**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”  
(УНИВЕРСИТЕТ ИТМО)**

**ЦЕНТР АВТОРИЗОВАННОГО ОБУЧЕНИЯ ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ**

**ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА**

Разработкаweb-приложения для автоматизации управления посуточной арендой квартир

Автор Маракуева Ольга Валериевна \_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия Имя Отчество) (Подпись)

**Центр авторизованного обучения информационным технологиям**

Наименование программы **«Python-разработчик»**

Руководитель Кузьмин К.М. \_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

**К защите допустить**

|  |  |  |  |
| --- | --- | --- | --- |
| *Заместитель директора ЦАО ИТ,*  *к.т.н.* |  | / *Т.В. Зудилова*/ |  |

Санкт-Петербург, 2023г.

Обучающийся Маракуева Ольга Валериевна Группа 124/02

(Фамилия, И. О.)

Работа принята «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Работа выполнена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Секретарь ИАК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, И. О.) (подпись)

Листов хранения \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Демонстрационных материалов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оглавление

[Введение 4](#_Toc141206827)

[1 Анализ предметной области 5](#_Toc141206828)

[2 Этап проектирования 8](#_Toc141206829)

[3 Описание системы 9](#_Toc141206830)

[3.1 Средства разработки и структура проекта 9](#_Toc141206831)

[3.2 Структура базы данных 12](#_Toc141206832)

[3.3 Реализованные функции системы 13](#_Toc141206833)

[3.4 Архитектура приложения 14](#_Toc141206834)

[3.5 Тестирование 15](#_Toc141206835)

[3.6 Руководство пользователя 17](#_Toc141206836)

[3.7 Руководство по развертыванию приложения 24](#_Toc141206837)

[Заключение 28](#_Toc141206838)

[Использованные источники 28](#_Toc141206839)

# Введение

В последнее время в России становится всё более популярной посуточная аренда квартир как инструмента для организации путешествий и деловых поездок. В связи с этим растет количество сервисов, предоставляющих свои услуги по размещению объявлений о сдаче в аренду и управлению бронированиями. Данные сервисы отличаются концепцией своей работы, часть приложений предоставляет возможность ведения календаря бронирования, другая часть является лишь площадкой по размещению объявлений.

Для увеличения количества клиентов арендодатели используют множество сервисов обеих категорий, что приводит к усложнению организации деятельности. Арендодателю необходимо следить за синхронизацией календарей на сайтах и вносить в них изменения вручную, если арендатор, осуществляющий бронирование, нашел объявление на сайте без функции календаря. Арендодатель тратит существенное количество времени для консультации арендаторов, которые обращаются через сайты без функции календаря, для подбора удобных дат проживания и указания стоимости. Для отслеживания статистики по доходам и бронированиям арендодателю приходится вести учет в ручном режиме или прибегать к услугам платных сервисов.

Целью работы является разработка web-приложения для автоматизации управления посуточной аренды квартир для сокращения времени работы арендодателя и минимизации ошибок.

В работе определены следующие задачи:

* провести анализ предметной области;
* построить логическую и физическую модель данных;
* создать физическую модель, заполнить базу данных;
* написать программу;
* протестировать программу;
* развернуть приложение на веб-хостинге.

Для реализации используются возможности языка программирования Python, фреймворка Django и СУБД MySQL, развертывание приложения осуществляется на сервисе pythonanywhere.

# Анализ предметной области

На сегодняшний момент посуточная аренда жилья – популярный и доступный инструмент в организации поездок для путешествий и командировок. Существует большое количество сервисов, на которых арендатор может найти объявление и забронировать квартиру. Все сервисы можно разделить на две большие категории:

1. Сервисы с функцией календаря. Арендодатель и арендатор имеют личный кабинет. Арендодатель управляет настройками календаря: стоимостью суток, минимальным сроком проживания, доступностью дат для бронирования, системой скидок. Арендатор просматривает календарь интересующей его квартиры, подбирает удобный период проживания по стоимости и доступности, добавляет заявку на бронирование или отменяет его. Данные сервисы берут комиссию за свою работу в диапазоне 15-25% стоимости бронирования. В качестве примеров можно привести следующие платформы sutochno.ru, avito.ru, twil.ru.
2. Сервисы без функции календаря. Арендодатель имеет личный кабинет, в котором он выставляет объявление о сдаче в аренду квартиры, её описание и стоимость проживания. Арендатор просматривает доску с объявлениями и связывается с арендодателем заинтересовавшей его квартиры через чат или по номеру телефона. Арендодатель консультирует о свободных датах, актуальных ценах и скидках. Преимуществом данных сервисов является отсутствие комиссий. Примерами являются cian.ru, Яндекс Недвижимость (realty.ya.ru).

Преимуществом в удобстве сдачи и поиска жилья обладают сервисы первой группы, но для расширения аудитории и экономии средств за счет отсутствия комиссий арендодатели также активно используют и сервисы второй группы. Также в России с 2022 года нет доминирующего сервиса посуточной аренды, поэтому арендодателям приходится размещаться на множестве площадок.

Из вышеизложенного вытекает наличие следующих проблем:

1. Арендодатель должен следить за синхронизацией календарей на всех сайтах, ошибки могут привести к овербукингу;
2. Арендодатель должен вручную вводить изменения в календарь, если бронирование пришло с сервиса из второй группы, что увеличивает возможность возникновения ошибок и овербукинга;
3. Арендодатель тратит много времени для консультации арендаторов, обращающихся через сервисы из второй группы;
4. У арендодателя нет единой базы для анализа статистики, приходится вести ручной учет (Excel и т.д.) или прибегать к платным сервисам;
5. У арендатора нет доступа к календарю, если он нашел объявление на сервисе второй группы;
6. В календарях сервисов первой группы цены указаны с учетом комиссий, поэтому они не могут использоваться для консультаций арендаторов с сервисов второй группы.

Для определения функциональных требований в данной предметной области выделим следующих актёров:

* Анонимный пользователь;
* Арендодатель;
* Арендатор.

Функциональные требования сформированы на базе пользовательских историй [1], приведенных ниже, и отображены на Use Case диаграмме (рисунок Рисунок 1.1)

Пользовательские истории

Как анонимный пользователь хочу создавать профиль в системе.

Как арендодатель хочу редактировать профиль.

Как арендодатель хочу добавлять объект(квартиру) в систему с указанием названия, адреса, системы скидок, ресурсов, на которых выставлено объявление, и комментария.

Как арендодатель хочу изменять данные объекта.

Как арендодатель хочу изменять удалять объект.

Как арендодатель хочу просматривать список всех своих объектов.

Как арендодатель хочу настраивать календарь для каждого объекта с указанием базовой цены проживания за сутки, минимального количества ночей проживания.

Как арендодатель хочу назначать доступные и недоступные дни для бронирования в календаре для каждого объекта.

Как арендодатель хочу просматривать календарь по месяцам для каждого объекта со всеми примененными настройками.

Как арендодатель хочу иметь возможность передачи ссылки арендатору для просмотра доступных дней, цен и скидок конкретного объекта.

Как арендодатель хочу иметь возможность передачи информации в формате .ical о забронированных периодах на сторонние сайты для синхронизации.

Как арендодатель хочу иметь возможность проверки доступности указанных дат, общей стоимости и подходящих скидок по периоду проживания.

Как арендодатель хочу бронировать доступные даты

Как арендодатель хочу указывать при бронировании имя и телефон арендатора и ресурса, через который обратился арендатор.

Как арендодатель хочу назначать произвольную скидку или стоимость проживания во время бронирования.

Как арендодатель хочу видеть весь список бронирования для конкретного объекта.

Как арендодатель хочу сортировать список бронирования для конкретного объекта по дате бронирования, дате заезда или статусу бронирования.

Как арендодатель хочу редактировать данные в бронировании.

Как арендодатель хочу менять статус бронирования.

Как арендодатель хочу удалять бронирования.

Как арендодатель хочу просматривать годовую статистику по каждому объекту с выводом дохода, средней стоимости суток, загрузки, количества бронирований по месяцам, а также процентное соотношение по ресурсам, на которых выставлено объявление.

Как арендатор хочу иметь доступ к календарю конкретной квартиры по переданной арендодателем ссылке с отображением доступности дат, ценам и минимальному периоду проживания.

Как арендатор хочу иметь возможность проверки доступности указанных дат и подходящих скидок по периоду проживания и общей стоимости.

