244716523
ПАСПОРТНЫЕ ДАННЫЕ: 2583691479
ПОЛ: Ж
АДРЕС ПРОЖИВАНИЯ: Г. МОСКВА УЛ. СТРОИТЕЛЕЙ 45 КВ. 147
В КВАРТИРЕ 47.7 КВ.М.
СЧЕТЧИК: УСТАНОВЛЕН
ДАТА УСТАНОВКИ СЧЕТЧИКА: 2007-05-24
ФИРМА СЧЕТЧИКА: МАГАЗИН ХОЗТОВАРОВ
ГЛАВНОЕ МЕНЮ

Рисунок 5.30 – Вывод информации о клиенте

7.0.0.1:8048 Изменение информации о клиенте

ИЗМЕНЕНИЕ ИНФОРМАЦИИ О КЛИЕНТЕ
ИЗМЕНЯЕМЫЕ ДАННЫЕ (ПАСПОРТНЫЕ ДАННЫЕ СОБСТВЕННИКА):
СЕРИЯ И НОМЕР ПАСПОРТА(ВВОД БЕЗ ПРОБЕЛА): 2583691479
НОВЫЕ ДАННЫЕ:
ДАТА РОЖДЕНИЯ (ПО ФОРМЕ ГГГГ.ММ.ДД): 2000.05.07
НОМЕР ТЕЛЕФОНА (ПО ФОРМЕ +7*****): +74566548932
СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА): 4567890321
<input type="button" value="Изменить"/>
ГЛАВНОЕ МЕНЮ

Рисунок 5.31 – Изменение информации о клиенте (корректные данные)

МАКСИМОВА ИРИНА ВЛАДИМИРОВНА

ФИО: МАКСИМОВА ИРИНА ВЛАДИМИРОВНА

ДАТА РОЖДЕНИЯ: 2000-05-07

НОМЕР ТЕЛЕФОНА: +74566548932

ПАСПОРТНЫЕ ДАННЫЕ: 4567890321

ПОЛ: Ж

АДРЕС ПРОЖИВАНИЯ: Г. МОСКВА УЛ. СТРОИТЕЛЕЙ 45 КВ. 147

В КВАРТИРЕ 47.7 КВ.М.

СЧЕТЧИК: УСТАНОВЛЕН

ДАТА УСТАНОВКИ СЧЕТЧИКА: 2007-05-24

ФИРМА СЧЕТЧИКА: МАГАЗИН ХОЗТОВАРОВ

ГЛАВНОЕ МЕНЮ

Рисунок 5.32 – Измененные данные

ДАННЫЕ КОМПАНИИ

НАЗВАНИЕ КОМПАНИИ: ГК ТАЙМ
ТАРИФ: 45 Р.
РЕКВИЗИТЫ: 40702810500000000001
ФИО РУКОВОДИТЕЛЯ: ИВАНОВ ПЕТР НИКОЛАЕВИЧ

НАЗВАНИЕ КОМПАНИИ: ГК ТЕПЛО
ТАРИФ: 70 Р.
РЕКВИЗИТЫ: 0000000000000000789
ФИО РУКОВОДИТЕЛЯ: ЗУБЕНКО МИХАИЛ ПЕТРОВИЧ

НАЗВАНИЕ КОМПАНИИ: ГК БАТАРЕИ
ТАРИФ: 44 Р.
РЕКВИЗИТЫ: 4070281050000000003
ФИО РУКОВОДИТЕЛЯ: РАССКАЗОВ АНАТОЛИЙ АНДРЕЕВИЧ

НАЗВАНИЕ КОМПАНИИ: ТВОЕ ТЕПЛО
ТАРИФ: 47 Р.
РЕКВИЗИТЫ: 4070281050000000004
ФИО РУКОВОДИТЕЛЯ: ПЕТРОВ АЛЕКСАНДР ВЛАДИМИРОВИЧ

ГЛАВНОЕ МЕНЮ

Рисунок 5.33 – Вывод информации о компаниях

7.0.0.1:8048 Изменение информации о компании

ИЗМЕНЕНИЕ ИНФОРМАЦИИ О КОМПАНИИ

В НАЗВАНИИ КОМПАНИИ МОГУТ НАХОДИТЬСЯ ТОЛЬКО БУКВЫ!

В ПОЛЕ НАЗВАНИЕ ПРИСУТСТВУЮТ ПОСТОРОННИЕ СИМВОЛЫ!

НАЗВАНИЕ: ГК Тепло1

ТАРИФ (СТОИМОСТЬ ЗА 1 КВ.М.): 88

РЕКВИЗИТЫ: 678

ФИО РУКОВОДИТЕЛЯ: test

Изменить

ГЛАВНОЕ МЕНЮ

This screenshot shows a web-based form for changing company information. At the top, it says 'Изменение информации о компании'. Below that is a section titled 'ИЗМЕНЕНИЕ ИНФОРМАЦИИ О КОМПАНИИ' with two error messages: 'В НАЗВАНИИ КОМПАНИИ МОГУТ НАХОДИТЬСЯ ТОЛЬКО БУКВЫ!' (Letters only) and 'В ПОЛЕ НАЗВАНИЕ ПРИСУТСТВУЮТ ПОСТОРОННИЕ СИМВОЛЫ!' (Non-alphabetic characters). The form fields are: 'НАЗВАНИЕ' (Name) containing 'ГК Тепло1', 'ТАРИФ (СТОИМОСТЬ ЗА 1 КВ.М.)' (Tariff) containing '88', 'РЕКВИЗИТЫ' (Requisites) containing '678', and 'ФИО РУКОВОДИТЕЛЯ' (Name of Director) containing 'test'. A 'Изменить' (Change) button is visible, along with a 'ГЛАВНОЕ МЕНЮ' (Main Menu) link.

Рисунок 5.34 – Ошибка: некоректный ввод названия компании

7.0.0.1:8048 Изменение информации о компании

ИЗМЕНЕНИЕ ИНФОРМАЦИИ О КОМПАНИИ

В НАЗВАНИИ КОМПАНИИ МОГУТ НАХОДИТЬСЯ ТОЛЬКО БУКВЫ!

ТАРИФ НЕ МОЖЕТ БЫТЬ ДОРОЖЕ 70Р!

НАЗВАНИЕ: ГК Тепло

ТАРИФ (СТОИМОСТЬ ЗА 1 КВ.М.): 88

РЕКВИЗИТЫ: 678

ФИО РУКОВОДИТЕЛЯ: test

Изменить

ГЛАВНОЕ МЕНЮ

This screenshot shows a similar web-based form for changing company information. It includes the same error message for the company name: 'В НАЗВАНИИ КОМПАНИИ МОГУТ НАХОДИТЬСЯ ТОЛЬКО БУКВЫ!' (Letters only). A new error message 'ТАРИФ НЕ МОЖЕТ БЫТЬ ДОРОЖЕ 70Р!' (Tariff cannot be higher than 70R) is displayed above the tariff input field. The form fields are identical to the previous screenshot: 'НАЗВАНИЕ' (Name) containing 'ГК Тепло', 'ТАРИФ (СТОИМОСТЬ ЗА 1 КВ.М.)' (Tariff) containing '88', 'РЕКВИЗИТЫ' (Requisites) containing '678', and 'ФИО РУКОВОДИТЕЛЯ' (Name of Director) containing 'test'. A 'Изменить' (Change) button is visible, along with a 'ГЛАВНОЕ МЕНЮ' (Main Menu) link.

Рисунок 5.35 – Ошибка: стоимость тарифа не может быть дороже 70р.

7.0.0.1:8048 Изменение информации о компании

ИЗМЕНЕНИЕ ИНФОРМАЦИИ О КОМПАНИИ

В НАЗВАНИИ КОМПАНИИ МОГУТ НАХОДИТЬСЯ ТОЛЬКО БУКВЫ!

В ПОЛЕ РЕКВИЗИТЫ ДОЛЖНО БЫТЬ 20 ЦИФР!

НАЗВАНИЕ: ГК Тепло

ТАРИФ (СТОИМОСТЬ ЗА 1 КВ.М.): 69

РЕКВИЗИТЫ: 678

ФИО РУКОВОДИТЕЛЯ: test

Изменить

ГЛАВНОЕ МЕНЮ

Рисунок 5.36 – Ошибка: в поле реквизиты должно быть 20 цифр

7.0.0.1:8048 Данные компании

ДАННЫЕ КОМПАНИИ

НАЗВАНИЕ КОМПАНИИ: ГК ТАЙМ
ТАРИФ: 45 Р.
РЕКВИЗИТЫ: 4070281050000000001
ФИО РУКОВОДИТЕЛЯ: ИВАНОВ ПЕТР НИКОЛАЕВИЧ

НАЗВАНИЕ КОМПАНИИ: ГК ТЕПЛО
ТАРИФ: 69 Р.
РЕКВИЗИТЫ: 67867867801234567890
ФИО РУКОВОДИТЕЛЯ: НОВИКОВ АНДРЕЙ АНАТОЛЬЕВИЧ

НАЗВАНИЕ КОМПАНИИ: ГК БАТАРЕИ
ТАРИФ: 44 Р.
РЕКВИЗИТЫ: 4070281050000000003
ФИО РУКОВОДИТЕЛЯ: РАССКАЗОВ АНАТОЛИЙ АНДРЕЕВИЧ

НАЗВАНИЕ КОМПАНИИ: ТВОЕ ТЕПЛО
ТАРИФ: 47 Р.
РЕКВИЗИТЫ: 4070281050000000004
ФИО РУКОВОДИТЕЛЯ: ПЕТРОВ АЛЕКСАНДР ВЛАДИМИРОВИЧ

ГЛАВНОЕ МЕНЮ

Рисунок 5.37 – Измененные данные компаний

7.0.0.1:8048 Изменение показаний счетчика

ИЗМЕНЕНИЕ ПОКАЗАНИЙ СЧЕТЧИКА
ДЛЯ ИЗМЕНЕНИЯ ПОКАЗАНИЙ СЧЕТЧИКА НЕОБХОДИМО ВВЕСТИ ПАСПОРТНЫЕ ДАННЫЕ СОБСТВЕННИКА!
В ПОЛЕ ПОКАЗАНИЯ СЧЕТЧИКА ПРИСУТСТВУЮТ ПОСТОРОННИЕ СИМВОЛЫ!

