

Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
«Владимирский государственный университет
Имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИВлГУ)

Факультет _____ ИТР
Кафедра _____ ПИИ

КУРСОВАЯ РАБОТА

по _____ Разработка корпоративных приложений

Тема _____ Автоматизированная информационная система домоуправления

(оценка)

Руководитель

Кульков Я.Ю.
(фамилия, инициалы)

(подпись) (дата)

Члены комиссии

Студент ПИИ-119
(группа)

(подпись) (Ф.И.О.)

Тюрина П.П.
(фамилия, инициалы)

(подпись) (Ф.И.О.)

(подпись) (дата)

Муром 2022

В данной курсовой работе была разработана распределенная ИС домоуправления для автоматизации предметной области с использованием фреймворка Spring. В ходе выполнения работы была проанализирована предметная область, выявлены требования к работе, разработаны модели данных и диаграммы. На основе разработанных моделей данных был реализован набор классов и разработано приложение на языке Java в среде разработки Intel Idea 2021. На заключительном этапе работы произведено тестирование разработанного продукта.

In this course work, a distributed house management system was developed to automate the subject area using the Spring framework. In the course of the work, the subject area was analyzed, the requirements for the work were identified, data models and diagrams were developed. Based on the developed data models, a set of classes was implemented and a Java application was developed in the Intel Idea 2021 development environment. At the final stage of the work, the developed product was tested.

Содержание

Введение	6
1. Анализ технического задания	8
2. Разработка моделей данных	13
2.1 Концептуальная модель данных	13
2.2 Логическая модель данных	16
2.3 Физическая модель данных	17
3. Проектирование работы системы	22
3.1 Диаграммы UML	22
3.2 Диаграммы SADT	28
4. Разработка и реализация системы	32
5. Тестирование системы	40
Заключение	50
Список использованной литературы	51
Приложение 1. Модели данных	52
Приложение 2. Текст программы	55
Приложение 3. Снимки окон программы (скриншоты программы)	56

					МИВУ 09.03.04-13.000 ПЗ					
Изм.	Лист	№ докум.	Подпись	Дата	АИС домоуправления			Лит.	Лист	Листов
Разраб.		Тюрина П.П.								
Провер.		Кульков Я.Ю.							5	74
Реценз.								МИ ВлГУ ПИИ-119		
Н. Контр.										
Утверд.										

Введение

В настоящее время в России активно развиваются новые подходы к управлению многоквартирными домами и жилищными хозяйствами. Эти изменения направлены на улучшение, оптимизацию, повышения качества обслуживания, сокращение расходов, активизацию деятельности собственников жилья.

Жилищно-управляющая компания осуществляет услуги по поддержанию и восстановлению надлежащего состояния зданий, сооружений, оборудования, коммуникаций и объектов коммунального назначения. Собственники жилья производят оплату за эти услуги, согласно установленным нормам и тарифам.

Актуальность данной работы заключается в автоматизации деятельности жилищно-управляющей компании. Сотрудникам компании достаточно сложно работать с большим количеством бумажной документации, что тормозит все бизнес-процессы компании. Как показывает практика, любой документ может затеряться или не дойти до ответственных лиц. Кроме того, перерасчет квартплаты для квартиры каждый месяц посредством обычного калькулятора достаточно длительный и трудоемкий процесс. К тому же, поход квартиросъемщиков каждый месяц в домоуправляющую компанию только для того, чтобы передать показания или предоставить чеки также занимает немало времени. Автоматизация деятельности данной организации позволит снизить затраты на обработку документов, повысит качество работы за счет ускорения процессов начисления квартплаты, поиска, обработки и предоставления нужной информации, что в результате, обеспечит повышение эффективности работы.

Функции, которые будут автоматизированы:

– централизованное хранение информации: это снижает риск ее потери и повышает скорость ее поиска;

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

– возможность передачи показаний и загрузка чеков пользователями, которые проживают в доме, что позволит быстрее обрабатывать информацию администратору;

– начисление платежей: система будет сама начислять платежи по различным тарифам в зависимости от наличия/отсутствия счетчиков, их показаний, площади квартиры и количества прописанных жильцов.

Целью курсовой работы является разработка распределенной ИС для автоматизации предметной области с использованием фреймворка Spring.

Для достижения поставленной цели были составлены следующие задачи курсовой работы:

- проанализировать предметную область;
- разработать модели данных и диаграммы;
- разработать и реализовать АИС;
- протестировать разработанное приложение.

1 Анализ Технического Задания

В данной курсовой работе необходимо разработать автоматизированную информационную систему домоуправления. Она используется с целью учета квартир, жильцов, и оплаты услуг. Для разработки такой системы необходимы база данных и среда разработки для создания интерфейса и функций по работе с БД.

1.1 Обоснование выбора средств реализации

Для разработки системы была выбрана СУБД MySQL Workbench - приложение для визуального проектирования баз данных, в составе которого инструменты проектирования, моделирования, создания и эксплуатации БД. Удобный графический интерфейс позволяет легко создавать структуру базы любой сложности, а наглядность позволяет легко ориентироваться. Также MySQL Workbench имеет удобный редактор SQL запросов, позволяющий сразу же отправлять их серверу и получать ответ в виде таблицы.

В качестве среды для написания программы по работе с созданной в базой данных использовалась интегрированная среда разработки ПО IntelliJ IDEA, язык программирования Java. Она делает процесс разработки программного обеспечения более простым, надежным и менее подверженным ошибкам.

1.2 Аналоги

Для наиболее точного анализа предоставленного технического задания мною были найдены несколько аналогов разрабатываемого программного средства.

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

1.2.1 Домовладелец

Домовладелец [1] – это программа для предприятий ЖКХ, позволяющая рассчитывать и пересчитывать коммунальные платежи, делать любые начисления, вести отдельный учет капремонта, учитывать всю историю лицевых счетов и хранить информацию о жилом фонде, паспортном столе, бухгалтерии. Для удобства есть отдельное приложение на iOS и Android, в котором можно оплачивать счета коммунальных услуг с комиссией 0.8%.

Программа интегрируется с сервисами ГИС, СБЕРБАНК, Почта России, ВТБ, Система Город, Коммунальные Платежи Онлайн, Яндекс.деньги, МИНБ, МКБ, с соцзащитой, фондом капитального ремонта и другими сервисами. Все перерасчеты в системе происходят автоматически на основе полученных данных, поэтому ничего не нужно перепроверять.

На рисунке 1 представлен экран программы с информацией о квартирах.

The screenshot shows the 'Домовладелец' program interface. At the top, there are tabs for 'Экспликация', 'ПостУсл', 'Ремонты', 'Акты', 'Электроценты', 'Характеристики', 'ВОЖ', 'Тарифы ВОЖ', 'Услуги', 'Тарифы на усл', and 'ВЫХОД'. Below the tabs, there are input fields for 'Улица' (Батайский проезд Москва), 'Дом' (555), 'ЖСК' (1), 'Год постр' (1981), 'Серия', 'Этажей' (5), 'Подъездов' (6), 'Жилфонд', 'ЖК', 'Категория услуг и удобства', and 'Категория здания'. A 'Список квартир' window is open, showing a table with columns: '№ кв.', 'Эт', 'Под', 'Тип', 'Количество комнат' (Всего, Своё, Испол), 'ФЛС', 'Сборная', 'Коммуналка', 'Жилая', 'Общая', 'Сво-бодная', and 'Опл-мая'. The table lists 13 apartments with their respective characteristics and payment data.

№ кв.	Эт	Под	Тип	Количество комнат	ФЛС	Сборная	Коммуналка	Жилая	Общая	Сво-бодная	Опл-мая
				Всего	Своё	Испол					
1-к	1	1	ж	3	0	3	1	45.5	74.6	0.0	74.6
2-к	2	1	ж	1	0	1	1	19.1	37.9	0.0	37.9
3-к	3	1	ж	2	0	2	1	30.7	50.8	0.0	50.8
4-к	4	1	ж	2	0	2	1	33.9	57.8	0.0	57.8
5-к	5	2	ж	3	0	3	1	45.3	76.8	0.0	74.3
6-к	6	2	ж	1	0	1	1	19.1	38.5	0.0	37.7
7-к	7	2	ж	2	0	2	1	30.6	51.6	0.0	50.8
Всего	8	2	ж	2	0	2	1	34.2	60.6	0.0	58.1
Сумма	9	3	ж	3	0	3	1	45.2	76.7	0.0	74.2
	10	3	ж	1	0	1	1	19.1	38.6	0.0	37.8
	11	3	ж	2	0	2	1	30.5	51.3	0.0	50.5
	12	3	ж	2	0	2	1	33.9	60.5	0.0	58.0
	13	4	ж	3	0	3	1	45.2	76.9	0.0	74.4

Рисунок 1 – Экран программы Домовладелец

1.2.2 ИнфоКрафт ЖКХ 365

ИнфоКрафт ЖКХ 365 [2] – это облачный сервис для управляющих компаний, состоящий из нескольких программ для ведения жилищно-коммунального хозяйства. С его помощью можно начислять квартплату, вести бухгалтерский учет, составлять сметы, рассчитывать налоги и автоматизировать работу с жильцами.

В сервис входят несколько модулей для разных целей: рассчитывать квартплату и начислять коммунальные услуги можно через модуль «Формула ЖКХ», а вести бухгалтерию и считать налоги через «Формула ЖКХ + Бухгалтерия». Доступны модули для ЕИРЦ (ЕРЦ, РКЦ) и иных расчетных центров, и программы для водоканалов, энергосбытовых, теплоснабжающих и газоснабжающих предприятий.

На рисунке 2 представлен экран приложения ИнфоКрафт ЖКХ 365 с информацией о лицевых счетах.

Код	Ответственный	Участок	Статус	Комментарий
00-00000001	Синчук Эдуард Викторович	Победа п. Лесная, уч. 001	Открыт	
00-00000002	Белов Артур Сергеевич	Победа п. Лесная, уч. 002	Открыт	
00-00000003	Водольнова Марфа Руслановна	Победа п. Лесная, уч. 003	Открыт	
00-00000004	Андреева Мария Владимировна	Победа п. Лесная, уч. 004	Открыт	
00-00000005	Абдулова Зинаида Сергеевна	Победа п. Лесная, уч. 005	Открыт	
00-00000006	Захаров Петр Иванович	Победа п. Лесная, уч. 006	Открыт	
00-00000007	Ким Лариса Леонидовна	Победа п. Лесная, уч. 007	Открыт	
00-00000008	Русакowa Наталья Алексеевна	Победа п. Лесная, уч. 008	Открыт	
00-00000009	Земляникин Леонид Олегович	Победа п. Лесная, уч. 009	Открыт	
00-00000010	Петров Виктор Николаевич	Победа п. Лесная, уч. 010	Открыт	
00-00000011	Алевино Галина Васильевна	Победа п. Ягодная, уч. 001	Открыт	
00-00000012	Кемли Павел Сергеевич	Победа п. Ягодная, уч. 002	Открыт	
00-00000013	Лукин Роман Андреевич	Победа п. Ягодная, уч. 003	Закрыт	
00-00000014	Ксенофонтов Гарик Эдуардович	Победа п. Ягодная, уч. 004	Открыт	
00-00000015	Павлов Анатолий Федорович	Победа п. Ягодная, уч. 005	Открыт	
00-00000016	Кемли Павел Сергеевич	Победа п. Ягодная, уч. 003	Открыт	

Рисунок 2 - Экран программы ИнфоКрафт ЖКХ 365

1.2 Функциональные возможности

Разрабатываемое программное средство должно обеспечивать получение из базы данных всей необходимой информации в полном объеме, а также возможность её редактирования и удаления.

Разрабатываемая АИС будет хранить следующие данные о квартире:

- номер подъезда;
- номер квартиры;
- лицевой счет квартиры;
- общая площадь;
- полезная площадь;
- количество комнат;
- количество прописанных жильцов;
- количество собственников жилья.

Также приложение будет хранить такие данные, как ФИО, дата прописки и количество членов семьи о квартиросъемщике. Кроме того, в программе хранятся действующие тарифы и нормативы на услуги, которые в случае их изменения можно поменять.

На основе данных, внесенных в приложение, АИС должна производить вычисления по оплатам услуг. Так как для правильного расчета необходимы данные о счетчиках, то для этого также нужно будет реализовать хранение данных о них, а также их показания. Для расчета квартплаты, необходимо действующие тарифы умножить на показатели по потребленным ресурсам ЖКХ. В случае отсутствия счетчиков любые коммунальные платежи будут зависеть от количества зарегистрированных жителей в квартире. В этом случае квартплата будет рассчитываться следующим образом: ежемесячный норматив на одного человека умножается на количество прописанных людей и на тариф. В случае отсутствия прописанных жителей берется число, равное количеству собственников жилья. Стоит также учитывать, что при вычислении платы за

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

отопление при отсутствии счетчиков учитывается полезная площадь квартиры: норматив потребления на 1 квадратный метр умножается на площадь. Также в том случае, когда показания по счетчикам не были переданы, начисления будут проводиться по нормативам.

Данной АИС будет пользоваться несколько пользователей. Они будут разделены на роли, одна из них – администратор, - это сотрудник жилищноуправляющей компании. Данный пользователь сможет добавлять и изменять практически все данные в базе данных. Другая роль – квартиросъемщик, - это житель дома, которым управляет данная компания. Данный пользователь сможет просматривать данные только по своим квартирам, а так передавать показания и загружать чеки по различным начислениям.

АИС домоуправления должна обеспечивать следующие функциональные возможности:

- внесение, изменение и удаление данных в БД о квартире и ее квартиросъемщике;
- расчет квартплаты для жителей каждой квартиры за предыдущий месяц;
- хранение чеков об оплате услуг;
- авторизация пользователей (хранений хранение пользователей в отдельной таблице).

При реализации приложения необходимо использовать фреймворк Spring, который редставляет собой просто контейнер внедрения зависимостей, с несколькими удобными слоями (например: доступ к базе данных или веб-инфраструктура MVC). Это все позволяет быстрее и удобнее создавать Java-приложения.

2 Разработка моделей данных

Разработка моделей данных является очень важным этапом в разработке АИС, на котором выделяются сущности, атрибуты сущностей и связи между ними.

Перед разработкой моделей данных были выявлены следующие ограничения предметной области:

- у квартиры может быть только один квартиросъемщик;
- квартиросъемщик может иметь несколько квартир;
- лицевой счет и номер квартиры всегда индивидуальны;
- у квартиры всегда должен иметься хотя бы один собственник;
- показания по счетчику квартиры, могут передаваться только за прошедший месяц;
- номер счетчика квартиры не может совпадать с номером того же типа счетчика другой квартиры;
- начисления также могут проводиться только за прошедший месяц;
- по каждому из начислений должен храниться чек по оплате.

2.1 Концептуальная модель данных

Концептуальная модель данных - модель предметной области, которая описывает общую картину: что будет содержать система, как она будет организована и какие бизнес-правила будут задействованы. Как правило, она включают классы сущностей, их характеристики и ограничения, отношения между сущностями, требования к безопасности и целостности данных.

В приложении 1 расположена концептуальная модель данных.

В таблице 1 представлена информация о полях таблицы «Квартира» базе данных.

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

Таблица 1

Информация о полях таблицы «Квартира»

Поле	Описание
Dom (лицевой счет)	{строка символов длиной не более 20, символами являются цифры}
Dom (полезная площадь)	{вещественное число}
Dom (номер подъезда)	{строка символов длиной не более 3, символами являются цифры}
Dom (номер квартиры)	{строка символов длиной не более 3, символами являются цифры}
Dom (общая площадь)	{вещественное число}
Dom (количество комнат)	{строка символов длиной не более 2, символами являются цифры}
Dom (количество прописанных жильцов)	{строка символов длиной не более 2, символами являются цифры}
Dom (количество собственников жилья)	{строка символов длиной не более 2, символами являются цифры}

В таблице 3 представлена информация о полях таблицы «Начисление» в базе данных.

Таблица 3

Информация о полях таблицы «Начисление»

Поле	Описание
Dom (чек)	{строка символов, длиной не более 50}
Dom (тариф)	{строка символов длиной не более 10, символами являются цифры}
Dom (норматив)	{строка символов длиной не более 10, символами являются цифры}

Продолжение таблицы 3	
Dom (период)	{строка символов длиной не более 15}
Dom (статус оплаты)	{строка символов длиной не более 1, символами могут быть 0 и 1}
Dom (сумма)	{вещественное число с двумя знаками после запятой}

В таблице 4 представлена информация о полях таблицы «Квартиросъемщик» в базе данных.

Таблица 4

Информация о полях таблицы «Квартиросъемщик»

Поле	Описание
Dom (ФИО)	{строка символов, длиной не более 50}
Dom (дата прописки)	{дата, в которой указан год, месяц и число}
Dom (норматив)	{строка символов длиной не более 10, символами являются цифры}
Dom (количество членов семьи)	{строка символов длиной не более 3, символами являются цифры}
Dom (email)	{строка символов, длиной не более 50}
Dom (номер телефона)	{строка символов длиной не более 11, символами являются цифры}

В таблице 5 представлена информация о полях таблицы «Показание» в базе данных.

Таблица 5

Информация о полях таблицы «Показание»

Поле	Описание
Dom (значение)	{строка символов длиной не более 3, символами являются цифры}
Dom (период)	{строка символов длиной не более 15}

В таблице 6 представлена информация о полях таблицы «Счетчик» в базе данных.

Таблица 6

Информация о полях таблицы «Счетчик»

Поле	Описание
Dom (тип счетчика)	{строка символов длиной не более 20}
Dom (номер счетчика)	{строка символов длиной не более 10, символами являются цифры}
Dom (используется ли)	{строка символов длиной не более 1, символами могут быть 0 и 1}

2.2 Логическая модель данных

Логическая модель данных предоставляет более подробную информацию о концепциях и взаимосвязях. Она содержит атрибуты данных и показывает отношения между сущностями. В приложении 1 представлена логическая модель данных создаваемой АИС домоуправления.

2.3 Физическая модель данных

Физическая модель данных представляет схему того, как данные будут храниться в базе. Они предлагают окончательный дизайн, который может быть реализована база данных, включающая ассоциативные таблицы, которые иллюстрируют отношения между сущностями, а также первичные и внешние ключи. В приложении 1 представлена физическая модель данных создаваемой АИС домоуправления

Построение физической модели БД производится на основе логической модели. В таблице 7 описано соответствие сущностей логической модели и таблиц физической. Пример подробного описания каждой сущности приведено в таблицах 8 - 14. Также в этих таблицах приведен расчет памяти, необходимой для хранения одной записи.

Таблица 7

Соответствие сущностей логической и физической моделей

Сущность	Таблица
Квартиросъемщик	Tenant
Квартира	Flat
Показания	Indications
Счетчик	Counter
Оплата	Payment
Тариф	Rate
Норматив	Normative

Таблица 8

Таблица Tenant

Имя столбца	Тип	Размер (байт)
id_tenant	INT	4
full_name	VARCHAR(50)	50

Продолжение таблицы 8		
date_of_registration	DATE	3
number_of_family_members	VARCHAR(3)	3
email	VARCHAR(30)	30
phone_number	VARCHAR(11)	11
Итого		101

Таблица 9

Таблица Flat

Имя столбца	Тип	Размер (байт)
id_flat	INT	4
personal_account	VARCHAR(20)	20
flat_number	VARCHAR(3)	3
total_area	FLOAT	4
usable_area	FLOAT	4
entrance_number	VARCHAR(3)	3
number_of_rooms	VARCHAR(3)	3
number_of_registered_residents	INT	4
number_of_owners	INT	4
id_tenant	INT	4
Итого		53

Таблица 10

Таблица Indications

Имя столбца	Тип	Размер (байт)
id_indications	INT	4
period	VARCHAR(15)	15
id_counter	INT	4

Продолжение таблицы 10		
value	INT	4
Итого		27

Таблица 11

Таблица Counter

Имя столбца	Тип	Размер (байт)
id_counter	INT	4
type	VARCHAR(30)	30
id_flat	INT	4
number	VARCHAR(20)	20
used	BOOL	1
Итого		59

Таблица 12

Таблица Payment

Имя столбца	Тип	Размер (байт)
id_payment	INT	4
period	VARCHAR(15)	15
id_flat	INT	4
id_rate	INT	4
id_normative	INT	4
amount	FLOAT	4
payment_status	BOOL	1
cheque	VARCHAR(50)	50
Итого		86

Таблица 13

Таблица Rate

Имя столбца	Тип	Размер (байт)
id_rate	INT	4
name	VARCHAR(20)	20
value	FLOAT	4
Итого		28

Таблица 14

Таблица Normative

Имя столбца	Тип	Размер (байт)
id_normative	INT	4
name	VARCHAR(20)	20
value	FLOAT	4
Итого		28

Определим объем внешней памяти, необходимой для размещения данных за год использования. Для того чтобы оценить объем, занимаемый таблицами базы данных, необходимо оценить объем каждой таблицы. Примерный расчет необходимо объема памяти приведен в таблице 15.

Таким образом, при максимальном заполнении БД объем таблиц составит:
 $V_{\text{данных}} = 13,79 \text{ Мбайт.}$

Расчет объема ПЗУ для хранения данных

Таблица	Размер записи, байт	Максимальное (оценочное) количество записей	Всего, Кбайт
Tenant	101	200	19,73
Flat	53	200	10,35
Counter	59	1000	57,62
Indications	27	12000	316,41
Normative	28	5	0,14
Rate	28	5	0,14
Payment	86	12000	1007,81
Итого		25610	1412,2

Актерами на данной диаграмме являются администратор и квартиросъемщик, они – внешняя часть системы. Администратор имеет доступ ко всем реализованным функциям. Квартиросъемщик имеет доступ только к тем данным, которые относятся к его квартире.

Диаграммы DFD

В соответствии с DFD методологией, модель системы определяется как иерархия диаграмм потоков данных, описывающих процессы преобразования информации от момента ее ввода в систему до выдачи конечному пользователю. Для такого описания модели системы была создана диаграмма потоков данных, которая показана на рисунке 4.

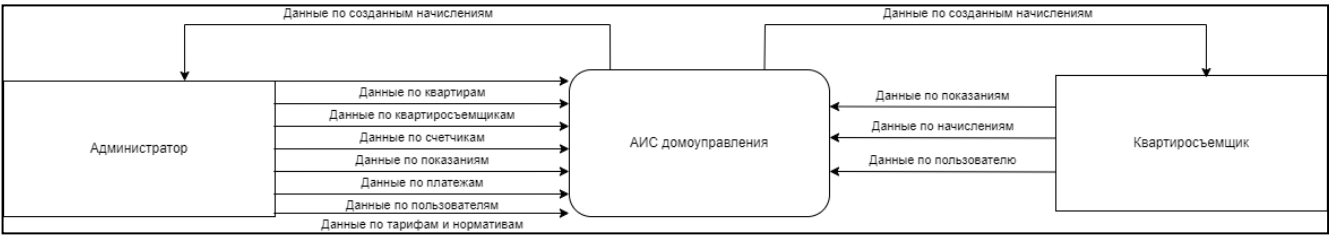


Рисунок 4 - Диаграмма потоков данных АИС домоуправления

Центральной здесь является работа АИС домоуправления. На ее вход поступают данные по квартирам, квартиросъемщикам, счетчикам, показаниям, платежам, пользователям, тарифам и нормативам. Выходом этой работы будут данные по созданным начислениям. Для раскрытия диаграммы потоков данных была построена ее подробная диаграмма. Она показана на рисунке 5.