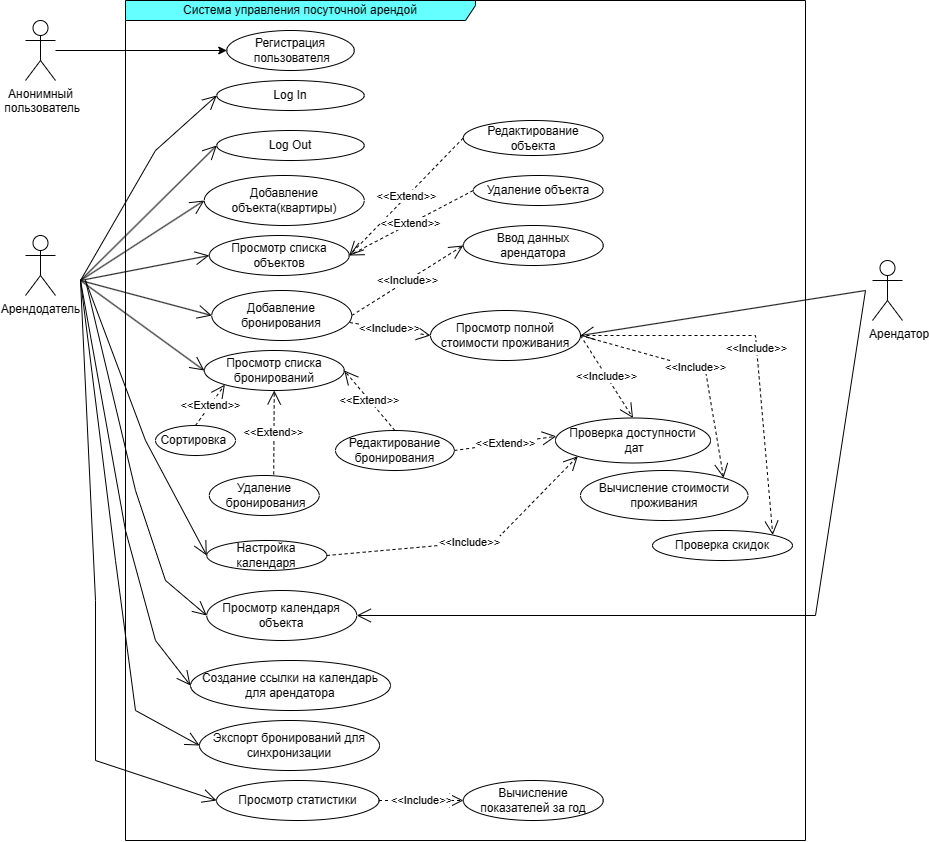


Рисунок . – Диаграмма Use Case

Основным аналогом разрабатываемой системы является платный сервис RealtyCalendar: *new.realtycalendar.ru* [2]. Данный сервис имеет ряд модулей:

1. Channel Manager - система управления рекламными каналами, которая синхронизирует работу всех площадок продаж:

* одновременное размещение объекта на всех подключенных площадках;
* единый центр управления заявками со всех каналов;
* настройка гибких тарифных планов — скидок, спеццен — на всех площадках одновременно;
* безопасное хранение данных в единой системе;
* исключение овербукинга.

1. Модуль бронирования: программного код, который добавляется на собственный ресурс или в соцсетях.
2. Модуль статистики позволяет анализировать:

* с каких площадок чаще всего приходят гости;
* из каких они регионов;
* какие квартиры пользуются популярностью, а какие простаивают;
* какие объекты приносят самую высокую прибыль;
* и др.

Из анализа функционала RealtyCalendar видно, что он решает большинство проблем.

Однако, в описании не упоминается возможность передачи доступа к календарю арендаторам. Минимальная стоимость подписки составляет 1100 р в месяц за объект.

# Этап проектирования

Для проектирования системы и описания логической структуры данных построена ER диаграмма (рисунок

Рисунок 2.1), отражающая результаты анализа предметной области проекта [3-4]. Выделены следующие сущности: Пользователь, Арендодатель (относящийся к Пользователю), Арендатор, Квартира и Ресурс (бронирования).

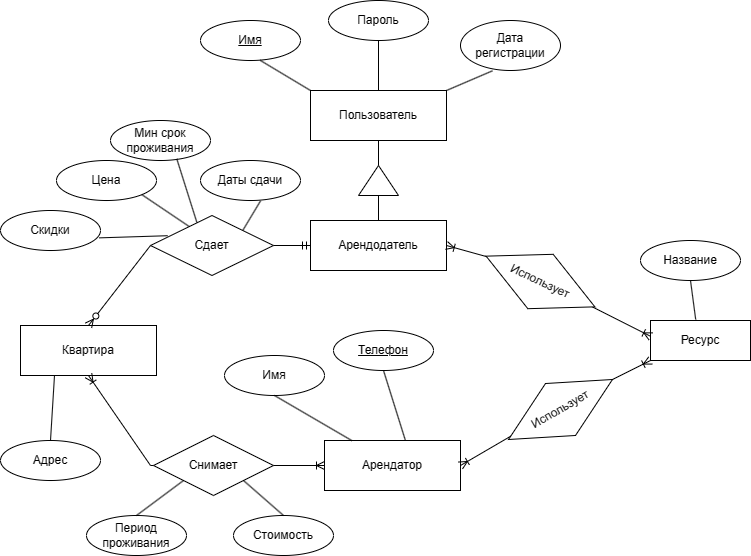


Рисунок . – ER диаграмма

На диаграмме активности (рисунок Рисунок 2.2) приведен процесс бронирования квартиры арендодателем.

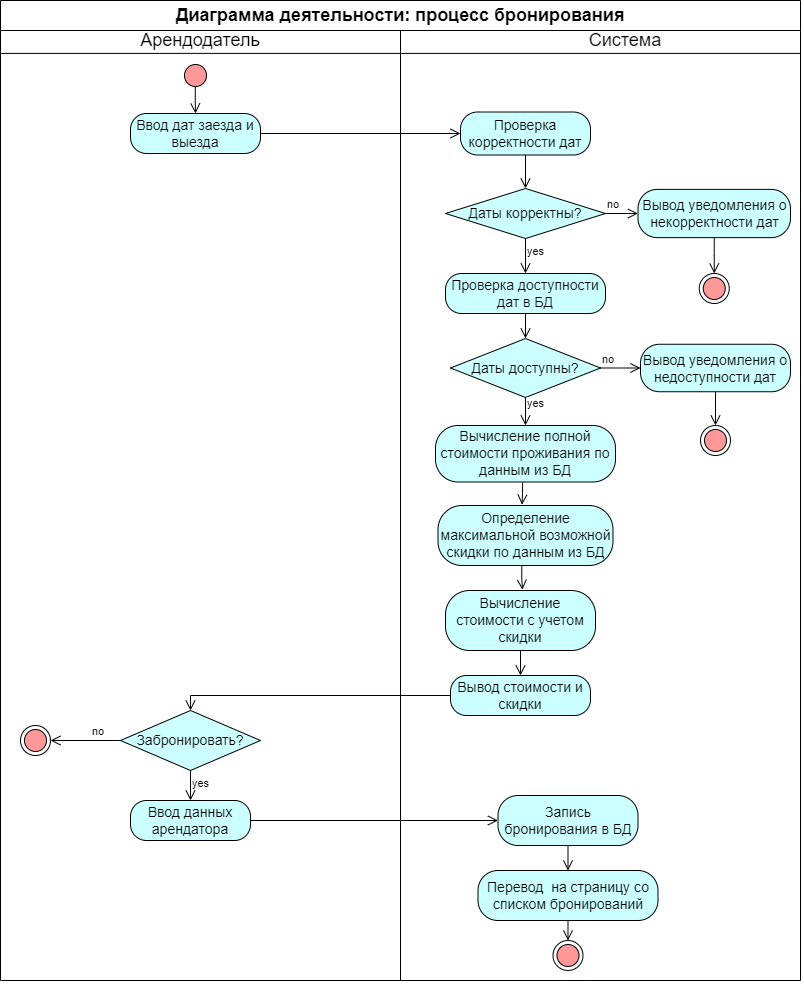


Рисунок . – Диаграмма активности

# Описание системы

## Средства разработки и структура проекта

Для арендодателей и арендаторов важна возможность доступа к системе с различных устройств, поэтому в качестве типа разрабатываемого приложения выбрано веб-приложение, т.к. оно позволяет получать доступ к функционалу сервиса только с использованием браузера и наличия интернет-соединения, нет необходимости скачивания и установки программы. Что также значительно экономит время пользователей.

Для реализации проекта выбран серверный веб-фреймворк Django 4.2.2 [5-8], позволяющий организовать маршрутизацию URL, рендеринг HTML, аутентификацию, администрированию и логику бэкенда. Фреймворк Django базируется на архитектурном паттерне Model-View-Template (MVT), являющимся модификацией распространенного в веб-программировании паттерна MVC (Model-View-Controller). Схематично архитектура MVT в Django изображена на рисунке Рисунок 3.1.