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

НОВЫЕ ПОКАЗАНИЯ СЧЕТЧИКА:

МЕСЯЦ:

НЕОБХОДИМО УКАЗЫВАТЬ ПОКАЗАНИЯ ЗА МЕСЯЦ, А НЕ ОБЩИЕ ПОКАЗАНИЯ! ИНАЧЕ СЧЕТ ЗА ОТОПЛЕНИЯ БУДЕТ НЕПРАВИЛЬНЫЙ!

Рисунок 5.38 – Ошибка: некорректный ввод показаний счетчика

7.0.0.1:8048 Изменение показаний счетчика

ИЗМЕНЕНИЕ ПОКАЗАНИЙ СЧЕТЧИКА
ДЛЯ ИЗМЕНЕНИЯ ПОКАЗАНИЙ СЧЕТЧИКА НЕОБХОДИМО ВВЕСТИ ПАСПОРТНЫЕ ДАННЫЕ СОБСТВЕННИКА!
МЕСЯЦ ПОКАЗАНИЙ УКАЗАН НЕВЕРНО!

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

НОВЫЕ ПОКАЗАНИЯ СЧЕТЧИКА:

МЕСЯЦ:

НЕОБХОДИМО УКАЗЫВАТЬ ПОКАЗАНИЯ ЗА МЕСЯЦ, А НЕ ОБЩИЕ ПОКАЗАНИЯ! ИНАЧЕ СЧЕТ ЗА ОТОПЛЕНИЯ БУДЕТ НЕПРАВИЛЬНЫЙ!

Рисунок 5.39 – Ошибка: некорректно указан месяц показаний

7.0.0.1:8048 Изменение показаний счетчика

ИЗМЕНЕНИЕ ПОКАЗАНИЙ СЧЕТЧИКА
ДЛЯ ИЗМЕНЕНИЯ ПОКАЗАНИЙ СЧЕТЧИКА НЕОБХОДИМО ВВЕСТИ ПАСПОРТНЫЕ ДАННЫЕ СОБСТВЕННИКА!
ИНФОРМАЦИЯ О ПОКАЗАНИЯХ ЗА ДАННЫЙ МЕСЯЦ НЕ ИМЕЮТСЯ!

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

НОВЫЕ ПОКАЗАНИЯ СЧЕТЧИКА:

МЕСЯЦ:

НЕОБХОДИМО УКАЗЫВАТЬ ПОКАЗАНИЯ ЗА МЕСЯЦ, А НЕ ОБЩИЕ ПОКАЗАНИЯ! ИНАЧЕ СЧЕТ ЗА ОТОПЛЕНИЯ БУДЕТ НЕПРАВИЛЬНЫЙ!

Рисунок 5.40 – Ошибка: информация о показаниях не найдена

7.0.0.1:8048 Добавление показаний счетчик

ДОБАВЛЕНИЕ ПОКАЗАНИЙ СЧЕТЧИК
ДЛЯ ВНЕСЕНИЯ ПОКАЗАНИЙ СЧЕТЧИКА НЕОБХОДИМО ВВЕСТИ ПАСПОРТНЫЕ ДАННЫЕ СОБСТВЕННИКА!

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПОКАЗАНИЯ СЧЕТЧИКА:

МЕСЯЦ:

НЕОБХОДИМО УКАЗЫВАТЬ ПОКАЗАНИЯ ЗА МЕСЯЦ, А НЕ ОБЩИЕ ПОКАЗАНИЯ! ИНАЧЕ СЧЕТ ЗА ОТОПЛЕНИЯ БУДЕТ НЕПРАВИЛЬНЫЙ!

Рисунок 5.41 – Корректно введенные данные

7.0.0.1:8048 Создание квитанции

СОЗДАНИЕ КВИТАНЦИИ

ДАТА ПО ФОРМАТУ ГГГГ.ММ.ДД:

СЕРИЯ И НОМЕР ПАСПОРТА (ВВОД БЕЗ ПРОБЕЛА):

ПЕРИОД ОПЛАТЫ (УКАЖИТЕ МЕСЯЦ):

ГЛАВНОЕ МЕНЮ

Рисунок 5.42 – Создание квитанции (данные корректны)

Квитанция на оплату Отопления

Извещение	Получатель платежа: ГК Тайм, Иванов Петр Николаевич р/с: 40702810500000000001 Плательщик: Максимова Ирина Владимировна Паспортные данные: 4567890321 Адрес: г. Москва ул. Строителей 45 кв. 147 Телефон: +74566548932 Период оплаты: май Площадь квартиры: 47.7 Средняя температура в квартире: +7 Счетчик в квартире: есть Дата установки счетчика: 2007-05-24 Фирма, устанавливающая счетчик: Магазин ХозТоваров Показания счетчика: 45 Скидка: 10% Наименование платежа: Оплата Отопления Сумма: 2025 руб. В том числе скидка: -202 руб. Итог к оплате: 1823 руб.
Кассир	Плательщик: _____ (подпись) Дата: 2021.03.15
Квитанция	Получатель платежа: ГК Тайм, Иванов Петр Николаевич р/с: 40702810500000000001 Плательщик: Максимова Ирина Владимировна Паспортные данные: 4567890321 Адрес: г. Москва ул. Строителей 45 кв. 147 Телефон: +74566548932 Период оплаты: май Площадь квартиры: 47.7 Средняя температура в квартире: +7 Счетчик в квартире: есть Дата установки счетчика: 2007-05-24 Фирма, устанавливающая счетчик: Магазин ХозТоваров Показания счетчика: 45 Скидка: 10% Наименование платежа: Оплата Отопления Сумма: 2025 руб. В том числе скидка: -202 руб. Итог к оплате: 1823 руб.
Кассир	Плательщик: _____ (подпись) Дата: 2021.03.15

Рисунок 5.43 – Выходная заполненная квитанция

6 Листинг программы

6.1 Листинг файла main.py

```
from flask import Flask, render_template, make_response, jsonify
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField
from wtforms.validators import DataRequired
from werkzeug.utils import redirect
from docxtpl import DocxTemplate # импорт библиотеки для работы с документом
import pypyodbc # импорт библиотеки sql

app = Flask(__name__) # создание приложения
app.config['SECRET_KEY'] = 'test'

mySQLServer = 'DESKTOP-M15245Q\SQLEXPRESS' # Имя сервера (в моем случае локального)
myDataBase = 'Otoplenie' # Имя БД

connection = pypyodbc.connect('Driver={SQL Server};" # подключение к БД, которая на MySqlServer
                                'Server=' + mySQLServer + ';'
                                'Database=' + myDataBase + ';')

cursor = connection.cursor() # будет выполнять запросы на sql

spisok = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
spisok_buk = ['ё', 'й', 'ц', 'у', 'к', 'е', 'н', 'г', 'ш', 'щ', 'з', 'х', 'ъ', 'ф', 'ы', 'в', 'а', 'п', 'р', 'о',
              'ө', 'л', 'д', 'ж', 'э', 'я', 'ч', 'с', 'м', 'и', 'т', 'ь', 'б', 'ю']

# Форма для добавления нового клиента
class NewClient(FlaskForm):
    data_birthday = StringField('Дата рождения (по форме ГГГГ.ММ.ДД):',
                               validators=[DataRequired()])
    phone = StringField('Номер телефона (по форме +7*****-*):',
                        validators=[DataRequired()])
    fio_sobstv = StringField('ФИО Собственника:', validators=[DataRequired()])
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',
                          validators=[DataRequired()])
    gender = StringField('Пол (м/ж):', validators=[DataRequired()])
    submit = SubmitField('Добавить')

# Форма для добавления новой квартиры
class NewFlat(FlaskForm):
    gorod = StringField('Город:', validators=[DataRequired()])
    street = StringField('Улица:', validators=[DataRequired()])
    dom = StringField('Номер дома:', validators=[DataRequired()])
```

```
id_flat = StringField('Номер квартиры:', validators=[DataRequired()])
other_info = StringField('Площадь квартиры (кв.м.):', validators=[DataRequired()])
temperatura = StringField('Средняя температура в квартире (от -100 до +100 градусов):',
validators=[DataRequired()])
schetchik = StringField('Наличие счетчика (да/нет):', validators=[DataRequired()])
company = StringField('Обслуживающая компания:', validators=[DataRequired()])
submit = SubmitField('Добавить')
```

```
# Форма для добавления нового счетчика
class NewSchetchik(FlaskForm):
    date_ustanovk = StringField('Дата установки счетчика (в формате ГГГГ.ММ.ДД):',
validators=[DataRequired()])
    firma = StringField('Фирма производства:', validators=[DataRequired()])
    submit = SubmitField('Добавить')
```

```
# Форма для удаления клиента
class DeleteClient(FlaskForm):
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',
validators=[DataRequired()])
    submit = SubmitField('Удалить')
```

```
# Форма для добавления показаний
class AddPokaz(FlaskForm):
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',
validators=[DataRequired()])
    pokaz = StringField('Показания счетчика:', validators=[DataRequired()])
    month_pokaz = StringField('Месяц:', validators=[DataRequired()])
    submit = SubmitField('Добавить')
```

```
# Форма для изменения показаний счетчика
class ChangePokaz(FlaskForm):
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',
validators=[DataRequired()])
    pokaz = StringField('Новые показания счетчика:', validators=[DataRequired()])
    month_pokaz = StringField('Месяц:', validators=[DataRequired()])
    submit = SubmitField('Изменить')
```

```
# Форма для изменения информации компаний
class ChangeInfoCompany(FlaskForm):
    nazv_old = StringField('Название:', validators=[DataRequired()])
    tariff = StringField('Тариф (стоимость за 1 кв.м.):', validators=[DataRequired()])
    recvizit = StringField('Реквизиты:', validators=[DataRequired()])
    fio_ruck = StringField('ФИО Руководителя:', validators=[DataRequired()])
    submit = SubmitField('Изменить')
```

```
# Форма для изменения информации клиента
class ChangeInfoClient(FlaskForm):
```

```

    pasport_old = StringField('Серия и номер паспорта(ввод без пробела):',
validators=[DataRequired()])
    data_birthday = StringField('Дата рождения (по форме ГГГГ.ММ.ДД):',
validators=[DataRequired()])
    phone = StringField('Номер телефона (по форме +7*****):',
validators=[DataRequired()])
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',
validators=[DataRequired()])
    submit = SubmitField('Изменить')

# Форма для формирования выходного документа
class Pay(FlaskForm):
    data_pay = StringField('Дата по формату ГГГГ.ММ.ДД:', validators=[DataRequired()])
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',
validators=[DataRequired()])
    month = StringField('Период оплаты (укажите месяц):', validators=[DataRequired()])
    submit = SubmitField('Создать')

# Форма для вывода информации о клиенте
class AboutClient(FlaskForm):
    pasport = StringField('Серия и номер паспорта (ввод без пробела):',
validators=[DataRequired()])
    submit = SubmitField('Найти')

# Запуск приложения через функцию
def main():
    app.run(port=8048, host='127.0.0.1')

# проверка только на наличие цифр
def chek_number(slovo):
    for i in str(slovo):
        if i not in spisok:
            if i != '.':
                return False

# проверка только на наличие букв
def chek_bukv(slovo):
    for i in str(slovo).lower():
        if (i != ' ') and (i != '-'):
            if i not in spisok_buk:
                return False

# проверка корректности ввода паспорта
def chek_pasport(pasport):
    if len(str(pasport)) != 10:
        return 'Неверное число символов в серии и номере паспорта!'