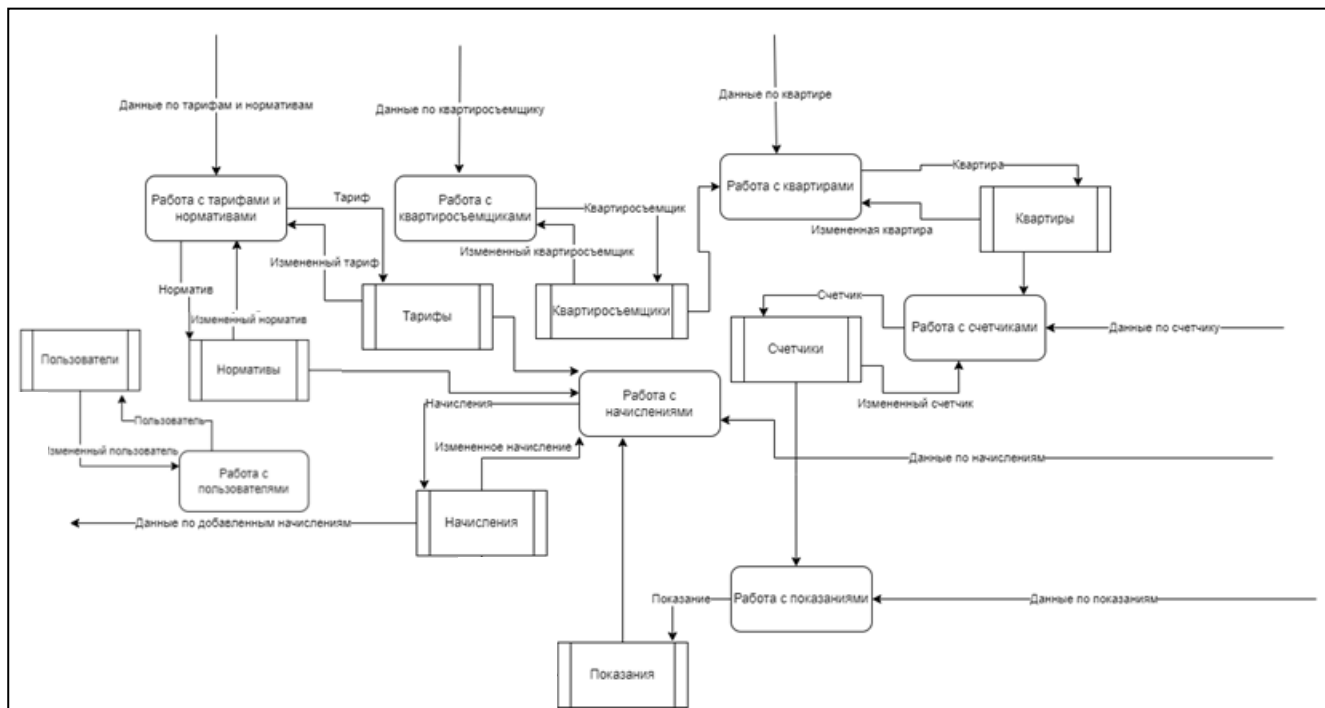


Рисунок 5 – Подробная диаграмма потоков данных

Работа "Работа с квартиросъемщиком" работает с данными по квартиросъемщику, полученными от администратора. Стрелка, соединяющая эту работу и хранилище данных "Квартиросъемщики" однонаправленная, т.к. работа только вносит информацию о квартиросъемщике. Из данного хранилища выходит стрелка, которая входит в работу «Работа с квартирами».

Работа "Работа с квартирами" работает с данными по квартире и квартиросъемщику, полученными от администратора и хранилища. Стрелка, соединяющая эту работу и хранилище данных "Квартиры" однонаправленная, т.к. работа только вносит информацию о квартире. Из данного хранилища выходит стрелка, которая входит в работы «Работа со счетчиками» и «Работа с начислениями».

Работа «Работа со счетчиками» работает с данными по счетчику, полученными от администратора и хранилища. Стрелка, соединяющая эту работу и хранилище данных "Счетчики" однонаправленная, т.к. работа только вносит информацию о счетчике. Из данного хранилища выходит стрелка, которая входит в работу «Работа с показаниями».

Работа «Работа с показаниями» работает с данными по показаниям, полученным от администратора, квартиросъемщика и хранилища. Стрелка, соединяющая эту работу и хранилище данных "Показания" однонаправленная, т.к. работа только вносит информацию о показании. Из данного хранилища выходит стрелка, которая входит в работу «Работа с начислениями».

Работа «Работа с тарифами и нормативами» работает с данными по тарифам и нормативам, полученным от администратора. Стрелки, соединяющая эту работу и хранилища данных "Тарифы" и "Нормативы" однонаправленная, т.к. работа только вносит информацию о тарифах и нормативах. Из данных хранилища выходят стрелки, которые входят в работу «Работа с начислениями».

Работа «Работа с начислениями» работает с данными по начислениям, полученным от администратора, квартиросъемщика и разных хранилищ. Стрелка, соединяющая эту работу и хранилище данных "Начисления" однонаправленная, т.к. работа только вносит информацию о начислении.

Работа «Работа с пользователями» работает с данными по пользователям, полученным от администратора, квартиросъемщика и разных хранилищ. Стрелка, соединяющая эту работу и хранилище данных "Пользователи" однонаправленная, т.к. работа только вносит информацию о счетчике.

Диаграммы последовательностей

Построим диаграммы последовательностей, которые позволяют изобразить поведение нескольких объектов в рамках одного прецедента.

Ниже на рисунке 6 изображена диаграмма последовательности работы с квартирой.

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		25



Рисунок 6 - Диаграмма последовательности работы с квартирой

На данной диаграмме администратор создает квартиру, после чего АИС сохраняет ее в БД. При просмотре своих квартир квартиросъемщиком АИС отправляет ему его квартиры. Администратор также может изменять или удалять квартиру, при этом АИС отправляет ему эту квартиру из БД, после чего происходит сохранение изменений.

Ниже на рисунке 7 показана диаграмма последовательности работы с показаниями.



Рисунок 7 - Диаграмма последовательности работы с показаниями

На данной диаграмме, как администратор, так и квартиросъемщик могут передавать показания, после чего АИС сохраняет их в БД. Также они могут просматривать различные показания, которые АИС отправляет им из БД. Администратор может удалить полученное показание, после чего БД сохранит эти изменения.

На рисунке 8 показана диаграмма последовательности работы с начислениями.

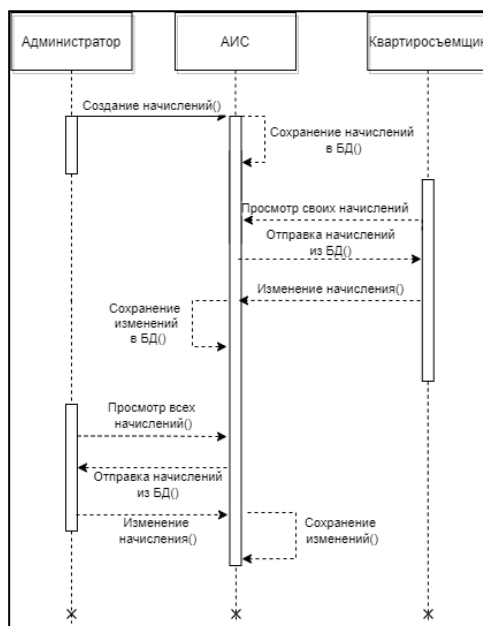


Рисунок 8 - Диаграмма последовательности работы с начислениями

На данном рисунке администратор может создать начисления, которые после будут сохранены АИС в базе данных. Также и администратор, и квартиросъемщик, могут просматривать начисления, которые АИС отправляет им из БД, а также они могут изменять их, после чего эти изменения будут сохранены.

3.2 Диаграммы SADT

Представим всю систему в виде простейших компонентов – одного блока и дуг, изображающих интерфейсы с функциями вне системы. Созданная функциональная диаграмма показана на рисунке 9.

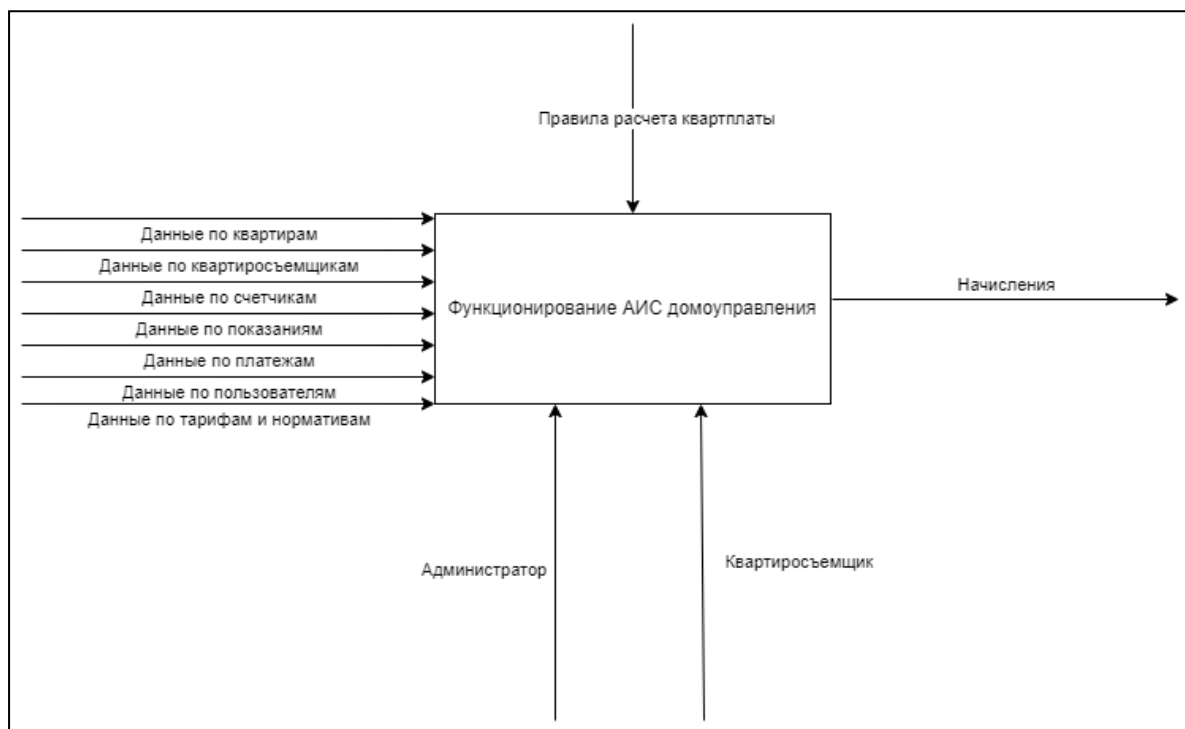


Рисунок 9 – Функциональная диаграмма АИС домоуправления

Управляющая информация (правила расчета квартплаты) входит в блок сверху, в то время как информация, которая подвергается обработке (данные по квартирам, квартиросъемщикам, счетчикам, показаниям, платежам, пользователям, тарифам и нормативам), показана с левой стороны блока, а результаты выхода (начисления) показаны с правой стороны. Механизмы (администратор и квартиросъемщик), которые осуществляют операцию, представляются дугой, входящей в блок снизу.

Для получения подробной информации о функционировании АИС необходимо создать подробную функциональную диаграмму. Она показана на рисунке 10.

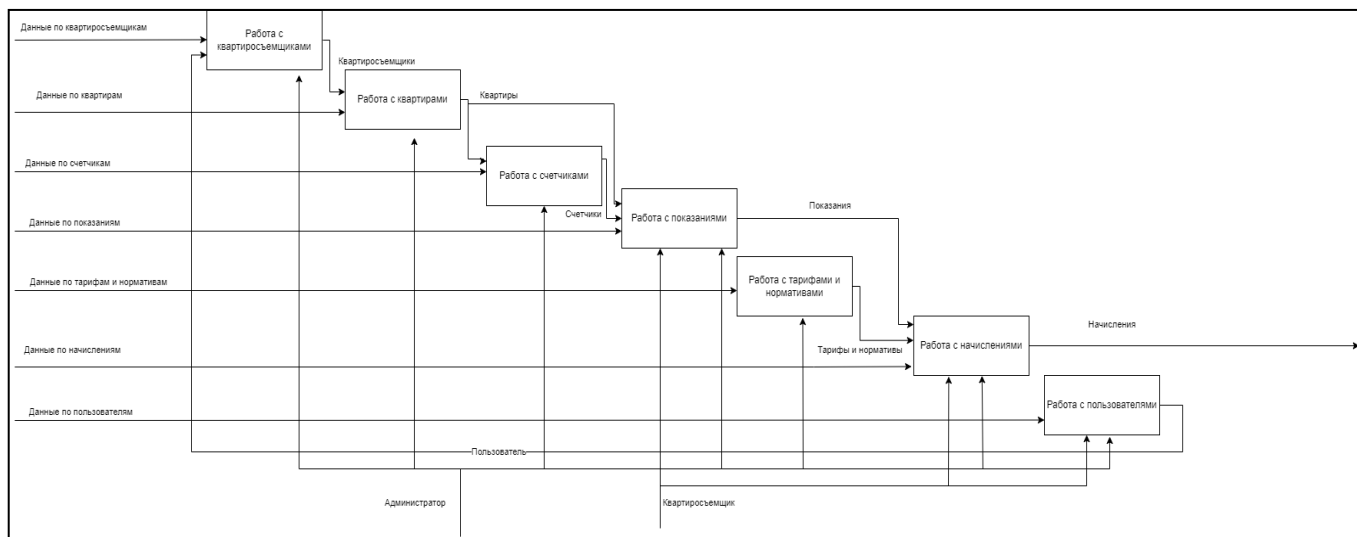


Рисунок 10 - Подробная функциональная диаграмма функционирования АИС домоуправления

Администратор может осуществлять работу со всеми блоками. Он передает следующее: информацию по квартиросъемщикам в блок «Работа с квартиросъемщиками» (этот блок подразумевает создание, изменение и удаление квартиросъемщика), информацию по квартирам в блок «Работа с квартирами» (этот блок подразумевает создание, изменение и удаление квартиры), информацию по счетчикам в блок «Работа со счетчиками» (этот блок подразумевает создание, изменение и удаление счетчика), информацию по показаниям в блок «Работа с показаниями» (этот блок подразумевает создание, изменение и удаление показания), информацию по тарифам и нормативам в блок «Работа с тарифами и нормативами» (этот блок подразумевает их изменение), информацию по пользователям в блок «Работа с пользователями» (этот блок подразумевает регистрацию и авторизацию пользователя). Также администратор может создавать или изменять начисления, которые находятся в блоке «Работа с начислениями». Квартиросъемщик на данной диаграмме может следующее: передавать показания в блок «Работа с показаниями», заносить информацию о начислениях в блок «Работа с начислениями», а также вносить изменения в блоке «Работа с пользователями».

Ниже, на рисунке 11 представлена более подробная реализация блока «Работа с показаниями».

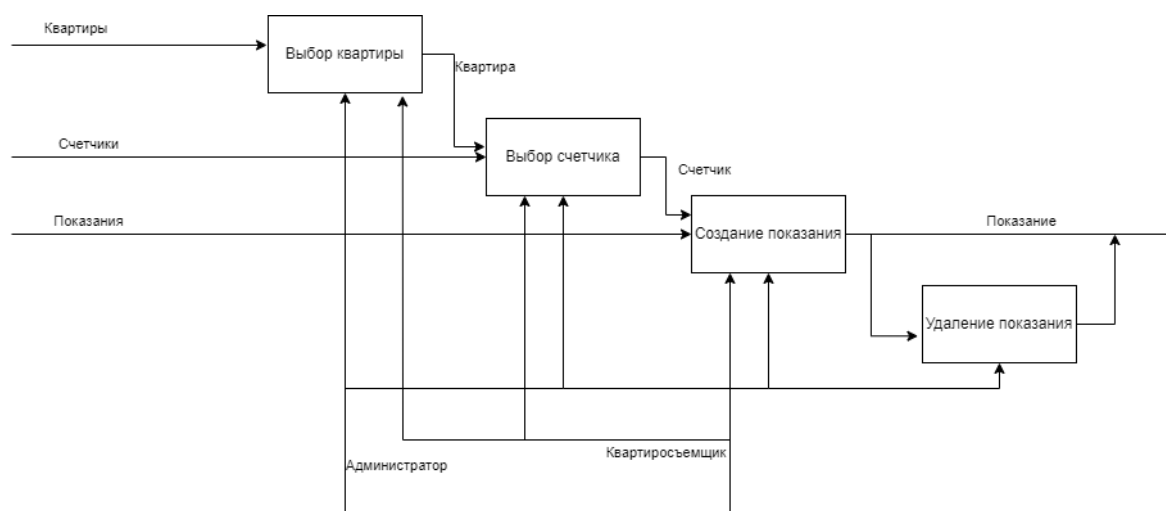


Рисунок 11 – Подробная функциональная диаграмма блока «Работа с показаниями»

На данной схеме администратор и квартиросъемщик вносят информацию в блоки «Выбор квартиры», «Выбор счетчика» и «Создание показания». Также администратор вносит изменения в блок «Удаление показания». Выходом данной диаграммы является показание.

4 Разработка и реализация АИС

После разработки моделей данных была создана сама база данных и из классов моделей.

Так как основу программы составляет взаимодействие с базой данных, то рассмотрим его на примере работы с квартирой.

Для каждой сущности был создан свой репозиторий, который позволяет оперировать с объектом в базе данных. Для работы с каждым из репозиторий были созданы классы Service – Java классы, которые представляют конечные данные для пользовательского интерфейса.

4.1 Классы модели данных

Листинг класса Flat.java:

```
@Entity
@Table(name = "Flat")
@NoArgsConstructor
@Getter
@Setter
@AllArgsConstructor
public class Flat {
    @Id
    @Column(name = "id")
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    @Size(min = 8, max = 20, message = "Длина данного поля должна быть от 8 до 20")
    @Column(name = "personal_account", length = 20, nullable = false, unique = true)
    private String personalAccount;
    @Pattern(regexp = "[0-9]{1,3}", message = "Значением данного поля должно быть число от 0 до 999")
    @Column(name = "flat_number", length = 3, nullable = false, unique = true)
    private String flatNumber;
    @NotNull(message = "Необходимо заполнить поле")
    @Column(name = "total_area", nullable = false)
    private Float total;
    @NotNull(message = "Необходимо заполнить поле")
    @Column(name = "usable_area", nullable = false)
    private Float usable;
    @Pattern(regexp = "[0-9]{1,3}", message = "Значением данного поля должно быть число от 0 до 999")
    @Column(name = "entrance_number", length = 3, nullable = false)
    private String entranceNumber;
    @Pattern(regexp = "[0-9]{1,3}", message = "Значением данного поля должно быть
```

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		32

```

число от 0 до 999")
@Column(name = "number_of_rooms", length = 3, nullable = false)
private String numberOfRooms;
@NotNull(message = "Необходимо заполнить поле")
@Range(min = 0, max = 999, message = "Значением данного поля должно быть число
от 0 до 999")
@Column(name = "number_of_registered_residents", nullable = false)
private int numberOfRegisteredResidents;
@NotNull(message = "Необходимо заполнить поле")
@Range(min = 1, max = 999, message = "Значением данного поля должно быть число
от 1 до 999")
@Column(name = "number_of_owners", nullable = false)
private int numberOfOwners;
@OneToOne
@JsonBackReference
private Tenant tenant;

```

Данный класс соответствует таблице «Flat» в базе данных. Здесь есть 9 полей: id – уникальный номер квартиры в базе данных, personal_account – лицевой счет квартиры, flat_number – номер квартиры, total_area – общая площадь квартиры, usable_area – полезная площадь квартиры, entrance_number – номер подъезда, number_of_rooms – количество комнат, number_of_registered_residents – количество зарегистрированных пользователей, number_of_owners – количество собственников жилья, tenant – квартиросъемщик, который проживает в данной квартире.

Перед классом и его полями, которые заносятся в БД, указываются аннотации данных – инструмент языка для настройки сущностей с помощью атрибутов.

@Entity - говорит о том, класс является сущностью и на основе этого класса будет создана таблица с именем из @Table

@NoArgsConstructor – генерирует конструктор без параметров

@AllArgsConstructor – генерирует конструктор с одним параметром для каждого поля класса

@Getter и @Setter – генерирует геттеры и сеттеры для каждого поля класса

@Column – указывает на имя колонки, которая отображается в свойство сущности

@Id – аннотация, которая указывается для установки поля в качестве первичного ключа

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33

@GeneratedValue - указывает, что данное свойство будет создаваться согласно указанной стратегии (в данном случае сгенерированное значение будет уникальным)

@Size – атрибут, который проверяет размер введенной строки.

@Pattern - атрибут, который используется для проверки соответствия шаблону.

@NotNull – проверяет значение поля на значение null.

@Range – аннотация, которая позволяет проверять введенное в поле целочисленное значение.

@OneToOne – атрибут, который показывает связь данной сущности с другой (в данном случае один к одному).

У некоторых из этих атрибутов есть свойство “ErrorMessage”, которое позволяет указывать текст сообщения об ошибке.

4.1 Вывод всех квартир

Для вывода всех запросов используется GET запрос, который возвращает нужный html файл.

Листинг метода mainPage():

```
@GetMapping("/list")
public String mainPage(Model model) {
    Authentication authentication =
SecurityContextHolder.getContext().getAuthentication();
    User user = userService.getUserByName(authentication.getName()).get();
    List<Flat> flats = new ArrayList<>();
    if(user.getAuthority().equals("ADMIN")) flats = flatService.list();
    else flats =
flatService.listByTenant(tenantService.getTenantsByEmail(authentication.getName())
.get(0));
    String role = null;
    if(user.getAuthority().equals("ADMIN")) role = "ADMIN";
    model.addAttribute("role", role);
    model.addAttribute("flats", flats);
    model.addAttribute("filterSearchData", new FilterSearchData());
    return "flat/flats";
}
```

В данном методе на html файл при помощи объекта класса Model передается список квартир. При помощи класса Authentication берется авторизованный пользователь: если это администратор – выводится весь список квартир, если квартиросъемщик – выводится список квартир, которые относятся только к данному пользователю. Также на модель помещается объект класса FilterSearchData, при помощи которого производится поиск по списку квартир.