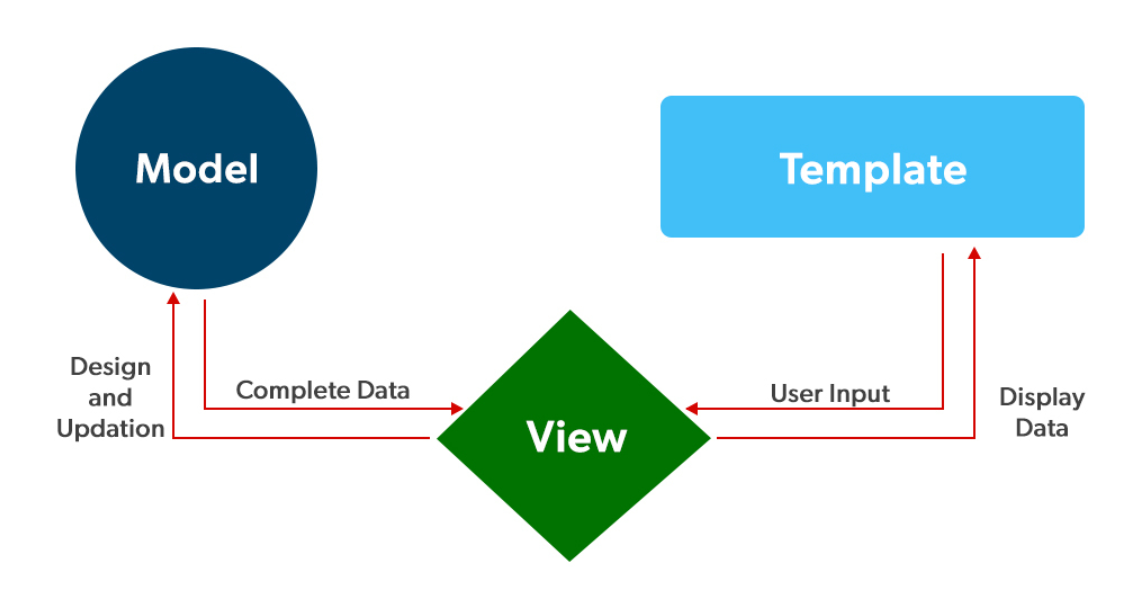


Рисунок . – Архитерктурный паттерн MVT

Основными элементы паттерна являются:

* Model: элемент доступа к данным. Как правило, отдельные классы соответствуют таблицам в базе данных;
* View: используется для выполнения бизнес-логики и взаимодействия с моделью для переноса данных и отображения шаблона:
* Template: уровень представления, полностью обрабатывает часть пользовательского интерфейса.

Когда в приложение приходит запрос, Django определяет, с каким ресурсом сопоставляется данный запрос и вызывает View, который получает запрос и определенным образом обрабатывает его. В процессе обработки View может обращаться к моделям и базе данных, получать из нее данные или сохранять. Результат обработки запроса представляется в виде сгенерированного html-кода с помощью Template. Результат обработки запроса отправляется обратно, и этот результат пользователь видит в своем браузере.

Django значительно упрощает процесс разработки веб-приложения [9], т.к. он содержит большой набор функционала:

* ORM;
* Миграции базы данных;
* Аутентификация пользователя;
* Панель администратора;
* Формы.

В Django стандартизированная структура проекта, что экономит время и ресурсы разработчика. Django включает механизмы защиты от, например, SQL-инъекций и подделки межсайтовых запросов (CSRF). Данный фреймворк имеет детализированную документацию[8] и обширное сообщество пользователей, что значительно упрощает поиск информации и средств решения возникающих при разработке проблем.

Для упрощения процесса верстки html страниц использовался открытый HTML, CSS и JS фреймворк Bootstrap [10]. С использованием данного фреймворка на сегодняшний момент верстается около 19% сайтов в мире [11]. Bootstrap включает в себя множество разных готовых компонентов для веб-сайтов: типографику, веб–формы, кнопки, блоки навигации и другое. Разработчику на нужно создавать верстку и стили для традиционных элементов веб-страницы или веб-приложения. У Bootstrap подробная документация с большим количеством примеров готового кода.

Для разработки реляционной базы данных выбрана СУБД MySQL, т.к. она поддерживает SQL, имеет удобный инструментарий для импорта и экспорта данных, известна своим быстродействием, бесплатно распространяется для некоммерческих целей, наиболее проста в установке, не требует сложной процедуры конфигурации, за счет большой популярности и распространенности существует огромное количество образовательных ресурсов, имеет детальную документацию. Также стоит отметить, что в Django присутствует встроенный интерфейс для подключения к БД MySQL.

Рассмотрим структуру созданного проекта Django *Flatrent\_website* и основные файлы.

1. Внешний каталог *FlatRent\_website* - контейнер для проекта.
2. Внутренний каталог *FlatRent\_website* – проект Django:

* *settings.py*: файл c настройками конфигурации проекта;

DATABASES – настройки подключения к базам данных

INSTALLED\_APPS – список приложений Django, используемых в проекте:

1. django.contrib.admin – интерфейс администратора;
2. django.contrib.auth – система аутентификации;
3. django.contrib.contenttypes – фреймворк типов данных;
4. django.contrib.sessions – фреймворк сессии;
5. django.contrib.messages – фреймворк сообщений;
6. django.contrib.staticfiles – работа со статическими файлами;
7. booking – разработанное приложение для бронирований;
8. user - разработанное приложение для регистрации и аутентификации пользователей;
9. widget-tweaks – приложение для стилизации форм.

STATIC\_URL - URL, указывающий на каталог со статическими файлами

* *urls.py*: конфигурация URL адресов проекта Django;
* *wsgi.py*: точка входа проекта для WSGI – совместных веб-серверов.

1. Внутренний каталог *booking* – приложение для бронирований, его функции приведены в таблице Таблица 1.

* *migrations*: каталог с файлами миграций, создающихся при изменении модели;
* *templates*: html-шаблоны приложения;
* *static*: каталог со статическими файлами, в данном приложении в нем содержатся изображения;
* *models.py*: модели проекта, каждый класс (рисунокРисунок 3.2) представляет таблицу в базе данных. На рисунке Рисунок 3.3. показана структура разработанной базы данных в MySQL (за исключением всех автоматически создаваемых таблиц Django, кроме auth\_user);
* *forms.py*: файл с формами. Форма для добавления/редактирования объекта, форма для добавления/редактирования бронирования, форма для добавления арендатора, форма для добавления дней в календаре при регистрации объекта, форма для проверки доступности, цен и скидок указанного периода;
* *admin.py*: настройка использования моделей в административной части проекта;
* *views.py:* файл который отвечает за отображение общедоступной части, за логику приложения. Отображение является функцией, принимающей и обрабатывающей http-запрос, при необходимости обращается к базе данных, после чего генерирует страницу html и возвращает её пользователю в виде http-ответа.
* *urls.py* – файл, отвечающий за связь с файлом urls.py проекта. Функция url() определяет URL-паттерн и соответствующую функцию во View.
* *apps.py* – файл который определяет конфигурацию приложения.

1. Внутренний каталог *users* – приложение для регистрации и аутентификации пользователей, функции приведены в таблице Таблица 2:

* *templates*: html-шаблоны приложения;
* *views.py:* отвечает за отображение, за логику приложения.
* *urls.py* – файл, отвечающий за связь с файлом urls.py проекта. Функция url() определяет URL-паттерн и соответствующую функцию во View.
* *apps.py* – файл который определяет конфигурацию приложения.

1. *manage.py –* основной скрипт взаимодействия с проектом.