```

```

else:
    if '.' in pasport:
        return 'В поле паспорт присутствуют посторонние символы (уберите ".")'
    if chek_number(pasport) == False:
        return 'В поле паспорт присутствуют посторонние символы!'
    return ""

# проверка корректности ввода даты
def chek_data(znach_data):
    if len(znach_data) == 10: # проверка длины введенной даты
        vrem = znach_data.split('.')
    if len(vrem) == 3: # проверка формата ввода
        if (len(vrem[0]) != 4) or (len(vrem[1]) != 2) or (len(vrem[2]) != 2):
            return "Неверный формат, введите по формату ГГГГ.ММ.ДД"
        else:
            message = chek_number(znach_data)
            if message == False:
                return 'В поле дата присутствуют посторонние символы'
            if int(vrem[0]) > 2021:
                return 'Год указан неверно!'
            if '00' == vrem[1]:
                return 'Введите месяц больше 0!'
            if '00' == vrem[2]:
                return 'Введите число дней больше 0!'
            if 0 > int(vrem[1]) or int(vrem[1]) > 12: # проверка корректности введенного
месяца
                return "Неверно введен месяц, введите по формату ГГГГ.ММ.ДД"
            else: # проверка числа дней в месяце
                mes = int(vrem[1])
                day = int(vrem[2])
                spisok_day_mes = [31, 0, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
                if mes == 2:
                    if int(vrem[0]) // 4 == 0:
                        if 0 < day > 30:
                            return "Количество дней не допустимое в месяце!"
                    else:
                        if 0 < day > 29:
                            return "Количество дней не допустимое в месяце!"
                else:
                    if 0 < day > spisok_day_mes[mes - 1]:
                        return "Количество дней не допустимое в месяце!"
                if chek_number(znach_data) == False:
                    return "В введенной дате имеются буквы!"
            else:
                return "Введите дату через точку! (ГГГГ.ММ.ДД)"
        else:
            return "Неверное число символов в дате!"
    return ""

```

проверка корректности ввода телефона

```

def chek_phone(phone):
    if phone[0] != '+': # проверка корректности введенного телефона
        return "Номер телефона начинается с не с +7!"
    if phone[1] != '7': # проверка корректности введенного телефона
        return "Номер телефона начинается с не с +7!"
    elif len(phone) == 12:
        if chek_number(phone[1::]) == False:
            return "В номере телефона присутствуют посторонние символы!"
    else:
        return "Неверное число символов в номере телефона!"
    return ""

# проверка ФИО человека
def chek_fio(fio):
    sobstv = fio.split()
    if 4 > len(sobstv) > 1:
        if chek_bukv(fio) == False:
            return "В поле ФИО Собственника присутствуют посторонние символы!"
    else:
        return "В поле ФИО Собственника укажите ФИО!"
    return ""

# Ссылка на открытие основного меню
@app.route("/", methods=['GET', 'POST'])
def index():
    return render_template("base.html", title='Учет платы за отопление')

# Ссылка на открытие меню работы с клиентами
@app.route("/client", methods=['GET', 'POST'])
def client():
    return render_template("client.html", title='Работа с клиентами (Добавление, изменение, просмотр, удаление информации)')

# Ссылка на открытие меню работы с компаниями
@app.route("/company", methods=['GET', 'POST'])
def company():
    return render_template("company.html", title='Работа с компаниями (Изменение, просмотр информации)')

# Ссылка на открытие меню работы со счетчиками
@app.route("/schetchik", methods=['GET', 'POST'])
def schetchik():
    return render_template("schetchik.html", title='Работа с счетчиками (Изменение, просмотр информации)')

# Ссылка на открытие страницы создания квитанции (чека на оплату)

```

```

@app.route("/itog", methods=['GET', 'POST'])
def itog():
    form = Pay() # обращение к классу формы
    temp = 'itog.html' # имя шаблона
    title = 'Создание квитанции' # заголовок страницы
    if form.validate_on_submit():
        message = chek_pasport(form.pasport.data) # проверка корректности паспорта
        if message != "":
            return render_template(temp, title=title, form=form, message=message)
        for_zapros = "Sobstv.Passport LIKE '%" + form.pasport.data + "%'"
        zapros = ("SELECT ID_sobstv FROM Sobstv WHERE(%s);%" % for_zapros)
        cursor.execute(zapros)
        info_sobstvinn_id = cursor.fetchall()
        if not info_sobstvinn_id:
            return render_template(temp, title=title, form=form,
                                  message="Клиент с такими паспортными данными не существует!")
        message = chek_data(form.data_pay.data) # проверка корректности паспорта
        if message != "":
            return render_template(temp, title=title, form=form, message='Неверно введена дата
оплаты! (ГГГГ.ММ.ДД)')
        all_month = ['январь', 'февраль', 'март', 'апрель', 'май', 'июнь', 'июль', 'август', 'сентябрь',
                    'октябрь', 'ноябрь', 'декабрь']
        month_pokaz = 13
        for i in range(12):
            if form.month.data.lower() == all_month[i]:
                month_pokaz = i + 1
                break
        if month_pokaz == 13:
            return render_template(temp, title=title, form=form, message="Месяц показаний указан
неверно!")

        for_zapros = "Sobstv.ID_sobstv LIKE'" + str(info_sobstvinn_id[0][0]) + "%'"
        zapros = ("SELECT FIO_sobstv, Phone FROM Sobstv WHERE(%s);%" % for_zapros)
        cursor.execute(zapros)
        info1_sobstvinn = cursor.fetchall()

        for_zapros = "Flat.ID_sobstv=" + str(info_sobstvinn_id[0][0]) + ""
        zapros = ("SELECT * FROM Flat WHERE(%s);%" % for_zapros)
        cursor.execute(zapros)
        info2_sobstvinn = cursor.fetchall()

        passport_pay = form.pasport.data
        fio_pay, phone_pay = info1_sobstvinn[0]
        id_flat, adres_pay, plosh_pay, tempa_pay, info_sobstvinn_id, id_company =
info2_sobstvinn[0]

        for_zapros = "Company.ID_company=" + str(id_company) + ""
        zapros = ("SELECT * FROM Company WHERE(%s);%" % for_zapros)
        cursor.execute(zapros)
        info3_sobstvinn = cursor.fetchall()
        id_company, nazv_pay, tarif, recvizit_pay, fio_ruc = info3_sobstvinn[0]

```

```

for_zapros = "Schetchik.ID_flat="" + str(id_flat) + """
zapros = ("SELECT * FROM Schetchik WHERE(%s);"" % for_zapros)
cursor.execute(zapros)
info2_sobstvinn = cursor.fetchall()
if info2_sobstvinn:
    id_shetchik, id_flat, ust_pay, firma_pay = info2_sobstvinn[0]
    shetchik_status_pay = 'есть'

    zapros = ("SELECT Adout_pokaz FROM Pokaz WHERE Month_pokaz=%s' AND
ID_schetchik=%s"" %
              str(form.month.data), str(id_flat)))
    cursor.execute(zapros)
    info_month_sobstvinn = cursor.fetchall()

    if info_month_sobstvinn == []:
        return render_template(temp, title=title, form=form, message="Информации за
данный месяц нет!")
        month_pokaz_pay = form.month.data
        about_pokaz_pay = info_month_sobstvinn[0][0]
        itog_pay = int(about_pokaz_pay) * int(tarif)
    else:
        shetchik_status_pay = 'нет'
        about_pokaz_pay, month_pokaz_pay, ust_pay, firma_pay = ('-', ',', '-', '-')
        itog_pay = int(plosh_pay) * int(tarif)
for_zapros = "Lgot.ID_sobstv LIKE'" + str(info_sobstvinn_id) + "%'"
zapros = ("SELECT Discount FROM Lgot WHERE(%s);"" % for_zapros)
cursor.execute(zapros)
info5_sobstvinn = cursor.fetchall()
if info5_sobstvinn:
    discount_pay = info5_sobstvinn[0][0]
    itog_discount_pay = itog_pay * 10 // 100
else:
    discount_pay = 'нет'
    itog_discount_pay = 0
oplata = itog_pay - itog_discount_pay
data_pay = form.data_pay.data

doc = DocxTemplate('C:/Users/79299/PycharmProjects/Praktika/shablon_kvit.docx')
znach = {'nazv_pay': str(nazv_pay),
         'fio_ruc': str(fio_ruc),
         'recvizit_pay': str(recvizit_pay),
         'fio_pay': str(fio_pay),
         'passport_pay': str(passport_pay),
         'adres_pay': str(adres_pay),
         'phone_pay': str(phone_pay),
         'month_pokaz_pay': str(month_pokaz_pay),
         'plosh_pay': str(plosh_pay),
         'tempa_pay': str(tempa_pay),
         'shetchik_status_pay': str(shetchik_status_pay),
         'ust_pay_data': str(ust_pay),
         'firma_pay': str(firma_pay),
         'about_pokaz_pay': str(about_pokaz_pay),