4.2 Поиск квартиры

Листинг метода SearchFlat() POST запроса:

```
@PostMapping("/search")
public String searchFlat(Model model, @ModelAttribute("filterSearchData")
FilterSearchData filterSearchData){
    Authentication authentication =
SecurityContextHolder.getContext().getAuthentication();
    User user = userService.getUserByName(authentication.getName()).get();
    List<Flat> flats = new ArrayList<>();
    if(user.getAuthority().equals("ADMIN")) flats = flatService.list();
    else flats =
flatService.listByTenant(tenantService.getTenantsByEmail(authentication.getName())
.get(0));
    if(filterSearchData.getSearch() != null) flats = flats.stream().filter(x ->
x.getFlatNumber().contains(filterSearchData.getSearch())).toList();
    String role = null;
    if(user.getAuthority().equals("ADMIN")) role = "ADMIN";
    model.addAttribute("role", role);
    model.addAttribute("flats",flats);
    return "flat/flats";
}
```

Данный метод отвечает на POST запрос, в теле которого передается объект модели FilterSearchData с введенным номером квартиры. После, аналогично предыдущему методу берется список квартир согласно пользователю и производится фильтрация по нему. В результате возвращается html страница со списком всех страниц.

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		35

4.3 Добавление квартиры

При добавлении квартиры есть 2 метода: с GET и POST запросами. GET запрос отображает нужную модель для ввода данных, а POST запрос обрабатывает данные, которые пользователь ввел.

Листинг метода CreatePage() GET запроса:

```
@GetMapping("/create")
public String createPage(Model model, @ModelAttribute("flat") Flat flat) {
    model.addAttribute("tenantsList", tenantService.list());
    Authentication authentication =
SecurityContextHolder.getContext().getAuthentication();
    User user = userService.getUserByName(authentication.getName()).get();
    if(user.getAuthority().equals("ADMIN"))
        return "flat/create";
    else return "redirect:/Flat/list";
}
```

В данном методе проверяется пользователь, если это квартиросъемщик – его перенаправляет на страницу со списком квартир, если администратор – то для него открывается страница с добавлением квартиры. Также в этом методе на форму с добавлением помещается список с квартиросъемщиками, так как при добавлении квартиры необходимо выбрать её жителя.

Листинг метода CreateFlat() POST запроса:

```
@PostMapping("/create")
public String createFlat(Model model, @ModelAttribute("flat") @Valid Flat flat,
BindingResult bindingResult){
    Authentication authentication =
SecurityContextHolder.getContext().getAuthentication();
    User user = userService.getUserByName(authentication.getName()).get();
    if(user.getAuthority().equals("ADMIN")){
        model.addAttribute("tenantsList", tenantService.list());
        if(flatService.getFlatsByPhoneNumber(flat.getFlatNumber()).size() != 0)
            bindingResult.addError(new FieldError("flat", "flatNumber",
flat.getFlatNumber(),
false, null, null, "Квартира с таким номером уже
существует"));
        if(flatService.getFlatsByPersonalAccount(flat.getPersonalAccount()).size()
!= 0)
            bindingResult.addError(new FieldError("flat", "personalAccount",
flat.getPersonalAccount(),
false, null, null, "Квартира с таким личным счетом уже
существует"));
        if((flat.getTotal() != null && flat.getUsable() != null) &&
flat.getTotal() < flat.getUsable())
```

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

```

        bindingResult.addError(new FieldError("flat", "total",
flat.getTotal(),
        false, null, null, "Общая площадь должна быть больше
полезной"));
        if(bindingResult.hasErrors()) return "flat/create";
        flatService.postFlat(flat);
    }
    return "redirect:/Flat/list";
}

```

В данном методе также проверяется роль пользователя, и, как можно увидеть, возможность работы с квартирой предоставляется только администратору. После проверки роли в программе проверяются введенные пользователем данные. Если присутствуют ошибки, то пользователю снова показывают страницу с добавлением квартиры, но уже с выведенными ошибками. Ниже представлен листинг поля и соответствующей ему ошибки:

```

<div class="input-group mb-3">
    <span class="input-group-text">Номер квартиры: </span>
    <input type="number" class="form-control" th:field="*{flatNumber}"/>
</div>

```

4.4 Изменение квартиры

В изменении квартиры есть также GET и POST запросы. Только в данном случае GET запрос принимает на вход id квартиры, которую необходимо изменить.

Листинг метода UpdatePage()GET запроса:

```

@GetMapping("/update/{id}")
public String updatePage(Model model, @ModelAttribute("flat") Flat flat,
@PathVariable("id") Integer id) {
    model.addAttribute("tenantsList", tenantService.list());
    model.addAttribute("flat", flatService.getFlatById(id).get());
    Authentication authentication =
SecurityContextHolder.getContext().getAuthentication();
    User user = userService.getUserByName(authentication.getName()).get();
    if(user.getAuthority().equals("ADMIN"))
        return "flat/update";
    else return "redirect:/Flat/list";
}

```

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		37

Можно увидеть, что изменение квартиры доступно также только администратору. На модель помещается выбранная квартира и список квартиросъемщиков. Все данные заполняются в соответствии с выбранной квартирой.

Листинг метода updateFlat() POST запроса:

```
@PostMapping("/update/{id}")
public String updateFlat(@ModelAttribute("flat") @Valid Flat flat, BindingResult
bindingResult, Model model, @PathVariable("id") Integer id){
    Authentication authentication =
SecurityContextHolder.getContext().getAuthentication();
    User user = userService.getUserByName(authentication.getName()).get();
    if (user.getAuthority().equals("ADMIN")) {
        model.addAttribute("tenantsList", tenantService.list());
        if (flatService.getFlatsByNumberNotId(id, flat.getFlatNumber()).size() !=
0)
            bindingResult.addError(new FieldError("flat", "flatNumber",
flat.getFlatNumber(),
false, null, null, "Квартира с таким номером уже
существует"));
        if (flatService.getFlatsByPersonalAccountNotId(id,
flat.getPersonalAccount()).size() != 0)
            bindingResult.addError(new FieldError("flat", "personalAccount",
flat.getPersonalAccount(),
false, null, null, "Квартира с таким лицевым счетом уже
существует"));
        if ((flat.getTotal() != null && flat.getUsablea() != null) &&
flat.getTotal() < flat.getUsablea())
            bindingResult.addError(new FieldError("flat", "total",
flat.getTotal(),
false, null, null, "Общая площадь должна быть больше
полезной"));
        if (bindingResult.hasErrors()) return "flat/update";
        flatService.postFlat(flat);
    }
    return "redirect:/Flat/details/" + id;
}
```

В данном методе также происходят проверки на уникальность номера и лицевого счета квартиры, а также на то, что полезная площадь не должна превышать общей площади квартиры. При возникновении ошибки пользователю под соответствующим полем она выводится, но если все верно – данные квартиры изменяются и пользователя перенаправляет на страницу с подробной информацией о квартире.

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		38

4.5 Удаление квартиры

В удалении квартиры, как и всегда участвуют два метода с разными запросами. Один из них GET, который возвращает нужную View модель с подтверждением удаления квартиры. Этот метод принимает на вход также, как и при изменении – id квартиры.

В POST запросе обрабатывается подтверждение пользователем удаления квартиры. Он также получает на вход id соответствующей квартиры. В этом методе происходит проверка на отсутствие связей данной квартиры с другими таблицами, если они есть – удаление не возможно и пользователю выводится сообщение об этом, если нет – происходит непосредственно удаление и пользователя перенаправляют на страницу со списком всех квартир.

Листинг deleteFlat() POST запроса:

```
@PostMapping("/delete/{id}")
public String deleteFlat(Model model, @ModelAttribute("flat") Flat flat,
BindingResult bindingResult, @PathVariable("id") Integer id){
    Authentication authentication =
SecurityContextHolder.getContext().getAuthentication();
    User user = userService.getUserByName(authentication.getName()).get();
    if (user.getAuthority().equals("ADMIN")) {
        Flat flatItem = flatService.getFlatById(flat.getId()).get();
        flat.setFlatNumber(flatItem.getFlatNumber());
        flat.setTenant(flatItem.getTenant());
        flat.setEntranceNumber(flatItem.getEntranceNumber());
        flat.setNumberOfOwners(flatItem.getNumberOfOwners());
        flat.setNumberOfRooms(flatItem.getNumberOfRooms());

        flat.setNumberOfRegisteredResidents(flatItem.getNumberOfRegisteredResidents());
        flat.setUsable(flatItem.getUsable());
        flat.setTotal(flatItem.getTotal());
        flat.setPersonalAccount(flatItem.getPersonalAccount());
        model.addAttribute("tenantsList", tenantService.list());
        if (counterService.findAllByFlat(id).size() != 0 ||
paymentService.findAllByFlat(flat.getId()).size() != 0)
            bindingResult.addError(new FieldError("flat", "tenant",
flat.getTenant(),
false, null, null, "Удаление невозможно, так как квартира
задействована в счетчиках или начислениях"));
        if (bindingResult.hasErrors()) return "flat/delete";
        flatService.deleteFlat(flat);
    }
    return "redirect:/Flat/list";
}
```

В данном методе также можно увидеть заполнение полей квартиры, которую необходимо вернуть в случае ошибки.

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		39

5. Тестирование

Одним из важнейших этапов создания АИС является ее тестирование. Тестирование позволяет выявить скрытые и явные недостатки программы, либо убедиться в ее пригодности для применения. Обнаруженные недостатки устраняются в ходе отладки.

Целью тестирования является проверка работоспособности АИС, правильности выполнения всех функций, а также правильности обработки всех исключений, возникающих в ходе работы программы.

Ниже, в таблице 16, представлена информация о тестировании приложения.

Таблица 16

Тестирование приложения

№	Описание теста	Ожидаемый результат	Полученный результат
1	Попытка перехода по контроллерам без авторизации	Отображение станицы авторизация	Результат соответствует ожиданию
2	Авторизация с введением неверного логина и пароля	Вывод ошибки о неверных данных	Результат соответствует ожиданию
3	Авторизация с данными администратора	Вход в приложение за администратора	Результат соответствует ожиданию
4	Авторизация за квартиросъемщика	Вход в приложение за квартиросъемщика	Результат соответствует ожиданию
	Попытка перехода на страницы за квартиросъемщика, к которым у него нет доступа	Перенаправление пользователя на страницы со списками	Результат соответствует ожиданию
Тестирование работы с квартиросъемщиками			
5	Переход на вкладку квартиросъемщиков в меню за администратора	Вывод списка всех квартиросъемщиков, присутствие на странице, присутствие на странице кнопок работы с квартиросъемщиками	Результат соответствует ожиданию

Продолжение таблицы 16

6	Переход на вкладку квартиросъемщиков в меню за квартиросъемщика	Вывод информации только об авторизованном квартиросъемщике	Результат соответствует ожиданию
7	Ввод в поле поиска строки, которая включает часть ФИО из присутствующих квартиросъемщиков в БД	Отображение списка квартиросъемщиков, ФИО которых включает введенную строку	Результат соответствует ожиданию
8	Ввод в поле поиска строки, которая не включает часть ФИО из присутствующих квартиросъемщиков в БД	Отображение пустого списка квартиросъемщиков	Результат соответствует ожиданию
9	Нажатие на кнопку «Подробнее»	Переход на страницу с подробной информации о квартиросъемщике	Результат соответствует ожиданию
10	Нажатие на кнопку «Добавить»	Открытие страницы добавления квартиросъемщика	Результат соответствует ожиданию
11	Ввод в какое – либо из полей пустого значения (кроме email) при добавлении квартиросъемщика	Вывод ошибки ввода информации в поле	Результат соответствует ожиданию
12	Ввод некорректного значения в поле номера телефона при добавлении квартиросъемщика	Вывод ошибки ввода информации в поле номера телефона	Результат соответствует ожиданию
13	Ввод некорректного значения в поле email при добавлении квартиросъемщика	Вывод ошибки ввода информации в поле email	Результат соответствует ожиданию
14	Ввод в поле email значения, которое уже есть в БД	Вывод ошибки о том, что такой email уже занят	Результат соответствует ожиданию
15	Ввод корректных значение во все поля и нажатие на кнопку «Добавить» при добавлении квартиросъемщика	Добавление в БД записи и возвращение администратора на страницу всех квартиросъемщиков	Результат соответствует ожиданию
16	Нажатие на кнопку «Изменить» в списке квартиросъемщиков	Переход на страницу изменения квартиросъемщика с уже заполненными полями	Результат соответствует ожиданию

Продолжение таблицы 16

17	Ввод некорректных данных в поля изменения квартиросъемщика	Вывод ошибки ввода информации в поле	Результат соответствует ожиданию
18	Ввод корректных данных в поля изменения квартиросъемщика и нажатие на кнопку «Изменить»	Успешное изменение записи в БД и переход на страницу подробной информации о квартиросъемщике	Результат соответствует ожиданию
19	Нажатие на кнопку «Удалить» в подробной информации о квартиросъемщике при отсутствии у него квартиры	Успешное удаление квартиросъемщика и переход на страницу всех квартиросъемщиков	Результат соответствует ожиданию
20	Нажатие на кнопку «Удалить» в подробной информации о квартиросъемщике при присутствии у него квартиры	Вывод ошибки о невозможности удаления	Результат соответствует ожиданию
21	Нажатие на кнопку «Аккаунт» в списке квартиросъемщиков	Переход на страницу редактирования аккаунта квартиросъемщика	Результат соответствует ожиданию
22	Ввод в поле email некорректного значения при изменении аккаунта	Вывод ошибки о некорректном вводе email	Результат соответствует ожиданию
23	Ввод в поле пароля значения, которое в длине меньше 10	Вывод ошибки о некорректном вводе пароля	Результат соответствует ожиданию
24	Ввод в поля паролей разных значений при изменении аккаунта	Вывод ошибки о несовпадении паролей	Результат соответствует ожиданию
25	Ввод корректных значений в поля при изменении аккаунта	Успешное изменение квартиросъемщика и возврат на страницу списка	Результат соответствует ожиданию

Тестирование работы с квартирами

26	Переход на вкладку квартир в меню за администратора	Вывод списка всех квартир, присутствие на странице присутствие на странице кнопок работы со квартирами	Результат соответствует ожиданию
----	---	--	----------------------------------

Продолжение таблицы 16

27	Переход на вкладку квартир в меню за квартиросъемщика	Вывод информации только об авторизованном квартиросъемщике	Результат соответствует ожиданию
28	Ввод в поле поиска строки, которая включает часть номера квартиры из присутствующих квартир в БД	Отображение списка квартир, номер которых включает введенную строку	Результат соответствует ожиданию
29	Ввод в поле поиска строки, которая не включает часть номера квартиры из присутствующих квартир в БД	Отображение пустого списка квартир	Результат соответствует ожиданию
30	Нажатие на кнопку «Добавить»	Открытие страницы добавления квартиры	Результат соответствует ожиданию
31	Ввод в какое – либо из полей пустого значения при добавлении квартиры	Вывод ошибки ввода информации в поле	Результат соответствует ожиданию
32	Ввод значения в поле общей площади, которое меньше, чем полезная площадь	Вывод ошибки ввода информации в поля общей и полезной площадей	Результат соответствует ожиданию
33	Ввод в поля номера квартиры и лицевого счета значений, которые уже есть в БД	Вывод ошибки о том, что квартира с такими данными уже существует	Результат соответствует ожиданию
34	Ввод корректных значение во все поля и нажатие на кнопку «Добавить» при добавлении квартиры	Добавление записи в БД и переход на страницу всех квартир	Результат соответствует ожиданию
35	Нажатие на «Подробнее» в списке квартир	Переход на страницу подробной информации о квартире	Результат соответствует ожиданию
36	Нажатие на кнопку «Изменить» в списке квартир	Переход на страницу изменения квартиры с уже заполненными полями	Результат соответствует ожиданию
37	Ввод некорректных данных в поля изменения квартиры	Вывод ошибки ввода информации в поле	Результат соответствует ожиданию
38	Ввод корректных данных в поля изменения квартиры и нажатие на кнопку «Изменить»	Успешное изменение записи в БД и переход на страницу подробной информации о квартире	Результат соответствует ожиданию

Продолжение таблицы 16

39	Нажатие на кнопку «Удалить» в подробной информации о квартире при отсутствии у нее счетчиков и начислений	Успешное удаление квартиры и переход на страницу всех квартир	Результат соответствует ожиданию
40	Нажатие на кнопку «Удалить» в подробной информации о квартире при присутствии у нее счетчиков и начислений	Вывод ошибки о невозможности удалении	Результат соответствует ожиданию
Тестирование работы со счетчиками			
41	Переход на вкладку счетчиков в меню за администратора	Вывод списка всех счетчиков, присутствие на странице кнопок работы со счетчиками	Результат соответствует ожиданию
42	Переход на вкладку счетчиков в меню за квартиросъемщика	Вывод информации только об авторизованном квартиросъемщике	Результат соответствует ожиданию
43	Применение фильтров поиска	Отображение отфильтрованного списка счетчиков	Результат соответствует ожиданию
44	Нажатие на кнопку «Добавить»	Открытие страницы выбора квартиры для добавления счетчика	Результат соответствует ожиданию
45	Ввод в поле поиска квартиры значение и выполнить поиск	Вывод отфильтрованного списка квартир	Результат соответствует ожиданию
46	Выбор какой-либо квартиры и нажатие «Выбрать»	Вывод страницы с добавлением информации о счетчике	Результат соответствует ожиданию
47	Ввод некорректного значения в поле номер счетчика при добавлении счетчика	Вывод ошибки ввода информации в поле номер счетчика	Результат соответствует ожиданию
48	Ввод в поле номера и типа счетчика значений, которые уже есть в БД	Вывод ошибки о том, что счетчик с такими данными уже существует	Результат соответствует ожиданию
49	Ввод корректных значение во все поля и нажатие на кнопку «Добавить» при добавлении счетчика	Добавление в БД записи и возвращение администратора на страницу всех счетчиков	Результат соответствует ожиданию
50	Нажатие на «Подробнее» в списке счетчиков	Переход на страницу подробной информации о счетчике	Результат соответствует ожиданию

Продолжение таблицы 16

51	Нажатие на кнопку «Изменить» в списке счетчиков	Переход на страницу изменения счетчика с уже заполненными полями	Результат соответствует ожиданию
52	Ввод некорректных данных в поля изменения счетчика	Вывод ошибки ввода информации в поле	Результат соответствует ожиданию
53	Ввод корректных данных в поля изменения счетчика и нажатие на кнопку «Изменить»	Успешное изменение записи в БД и переход на страницу подробной информации о счетчике	Результат соответствует ожиданию
54	Нажатие на кнопку «Удалить» в подробной информации о счетчике при отсутствии у него показаний	Успешное удаление счетчика и переход на страницу всех счетчиков	Результат соответствует ожиданию
55	Нажатие на кнопку «Удалить» в подробной информации о счетчике при присутствии у него показаний	Вывод ошибки о невозможности удаления	Результат соответствует ожиданию

Тестирование работы с показаниями

56	Переход на вкладку показаний в меню за администратора	Вывод списка всех показаний, присутствие на странице кнопок работы с показаниями	Результат соответствует ожиданию
57	Переход на вкладку показаний в меню за квартиросъемщика	Вывод информации только об авторизованном квартиросъемщике, присутствие на странице кнопки добавления показания	Результат соответствует ожиданию
58	Применение фильтров поиска	Отображение отфильтрованного списка показаний	Результат соответствует ожиданию
59	Нажатие на кнопку «Передать показание»	Открытие страницы выбора квартиры для передачи показания	Результат соответствует ожиданию
60	Выполнение поиска на страницу выбора квартир	Вывод отфильтрованного списка квартир	Результат соответствует ожиданию
61	Нажатие на кнопку «Выбрать» с списке квартир	Открытие страницы выбора счетчика по квартире при передаче показания	Результат соответствует ожиданию

Продолжение таблицы 16

62	Ввод некорректного значения в поле значения показания	Вывод ошибки ввода информации в поле значение показания	Результат соответствует ожиданию
63	Ввод корректных значение во все поля и нажатие на кнопку «Передать» при уже переданном показании по этому счетчику и отсутствии начислений за передаваемый период	Вывод ошибки о невозможности передачи показания	Результат соответствует ожиданию
64	Ввод корректных значение во все поля и нажатие на кнопку «Передать» при не переданном показании по этому счетчику и присутствии начислений за передаваемый период	Вывод ошибки о невозможности передачи показания	Результат соответствует ожиданию
65	Ввод корректных значение во все поля и нажатие на кнопку «Передать» при не переданном показании по этому счетчику и отсутствии начислений за передаваемый период	Успешное добавление записи в БД и переход на страницу всех показаний	Результат соответствует ожиданию
66	Нажатие на кнопку «Изменить» в списке показаний	Переход на страницу изменения показания с уже заполненными полями	Результат соответствует ожиданию
67	Ввод некорректных данных в поля изменения показания	Вывод ошибки ввода информации в поле	Результат соответствует ожиданию
68	Ввод корректных данных в поля изменения показания и нажатие на кнопку «Изменить» при уже переданном показании, либо уже выполненных начислениях за период	Вывод ошибки о невозможности передачи показания	
69	Ввод корректных данных в поля изменения показания и нажатие на кнопку «Изменить»	Успешное изменение записи в БД и переход на страницу всех показаний	Результат соответствует ожиданию
70	Нажатие на кнопку «Удалить» в списке при отсутствии начислений за период показания	Успешное удаление показания и переход на страницу всех показаний	Результат соответствует ожиданию

Продолжение таблицы 16			
71	Нажатие на кнопку «Удалить» в списке информации о счетчике при присутствии начислений за период показания	Вывод ошибки о невозможности удаления	Результат соответствует ожиданию
Тестирование работы с тарифами и нормативами			
72	Переход на вкладку тарифов и нормативов в меню за администратора	Вывод списка всех тарифов и нормативов и присутствие кнопок их изменения	Результат соответствует ожиданию
73	Нажатие на кнопку «Изменить» в списках тарифов или нормативов	Переход на страницу изменения тарифа или норматива с уже заполненными полями	Результат соответствует ожиданию
74	Ввод некорректных данных в поля изменения тарифа или норматива	Вывод ошибки ввода информации в поле	Результат соответствует ожиданию
75	Ввод корректных данных в поля изменения тарифа или норматива и нажатие на кнопку «Изменить»	Успешное изменение записи в БД и переход на страницы со списками тарифов или нормативов	Результат соответствует ожиданию
Тестирование работы с начислениями			
76	Переход на вкладку начислений в меню за администратора	Вывод списка всех начислений, присутствие на странице кнопки добавления и изменения начислений	Результат соответствует ожиданию
77	Переход на вкладку начислений в меню за квартиросъемщика	Вывод информации только об авторизованном квартиросъемщике, присутствие кнопки изменения начисления	Результат соответствует ожиданию
78	Применение фильтров поиска	Отображение отфильтрованного списка начислений	Результат соответствует ожиданию
79	Нажатие на кнопку «Добавить начисления»	Открытие страницы добавления начислений	Результат соответствует ожиданию
80	Нажатие на кнопку «Начислить» при отсутствии начислений за период	Успешное добавление записей в БД и переход на страницу всех начислений	Результат соответствует ожиданию

Продолжение таблицы 16

81	Нажатие на кнопку «Начислить» при присутствии начислений за период	Вывод ошибки о невозможности начислений	Результат соответствует ожиданию
82	Нажатие на «Подробнее» в списке начислений	Переход на страницу подробной информации о начислении	Результат соответствует ожиданию
83	Нажатие на кнопку «Изменить» в списке начислений за администратора	Переход на страницу изменения начисления с возможностью редактирования суммы, чека и статуса оплаты	Результат соответствует ожиданию
84	Нажатие на кнопку «Изменить» в списке за квартиросъемщика	Переход на страницу изменения начисления с возможностью редактирования только чека	Результат соответствует ожиданию
85	Ввод некорректных данных в поля изменения начисления	Вывод ошибки ввода информации в поле	Результат соответствует ожиданию
86	Ввод корректных данных в поля изменения начисления и нажатие на кнопку «Изменить»	Успешное изменение записи в БД и переход на страницу подробной информации о начислении	Результат соответствует ожиданию
Тестирование работы с аккаунтом			
87	Переход на вкладку аккаунта в меню за администратора	Вывод информации об аккаунте администратора	Результат соответствует ожиданию
88	Переход на вкладку аккаунта в меню за квартиросъемщика	Вывод информации об аккаунте квартиросъемщика	Результат соответствует ожиданию
89	Нажатие на кнопку «Выйти»	Выход из аккаунта и переход на страницу выхода из аккаунта	Результат соответствует ожиданию
90	Нажатие на кнопку «Редактировать email»	Переход на страницу редактирования email аккаунта	Результат соответствует ожиданию

Продолжение таблицы 16			
91	Ввод в поле email некорректного значения при изменении аккаунта	Вывод ошибки о некорректном вводе email	Результат соответствует ожиданию
92	Ввод корректных значений в поле email	Успешное изменение аккаунта и возврат на страницу аккаунта пользователя	Результат соответствует ожиданию
93	Нажатие на кнопку «Редактировать пароль»	Переход на страницу редактирования пароля аккаунта	Результат соответствует ожиданию
94	Ввод в поле пароля значения, которое в длине меньше 10	Вывод ошибки о некорректном вводе пароля	Результат соответствует ожиданию
95	Ввод корректных значений в поле пароля	Успешное изменение аккаунта и возврат на страницу аккаунта пользователя	Результат соответствует ожиданию

Заключение

В данной курсовой работе в соответствии с заданием была разработана распределенная ИС для автоматизации предметной области с использованием фреймворка Spring.