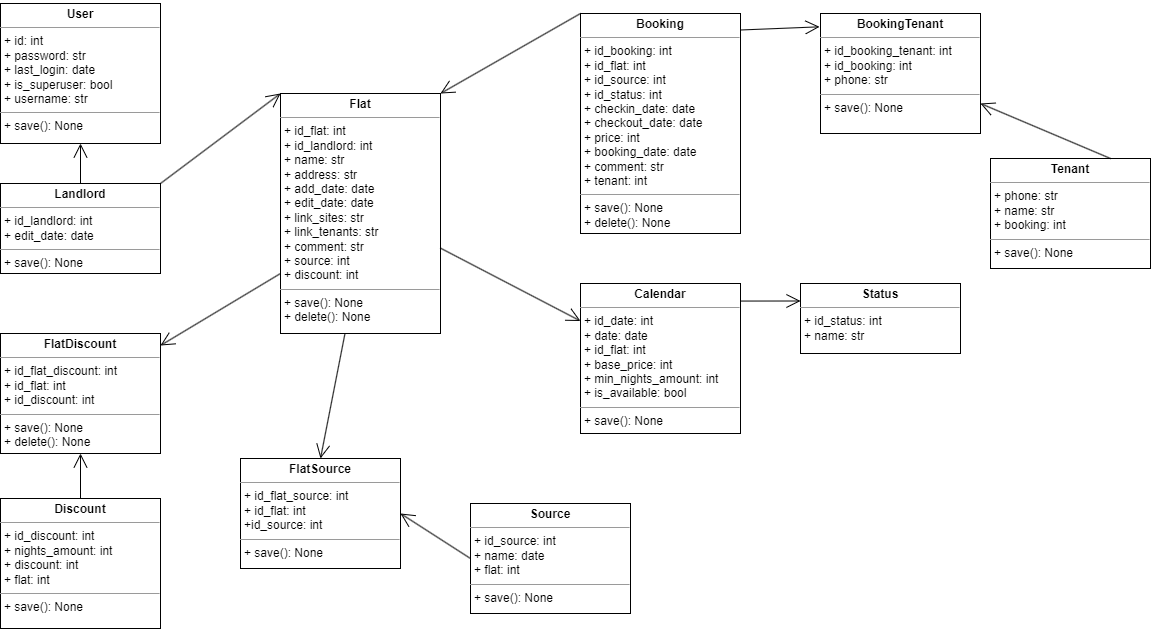


Рисунок . – Диаграмма классов

## Структура базы данных

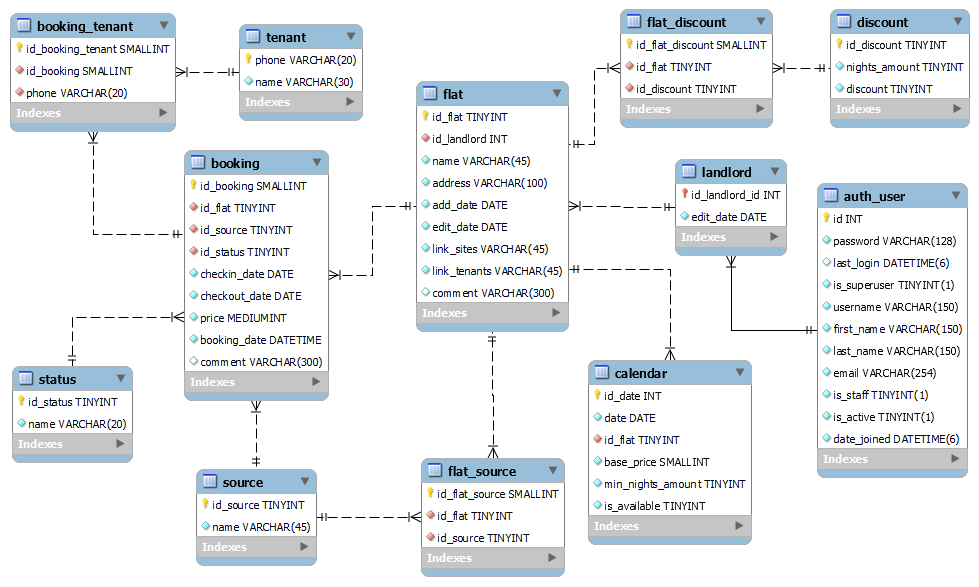


Рисунок . – Структура базы данных MySQL

В рамках базы данных запущено два события для регулярной обработки и изменения данных в таблице, SQL команды по созданию событий приводятся на рисункеРисунок 3.4. Одно из событий каждый день проверяет соответствие статуса бронирования текущей дате и изменяет статус с “ожидается” на “в процессе” или с “в процессе” на “завершено”. Второе событие каждый день для каждой квартиры добавляет 1 день в таблицу “календарь”.

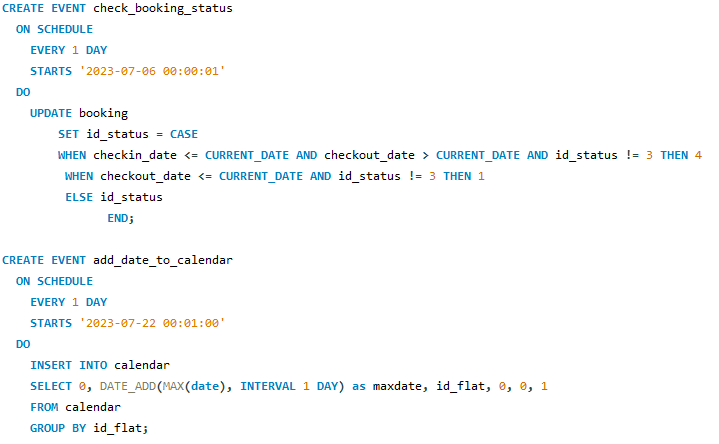
**

Рисунок . – Создание событий в базе данных MySQL

## Реализованные функции системы

Таблица 1 – Функции приложения booking

|  |  |
| --- | --- |
| 1 | Добавление объекта |
| 1.1 | Проверка отсутствия объекта в базе с таким именем |
| 1.2 | Проверка корректности введенной системы скидок |
| 1.3 | Проверка указанной скидки в базе данных |
| 1.4 | Добавление скидки в базу данных, если отсутствует |
| 1.5 | Проверка наличия источников в базе данных |
| 1.6 | Добавление источника в базу данных, если отсутствуют |
| 1.7 | Формирование ссылки на календарь объекта с ограничением доступа для арендаторов |
| 1.8 | Экспорт данных в формат .ical для синхронизации со сторонними сайтами |
| 1.9 | Добавление объекта в базу данных |
| 2 | Редактирование объекта |
| 2.1 | Проверка корректности введенной системы скидок |
| 2.2 | Проверка указанной скидки в базе данных |
| 2.3 | Добавление скидки в базу данных, если отсутствует |
| 2.4 | Проверка наличия источников в базе данных |
| 2.5 | Добавление источника в базу данных, если отсутствуют |
| 2.6 | Изменение параметров объекта в базе данных |
| 3 | Вывод списка всех объектов |
| 4 | Удаление объекта |
| 5 | Отображение календаря |
| 5.1 | Генерация календаря для заданного месяца и года |
| 5.2 | Получение статуса дня, цены и скидки из базы данных |
| 6 | Настройка календаря |
| 6.1 | Проверка доступности дат |
| 6.2 | Внесение параметров в базу данных, если даты доступны |
| 7 | Бронирование |
| 7.1 | Проверка корректности дат |
| 7.2 | Проверка доступности дат |
| 7.3 | Вычисление стоимости проживания |
| 7.4 | Поиск подходящих скидок |
| 7.5 | Внесение данных арендатора |
| 7.5.1 | Проверка длины номера телефона |
| 7.6 | Указание с какого сайта обратился арендатор |
| 7.7 | Редактирование цены бронирования/скидки |
| 7.7.1 | Пересчет скидки/цены |
| 7.8 | Проверка наличия арендатора в базе |
| 7.9 | Добавление арендатора в базу, если отсутствует |
| 7.10 | Добавление бронирования в базу |
| 8 | Просмотр списка бронирований |
| 8.1 | Сортировка списка по выбранному параметру: дата бронирования, статус бронирования, дата заезда |
| 9 | Редактирование бронирования |
| 9.1 | Изменение дат бронирования |
| 9.1.1 | Проверка корректности введенных дат |
| 9.1.2 | Проверка доступности дат |
| 9.2 | Изменение имени арендатора |
| 9.3 | Изменение телефона арендатора |
| 9.3.1 | Поиск арендатора с указанным телефоном в базе |
| 9.3.2 | Внесение арендатора в базу, если отсутствует |
| 9.4 | Изменение статуса бронирования |
| 9.4.1 | Проверка возможности изменения статуса |
| 10 | Удаление бронирования |
| 11 | Просмотр статистики за год |
| 11.2 | Поиск бронирований за выбранный год |
| 11.3 | Вычисление дохода |
| 11.4 | Вычисление средней цены за сутки |
| 11.5 | Вычисление среднего срока проживания |
| 11.6 | Вычисление загруженности |
| 11.7 | Вычисление процентного соотношения по ресурсам |
| 11.8 | Генерация диаграмм |
| 12 | Редактирование профиля |

Таблица 2 – Функции приложения user

|  |  |
| --- | --- |
| Приложение user | |
| 1 | Регистрация пользователя |
| 1.1 | Добавление арендодателя в базу |
| 2 | Аутентификация пользователя |
| 3 | Выход из профиля |

## Архитектура приложения

На рисункеРисунок 3.5 изображена архитектура проекта. С помощью сервиса pythonanywhere развернуто приложение и запущена база данных. С помощью HTTP запросов происходит обмен информацией между сервером pythonanywhere и браузером клиента.