```

```

'discount_pay': str(discount_pay),
'itog_pay': str(itog_pay),
'itog_discount_pay': str(itog_discount_pay),
'oplata': str(oplata),
'data_pay': str(data_pay)}
doc.render(context=znach)
doc.save('C:/Users/79299/PycharmProjects/Praktika/otchet/kvit%s.docx' %
info_sobstvinn_id)
return redirect('/')
return render_template(temp, title=title, form=form)

# Ссылка на открытие страницы добавления нового клиента (собственника)
@app.route("/new_client", methods=['GET', 'POST']) # Добавление клиента(собственника)
def new_client():
    form = NewClient() # обращениек классу формы
    temp = 'new_client.html' # имя шаблона
    title = 'Добавление клиента' # заголовок страницы
    if form.validate_on_submit():
        message = chek_pasport(form.pasport.data) # проверка корректности паспорта
        if message != "":
            return render_template(temp, title=title, form=form, message=message)

        for_zapros = "Sobstv.Passport LIKE '%" + form.pasport.data + "%'"
        zapros = ("SELECT FIO_sobstv FROM Sobstv WHERE(%s);%" % for_zapros) # запрос на
        наличие собственника в БД
        cursor.execute(zapros)
        info_sobstvinn = cursor.fetchall()

        if info_sobstvinn:
            return render_template(temp, title=title, form=form, message="Клиент с такими
            паспортными данными существует!")
            message = chek_data(form.data_birthday.data) # проверка корректности введенной даты
            if message != "":
                return render_template(temp, title=title, form=form, message=message)
            message = chek_phone(form.phone.data)
            if message != "":
                return render_template(temp, title=title, form=form, message=message)
            message = chek_fio(form.fio_sobstv.data)
            if message != "":
                return render_template(temp, title=title, form=form, message=message)
            if len(form.fio_sobstv.data) > 40:
                return render_template(temp, title=title, form=form,
                message="В поле ФИО Собственника может быть не больше 40
                символов! Просьба сократить инициалы!")
            if (form.gender.data.lower() != 'м') and (form.gender.data.lower() != 'ж'):
                return render_template(temp, title=title, form=form, message="Пол необходимо
                указать одной буквой: м/ж!")

        zapros = ("WITH SRC AS (SELECT TOP (1) ID_Sobstv FROM Sobstv ORDER BY
        ID_Sobstv DESC
        SELECT ID_Sobstv FROM SRC ORDER BY ID_sobstv")
```

```

cursor.execute(zapros)
info_sobstvinn = cursor.fetchall()
id_for_new_sobstv = int(info_sobstvinn[0][0]) + 1
sobstv = (form.fio_sobstv.data.strip()).split()
fio_new_client = sobstv[0][0].upper() + sobstv[0][1::].lower() + ' ' + sobstv[1][0].upper() +
sobstv[1][1::].lower()
if len(sobstv) == 3:
    fio_new_client += ' ' + (sobstv[2][0].upper() + sobstv[2][1::].lower())
zapros = ("INSERT INTO Sobstv(ID_sobstv, Data_birthday, Phone, FIO_sobstv, Passport,
Gender)
VALUES (%s, '%s', '%s', '%s', '%s')"" %
(id_for_new_sobstv, str(form.data_birthday.data), str(form.phone.data),
str(fio_new_client),
str(form.pasport.data), str(form.gender.data.lower())))
cursor.execute(zapros)
connection.commit()
return redirect("/new_flat")
return render_template(temp, title=title, form=form)

```

```

# Ссылка на открытие страницы добавление новой квартиры
@app.route("/new_flat", methods=['GET', 'POST']) # Добавление квартиры
def new_flat():
    form = NewFlat() # обращениек классу формы
    temp = 'new_flat.html' # имя шаблона
    title = 'Добавление квартиры' # заголовок страницы
    if form.validate_on_submit():
        if chek_bukv(form.gorod.data) == False:
            return render_template(temp, title=title, form=form,
                                   message="В поле Город присутствуют посторонние символы!")
        if chek_bukv(form.street.data) == False:
            return render_template(temp, title=title, form=form,
                                   message="В поле Улица присутствуют посторонние символы!")
        if '--' in form.gorod.data:
            return render_template(temp, title=title, form=form,
                                   message="В поле Город присутствуют два --!")
        if '--' in form.street.data:
            return render_template(temp, title=title, form=form,
                                   message="В поле Улица присутствуют два --!")
        if str(form.gorod.data)[0] == '-':
            return render_template(temp, title=title, form=form, message='В поле Город квартиры
не может быть "-" первым!')
        if str(form.street.data)[0] == '-':
            return render_template(temp, title=title, form=form, message='В поле Улица квартиры
не может быть "-" первым!')
        if chek_number(form.dom.data) == False:
            return render_template(temp, title=title, form=form,
                                   message="В поле Номер дома присутствуют посторонние символы!")
        if chek_number(str(form.dom.data)[0]) == False:
            return render_template(temp, title=title, form=form,
                                   message='В поле Номер дома присутствуют посторонние символы!')
        if str(form.dom.data)[0] == ':':

```

```

        return render_template(temp, title=title, form=form,
                               message='В поле Номер дома присутствуют посторонние символы!')
    if str(form.dom.data)[0] == '0':
        render_template(temp, title=title, form=form,
                        message='Номер дома не может начинаться с 0!')

    if chek_number(form.id_flat.data) == False:
        return render_template(temp, title=title, form=form,
                               message="В поле Номер квартиры присутствуют посторонние
                                         символы!")
    if chek_number(str(form.id_flat.data)[0]) == False:
        return render_template(temp, title=title, form=form,
                               message='В поле Номер квартиры присутствуют посторонние
                                         символы!')
    if str(form.id_flat.data)[0] == ':':
        return render_template(temp, title=title, form=form,
                               message='В поле Номер квартиры присутствуют посторонние
                                         символы!')
    if str(form.id_flat.data)[0] == '0':
        return render_template(temp, title=title, form=form, message='Номер квартиры не
                                         может начинаться с 0!!')
    if '.' in form.id_flat.data:
        return render_template(temp, title=title, form=form, message='В поле номер квартиры
                                         нельзя использовать "."')
    if chek_number(form.other_info.data) == False:
        return render_template(temp, title=title, form=form,
                               message="В поле Площадь квартиры присутствуют посторонние
                                         символы!")
    if chek_number(str(form.other_info.data)[0]) == False:
        return render_template(temp, title=title, form=form,
                               message='В поле Площадь квартиры присутствуют посторонние
                                         символы!')
    if str(form.other_info.data)[0] == '0':
        return render_template(temp, title=title, form=form, message='В поле Площадь
                                         квартиры не может быть 0 первым!')
    if str(form.other_info.data)[0] == ':':
        return render_template(temp, title=title, form=form, message='В поле Площадь
                                         квартиры не может быть ":" первым!')
    if len(form.other_info.data) > 4:
        return render_template(temp, title=title, form=form,
                               message="В поле Площадь можно указать 4 символа!")
    if '.' in str(form.other_info.data):
        flat_sqr = (form.other_info.data).split('.')
        if len(flat_sqr) == 2:
            flat_sqr = '.'.join(flat_sqr)
        else:
            return render_template(temp, title=title, form=form,
                                   message='В поле Площадь можно использовать одну "."')
    else:
        flat_sqr = form.other_info.data
    if str(form.temperatura.data)[0] != '+' and str(form.temperatura.data)[0] != '-':
        return render_template(temp, title=title, form=form,

```

```

        message="В поле Температура необходимо указать +/- перед
значением температуры!")
    if str(form.temperatura.data)[0] == '.':
        return render_template(temp, title=title, form=form, message='В поле Температура "."
не может быть первым символом!')
    if '.' in str(form.temperatura.data):
        temperatura = (form.temperatura.data).split('.')
        if len(temperatura) == 2:
            temperatura = '.join(temperatura)
        else:
            return render_template(temp, title=title, form=form,
                                   message='В поле Температура можно использовать одну ".")'
    else:
        temperatura = form.temperatura.data
    if len(temperatura.split('+')) > 2:
        return render_template(temp, title=title, form=form, message="Недопустимое значение
температуры!")
    if len(temperatura.split('-')) > 2:
        return render_template(temp, title=title, form=form, message="Недопустимое значение
температуры!")
    if len(temperatura) > 3:
        return render_template(temp, title=title, form=form, message="В поле температура
можно писать не более 3 символов!")
    if '+' in temperatura:
        if '-' in temperatura:
            return render_template(temp, title=title, form=form, message="Недопустимое
значение температуры!")
        if '-' in temperatura:
            if '+' in temperatura:
                return render_template(temp, title=title, form=form, message="Недопустимое
значение температуры!")
            if float(temperatura) >= 100 or float(temperatura) <= -100:
                return render_template(temp, title=title, form=form, message="Недопустимое значение
температуры!")
        if (str(form.schetchik.data).lower() != 'да') and (str(form.schetchik.data).lower() != 'нет'):
            return render_template(temp, title=title, form=form,
                                   message="В поле Наличие счетчика необходимо написать да/нет!")
    zapros = ("SELECT ID_company FROM Company WHERE(Company.Nazv=%s)" % (""
+ form.company.data + ""))
    cursor.execute(zapros)
    info_id_company = cursor.fetchall()
    if not info_id_company:
        return render_template(temp, title=title, form=form, message="Ошибка в названии
обслуживающей компании!")
    zapros = ("WITH SRC AS (SELECT TOP (1) ID_Sobstv FROM Sobstv ORDER BY
ID_Sobstv DESC)
SELECT ID_Sobstv FROM SRC ORDER BY ID_sobstv")
    cursor.execute(zapros)
    info_sobstvinn = cursor.fetchall()
    gorod = str(form.gorod.data).strip()
    street = str(form.street.data).strip()