В ходе выполнения курсовой работы были выполнены следующие задачи:

- выявлены требования к программе;
- разработаны модели данных;
- разработана программа;
- осуществлено ее тестирование.

Разработанная программа обеспечивает осуществление следующих функций:

1. добавление данных о квартирах, их счетчиках и квартиросъемщиков;
2. передача показаний;
3. возможность изменения информации;
4. возможность удаления информации;
6. предоставление информации на форме в табличном виде;
7. возможность начислять платежи по каждой квартире за предыдущий месяц согласно тарифам и нормативам;
8. авторизация и регистрация пользователей.

Список литературы

1. Домовладелец: [сайт]. URL: <https://dmv1.ru/> (дата обращения: 06.10.2022).
2. Инфокрафт ЖКХ 365: [сайт]. URL: <https://www.gkh365.ru/> (дата обращения: 06.10.2022).
3. Руководство по аннотациям Spring Framework [сайт]. URL: <https://coderlessons.com/articles/java/rukovodstvo-po-annotatsiiam-spring-framework> (дата обращения: 17.10.2022).
4. Мухамедзянов, Р. Р. JAVA. Серверные приложения / Р. Р. Мухамедзянов. — Москва: СОЛОН-Р, 2016. — 336 с. — ISBN 5-93455-134-5. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/90352.html> (дата обращения: 06.11.2022).
5. Руководство по MySQL Workbench - [сайт]. URL: <https://dev.mysql.com/doc/workbench/en/> (дата обращения: 17.11.2022).

Приложение 1. Модели данных

1. Концептуальная модель данных

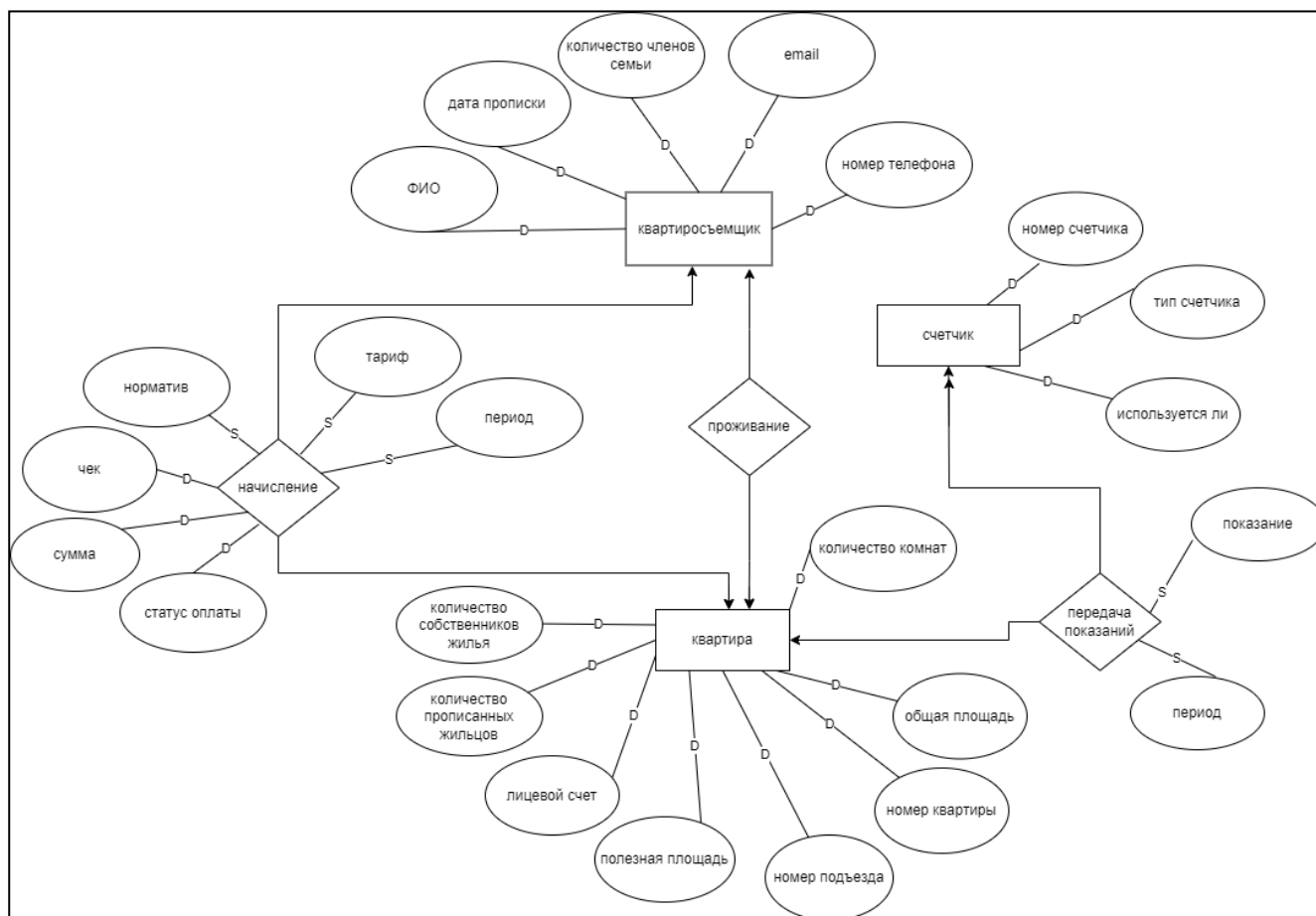


Рисунок 1 - Концептуальная модель данных

2. Логическа модель данных

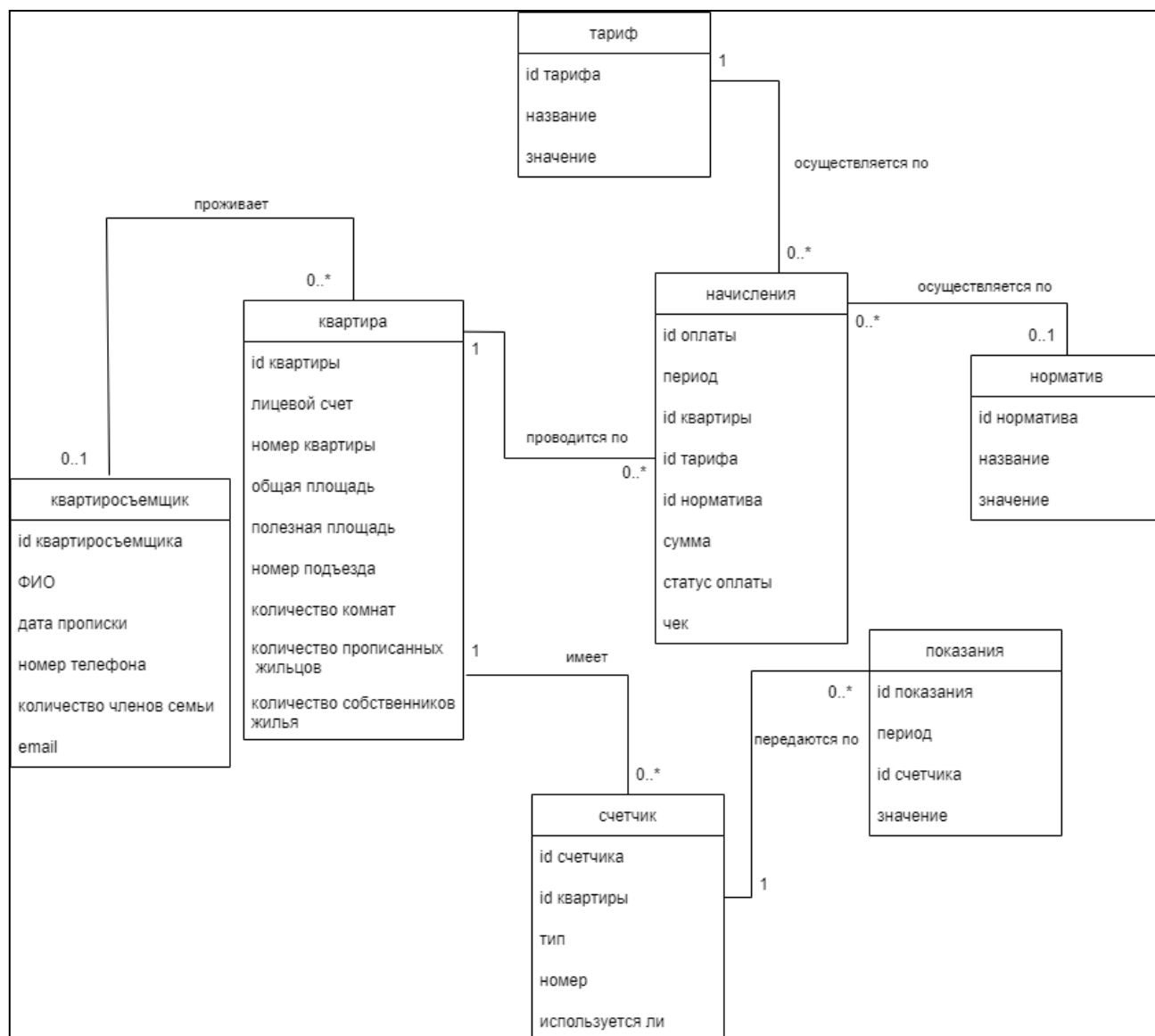


Рисунок 2 - Логическая модель данных

3. Физическая модель данных

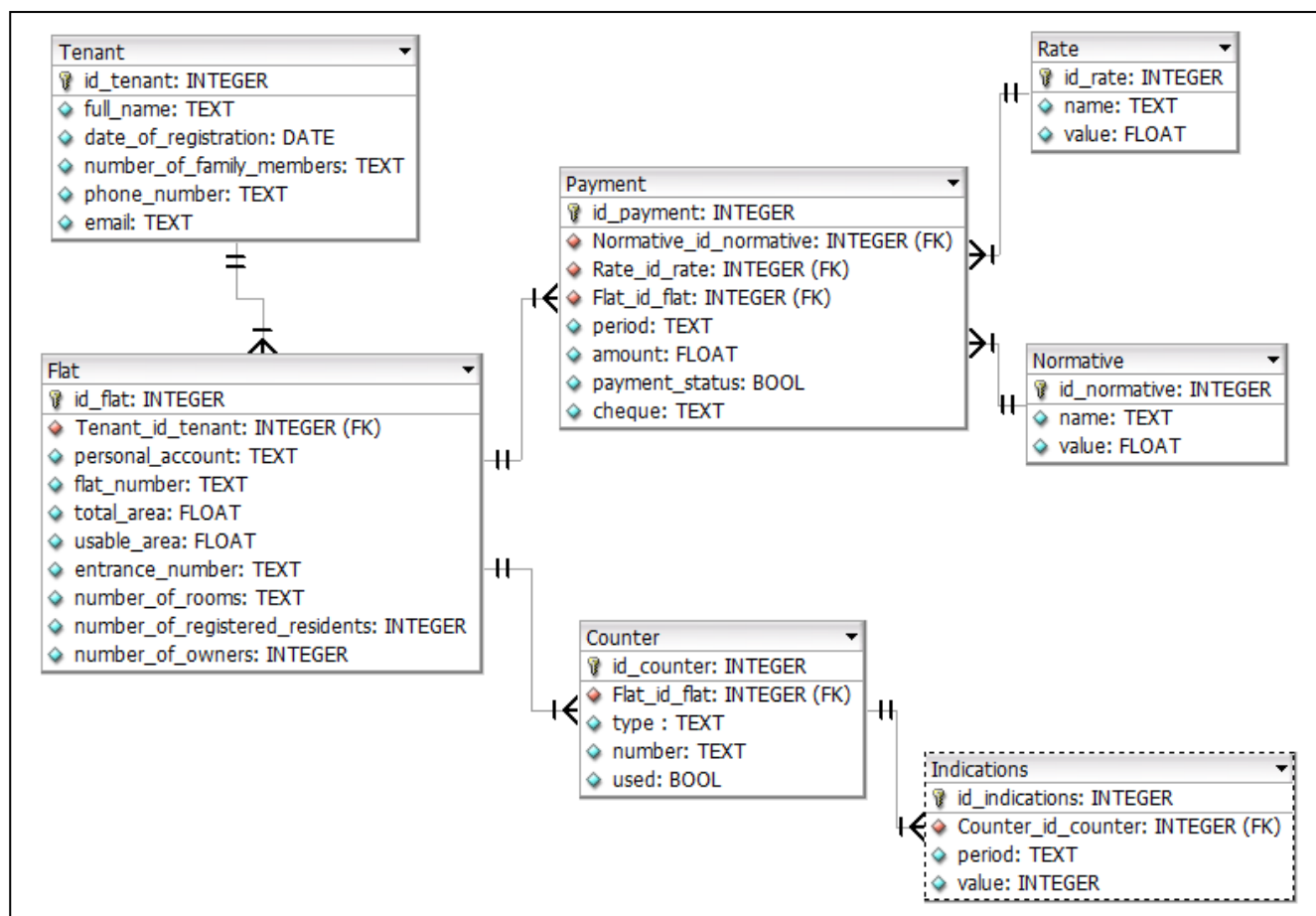


Рисунок 3 - Физическая модель данных

Приложение 2. Текст программы

Полный текст программы можно получить по ссылке на сайте GitHub:
[<https://github.com/Menma162/HouseManagementSpring>].

					МИВУ 09.03.04-13.000 ПЗ	Лист
						55
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение 3. Снимки окон программы (скриншоты программы)