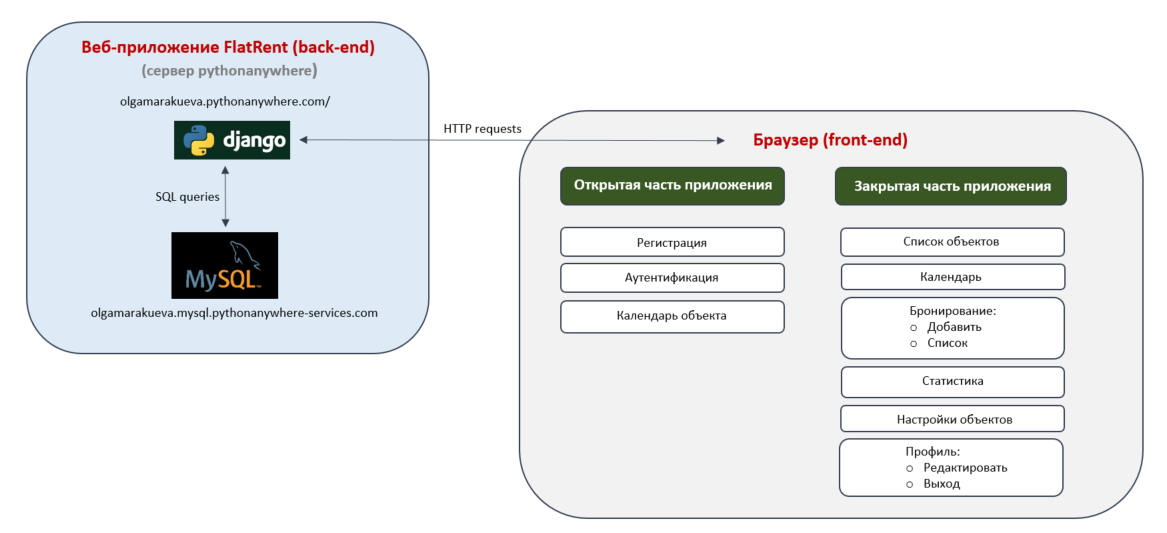


Рисунок . – Архитектура приложения

## Тестирование

Для проверки функционала веб-приложения, определены следующие задачи, которые должны успешно выполняться:

* зайти в приложение;
* создать нового пользователя;
* создать несколько новых объектов;
* выполнить редактирование объекта по всем полям формы;
* удалить объект;
* в календаре изменить месяц отображения;
* в календаре в списке изменить выбранный объект;
* в календаре применить операцию редактирования дней по всем полям формы;
* перейти в бронирование и ввести некорректные даты;
* в бронировании ввести свободные даты и перейти по кнопке «Забронировать»
* в бронировании ввести данные арендатора;
* в бронировании изменить цену;
* в бронировании изменить скидку;
* сохранить бронирование;
* перейти в бронирование и ввести даты предыдущего бронирования;
* перейти в список бронирований;
* отредактировать бронирование по всем полям формы;
* перейти в статистику, изменить год;
* вернуться в список бронирований и удалить бронирование;
* перейти в редактирование профиля и изменить имя;
* зайти в список объектов и перейти по ссылке для синхронизации с другими сайтами.
* зайти в список объектов и перейти по ссылке для арендаторов в другом браузере (для того, чтобы пользователь в используемом браузере не был аутентифицирован);
* изменить месяц в календаре по ссылке для арендаторов;
* проверить доступность дат в календаре по ссылке для арендаторов.

Результаты тестирования

* успешный вход в приложение, отображается страница входа и регистрации;
* новый пользователь создается успешно;
* новые объекты создаются успешно;
* редактирование объекта по всем полям формы выполняется успешно, появляется сообщение при попытке сохранения объекта с именем уже имеющегося в базе объекта;
* удаление объекта выполнено успешно;
* календарь отображается корректно при изменении месяца, происходит переключение года при изменении месяца с Декабря на Январь и наоборот;
* в календаре успешно происходит переключение с одного объекта на другой;
* в календаре изменение параметров дней выполняется корректно, дни закрываются и открываются;
* при переходе в бронирование и вводе некорректных дат выдается ошибка;
* в бронировании при вводе свободных дат выводится полная стоимость проживания, стоимость с учетом скидки, величина скидки. Появляется кнопка «Забронировать», при нажатии на неё выводится дополнительная форма для заполнения деталей бронирования;
* при бронировании данные арендатора вводятся успешно, возникает ошибка при вводе короткого телефона (до 5 цифр);
* цена в бронировании меняется успешно, пересчитывается скидка;
* скидка в бронировании меняется успешно, цена пересчитывается;
* бронирование сохраняется успешно;
* при попытке бронирования с датами предыдущего бронирования появляется уведомление о недоступности дат;
* список бронирований отображается корректно;
* редактирование бронирования осуществляется успешно, возникает ошибка при вводе некорректных или занятых дат, статус бронирования успешно меняется;
* статистика отображается корректно при изменении года и объекта;
* бронирование успешно удаляется;
* изменение имени в профиле выполняется корректно;
* календарь успешно отображается при переходе по ссылке для арендаторов;
* изменение месяца в календаре для арендаторов проводится успешно;
* проверка дат на доступность выполняется корректно, появляется сообщение при вводе неправильного срока;
* при переходе по ссылке для синхронизации с другими сайтами успешно сохраняется файл .ical.

Проведенное тестирование свидетельствует об успешной реализации веб-приложения.

## Руководство пользователя

**Руководство для арендатора**

При переходе по адресу приложения пользователь попадает на страницу аутентификации/регистрации. Для начала работы с приложением новому пользователю необходимо зарегистрироваться.

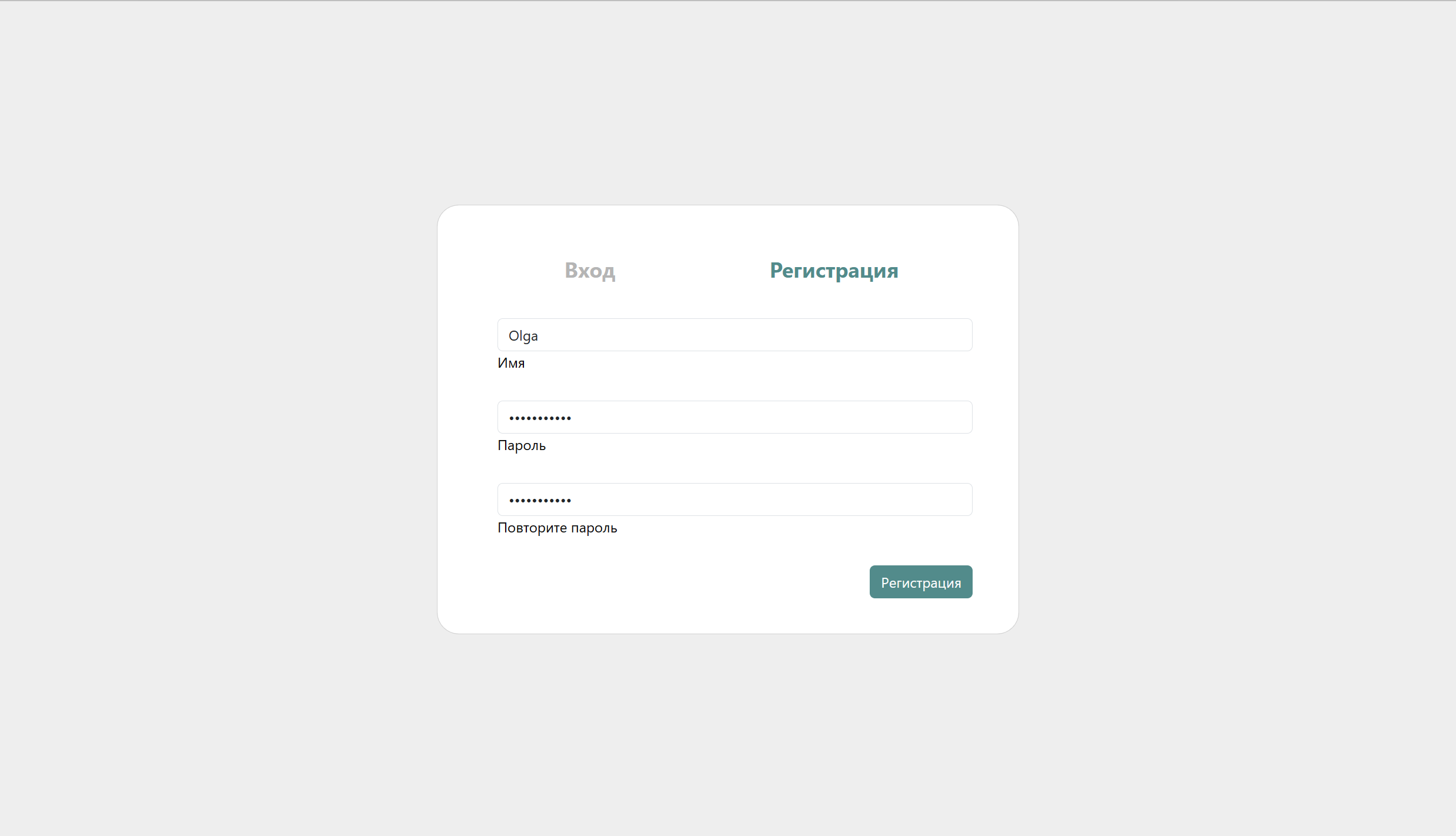
 

Рисунок . – Окна регистрации и аутентификации стартовой страницы

После удачной регистрации и входа пользователь попадает на домашнюю страницу. Для начала работы с календарем и бронированием необходимо добавить объекты. Перейдите во вкладку *Настройки объектов.*