```

```

    new_adress = 'г. ' + gorod[0].upper() + gorod[1::].lower() + ' ул. ' + street[0].upper() +
street[1::].lower() + \
        '' + str(form.dom.data) + ' кв. ' + str(form.id_flat.data)
    if len(new_adress) > 40:
        return render_template(temp, title=title, form=form, message="Число символов в
адресе не должно привышать 40!")
            " Введите сокращенно поле город, улица!")
    zapros = ("INSERT INTO Flat(ID_flat, Adress, Other_info, Temperatura, ID_sobstv,
ID_company)
VALUES (%s, '%s', '%s', '%s', %s, %s)" % (info_sobstvinn[0][0], str(new_adress),
str(flat_sqr),
str(temperatura), info_sobstvinn[0][0], info_id_company[0][0]))
    cursor.execute(zapros)
    connection.commit()
    if str(form.temperatura.data)[-1] == ':':
        return render_template(temp, title=title, form=form, message="В поле Температура ::
не может быть последней!")
    if int(form.temperatura.data) < 18:
        zapros = ("INSERT INTO Lgot(ID_lgot, ID_sobstv, Discount) VALUES (%s, %s, %s);"
%
            (info_sobstvinn[0][0], info_sobstvinn[0][0], "'10%')")
        cursor.execute(zapros)
        connection.commit()
    if form.schetchik.data.lower() == 'нет':
        return redirect('/')
    elif form.schetchik.data.lower() == 'да':
        return redirect('/new_schetchik')
    return render_template(temp, title=title, form=form)

```

```

# Ссылка на открытие страницы на добавление нового счетчика
@app.route("/new_schetchik", methods=['GET', 'POST']) # Добавление счетчика
def new_schetchik():
    form = NewSchetchik() # обращениек классу формы
    temp = 'new_schetchik.html' # имя шаблона
    title = 'Добавление счетчика' # заголовок страницы
    if form.validate_on_submit():
        message = chek_data(form.date_ustanovk.data)
        if message != "":
            return render_template(temp, title=title, form=form, message=message)
        if chek_bukv(form.firma.data) == False:
            return render_template(temp, title=title, form=form, message='В поле Фирма
присутствуют посторонние символы!')
        if len(form.firma.data) == 25:
            return render_template(temp, title=title, form=form,
                message='В поле Фирма не может быть больше 25 символов! Ведите
с сокращением!')
        zapros = ("WITH SRC AS (SELECT TOP (1) ID_flat FROM Flat ORDER BY ID_sobstv
DESC)
            SELECT ID_flat FROM SRC ORDER BY ID_flat")
        cursor.execute(zapros)
        info_about_flat = int(cursor.fetchall()[0][0])

```

```

zapros = ("INSERT INTO Schetchik(ID_schetchik, ID_flat, Date_ustanovk, Firma)
VALUES (%s, %s, %s, %s)"
         % (info_about_flat, info_about_flat, str(form.date_ustanovk.data),
            str(form.firma.data)))
cursor.execute(zapros)
connection.commit()
return redirect('/add_pokaz')
return render_template(temp, title=title, form=form)

# Ссылка на открытие страницы для добавления показаний счетчика
@app.route("/add_pokaz", methods=['GET', 'POST']) # Добавление показаний счетчика
def add_pokaz():
    form = AddPokaz() # обращениек классу формы
    temp = 'add_pokaz.html' # имя шаблона
    title = 'Добавление показаний счетчик' # заголовок страницы
    all_month = ['январь', 'февраль', 'март', 'апрель', 'май', 'июнь', 'июль', 'август', 'сентябрь',
                'октябрь', 'ноябрь', 'декабрь']
    if form.validate_on_submit():
        message = chek_pasport(form.pasport.data)
        if message != "":
            return render_template(temp, title=title, form=form, message=message)
        if chek_number(form.pokaz.data) == False:
            return render_template(temp, title=title, form=form,
                                   message="В поле Показания счетчика присутствуют посторонние
символы!")
        if chek_number(str(form.pokaz.data)[0]) == False:
            return render_template(temp, title=title, form=form,
                                   message='В поле Показания счетчика посторонние символы!')
        if str(form.pokaz.data)[0] == ':':
            return render_template(temp, title=title, form=form,
                                   message='В поле Показания счетчика посторонние символы!')
        if len(form.pokaz.data.split('.')) > 2:
            return render_template(temp, title=title, form=form, message='В поле Показания
счетчика присутствуют несколько "."!')
        if str(form.pokaz.data)[0] == '0':
            return render_template(temp, title=title, form=form,
                                   message='Показания счетчика не могут начинаться с 0!')
        if len(form.pokaz.data) > 10:
            return render_template(temp, title=title, form=form,
                                   message='В поле Показания счетчика введено большое значение!')
    month_pokaz = 13
    for i in range(12):
        if form.month_pokaz.data.lower() == all_month[i]:
            month_pokaz = i+1
            break
    if month_pokaz == 13:
        return render_template(temp, title=title, form=form, message="Месяц показаний указан
неверно!")

for_zapros = "Sobstv.Passport LIKE '%" + form.pasport.data + "%'"
zapros = ("SELECT ID_flat FROM Flat WHERE Flat.ID_sobstv =

```

```

(SELECT ID_sobstv FROM Sobstv WHERE(%s))"" % for_zapros)
cursor.execute(zapros)
about_client = cursor.fetchall()
if about_client == []:
    return render_template(temp, title=title, form=form, message="Информация о данном
собственнике не имеется!")

zapros = ("SELECT ID_schetchik FROM Schetchik WHERE Schetchik.ID_flat = %s"" %
about_client[0][0])
cursor.execute(zapros)
id_schetchik = cursor.fetchall()
if id_schetchik != []:
    zapros = ("SELECT Month_pokaz FROM Pokaz WHERE Pokaz.ID_schetchik=%s"" %
id_schetchik[0][0])
    cursor.execute(zapros)
    about_month = cursor.fetchall()
    if about_month != []:
        if form.month_pokaz.data.lower() == str(about_month[0][0]).lower():
            return render_template(temp, title=title, form=form,
                                   message="Информация о показаниях за данный месяц
имеются!!")
        id_pokaz = int(str(id_schetchik[0][0]) + '0' + str(month_pokaz))
        zapros = ("INSERT INTO Pokaz(ID_pokaz, Adout_pokaz, ID_schetchik, Month_pokaz)
VALUES (%s, '%s', '%s', '%s')"" %
(id_pokaz, str(form.pokaz.data), id_schetchik[0][0],
(str(form.month_pokaz.data)[0].upper() +
str(form.month_pokaz.data)[1::].lower())))
        cursor.execute(zapros)
        connection.commit()
        return redirect('/')
    else:
        return render_template(temp, title=title, form=form,
                               message="Информация о счетчике в данной квартире не имеется!")
return render_template(temp, title=title, form=form)

# Ссылка на открытие страницы изменение показаний счетчика
@app.route("/change_pokaz", methods=['GET', 'POST']) # Изменение показаний счетчика
def change_pokaz():
    form = ChangePokaz()
    temp = "change_pokaz.html"
    title = 'Изменение показаний счетчика'
    if form.validate_on_submit():
        message = chek_pasport(form.pasport.data)
        if message != "":
            return render_template(temp, title=title, form=form, message=message)

        if chek_number(form.pokaz.data) == False:
            return render_template(temp, title=title, form=form,
                                   message="В поле Показания счетчика присутствуют посторонние
символы!")

```

```

if chek_number(str(form.pokaz.data)[0]) == False:
    return render_template(temp, title=title, form=form,
                           message='В поле Показания счетчика посторонние символы!')
if str(form.pokaz.data)[0] == ':':
    return render_template(temp, title=title, form=form,
                           message='В поле Показания счетчика посторонние символы!')
if len(form.pokaz.data.split('.')) > 2:
    return render_template(temp, title=title, form=form, message='В поле Показания
счетчика присутствуют несколько ".!"')
if str(form.pokaz.data)[0] == '0':
    return render_template(temp, title=title, form=form,
                           message='Показания счетчика не могут начинаться с 0!')
if len(form.pokaz.data) > 10:
    return render_template(temp, title=title, form=form,
                           message='В поле Показания счетчика введено большое значение!')

all_month = ['январь', 'февраль', 'март', 'апрель', 'май', 'июнь', 'июль', 'август', 'сентябрь',
             'октябрь', 'ноябрь', 'декабрь']
month_pokaz = 13
for i in range(12):
    if form.month_pokaz.data.lower() == all_month[i]:
        month_pokaz = i + 1
        break
    if month_pokaz == 13:
        return render_template(temp, title=title, form=form, message="Месяц показаний указан
неверно!")

for_zapros = "Sobstv.Passport LIKE '%" + form.pasport.data + "%'"
zapros = ("SELECT ID_flat FROM Flat WHERE Flat.ID_sobstv =
(SELECT ID_sobstv FROM Sobstv WHERE(%s))" % for_zapros)
cursor.execute(zapros)
about_client = cursor.fetchall()
if about_client == []:
    return render_template(temp, title=title, form=form, message="Информация о данном
собственнике не имеется!")

zapros = ("SELECT ID_schetchik FROM Schetchik WHERE Schetchik.ID_flat = %s" % about_client[0][0])
cursor.execute(zapros)
id_schetchik = cursor.fetchall()
if id_schetchik != []:
    zapros = ("SELECT Month_pokaz FROM Pokaz WHERE Pokaz.ID_schetchik=%s" % id_schetchik[0][0])
    cursor.execute(zapros)
    about_month = cursor.fetchall()
    if form.month_pokaz.data.lower() != str(about_month[0][0]).lower():
        return render_template(temp, title=title, form=form,
                               message="Информация о показаниях за данный месяц не
имеются!")
    id_pokaz = int(str(id_schetchik[0][0]) + '0' + str(month_pokaz))
    zapros = ("UPDATE Pokaz SET Adout_pokaz=%s WHERE (ID_schetchik=%s AND
ID_pokaz=%s)"
```

```

        % (str(form.pokaz.data), id_schetchik[0][0], id_pokaz))
    cursor.execute(zapros)
    connection.commit()
    return redirect('/')
else:
    return render_template(temp, title=title, form=form,
                           message="Информация о счетчике в данной квартире не имеется!")
return render_template(temp, title=title, form=form)

# Ссылка на открытие страницы информации о компаниях
@app.route("/about_company") # Информация о компаниях
def about_company():
    zapros = "SELECT * FROM Company"
    cursor.execute(zapros)
    info_all_company = cursor.fetchall()
    return render_template("about_company.html", title='Данные компаний',
all_company=info_all_company)

# Ссылка на открытие страницы изменение информации о компании
@app.route("/change_info_company", methods=['GET', 'POST']) # Изменение информации о
компании
def change_info_company():
    form = ChangeInfoCompany() # обращение к классу формы
    temp = 'change_info_company.html' # имя шаблона
    title = 'Изменение информации о компании' # заголовок страницы
    if form.validate_on_submit():
        message = chek_bukv((form.nazv_old.data).lower())
        if message == False:
            return render_template(temp, title=title, form=form,
                                   message="В поле Название присутствуют посторонние символы!")
        for_zapros = """ + str(form.nazv_old.data) + """
        zapros = ("SELECT ID_company FROM Company WHERE Company.Nazv=%s" %
for_zapros)
        cursor.execute(zapros)
        company_id = cursor.fetchall()
        if company_id == []:
            return render_template(temp, title=title, form=form, message='Не найдена компания с
таким названием!')
        if chek_number(form.tariff.data) == False:
            return render_template(temp, title=title, form=form,
                                   message="В поле Тариф присутствуют посторонние символы!")
        if chek_number(str(form.tariff.data)[0]) == False:
            return render_template(temp, title=title, form=form,
                                   message='В поле Тариф присутствуют посторонние символы!')
        if str(form.tariff.data)[0] == ':':
            return render_template(temp, title=title, form=form,
                                   message="В поле Тариф присутствуют посторонние символы!")
        if str(form.tariff.data)[0] == '0':
            return render_template(temp, title=title, form=form,
                                   message='Тариф не может быть 0p.!')