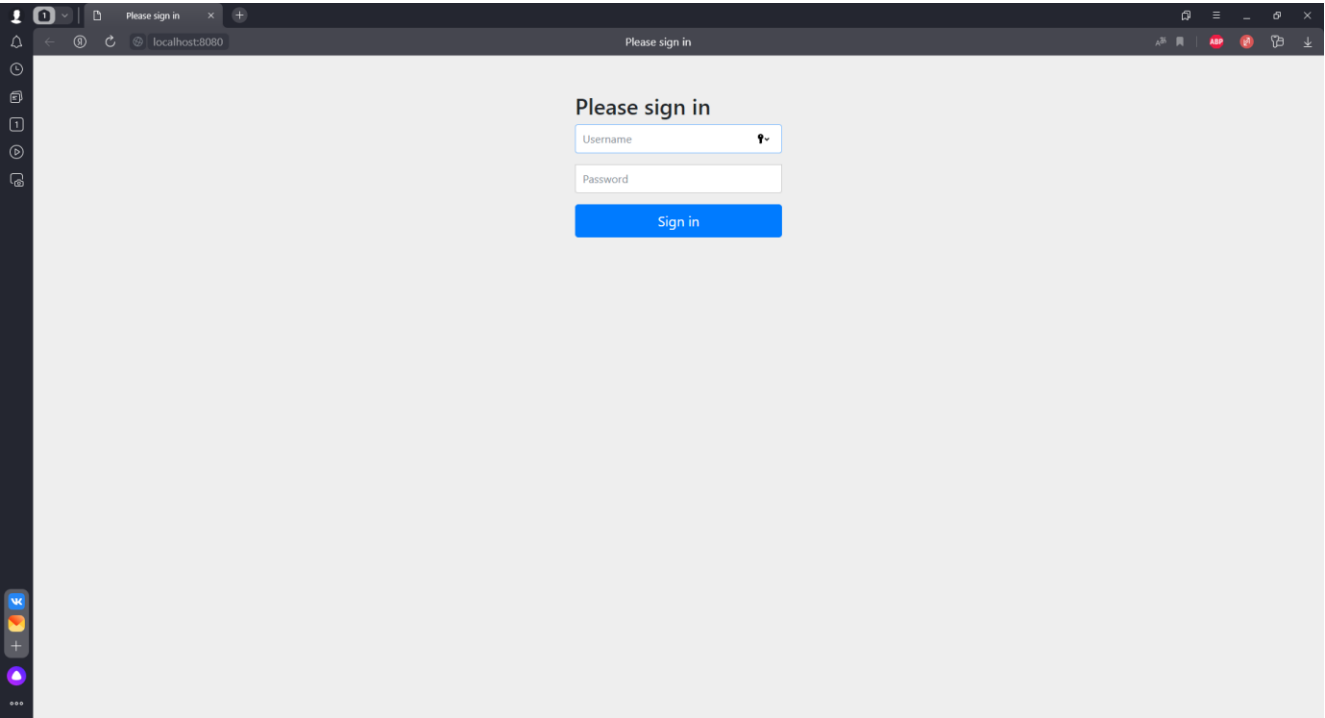


Рисунок 4 – Скриншот страницы с авторизацией

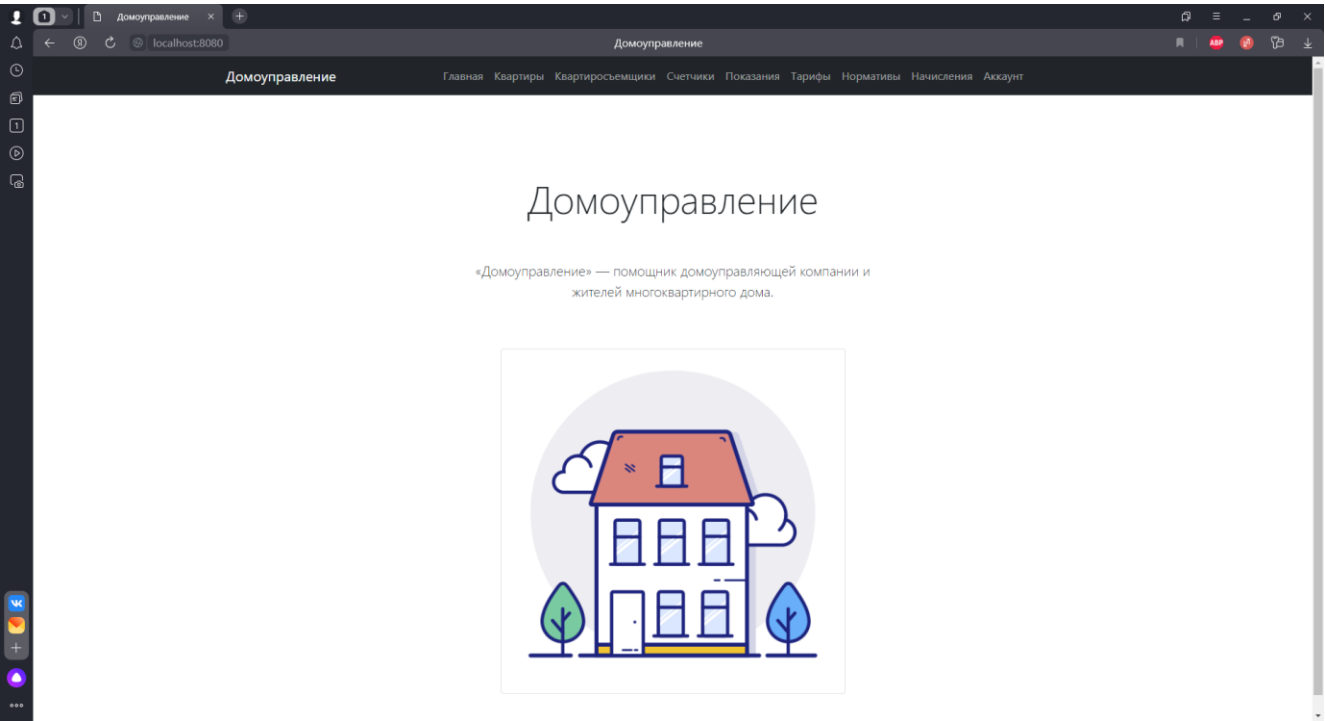


Рисунок 5 - Скриншот главной страницы

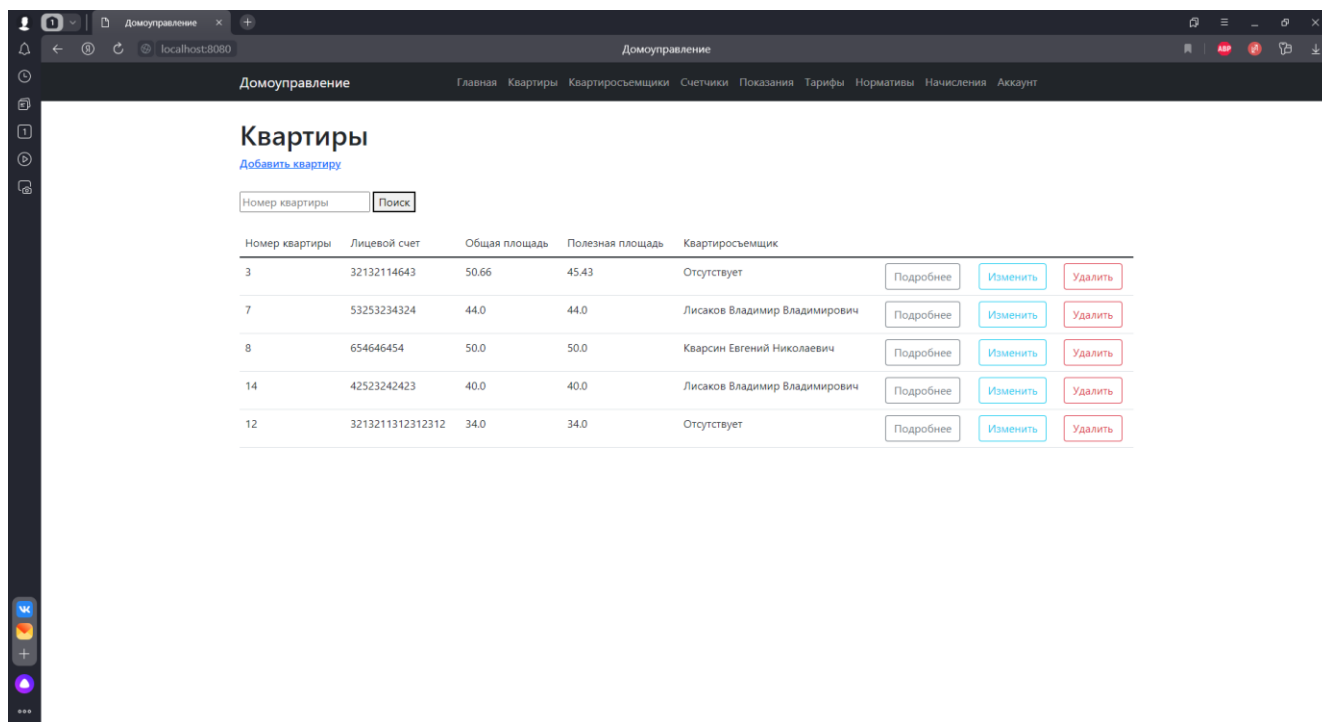


Рисунок 6 – Скриншот страницы с квартирами

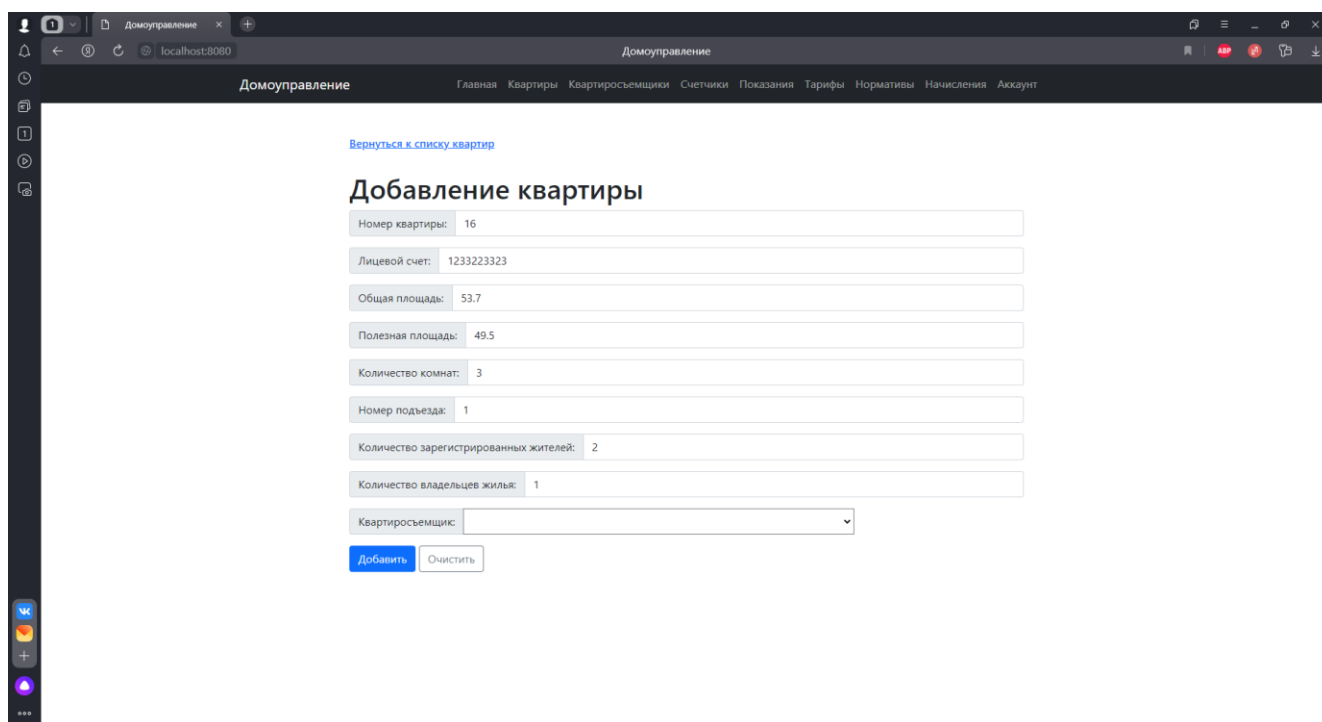


Рисунок 7 – Скриншот страницы с добавлением квартиры

Домоуправление

[Вернуться к списку квартир](#)

Изменение квартиры

Номер квартиры: 8

Лицевой счет: 654646454

Общая площадь: 50,0

Полезная площадь: 50,0

Количество комнат: 2

Номер подъезда: 3

Количество зарегистрированных жителей: 0

Количество владельцев жилья: 2

Квартиросъемщик: ФИО: Кварсин Евгений Николаевич, Номер телефона: +79075442345

[Сохранить](#)

Рисунок 8 - Скриншот страницы с изменением квартиры

Домоуправление

[Вернуться к списку квартир](#)

Удаление квартиры

Номер квартиры: 3

Лицевой счет: 32132114643

Общая площадь: 50,66

Полезная площадь: 45,43

Количество комнат: 2

Номер подъезда: 1

Количество зарегистрированных жителей: 3

Количество владельцев жилья: 2

Квартиросъемщик:

[Удалить](#)

Рисунок 9 - Скриншот страницы с удалением квартиры

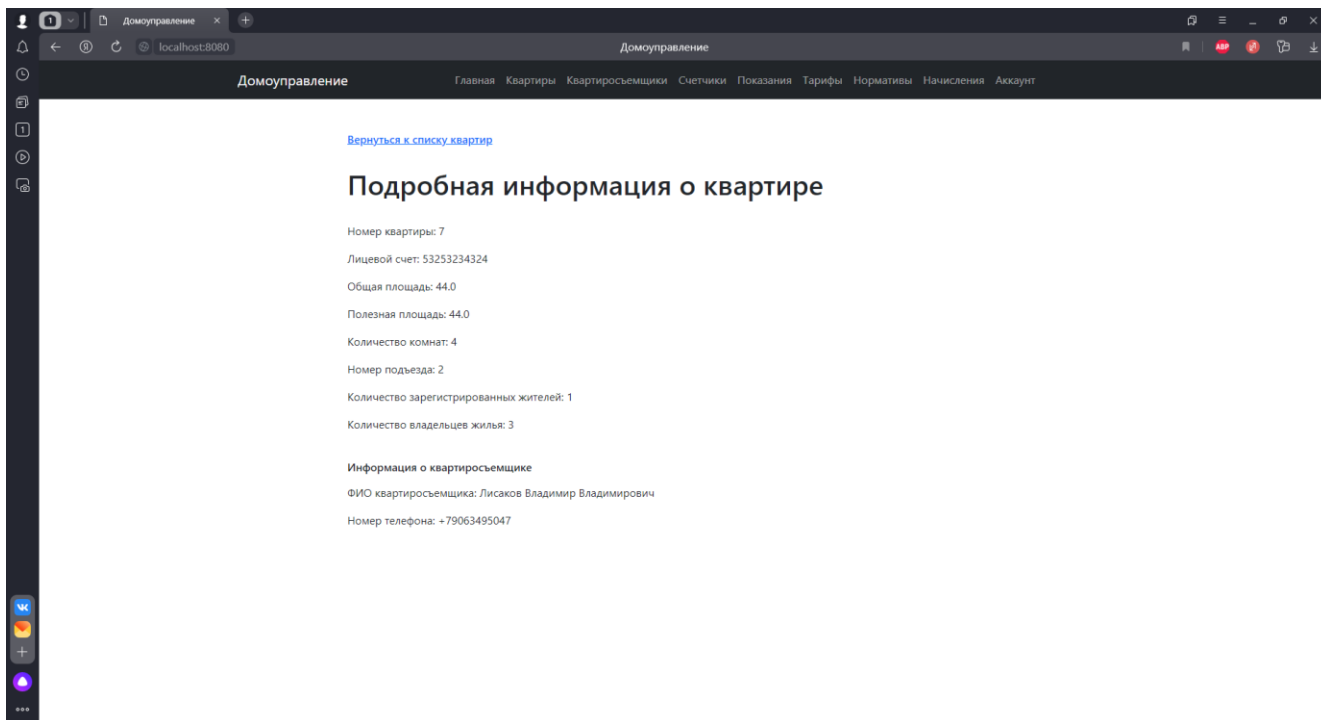


Рисунок 10 - Скриншот страницы с подробной информацией о квартире

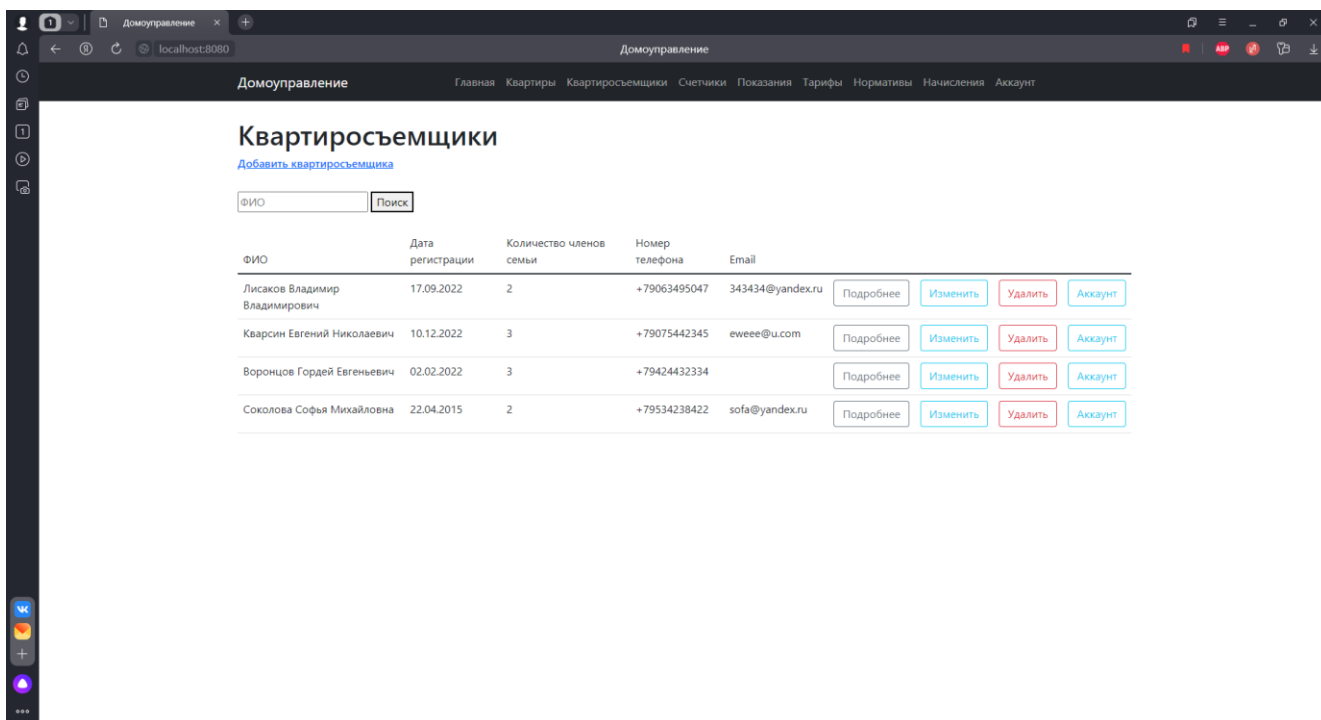


Рисунок 11 – Скриншот страницы с квартиросъемщиками

Домоуправление

[Вернуться к списку квартиросъемщиков](#)

Добавление квартиросъемщика

ФИО:

Дата регистрации:

Количество членов семьи:

Номер телефона:

Email:

Рисунок 12 – Скриншот страницы с добавлением квартиросъемщика

Домоуправление

[Вернуться к списку квартиросъемщиков](#)

Изменение квартиросъемщика

ФИО:

Дата регистрации:

Количество членов семьи:

Номер телефона:

Email:

Рисунок 13– Скриншот страницы с изменением квартиросъемщика

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		60

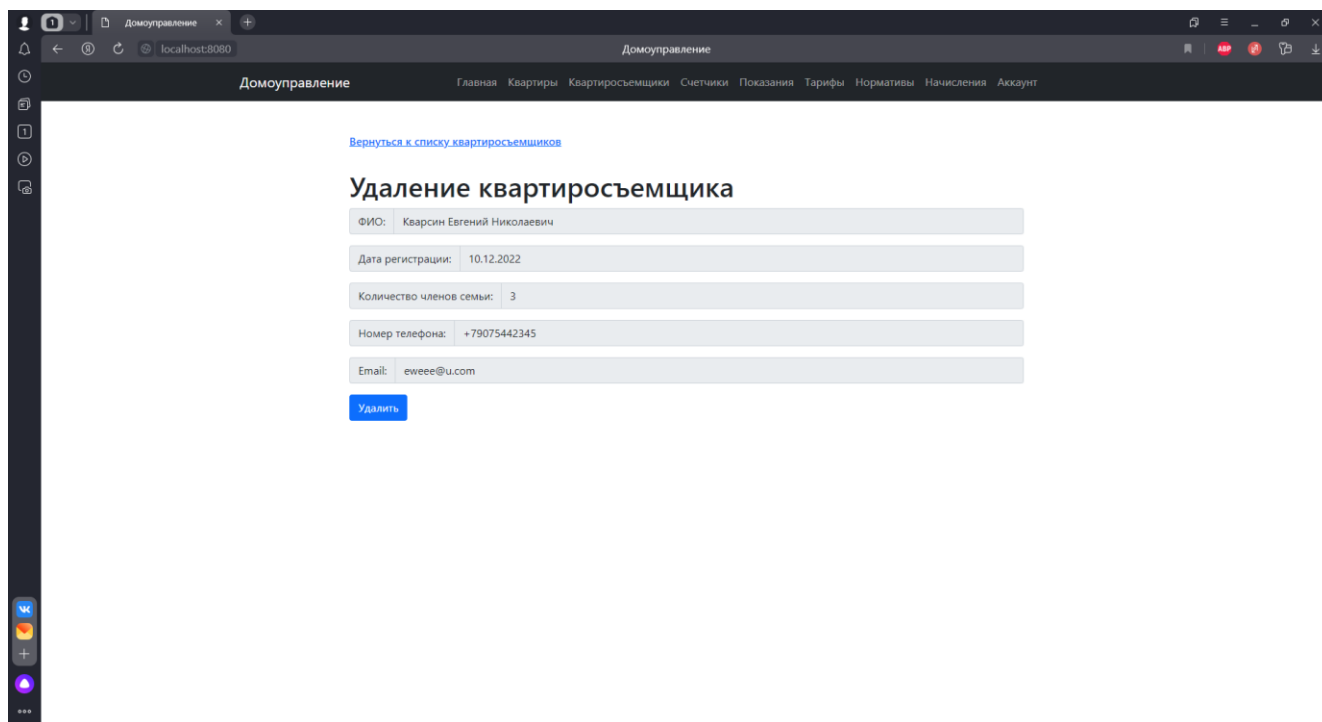


Рисунок 14 – Скриншот страницы с удаление квартиросъемщика

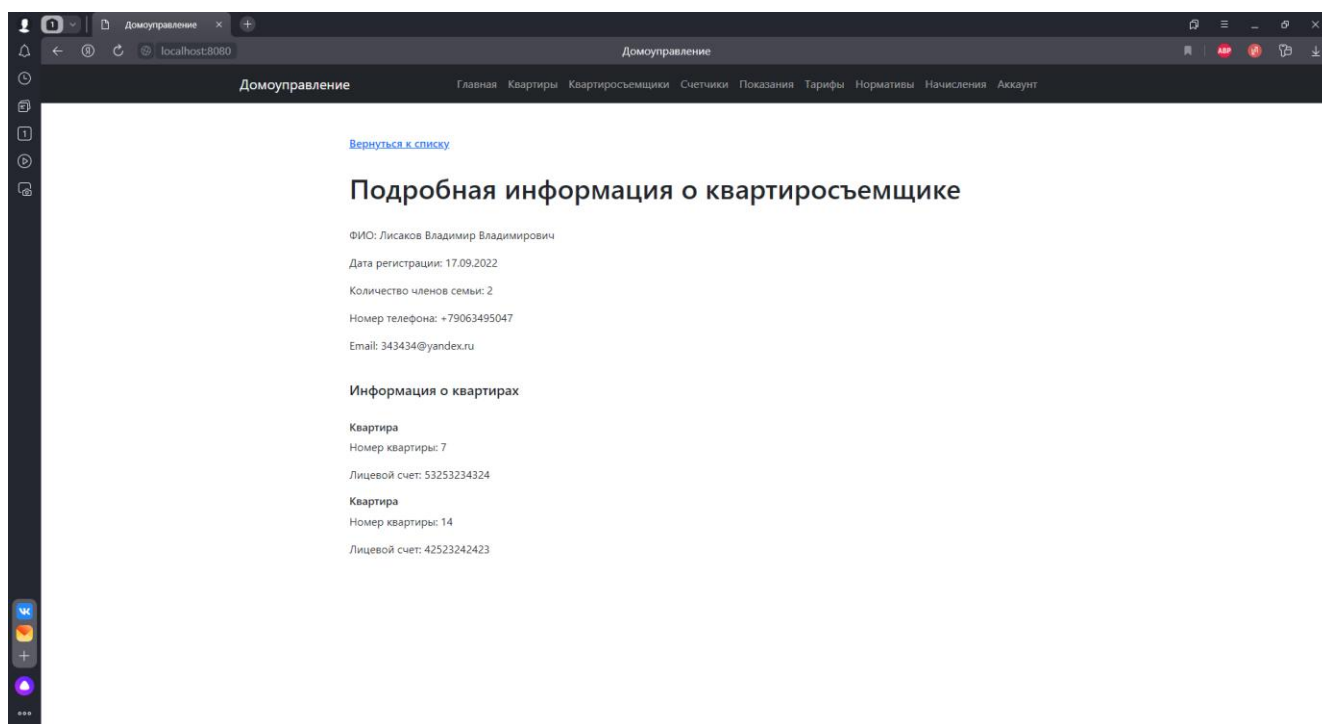


Рисунок 15 – Скриншот страницы с подробной информацией о квартиросъемщике

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		61

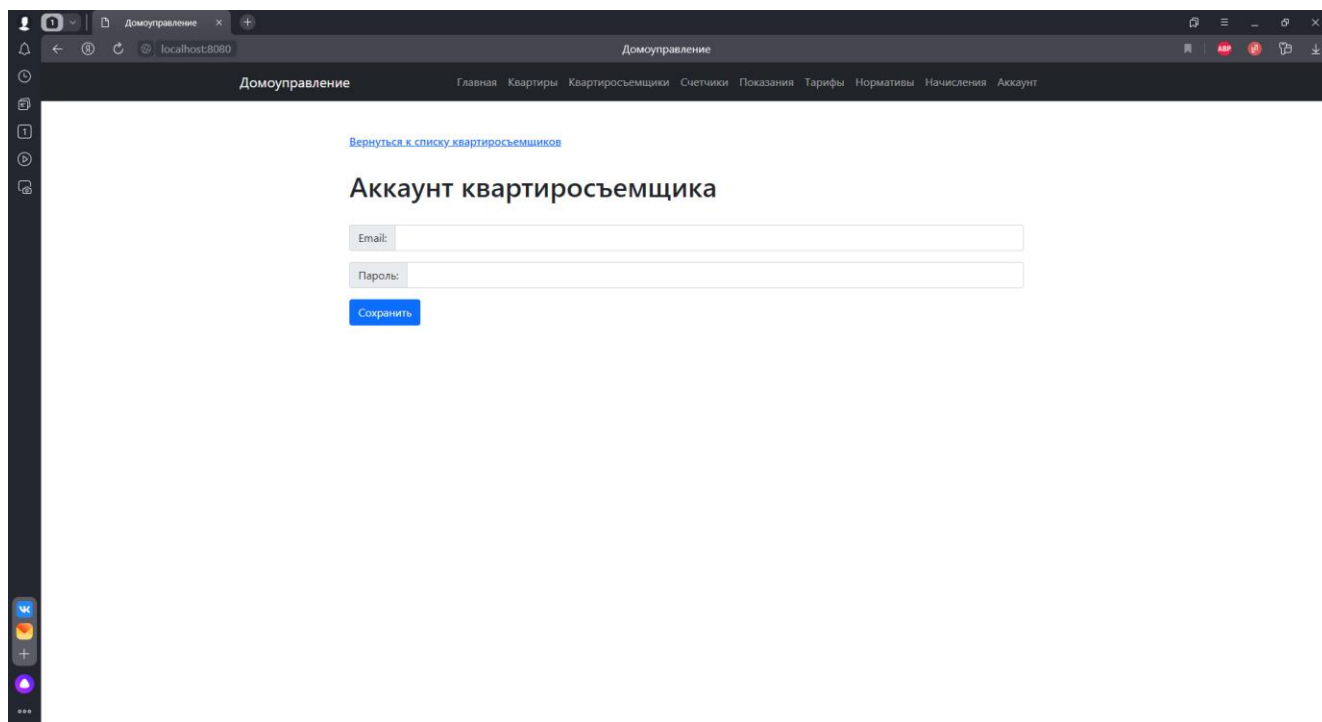


Рисунок 16 – Скриншот страницы с изменением аккаунта квартиросъемщика

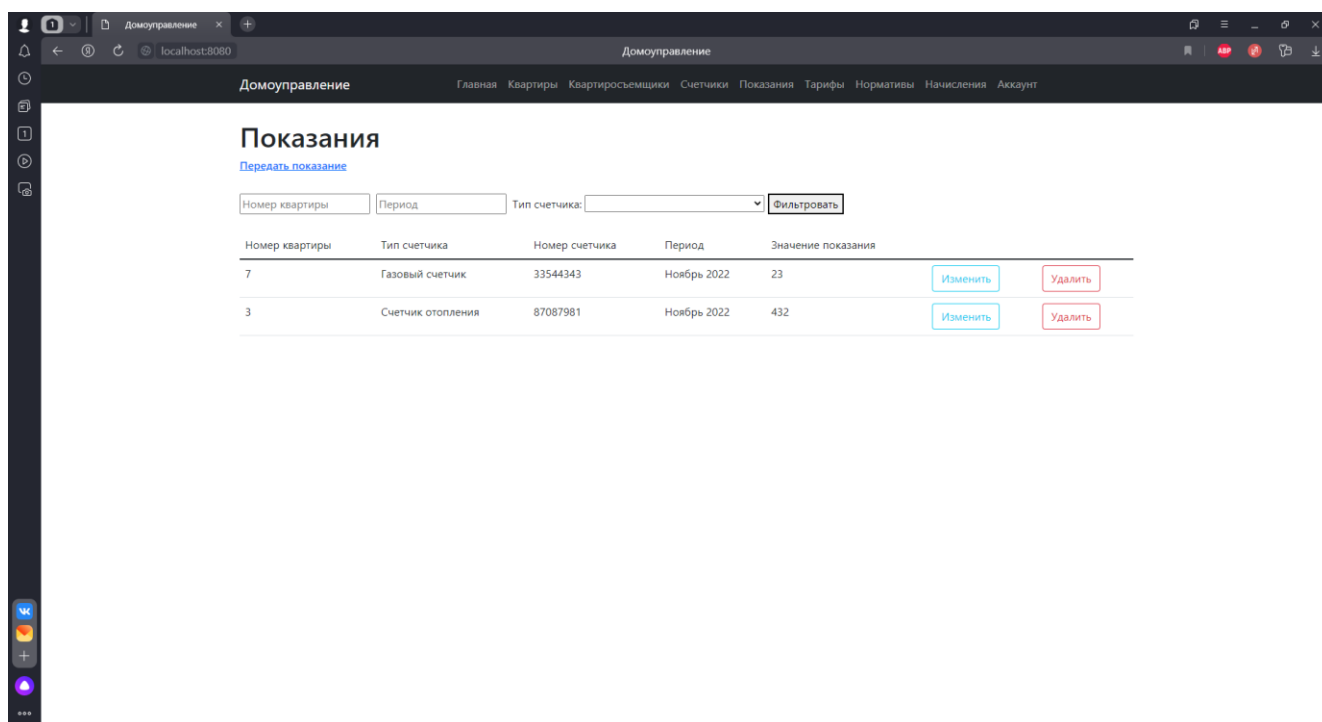


Рисунок 50 – Скриншот страницы с показаниями

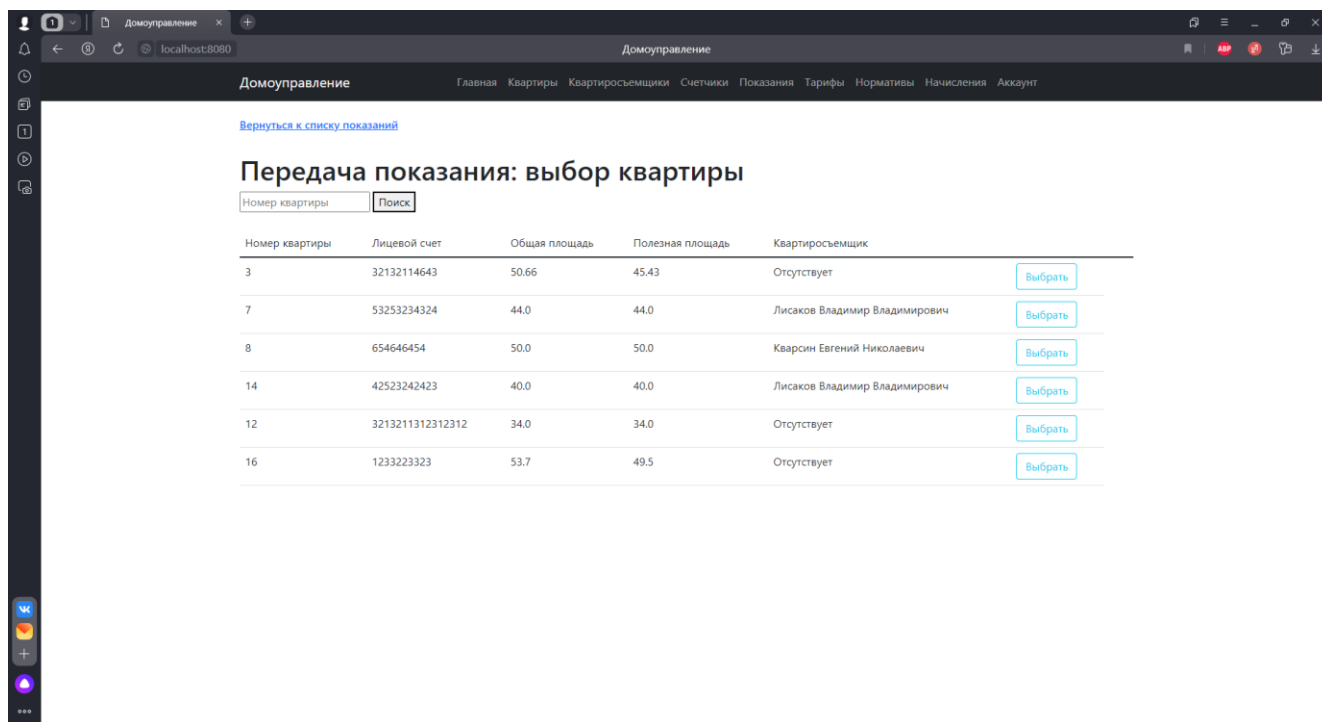


Рисунок 17 – Скриншот формы с передачей показаний: выбор квартиры

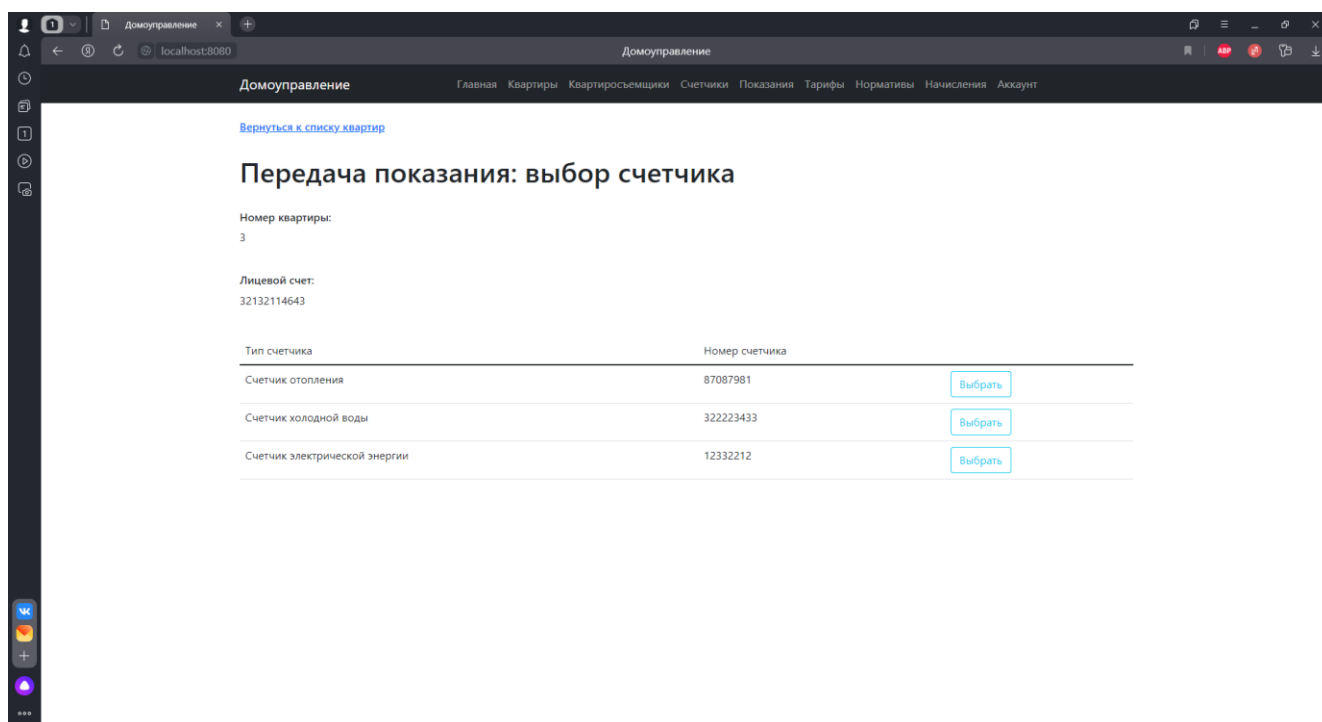


Рисунок 18 – Скриншот формы с передачей показаний: выбор счетчика

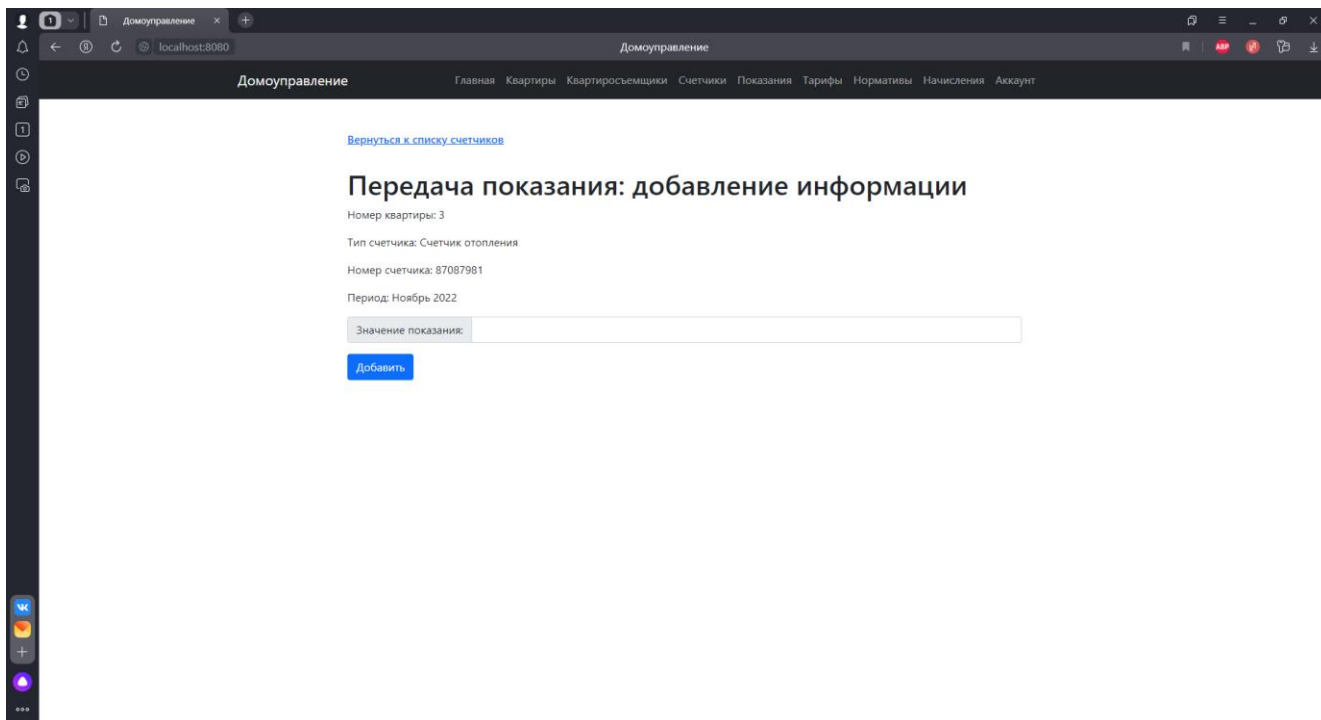


Рисунок 19 – Скриншот формы с передачей показаний: добавление информации

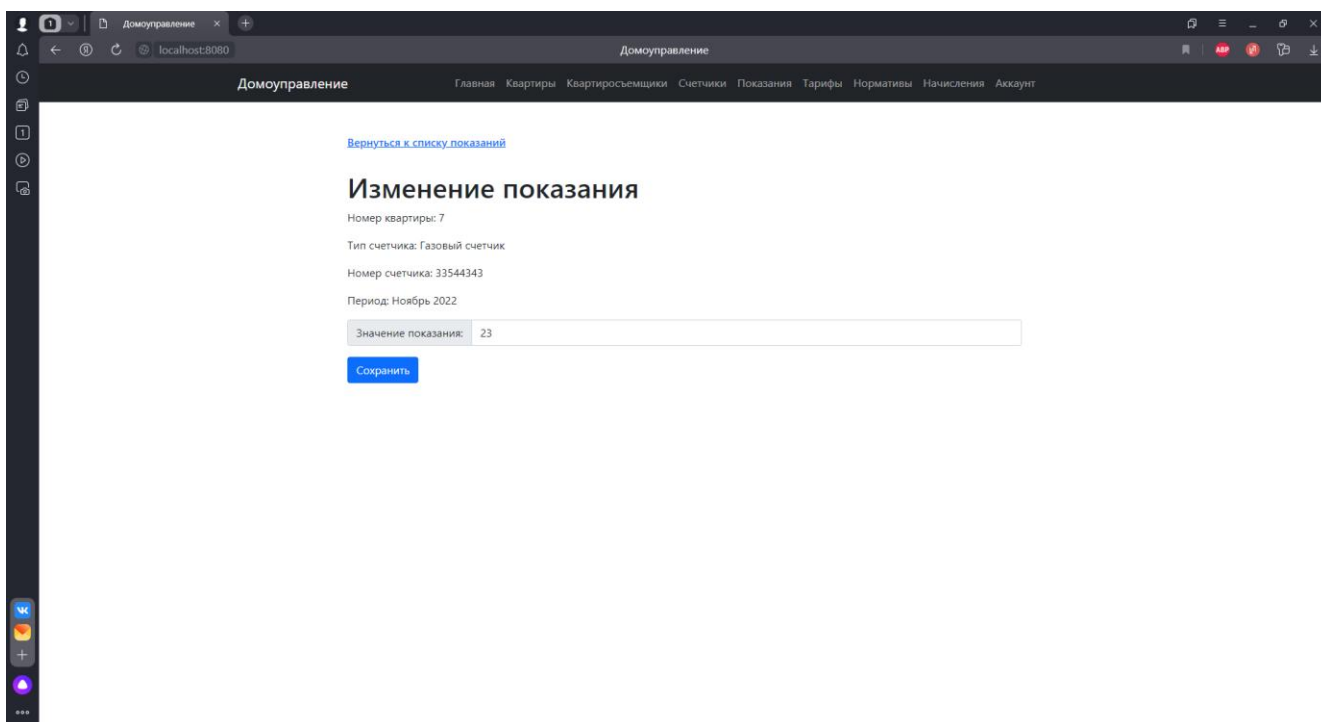


Рисунок 20 - Скриншот страницы с изменением показания

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		64

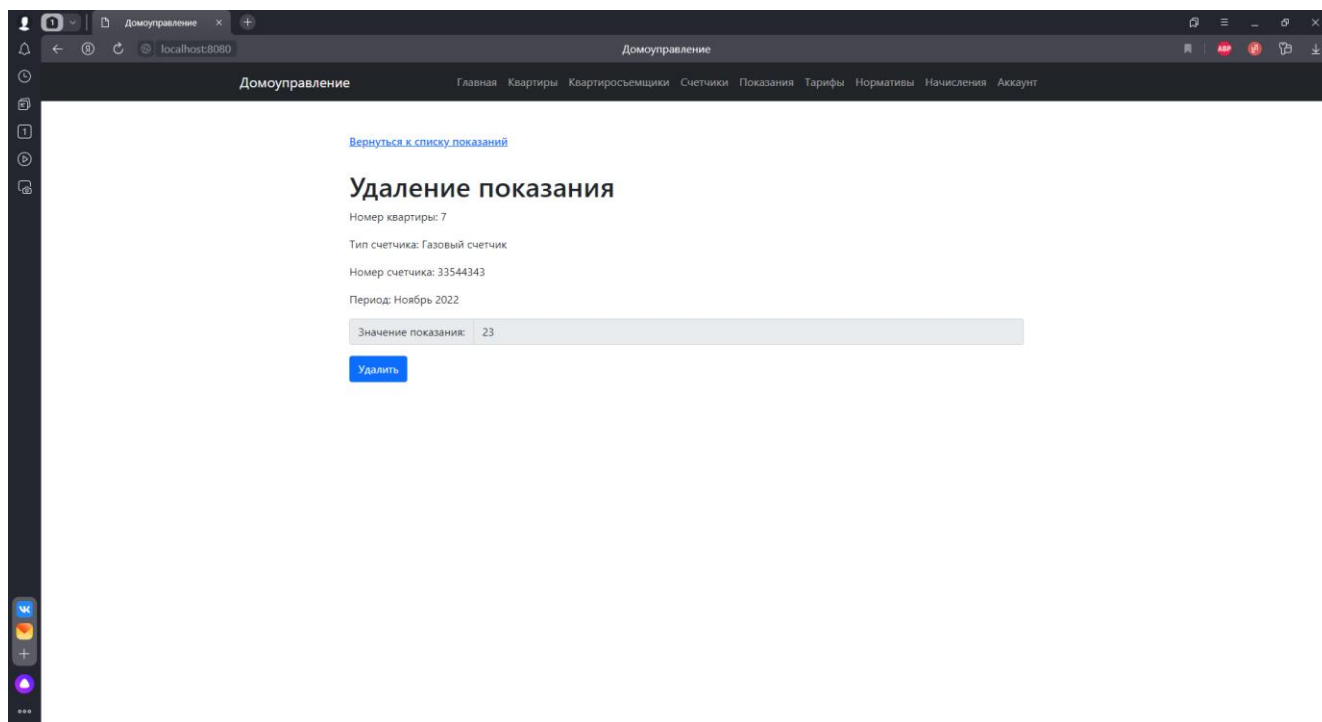


Рисунок 21 - Скриншот страницы с удалением показания

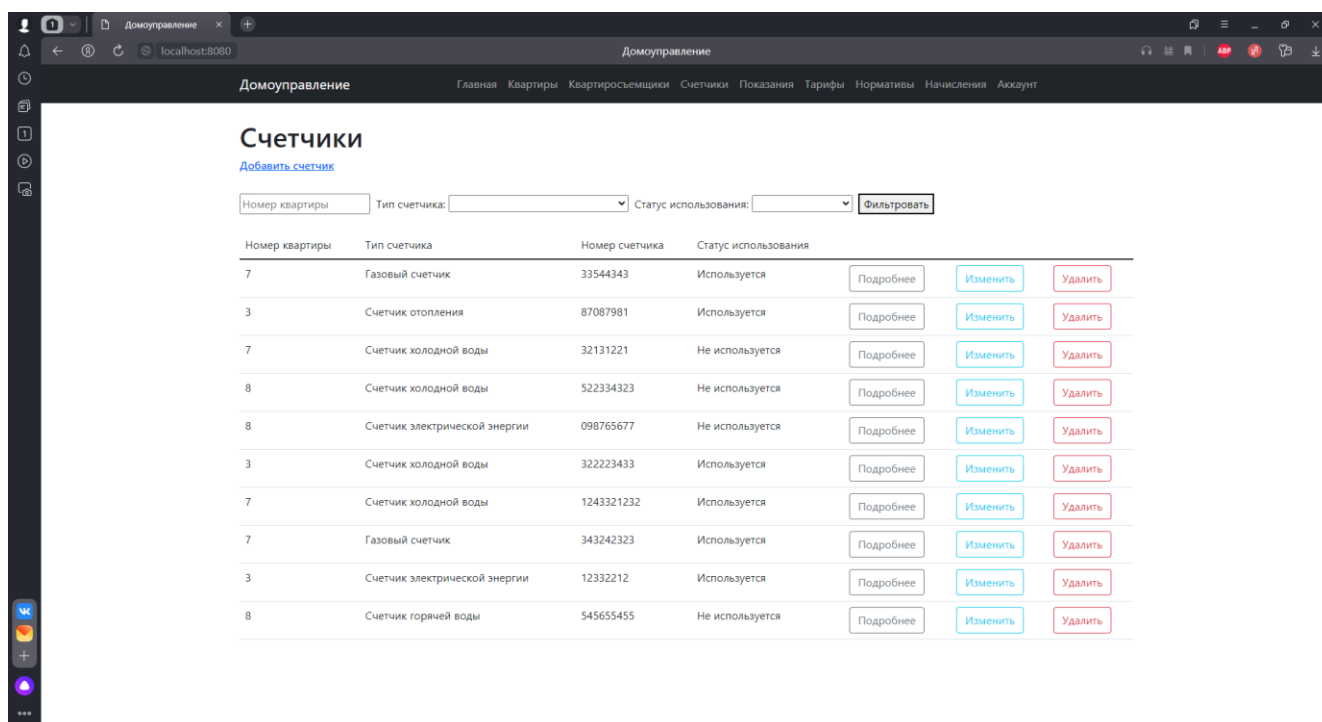


Рисунок 22 – Скриншот страницы со счетчиками