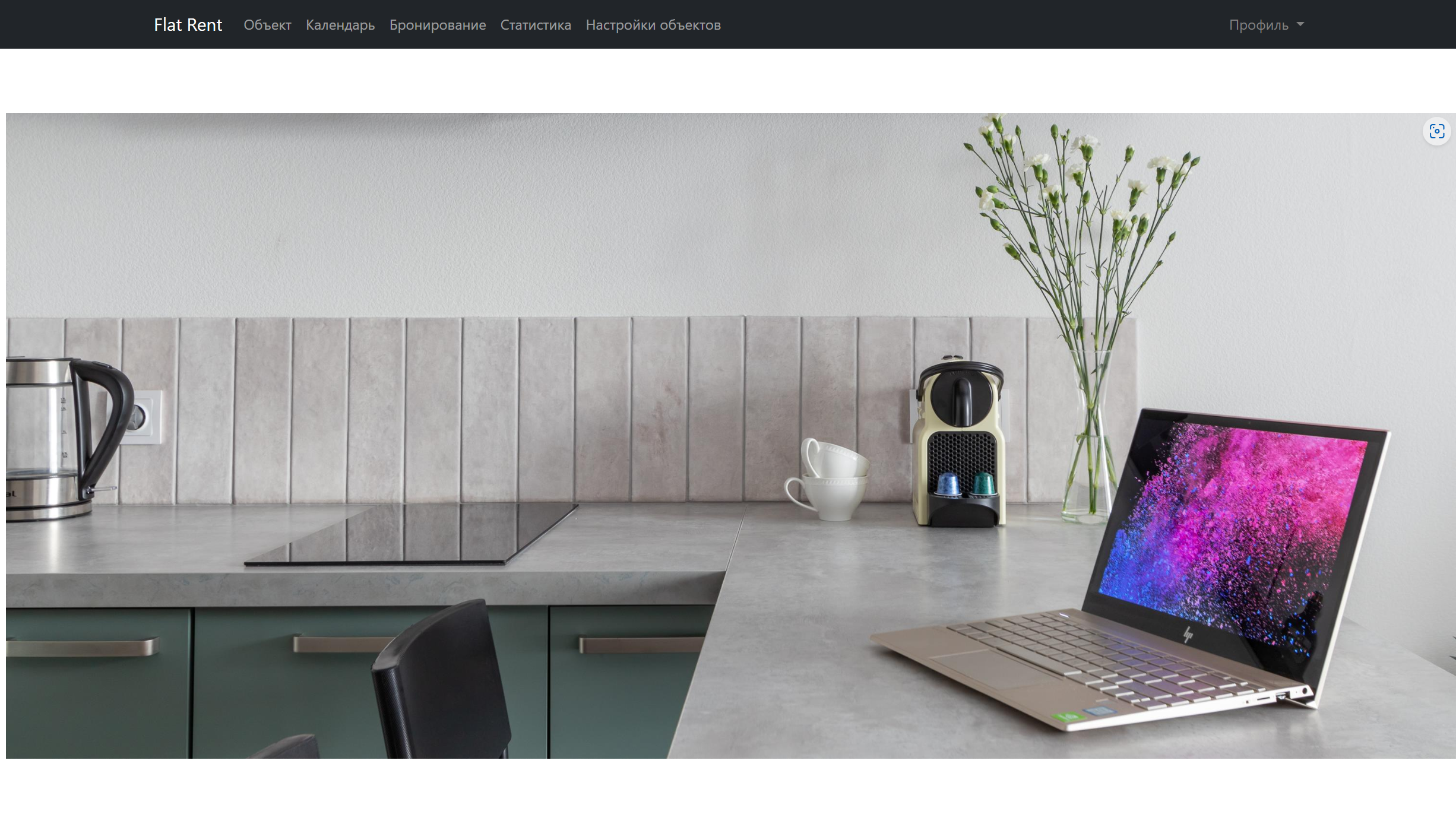


Рисунок . – Домашняя страница приложения

На странице настроек объектов нажмите *Добавить.*

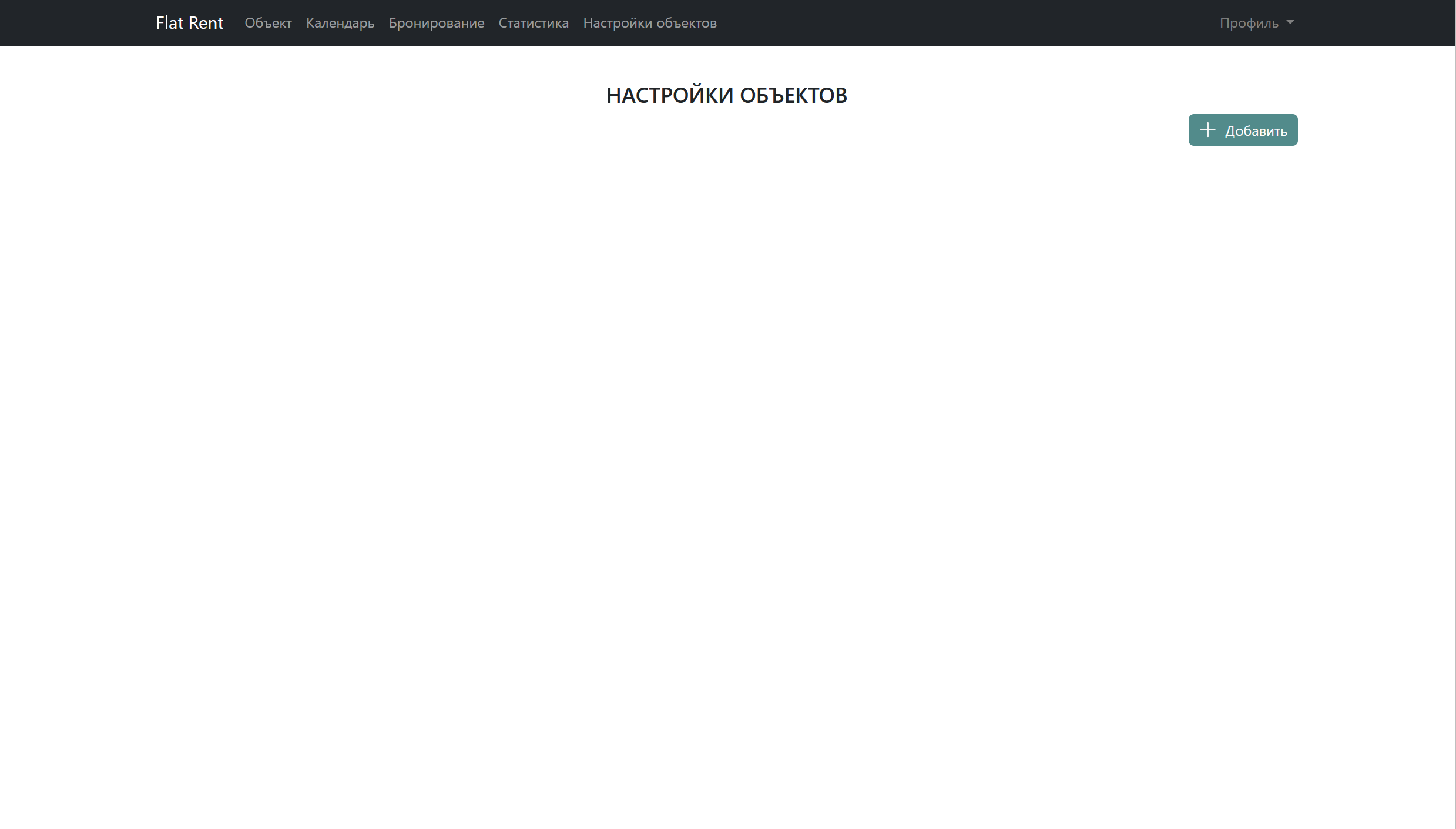


Рисунок . – Страница настройки и добавления объектов

Заполните открывшуюся форму, обязательные поля отмечены \*. Каждая скидка должна быть уникальна по количеству ночей и величине скидке. Для сохранения нажмите *Сохранить*