```

```

if len(form.tariff.data.split('.')) > 2:
    return render_template(temp, title=title, form=form, message='В поле Тариф
присутствуют несколько ".!"')
if float(form.tariff.data) > 70:
    return render_template(temp, title=title, form=form,
                           message='Тариф не может быть дороже 70p.!')
if '.' in str(form.tariff.data):
    company_tarif = (form.tariff.data).split('.')
    if len(company_tarif) == 2:
        company_tarif = '.join(company_tarif)
    else:
        return render_template(temp, title=title, form=form,
                               message='В поле Тариф можно использовать одну ".!"')
else:
    company_tarif = form.tariff.data
if '.' in form.recvizit.data:
    return render_template(temp, title=title, form=form, message='В поле Реквизиты
присутствуют посторонние символы!')
if len(form.recvizit.data) != 20:
    return render_template(temp, title=title, form=form, message='В поле Реквизиты
должно быть 20 цифр!')
message = chek_number(form.recvizit.data)
if message == False:
    return render_template(temp, title=title, form=form,
                           message='В поле Реквизиты присутствуют посторонние символы!')
message = chek_fio(form.fio_ruck.data)
if message != '':
    return render_template(temp, title=title, form=form, message=message)
if len(form.fio_ruck.data) > 40:
    return render_template(temp, title=title, form=form,
                           message='В поле ФИО Руководителя может быть не больше 40
символов! Просьба сократить инициалы!')
zapros = ("UPDATE Company SET Tariff=%s, Recvizit='%s', FIO_Ruck='%s' WHERE
Nazv=%s")
    % (company_tarif, form.recvizit.data, form.fio_ruck.data.strip(), for_zapros))
cursor.execute(zapros)
connection.commit()
return redirect('/')
return render_template(temp, title=title, form=form)

```

```

# Ссылка на открытие страницы изменение информации о клиента
@app.route("/change_info_client", methods=['GET', 'POST']) # Изменение информации о
клиенте(собственнике)
def change_info_client():
    form = ChangeInfoClient() # обращение к классу формы
    temp = 'change_info_client.html' # имя шаблона
    title = 'Изменение информации о клиенте' # заголовок страницы
    if form.validate_on_submit():
        message = chek_pasport(form.pasport_old.data)
        if message != '':
            return render_template(temp, title=title, form=form, message=message)

```

```

for_zapros = "Sobstv.Passport LIKE%" + form.pasport_old.data + "%"
zapros = ("SELECT ID_sobstv FROM Sobstv WHERE(%s);" % for_zapros)
cursor.execute(zapros)
info_sobstvinn_id = cursor.fetchall()
if not info_sobstvinn_id:
    return render_template(temp, title=title, form=form,
                           message="Клиент с такими паспортными данными не существует!")
message = chek_data(form.data_birthday.data) # проверка корректности введенной даты
if message != "":
    return render_template(temp, title=title, form=form, message=message)

message = chek_phone(form.phone.data)
if message != "":
    return render_template(temp, title=title, form=form, message=message)

message = chek_pasport(form.pasport.data)
if message != "":
    return render_template(temp, title=title, form=form, message=message)

new_data_birthday = "" + form.data_birthday.data + ""
new_phone = "" + form.phone.data + ""
new_pasport = "" + form.pasport.data + ""
zapros = ("UPDATE Sobstv SET Data_birthday=%s, Phone=%s, Passport=%s WHERE
ID_sobstv=%s"
         % (new_data_birthday, new_phone, new_pasport, str(info_sobstvinn_id[0][0])))
cursor.execute(zapros)
connection.commit()
return redirect('/')
return render_template(temp, title=title, form=form)

```

```

# Ссылка на открытие страницы удаление клиента
@app.route("/delete_client", methods=['GET', 'POST']) # Удаление клиента(собственника)
def delete_client():
    form = DeleteClient() # обращение к классу формы
    temp = 'delete_client.html' # имя шаблона
    title = 'Удаление клиента' # заголовок страницы
    if form.validate_on_submit():
        message = chek_pasport(form.pasport.data) # проверка корректности паспорта
        if message != "":
            return render_template(temp, title=title, form=form, message=message)
        for_zapros = "Sobstv.Passport LIKE%" + form.pasport.data + "%"
        zapros = ("SELECT ID_sobstv FROM Sobstv WHERE(%s);" % for_zapros) # запрос на
        наличие собственника в БД
        cursor.execute(zapros)
        info_sobstvinn = cursor.fetchall()

        if info_sobstvinn:
            # проверка наличия счетчиков
            zapros1 = ("SELECT ID_flat FROM Flat WHERE Flat.ID_sobstv = %s" %
info_sobstvinn[0][0])
            cursor.execute(zapros1)

```

```

id_flat_schetchik = cursor.fetchall()
# удаление информации о счетчиках
if id_flat_schetchik:
    for_zapros = """ + str(id_flat_schetchik[0][0]) + """
    zapros = ("DELETE FROM Pokaz WHERE ID_schetchik = %s" % for_zapros) #
    удаление показаний счетчика
    cursor.execute(zapros)
    connection.commit()

    zapros = ("DELETE FROM Schetchik WHERE ID_flat = %s" % for_zapros) #
    удаление информации о счетчике
    cursor.execute(zapros)
    connection.commit()
    # удаление информации о квартире
    zapros = ("DELETE FROM Flat WHERE ID_flat = %s" % info_sobstvinn[0][0]) #
    удаление инф-ции о квартире
    cursor.execute(zapros)
    connection.commit()

    zapros1 = ("SELECT ID_lgot FROM Lgot WHERE Lgot.ID_sobstv = %s" % info_sobstvinn[0][0])
    cursor.execute(zapros1)
    id_lgot = cursor.fetchall()
    # удаление информации о льготах
    if id_lgot:
        zapros = ("DELETE FROM Lgot WHERE ID_sobstv = %s" % id_lgot[0][0])
        cursor.execute(zapros)
        connection.commit()
        # удаление клиента(собственника)
        for_zapros = """ + form.pasport.data + """
        zapros = ("DELETE FROM Sobstv WHERE Passport = %s" % for_zapros) # удаление
        собственника (клиента)
        cursor.execute(zapros)
        connection.commit()
        return redirect('/') # переход на основную страницу
    else:
        return render_template(temp, title=title, form=form,
                               message='Клиент с такими паспортными данными не существует!')
return render_template(temp, title=title, form=form)

```

```

# Ссылка на открытие страницы информации о клиенте
@app.route("/about_client", methods=['GET', 'POST']) # Информация о
клиенте(собственнике)
def about_client():
    form = AboutClient() # обращение к классу формы
    temp = 'about_client.html' # имя шаблона
    title = 'Данные клиента' # заголовок страницы
    if form.validate_on_submit():
        message = chek_pasport(form.pasport.data) # проверка корректности паспорта
        if message != '':
            return render_template(temp, title=title, form=form, message=message)

```

```

for_zapros = "Sobstv.Passport LIKE%" + form.pasport.data + "%"
zapros = ("SELECT FIO_sobstv FROM Sobstv WHERE(%s);%" % for_zapros) # запрос на
наличие собственника в БД
cursor.execute(zapros)
info_sobstvinn = cursor.fetchall()

if info_sobstvinn == []: # проверка наличия собственника в БД
    message = "Клиент с такими паспортными данными не существует!"
    return render_template(temp, title=title, form=form, message=message)
# запросы для получения информации о собственнике
for_zapros = "" + form.pasport.data + ""
zapros = ("SELECT FIO_sobstv, Data_birthday, Phone, Passport, Gender FROM Sobstv
WHERE Sobstv.Passport=%s"
         % for_zapros)
cursor.execute(zapros)
info = cursor.fetchall()[0]

zapros = ("SELECT ID_sobstv FROM Sobstv WHERE Sobstv.Passport=%s" %
for_zapros)
cursor.execute(zapros)
id_sobstv = cursor.fetchall()[0][0]

for_zapros = "" + str(id_sobstv) + ""
zapros = ("SELECT Adress, Other_info FROM Flat WHERE Flat.ID_sobstv=%s" %
for_zapros)
cursor.execute(zapros)
info += cursor.fetchall()[0]

zapros = ("SELECT ID_flat FROM Flat WHERE Flat.ID_sobstv=%s" % for_zapros)
cursor.execute(zapros)
id_flat = cursor.fetchall()[0][0]

for_zapros = "" + str(id_flat) + ""
zapros = ("SELECT Date_ustanovk, Firma FROM Schetchik WHERE
Schetchik.ID_flat=%s" % for_zapros)
cursor.execute(zapros)
vrem = cursor.fetchall()
if vrem:
    info += vrem[0]
return render_template('info_client.html', title=info_sobstvinn[0][0], item=info,
dlina=len(info))
return render_template(temp, title=title, form=form)

# Обработка 404 ошибки - страница не найдена
@app.errorhandler(404)
def not_found(error):
    return make_response(jsonify({'error': 'Not found'}), 404)

if __name__ == '__main__':
    main()