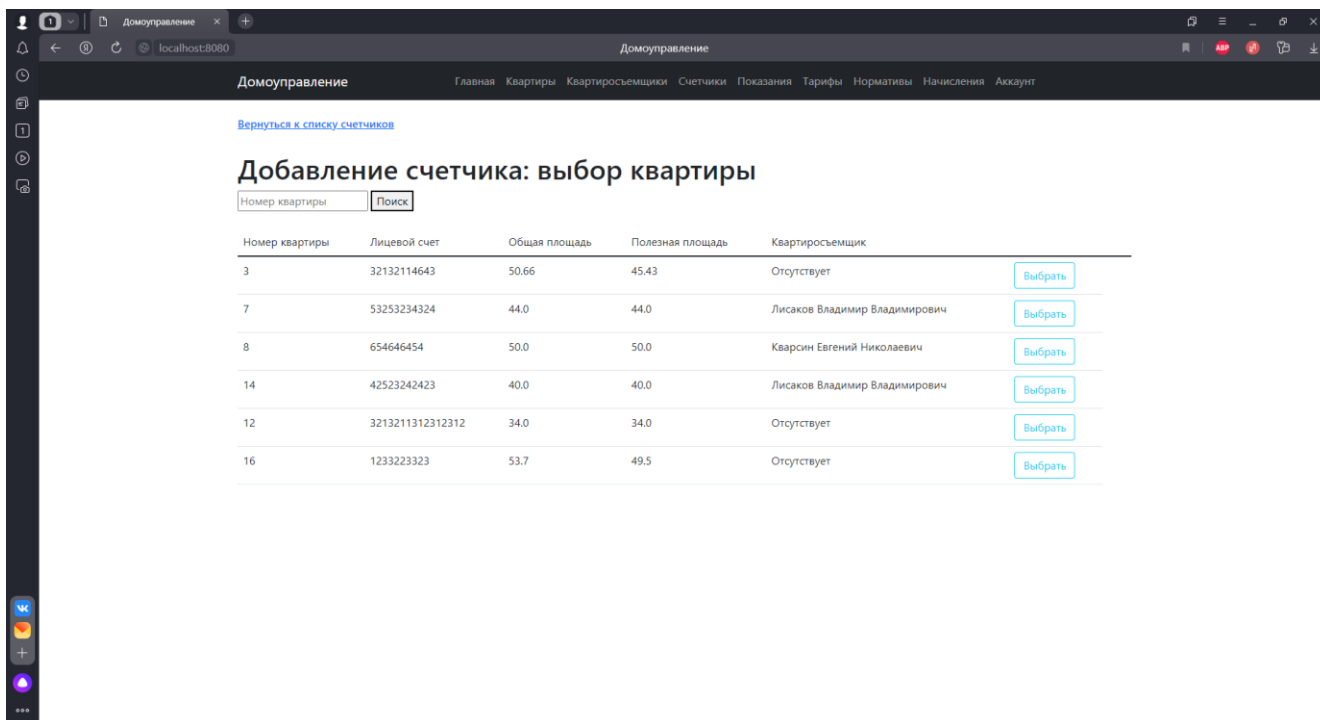


Рисунок 23 - Скриншот страницы с созданием счетчика: выбор квартиры

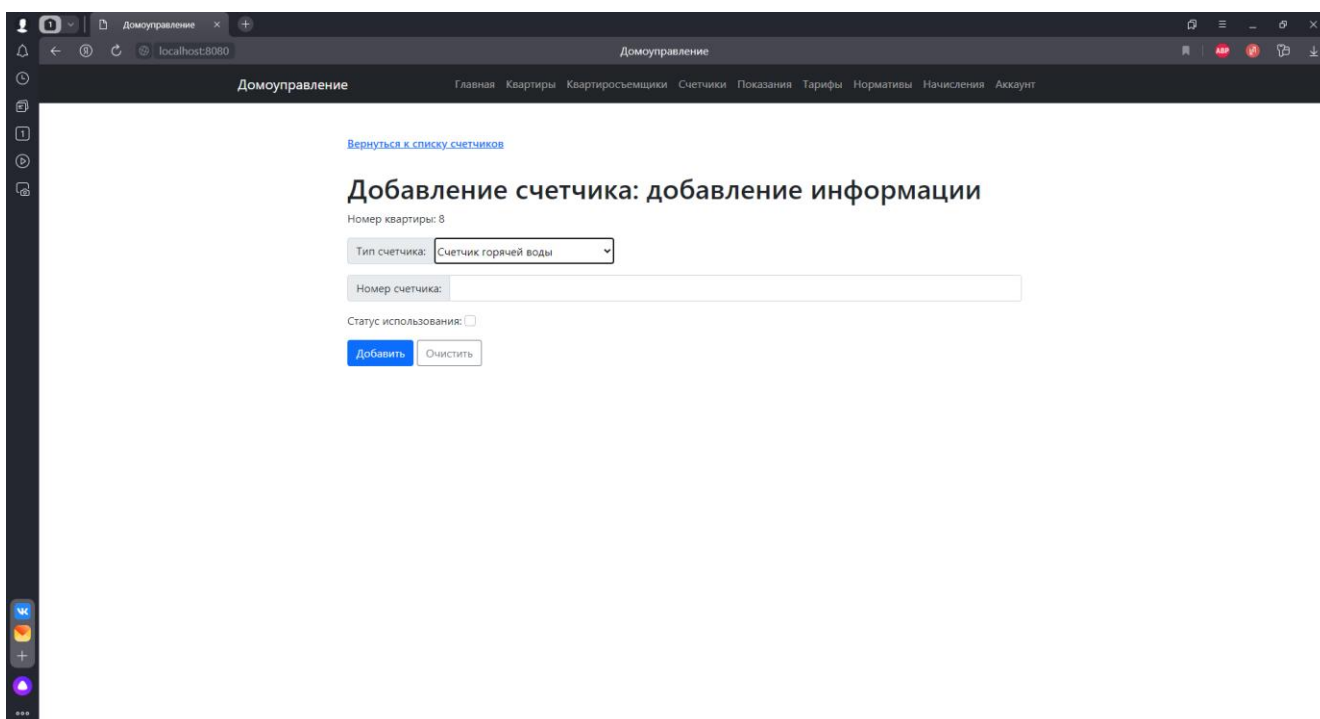


Рисунок 24 - Скриншот страницы с созданием счетчика: добавление информации

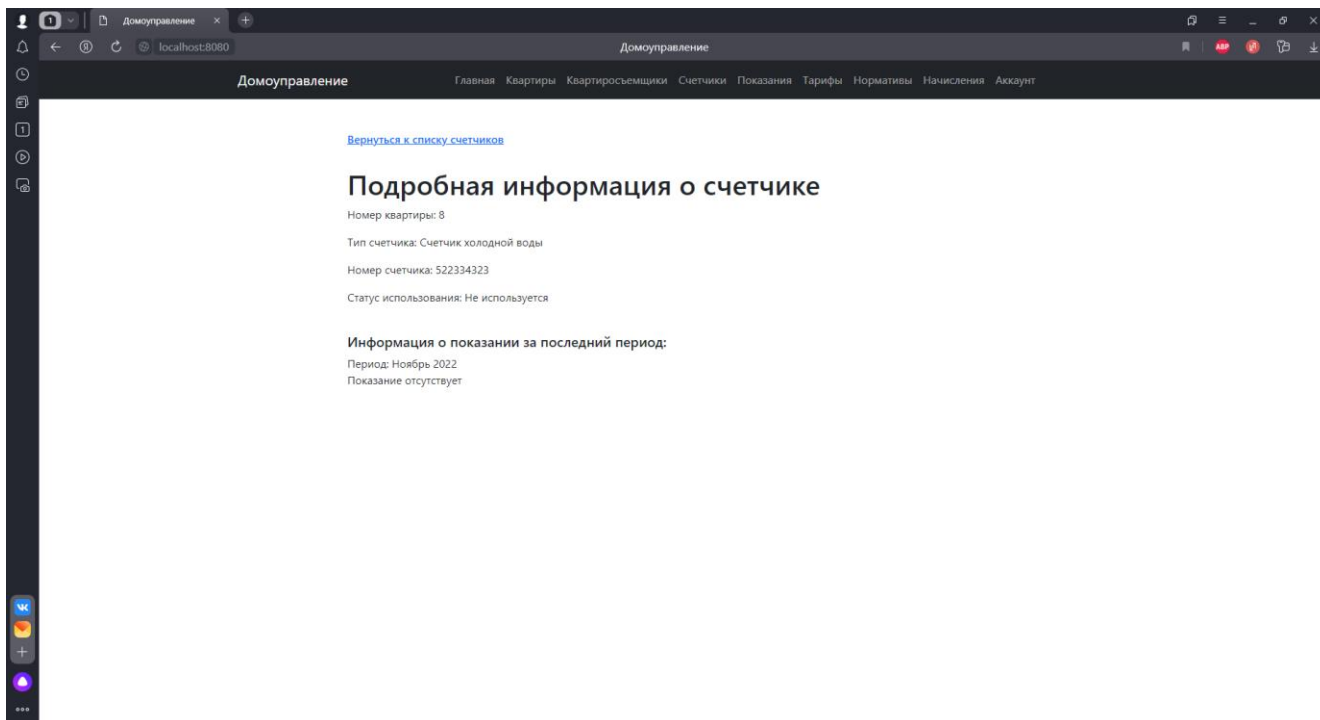


Рисунок 25 - Скриншот страницы с побробной информацией о счетчике

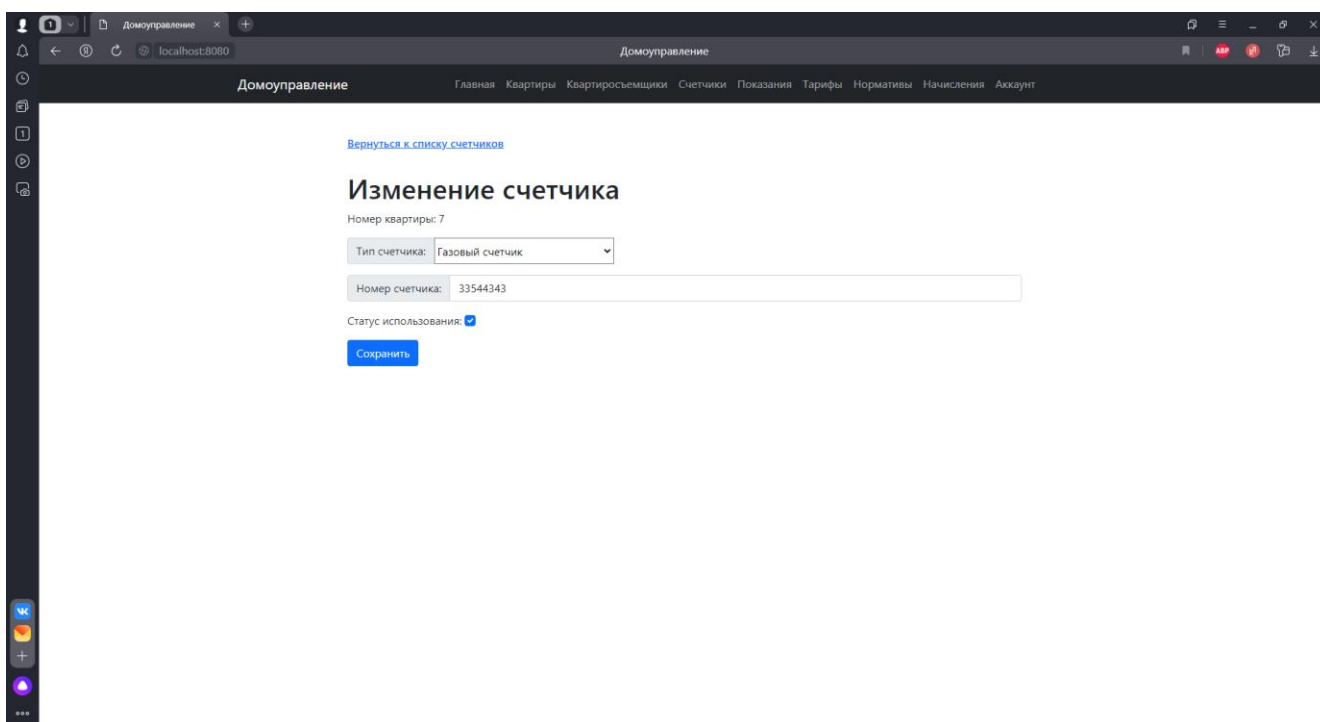


Рисунок 26 - Скриншот страницы с изменением счетчика

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		67

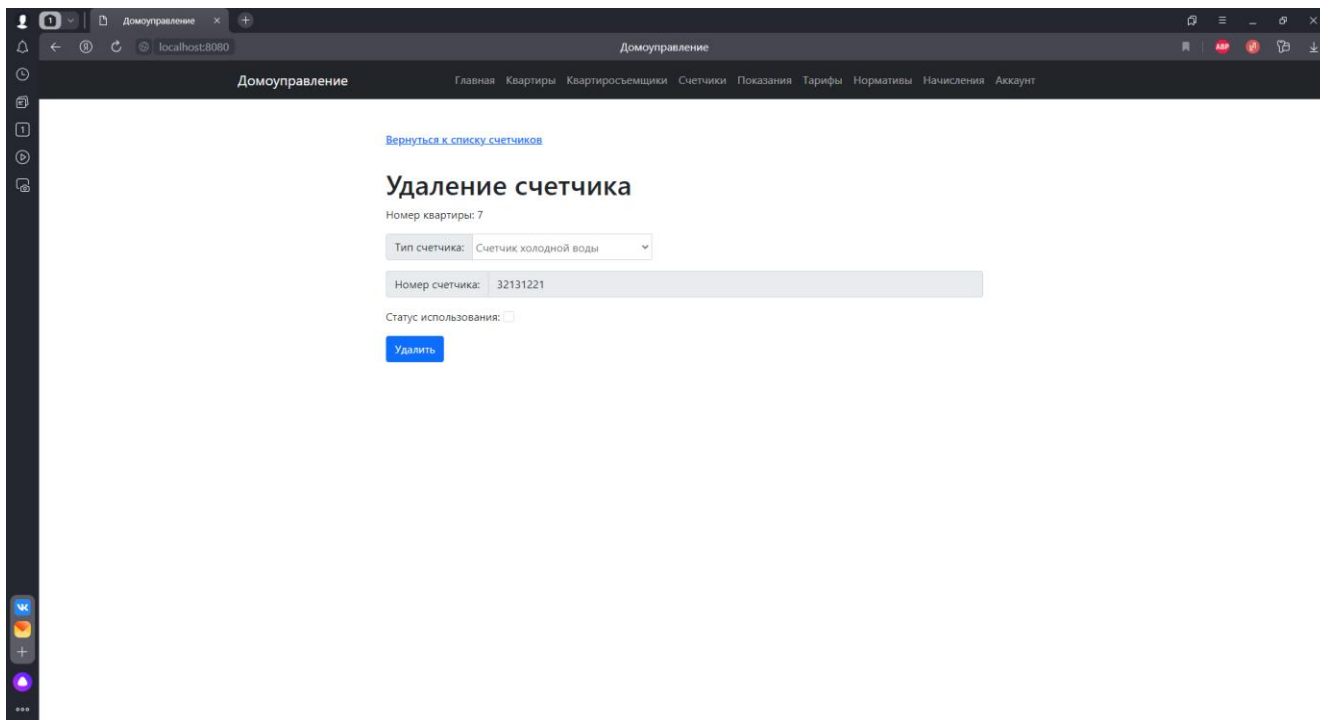


Рисунок 27 - Скриншот страницы с удалением счетчика

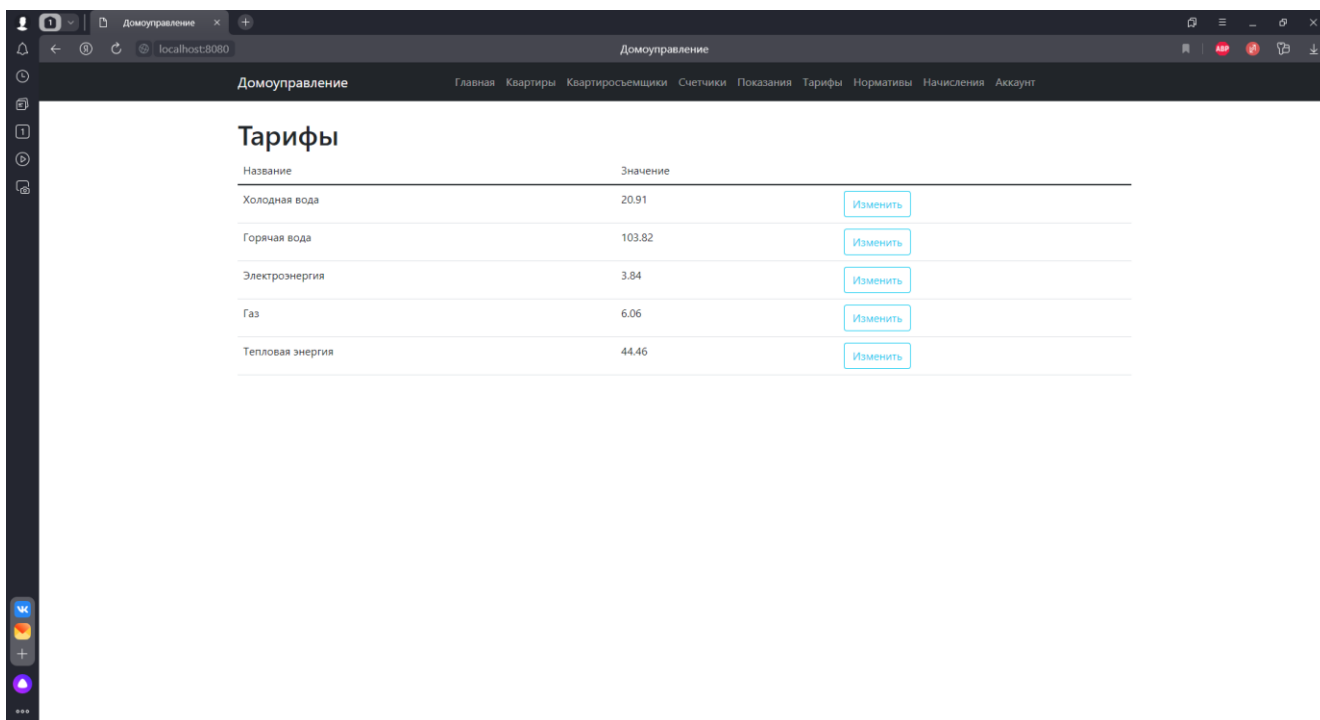


Рисунок 28 - Скриншот страницы с тарифами

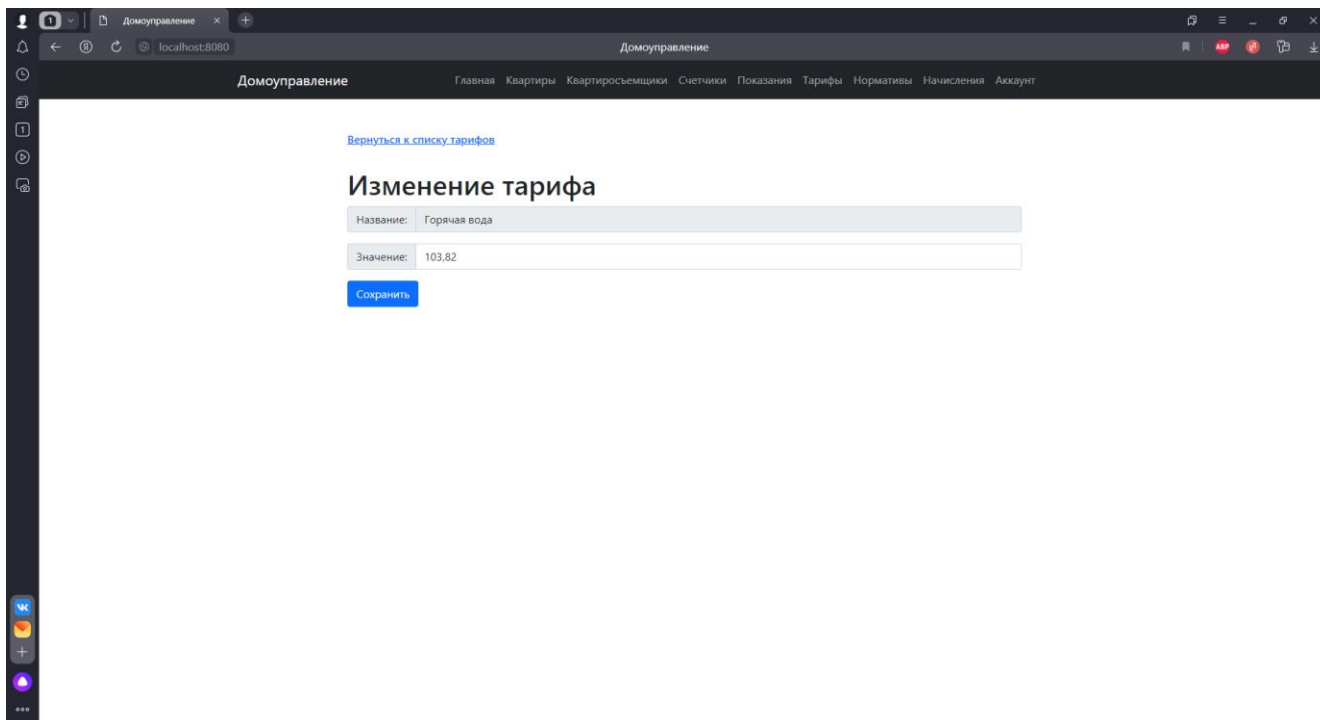


Рисунок 29 - Скриншот страницы с изменением тарифа

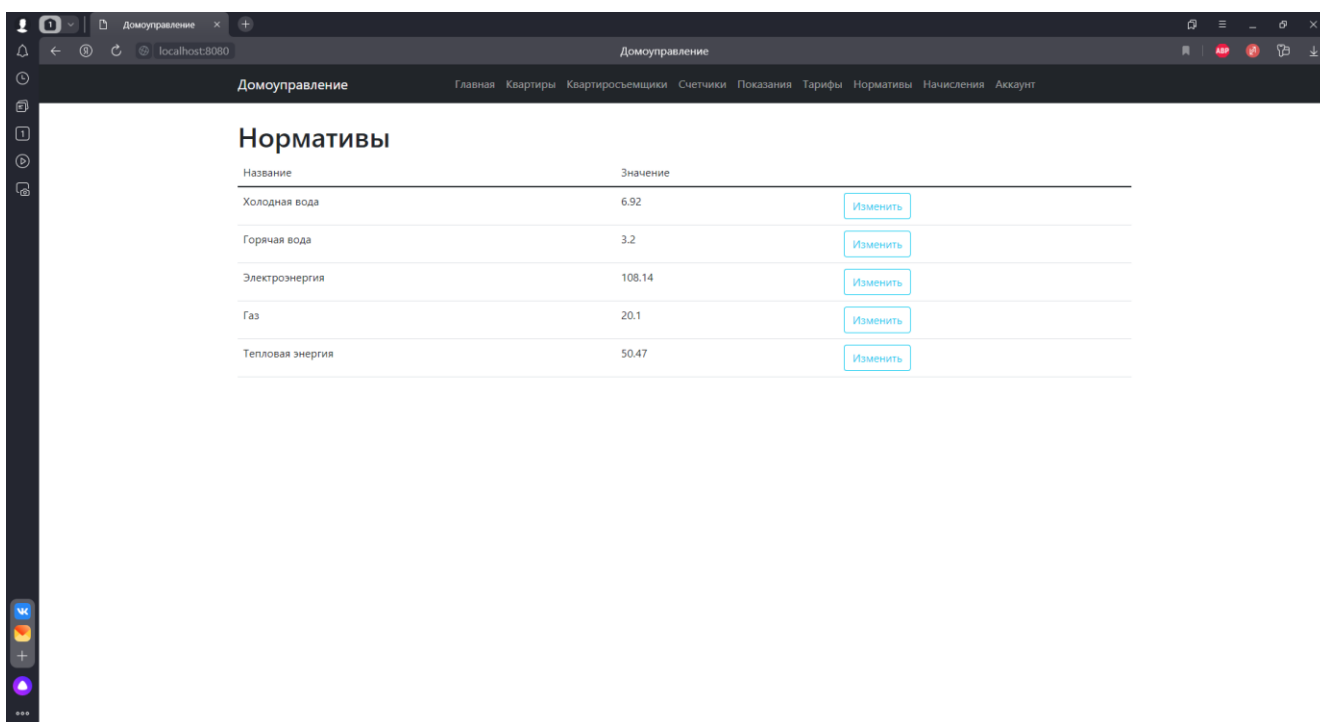


Рисунок 30 - Скриншот страницы с нормативами

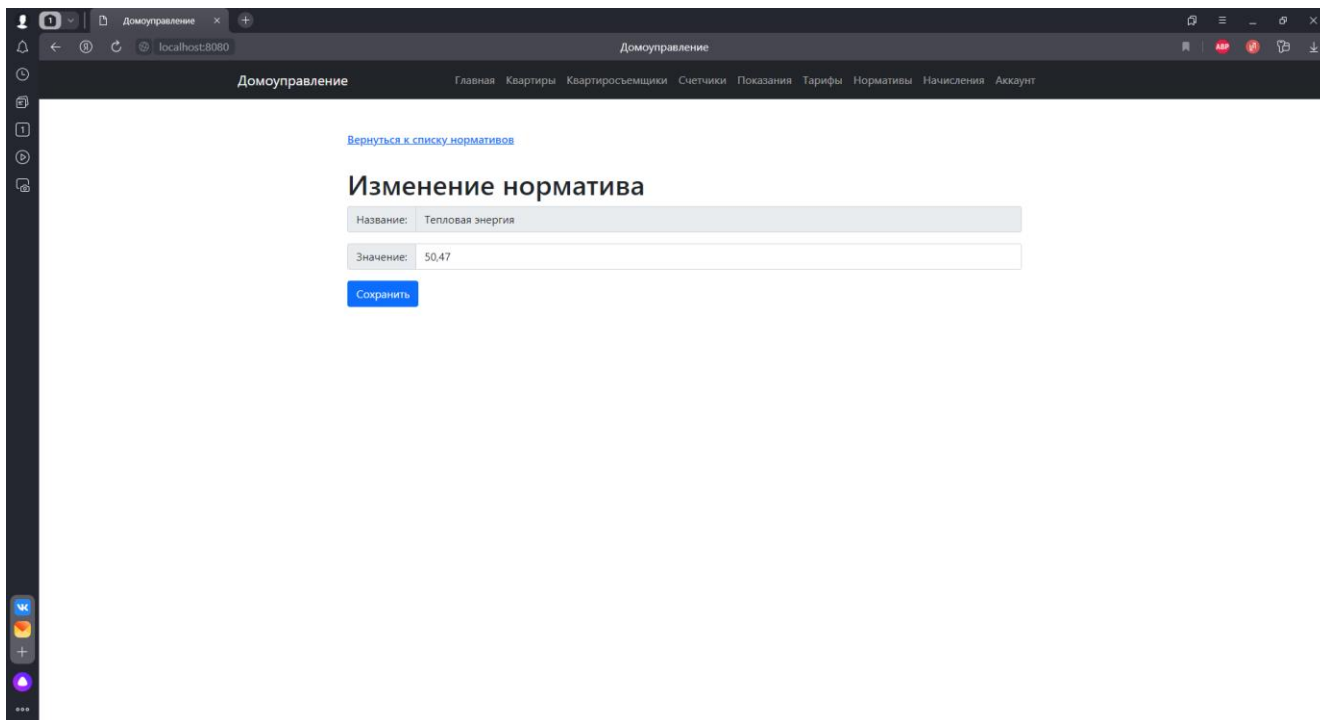


Рисунок 31 - Скриншот страницы с изменением норматива

Номер квартиры	Услуга	Период	Сумма	Статус оплаты	Подробнее	Изменить
3	Холодная вода	Ноябрь 2022	434.1	Не оплачено	Подробнее	Изменить
3	Горячая вода	Ноябрь 2022	996.7	Оплачено	Подробнее	Изменить
3	Электроэнергия	Ноябрь 2022	1245.8	Не оплачено	Подробнее	Изменить
3	Газ	Ноябрь 2022	365.4	Не оплачено	Подробнее	Изменить