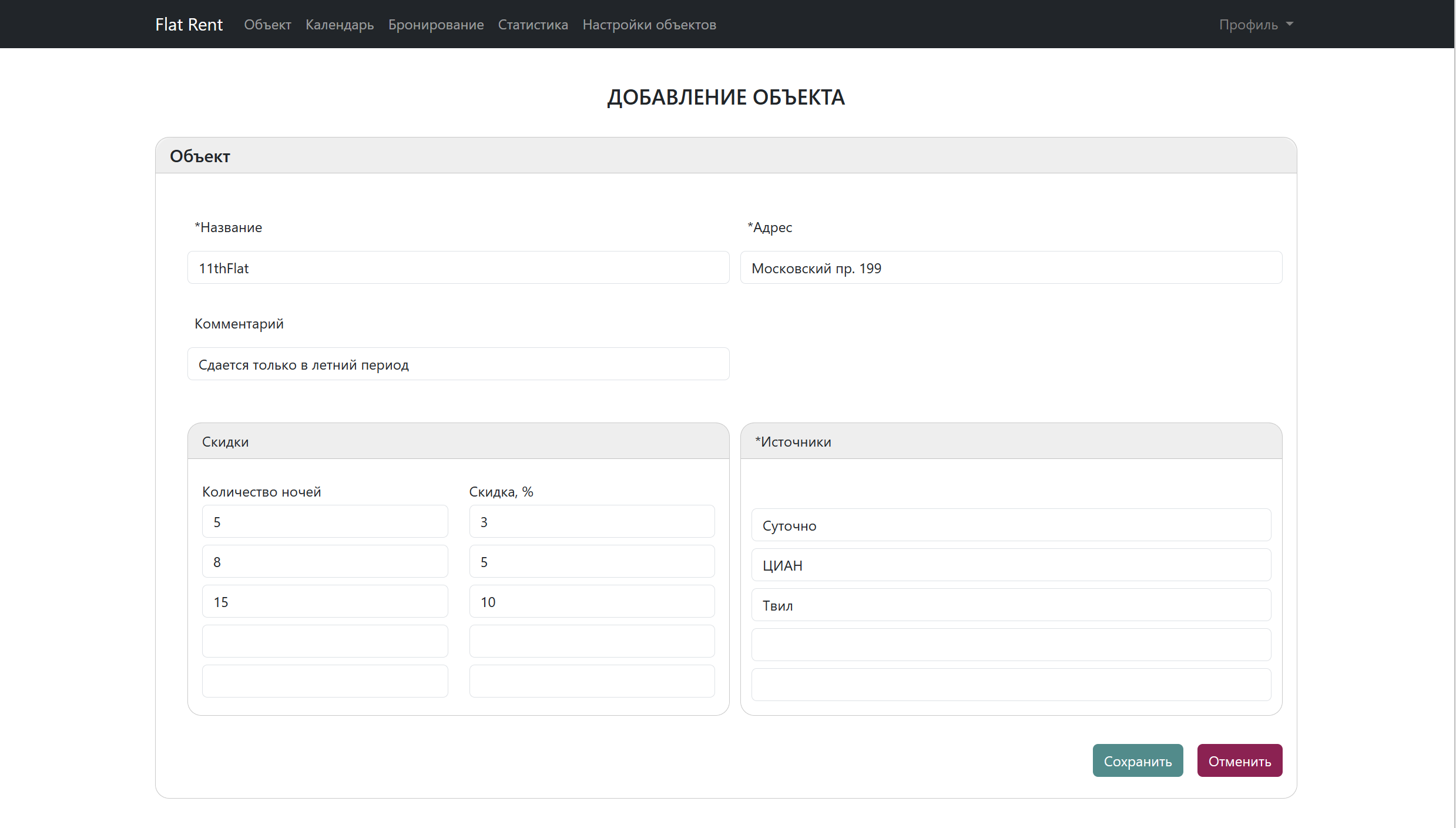


Рисунок . – Страница настройки и добавления объектов

После сохранения на странице отображаются карточки объектов. На скриншоте снизу добавлен дополнительный объект. В каждой карточке объекта, помимо, ранее введенной информации содержится две ссылки: *ссылка для сайтов* – которую необходимо указать на стороннем сайте, поддерживающем формат .ical, для синхронизации, *ссылка для арендаторов* – передается арендатору для самостоятельной проверки календаря и подбора удобных дат проживания по сроку и цене. Все ссылки уникальны.

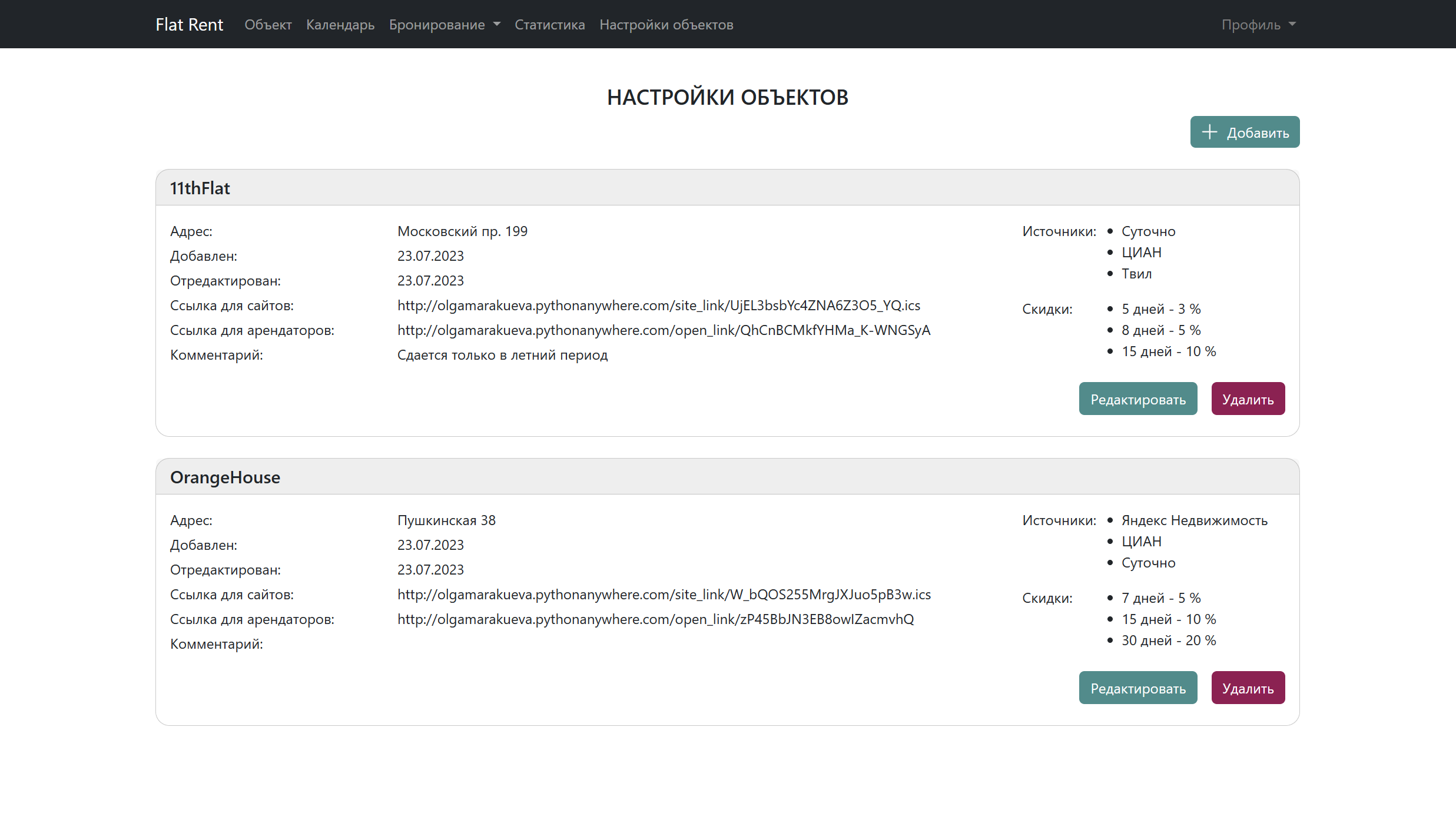


Рисунок . – Страница настройки объектов после добавления нескольких объектов

При необходимости изменить данные объекта нажмите на *Редактировать*. Внесите изменения, выберите *Сохранить* для подтверждения или *Отменить* для выхода без сохранения.