```

6.2 Листинг файла about.html

```
<!doctype html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="/static/css/style.css"/>
    <meta charset="utf-8">
    <title>{ { title } }</title>
</head>
<body>
<div class="bordered-link">
<main role="main">
    <div class="dобр block_align">{ { title } }</div>
    <div>{ % block content % }{ % endblock % }</div>
</main>
</div>
</body>
</html>
```

6.3 Листинг файла about_client.html

```
<!doctype html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="/static/css/style.css"/>
    <meta charset="utf-8">
    <title>{ { title } }</title>
</head>
<body>
<div class="bordered-link">
    <div class="dобр block_align">
        <div>{ { title } }</div>
        <br>
        <div class="error">{ { message } }</div>
    </div>
    <form action="" method="post">
        {{ form.hidden_tag() }}
        {{ form.csrf_token }}
        <div class="block block_padding">
            {{ form.pasport.label }}
            {{ form.pasport(class="form-control") }}
            {% for error in form.pasport.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div>{{ form.submit(type="submit", class="style_btn") }}</div>
    </form>
    <a href="/"><div class="block dobr block_align">Главное меню</div></a>
</div>
</body>
</html>
```

6.4 Листинг файла about_company.html

```
{% extends "about.html" %}

{% block content %}
<div class="cont">
    {% for item in all_company%}
        <div class="block block_padding">
            <div>Название компании: {{item[1]}}</div>
            <div>Тариф: {{item[2]}} р.</div>
            <div>Реквизиты: {{item[3]}}</div>
            <div>ФИО Руководителя: {{item[4]}}</div>
        </div>
        <h1></h1>
    {% endfor %}
</div>
<a href="/"><div class="block dobr block_align">Главное меню</div></a>
{% endblock %}
```

6.5 Листинг файла add_pokaz.html

```
<!doctype html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="/static/css/style.css"/>
    <meta charset="utf-8">
    <title>{{ title }}</title>
</head>
<body>
<div class="bordered-link">
    <div class="dобр block_align">
        <div>{{ title }}</div>
        <br>
        <div>Для внесения показаний счетчика необходимо ввести паспортные данные
собственника!</div>
        <br>
        <div class="error">{{ message }}</div>
    </div>
    <form action="" method="post">
        {{ form.hidden_tag() }}
        {{ form.csrf_token }}
        <div class="block block_padding">
            {{ form.pasport.label }}
            {{ form.pasport(class="form-control") }}
            {% for error in form.pasport.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.pokaz.label }}
            {{ form.pokaz(class="form-control") }}
            {% for error in form.pokaz.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.month_pokaz.label }}
            {{ form.month_pokaz(class="form-control") }}
            {% for error in form.month_pokaz.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding dobr block_align">Необходимо указывать показания
за месяц, а не общие показания! Иначе счет за отопления будет неправильный!</div>
        <div>{{ form.submit(type="submit", class="style_btn") }}</div>
    </form>
    <a href="/"><div class="block dobr block_align">Главное меню</div></a></div>
</html>
```

6.6 Листинг файла base.html

```
<!doctype html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="/static/css/style.css"/>
    <meta charset="utf-8">
    <title>{ { title } }</title>
</head>
<body>
<div class="bordered-link block_align">
<div class="dобр">Добро пожаловать!</div>
<div>
    <a href="/itog"><div class="block">Расчет стоимости отопления (формирование документа)</div></a>
    <a href="/client"><div class="block">Работа с клиентами (Добавление, изменение, просмотр, удаление информации)</div></a>
    <a href="/company"><div class="block">Работа с компаниями (Изменение, просмотр информации)</div></a>
    <a href="/schetchik"><div class="block">Работа с показаниями на счетчиках (Добавление, изменение информации)</div></a>
</div>
</div>
</body>
</html>
```

6.7 Листинг файла change_info_client.html

```
<!doctype html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="/static/css/style.css"/>
    <meta charset="utf-8">
    <title>{{ title }}</title>
</head>
<body>
<div class="bordered-link">
    <div class="dобр block_align">
        <div>{{ title }}</div>
        <br>
        <div class="error">{{ message }}</div>
    </div>
    <form action="" method="post">
        {{ form.hidden_tag() }}
        {{ form.csrf_token }}
        <div class="dобр">Изменяемые данные (паспортные данные собственника):</div>
        <div class="block block_padding">
            {{ form.pasport_old.label }}
            {{ form.pasport_old(class="form-control") }}
            {% for error in form.pasport_old.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="dобр">Новые данные:</div>
        <div class="block block_padding">
            {{ form.data_birthday.label }}
            {{ form.data_birthday(class="form-control") }}
            {% for error in form.data_birthday.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.phone.label }}
            {{ form.phone(class="form-control") }}
            {% for error in form.phone.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.pasport.label }}
            {{ form.pasport(class="form-control") }}
            {% for error in form.pasport.errors %}

```

```
<div class="alert alert-danger" role="alert">
    {{ error }}
</div>
{% endfor %}
</div>
<div>{ { form.submit(type="submit", class="style_btn") } }</div>
</form>
<a href="/"><div class="block dobr block_align">Главное меню</div></a>
</div>
</body>
</html>
```

6.8 Листинг файла change_info_company.html

```
<!doctype html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="/static/css/style.css"/>
    <meta charset="utf-8">
    <title>{ { title } }</title>
</head>
<body>
<div class="bordered-link">
    <div class="dобр block_align">
        <div>{ { title } }</div>
        <br>
        <div class="block block_padding">В названии компании могут находиться только
        буквы!</div>
        <br>
        <div class="error">{ { message } }</div>
    </div>
    <form action="" method="post">
        {{ form.hidden_tag() }}
        {{ form.csrf_token }}
        <div class="block block_padding">
            {{ form.nazv_old.label }}
            {{ form.nazv_old(class="form-control") }}
            {% for error in form.nazv_old.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.tariff.label }}
            {{ form.tariff(class="form-control") }}
            {% for error in form.tariff.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.recvizit.label }}
            {{ form.recvizit(class="form-control") }}
            {% for error in form.recvizit.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.fio_ruck.label }}
            {{ form.fio_ruck(class="form-control") }}
```

```
{% for error in form.fio_ruck.errors %}  
<div class="alert alert-danger" role="alert">  
    {{ error }}  
</div>  
{% endfor %}  
</div>  
<div>{{ form.submit(type="submit", class="style_btn") }}</div>  
</form>  
<a href="/"><div class="block dobr block_align">Главное меню</div></a>  
</div>  
</body>  
</html>
```

6.9 Листинг файла change_pokaz.html

```
<!doctype html><html lang="en"><head>
    <link rel="stylesheet" type="text/css" href="/static/css/style.css"/>
    <meta charset="utf-8">
    <title>{ {title} }</title>
</head><body>
<div class="bordered-link">
    <div class="dopr block_align">
        <div>{ {title} }</div><br>
        <div>Для изменения показаний счетчика необходимо ввести паспортные данные
собственника!</div>
        <br>
        <div class="error">{ {message} }</div>
    </div>
    <form action="" method="post">
        {{ form.hidden_tag() }}
        {{ form.csrf_token }}
        <div class="block block_padding">
            {{ form.pasport.label }}
            {{ form.pasport(class="form-control") }}
            {% for error in form.pasport.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.pokaz.label }}
            {{ form.pokaz(class="form-control") }}
            {% for error in form.pokaz.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.month_pokaz.label }}
            {{ form.month_pokaz(class="form-control") }}
            {% for error in form.month_pokaz.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding dobr block_align">Необходимо указывать показания
за месяц, а не общие показания! Иначе счет за отопления будет неправильный!</div>
        <div>{{ form.submit(type="submit", class="style_btn") }}</div>
    </form>
    <a href="/"><div class="block dobr block_align">Главное
меню</div></a></div></body></html>
```

6.10 Листинг файла client.html

```
{% extends "about.html" %}

{% block content %}
<a href="/new_client"><div class="block">Добавление новых клиентов</div></a>
<a href="/change_info_client"><div class="block">Изменение данных существующих
клиентов</div></a>
<a href="/about_client"><div class="block">Просмотр данных клиента</div></a>
<a href="/delete_client"><div class="block">Удаление клиента</div></a>
{ % endblock %}
```

6.11 Листинг файла company.html

```
{% extends "about.html" %}

{% block content %}
<a href="/about_company"><div class="block">Просмотр данных компании</div></a>
<a href="/change_info_company"><div class="block">Изменение данных
компании</div></a>
{% endblock %}
```

6.12 Листинг файла delete_client.html

```
<!doctype html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="/static/css/style.css"/>
    <meta charset="utf-8">
    <title>{ { title } }</title>
</head>
<body>
<div class="bordered-link">
    <div class="dобр block_align">
        <div>{ { title } }</div>
        <br>
        <div class="dобр block_align">Для удаления клиента необходимо ввести его
паспортные данные!</div>
        <div class="dобр block_align">При удалении клиента информация о квартире и
счетчиках будет уничтожена!</div>
        <br>
        <div class="error">{ { message } }</div>
    </div>
    <form action="" method="post">
        {{ form.hidden_tag() }}
        {{ form.csrf_token }}
        <div class="block block_padding">
            {{ form.pasport.label }}
            {{ form.pasport(class="form-control") }}
            {% for error in form.pasport.errors %}
            <div class="alert alert-danger" role="alert">
                {{ error }}
            </div>
            {% endfor %}
        </div>
        <div>{{ form.submit(type="submit", class="style_btn") }}</div>
    </form>
    <a href="/"><div class="block dobr block_align">Главное меню</div></a>
</div>
</body>
</html>
```