3	Тепловая энергия	Ноябрь 2022	19206.7	Не оплачено	Подробнее	Изменить
7	Холодная вода	Ноябрь 2022	144.7	Не оплачено	Подробнее	Изменить
7	Горячая вода	Ноябрь 2022	332.2	Не оплачено	Подробнее	Изменить
7	Электроэнергия	Ноябрь 2022	415.3	Не оплачено	Подробнее	Изменить
7	Газ	Ноябрь 2022	261.2	Не оплачено	Подробнее	Изменить
7	Тепловая энергия	Ноябрь 2022	2220.7	Не оплачено	Подробнее	Изменить
8	Холодная вода	Ноябрь 2022	289.4	Не оплачено	Подробнее	Изменить
8	Горячая вода	Ноябрь 2022	664.4	Не оплачено	Подробнее	Изменить

Рисунок 32 – Скриншот страницы с начислениями

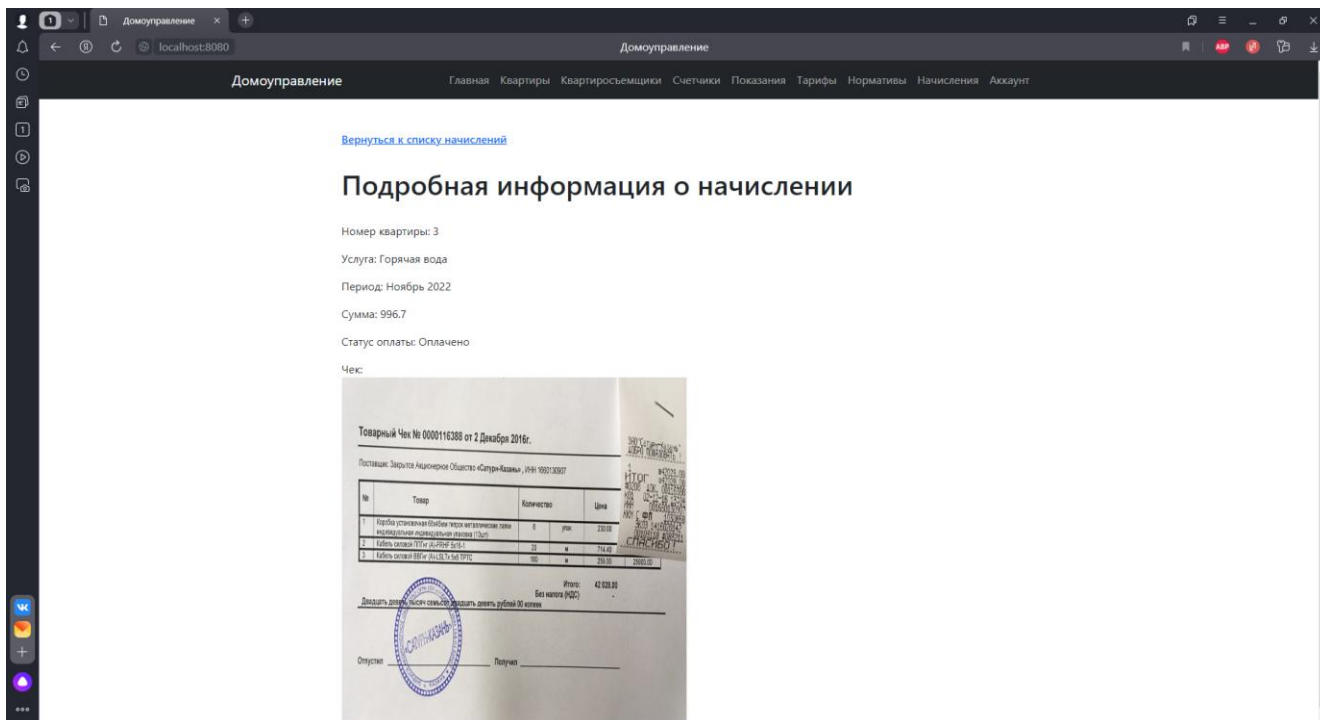


Рисунок 35 - Скриншот страницы с подробной информацией о начислении

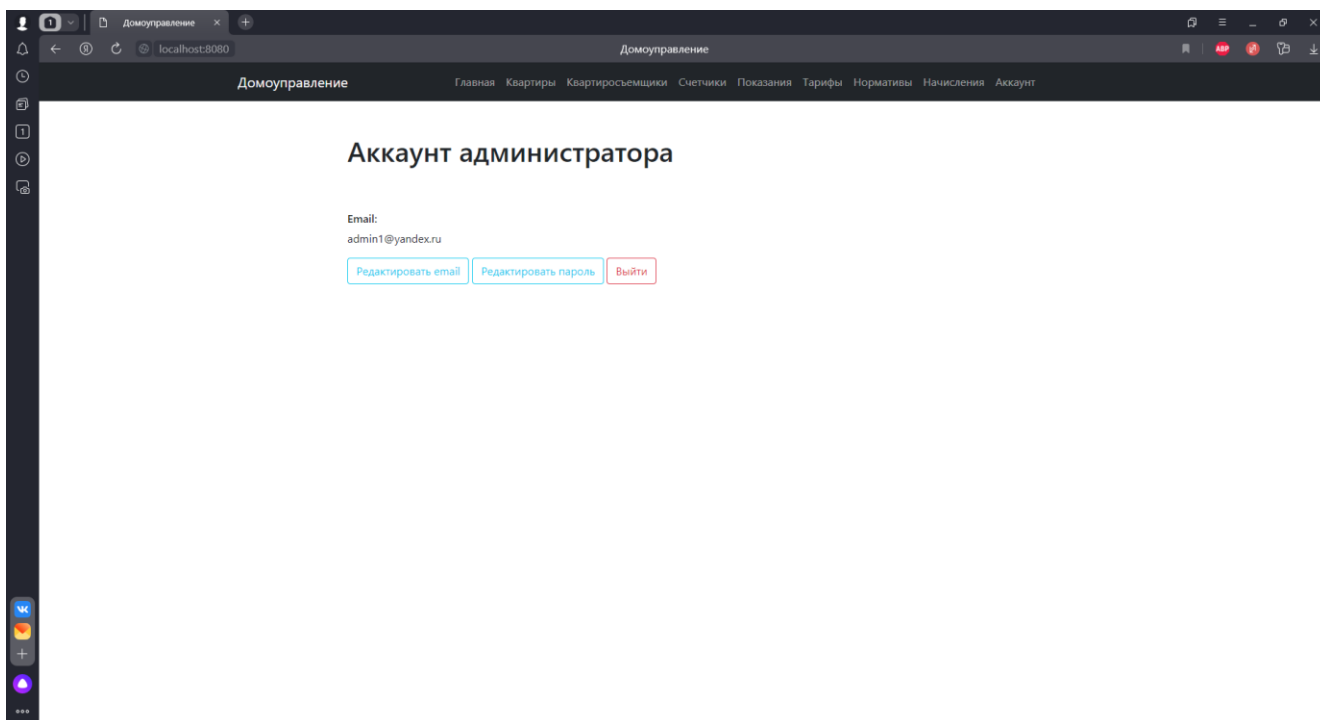


Рисунок 36 - Скриншот страницы с информацией о профиле

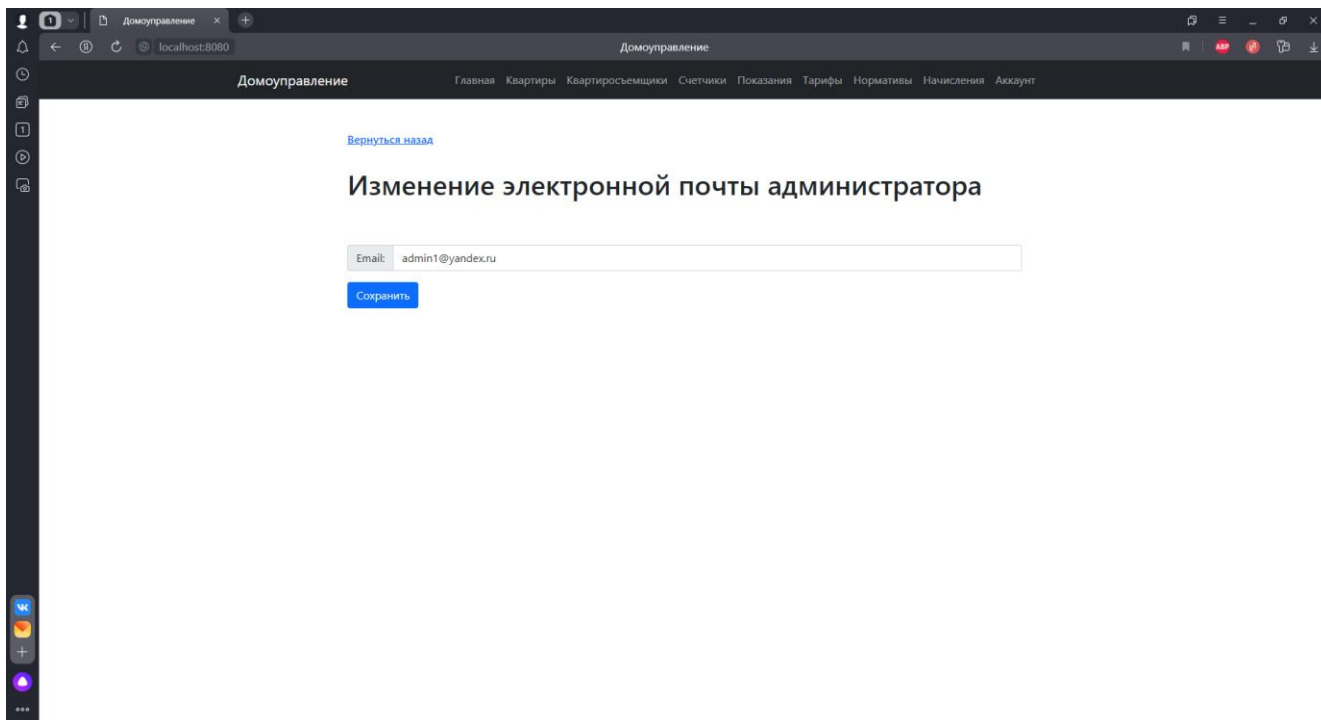


Рисунок 37 - Скриншот страницы с изменением логина профиля

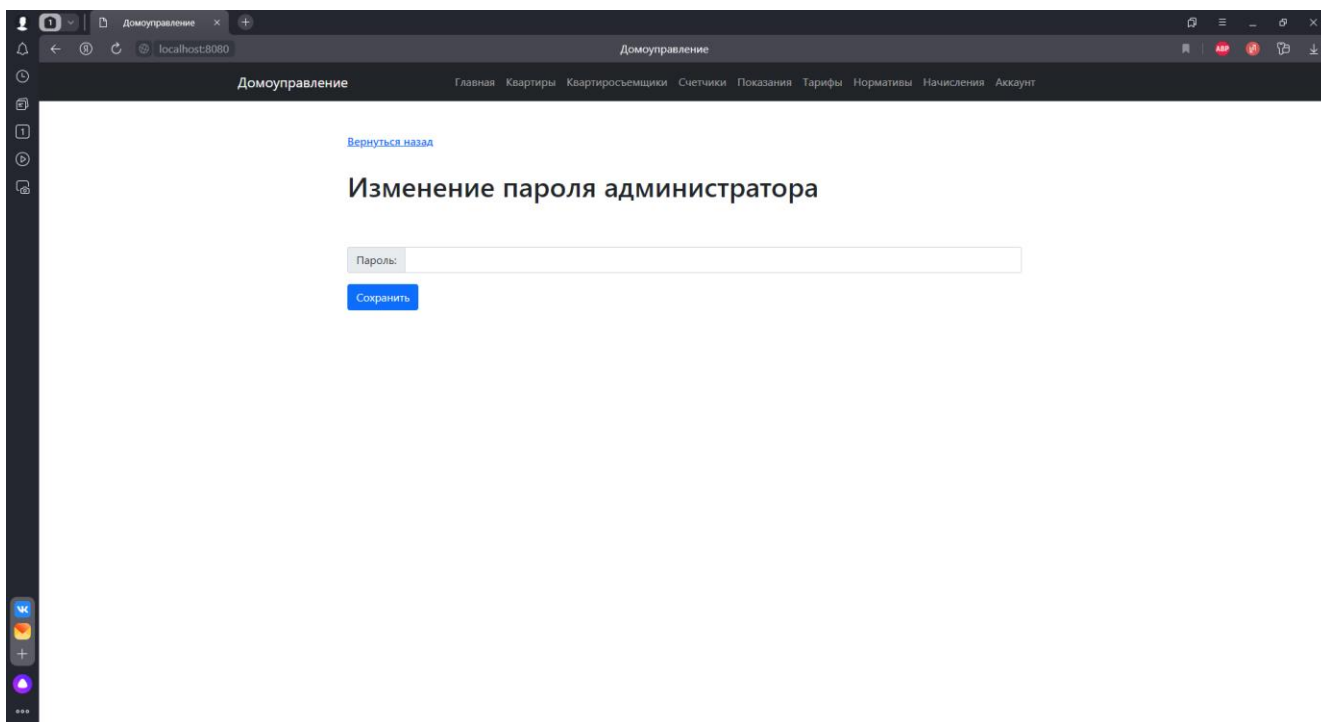


Рисунок 38 - Скриншот страницы с изменением пароля

					МИВУ 09.03.04-13.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		73

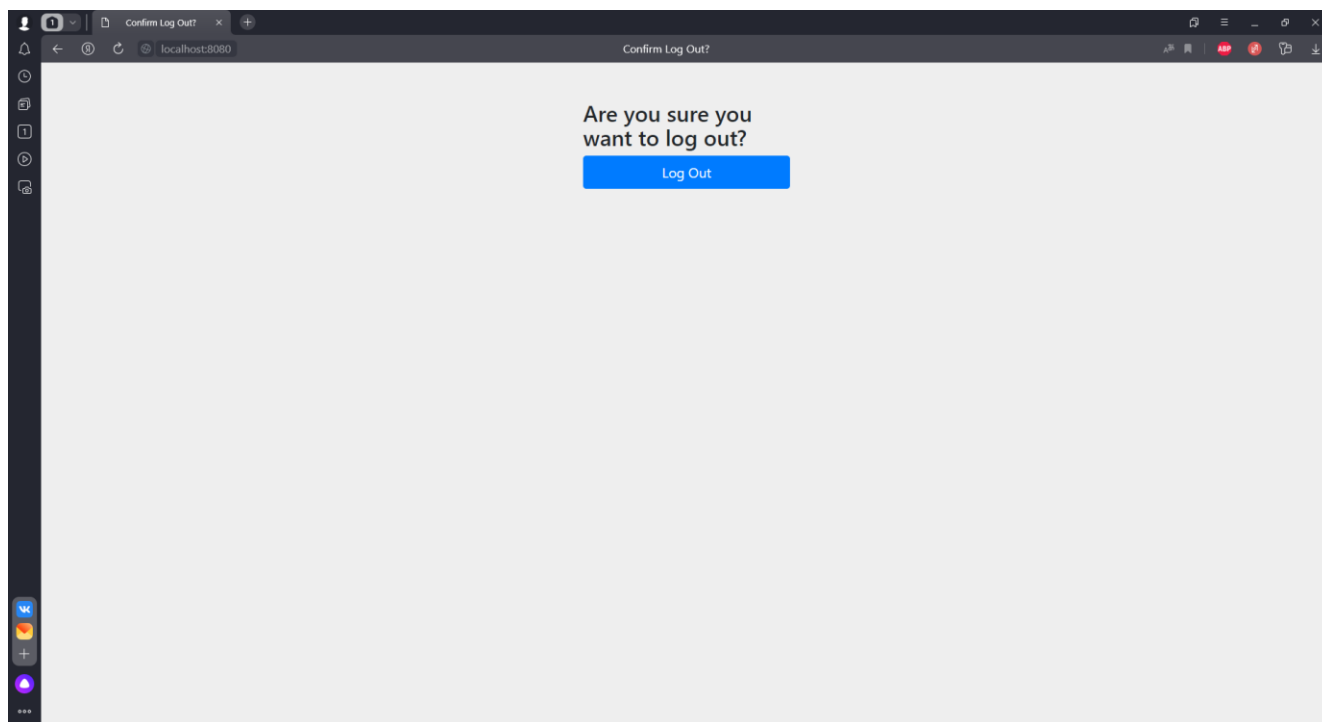


Рисунок 39 - Скриншот страницы с выходом из профиля

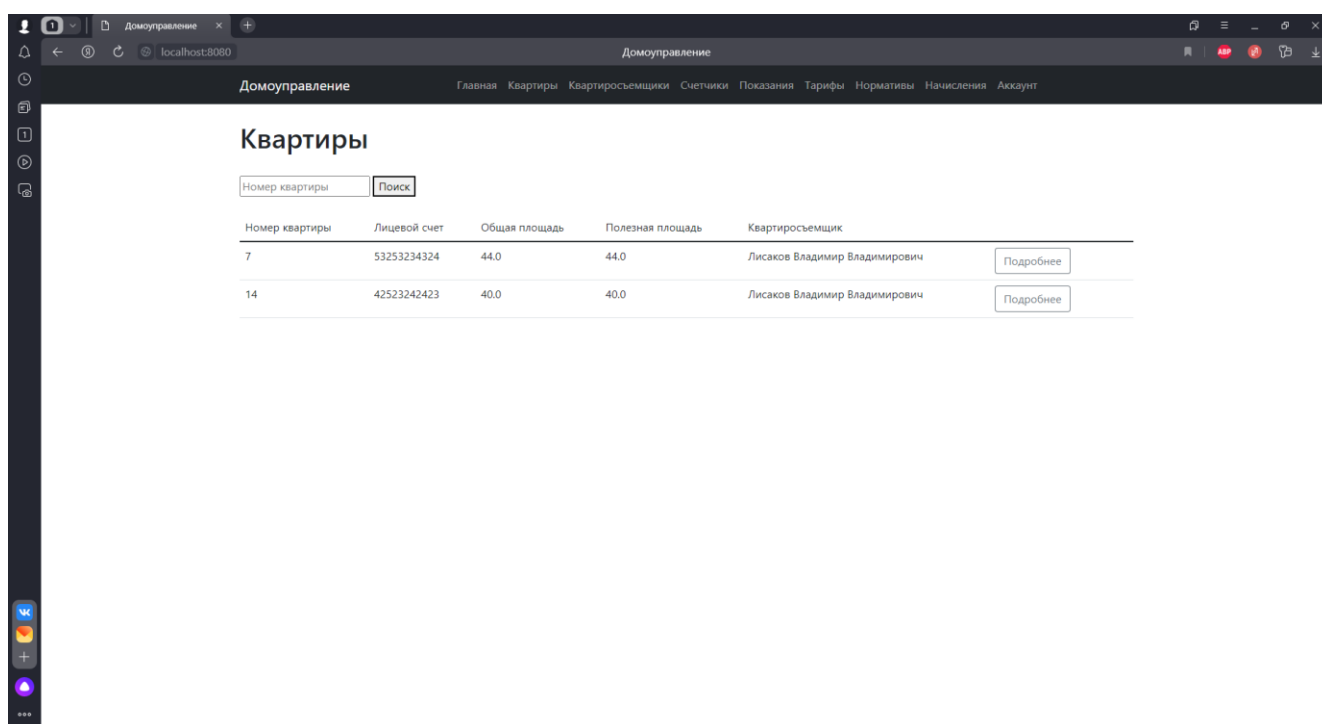


Рисунок 40 - Скриншот примера страницы за профилем квартиросъемщика