6.13 Листинг файла info_client.html

```
{% extends "about.html" %}

{% block content %}
<div class="cont">
    <div>
        <div class="block block_padding">ФИО: {{item[0]}}</div>
        <div class="block block_padding">Дата Рождения: {{item[1]}}</div>
        <div class="block block_padding">Номер телефона: {{item[2]}}</div>
        <div class="block block_padding">Паспортные данные: {{item[3]}}</div>
        <div class="block block_padding">Пол: {{item[4]}}</div>
        <div class="block block_padding">Адрес проживания: {{item[5]}}</div>
        <div class="block block_padding">В квартире {{item[6]}} кв.м.</div>
        {% if dlina > 7 %}
            <div class="block block_padding">Счетчик: установлен</div>
            <div class="block block_padding">Дата установки счетчика: {{item[7]}}</div>
            <div class="block block_padding">Фирма счетчика: {{item[8]}}</div>
        {% else %}
            <div class="block block_padding">Счетчик: не установлен</div>
        {% endif %}
    </div>
    <a href="/"><div class="block dobr block_align">Главное меню</div></a>
</div>
{% endblock %}
```

6.14 Листинг файла itog.html

```
<!doctype html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="/static/css/style.css"/>
    <meta charset="utf-8">
    <title>{{ title }}</title>
</head>
<body>
<div class="bordered-link">
    <div class="dобр block_align">
        <div>{{ title }}</div>
        <br>
        <div class="error">{{ message }}</div>
    </div>
    <form action="" method="post">
        {{ form.hidden_tag() }}
        {{ form.csrf_token }}
        <div class="block block_padding">
            {{ form.data_pay.label }}
            {{ form.data_pay(class="form-control") }}
            {% for error in form.data_pay.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.pasport.label }}
            {{ form.pasport(class="form-control") }}
            {% for error in form.pasport.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.month.label }}
            {{ form.month(class="form-control") }}
            {% for error in form.month.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div>{{ form.submit(type="submit", class="style_btn") }}</div>
    </form>
    <a href="/"><div class="block dobr block_align">Главное меню</div></a>
</div>
</body></html>
```

6.15 Листинг файла new_client.html

```
<!doctype html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="/static/css/style.css"/>
    <meta charset="utf-8">
    <title>{ { title } }</title>
</head>
<body>
<div class="bordered-link">
    <div class="dобр block_align">
        <div>{ { title } }</div>
        <br>
        <div class="error">{ { message } }</div>
    </div>
    <form action="" method="post">
        {{ form.hidden_tag() }}
        {{ form.csrf_token }}
        <div class="block block_padding">
            {{ form.data_birthday.label }}
            {{ form.data_birthday(class="form-control") }}
            {% for error in form.data_birthday.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.phone.label }}
            {{ form.phone(class="form-control") }}
            {% for error in form.phone.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.fio_sobstv.label }}
            {{ form.fio_sobstv(class="form-control") }}
            {% for error in form.fio_sobstv.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.pasport.label }}
            {{ form.pasport(class="form-control") }}
            {% for error in form.pasport.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
```

```
</div>
{ % endfor %
</div>
<div class="block block_padding">
{{ form.gender.label }}
{{ form.gender(class="form-control") }}
{ % for error in form.gender.errors % }
<div class="alert alert-danger" role="alert">
{{ error }}
</div>
{ % endfor %
</div>
<div>{{ form.submit(type="submit", class="style_btn") }}</div>
</form>
<a href="/"><div class="block dobr block_align">Главное меню</div></a>
</div>
</body>
</html>
```

6.16 Листинг файла new_flat.html

```
<!doctype html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="/static/css/style.css"/>
    <meta charset="utf-8">
    <title>{{ title }}</title>
</head>
<body>
<div class="bordered-link">
    <div class="dобр block_align">
        <div>{{ title }}</div>
        <div class="error">{{ message }}</div>
    </div>
    <form action="" method="post">
        <div class="dобр block_align">Необходимо ввести информацию о квартире для ранее
        добавленного собственника</div>
        {{ form.hidden_tag() }}
        {{ form.csrf_token }}
        <div class="block block_padding">
            {{ form.gorod.label }}
            {{ form.gorod(class="form-control") }}
            {% for error in form.gorod.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.street.label }}
            {{ form.street(class="form-control") }}
            {% for error in form.street.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.dom.label }}
            {{ form.dom(class="form-control") }}
            {% for error in form.dom.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.id_flat.label }}
            {{ form.id_flat(class="form-control") }}
            {% for error in form.id_flat.errors %}
                <div class="alert alert-danger" role="alert">

```

```

        {{ error }}
    </div>
    {% endfor %}
</div>
<div class="block block_padding">
    {{ form.other_info.label }}
    {{ form.other_info(class="form-control") }}
    {% for error in form.other_info.errors %}
        <div class="alert alert-danger" role="alert">
            {{ error }}
        </div>
    {% endfor %}
</div>
<div class="block block_padding">
    {{ form.temperatura.label }}
    {{ form.temperatura(class="form-control") }}
    {% for error in form.temperatura.errors %}
        <div class="alert alert-danger" role="alert">
            {{ error }}
        </div>
    {% endfor %}
</div>
<div class="block block_padding">
    {{ form.schetchik.label }}
    {{ form.schetchik(class="form-control") }}
    {% for error in form.schetchik.errors %}
        <div class="alert alert-danger" role="alert">
            {{ error }}
        </div>
    {% endfor %}
</div>
<div class="block block_padding">
    {{ form.company.label }}
    {{ form.company(class="form-control") }}
    {% for error in form.company.errors %}
        <div class="alert alert-danger" role="alert">
            {{ error }}
        </div>
    {% endfor %}
</div>
<div>{{ form.submit(type="submit", class="style_btn") }}</div>
</form>
</div>
</body>
</html>
```

6.17 Листинг файла new_schetchik.html

```
<!doctype html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="/static/css/style.css"/>
    <meta charset="utf-8">
    <title>{ { title } }</title>
</head>
<body>
<div class="bordered-link">
    <div class="dобр block_align">
        <div>{ { title } }</div>
        <br>
        <div class="error">{ { message } }</div>
    </div>
    <form action="" method="post">
        {{ form.hidden_tag() }}
        {{ form.csrf_token }}
        <div class="block block_padding">
            {{ form.date_ustanovk.label }}
            {{ form.date_ustanovk(class="form-control") }}
            {% for error in form.date_ustanovk.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div class="block block_padding">
            {{ form.firma.label }}
            {{ form.firma(class="form-control") }}
            {% for error in form.firma.errors %}
                <div class="alert alert-danger" role="alert">
                    {{ error }}
                </div>
            {% endfor %}
        </div>
        <div>{{ form.submit(type="submit", class="style_btn") }}</div>
    </form>
</div>
</body>
</html>
```

6.18 Листинг файла schetchik.html

```
{% extends "about.html" %}

{% block content %}
<a href="/add_pokaz"><div class="block">Добавление показаний счетчика</div></a>
    <a href="/change_pokaz"><div class="block">Изменение показаний счетчика</div></a>
{% endblock %}
```

6.19 Листинг файла style.css

```
.bordered-link {  
    width: 70%;  
    height: 92%;  
    display: inline-block;  
    padding: 8px;  
    border: 2px solid #9479fa;  
    border-radius: 6px;  
    box-shadow:  
        0 2px 1px rgba(0, 0, 0, 0.2),  
        inset 0 2px 1px rgba(0, 0, 0, 0.2);  
    /* Шрифт */  
    font-size: 20px;  
    text-transform: uppercase;  
    color: #222;  
}  
  
.bordered-link:hover {  
    border-color: #FDD68B  
}  
  
.bordered-link:active {  
    border-color: #FEE8BD  
}  
  
.block{  
    padding: 5px;  
    margin: 30px;  
    -webkit-box-shadow: 5px 5px 20px 0px rgba(34, 60, 80, 0.3);  
    -moz-box-shadow: 5px 5px 20px 0px rgba(34, 60, 80, 0.3);  
    box-shadow: 5px 5px 20px 0px rgba(34, 60, 80, 0.3);  
}  
  
.block_align{  
    text-align: center;  
}  
  
a {  
    text-decoration: none; /* Отмена подчеркивание у ссылки */  
    color: #222;  
}  
  
body{  
    position: fixed; /* or absolute */  
    top: 2%;  
    left: 13%;  
    width: 100%;  
    height: 100%;  
}  
.dобр{  
    padding: 5px;
```

```

        margin: 15px;
    }

.block_padding{
    padding-left: 3%;
}

.style_btn{
    background: linear-gradient(to bottom, #FFFFFF, #E6E6E6) #F5F5F5 repeat-x;
    border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) #B3B3B3;
    border-radius: 4px;
    border-style: solid;
    border-width: 1px;
    box-shadow: 0 1px 0 rgba(255, 255, 255, 0.2) inset, 0 1px 2px rgba(0, 0, 0, 0.05);
    color: #333333;
    text-decoration: none;
    display: block;
    font-size: 14px;
    width: 120px;
    line-height: 20px;
    margin: 20px auto;
    padding: 4px 12px;
    text-align: center;
    text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75);
    vertical-align: middle;
    position: relative;
    -webkit-transition-duration: 0.3s;
    transition-duration: 0.3s;
    -webkit-transition-property: -webkit-transform;
    transition-property: transform;
    -webkit-tap-highlight-color: rgba(0, 0, 0, 0);
    -webkit-transform: translateZ(0);
    -ms-transform: translateZ(0);
    transform: translateZ(0);
    box-shadow: 0 0 1px rgba(0, 0, 0, 0);
}
}

.style_btn:before {
    pointer-events: none;
    position: absolute;
    z-index: -1;
    content: "";
    top: 100%;
    left: 5%;
    height: 10px;
    width: 90%;
    opacity: 0;
    background: -webkit-radial-gradient(center, ellipse, rgba(0, 0, 0, 0.35) 0%, rgba(0, 0, 0, 0) 80%);
    background: radial-gradient(ellipse at center, rgba(0, 0, 0, 0.35) 0%, rgba(0, 0, 0, 0) 80%);
    -webkit-transition-duration: 0.3s;
    transition-duration: 0.3s;
}

```

```
-webkit-transition-property: -webkit-transform, opacity;  
transition-property: transform, opacity;  
}  
  
.style_btn:hover {  
-webkit-transform: translateY(-5px);  
-ms-transform: translateY(-5px);  
transform: translateY(-5px);  
}  
.style_btn:hover:before {  
opacity: 1;  
-webkit-transform: translateY(5px);  
-ms-transform: translateY(5px);  
transform: translateY(5px);  
}  
  
.error {  
color: #ff0000;  
}
```