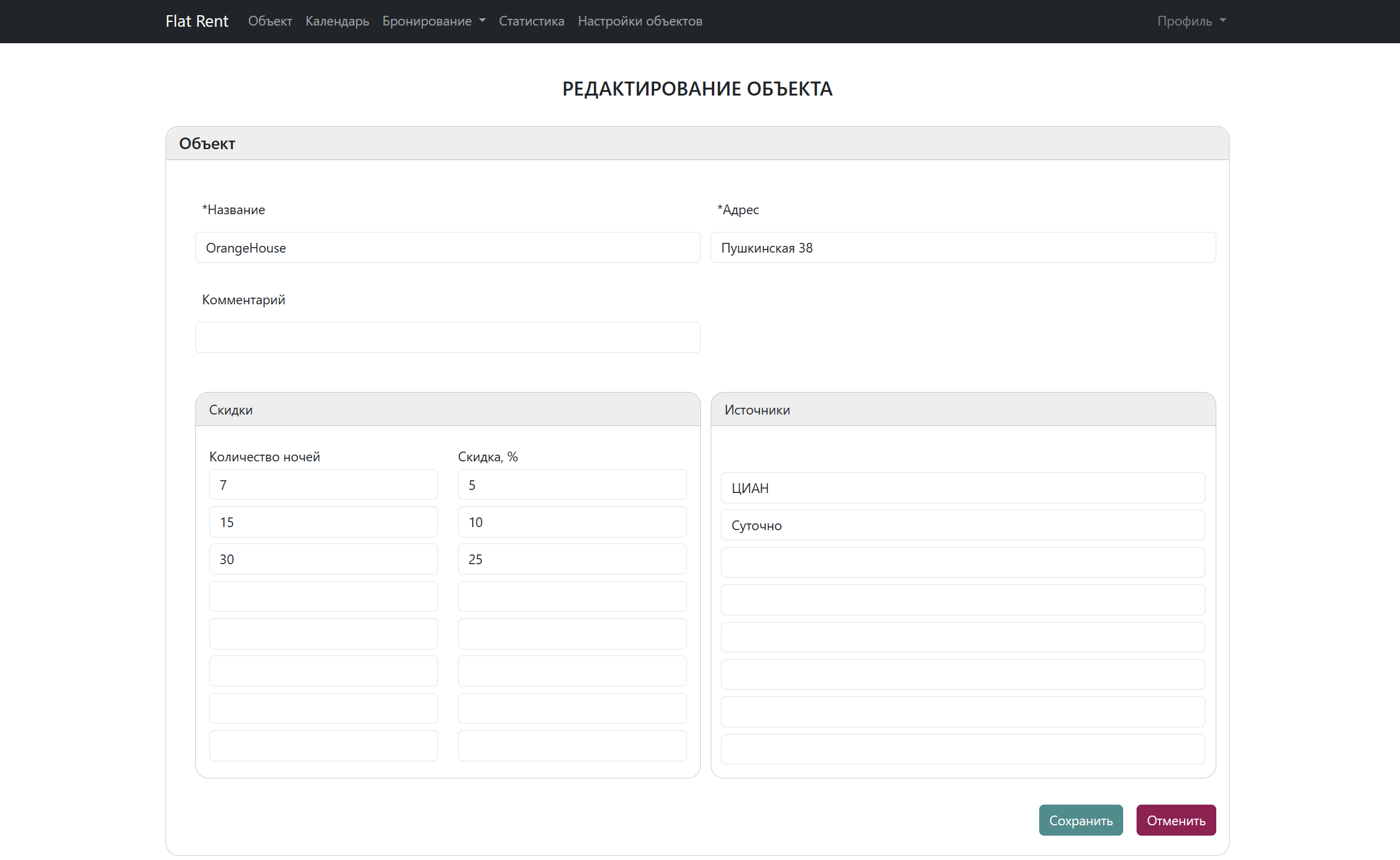


Рисунок . – Страница редактирования объекта

После добавления объектов можно переходить к работе с календарем, для настройки параметров календаря перейдите в главном меню в *Календарь.* Откроется вкладка с отображением календаря и формы для ввода настроек.Вглавном меню находится окно, в котором можно выбирать, для какого объекта проводятся настройки.

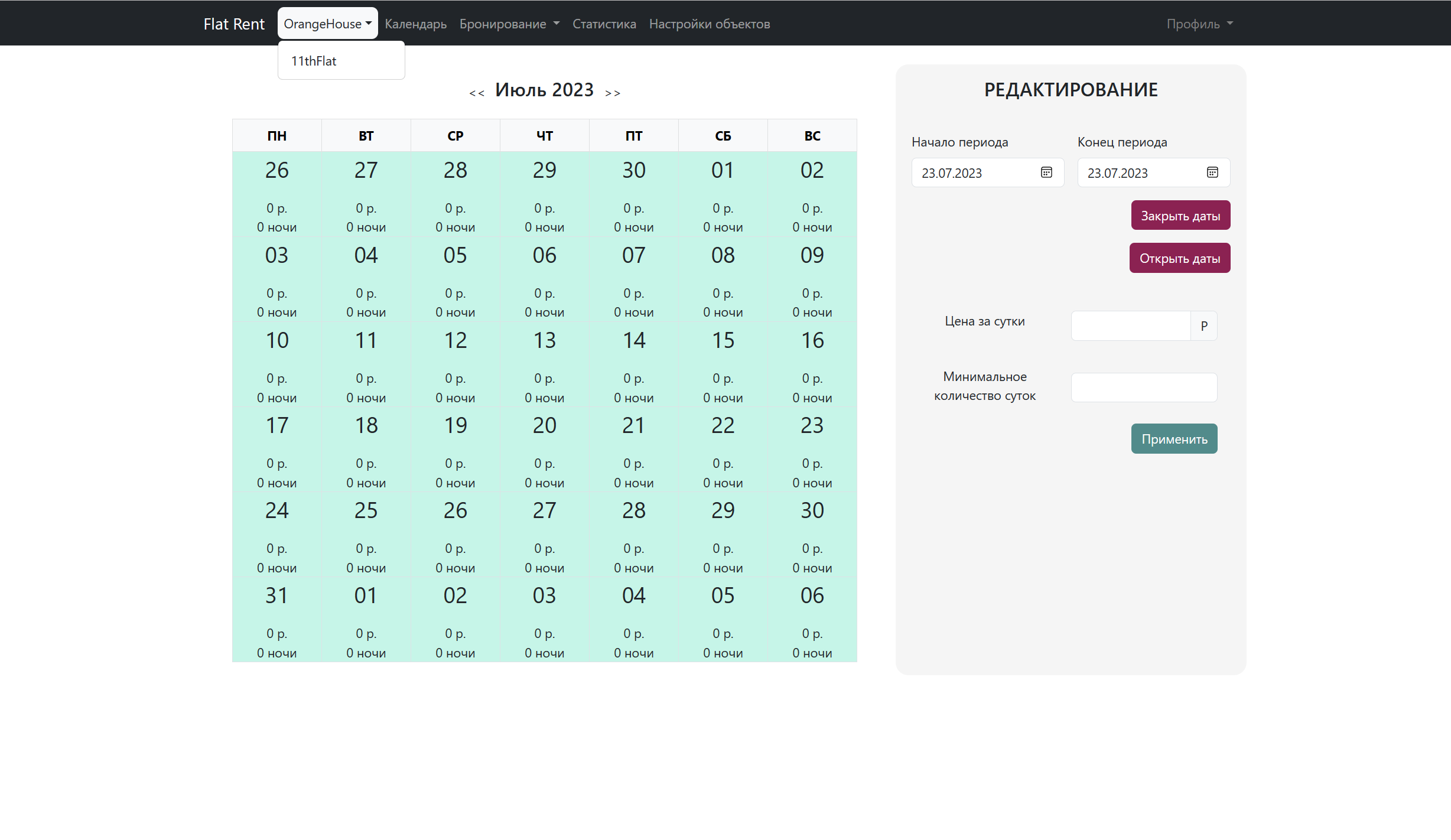


Рисунок . – Страница календаря

После выбора объекта введите период, для которого будут проводиться настройки. Заполните форму с параметрами: цена за сутки и минимальное количество суток, оба поля не являются обязательными. Нажмите *Применить* для сохранения настроек, после чего настройки отобразятся на календаре.



Рисунок . – Страница календаря после применения указанных настроек

Функция *Закрыть даты* позволяет сделать указанный период недоступным для бронирования, после применения даты отображаются на календаре серым цветом.

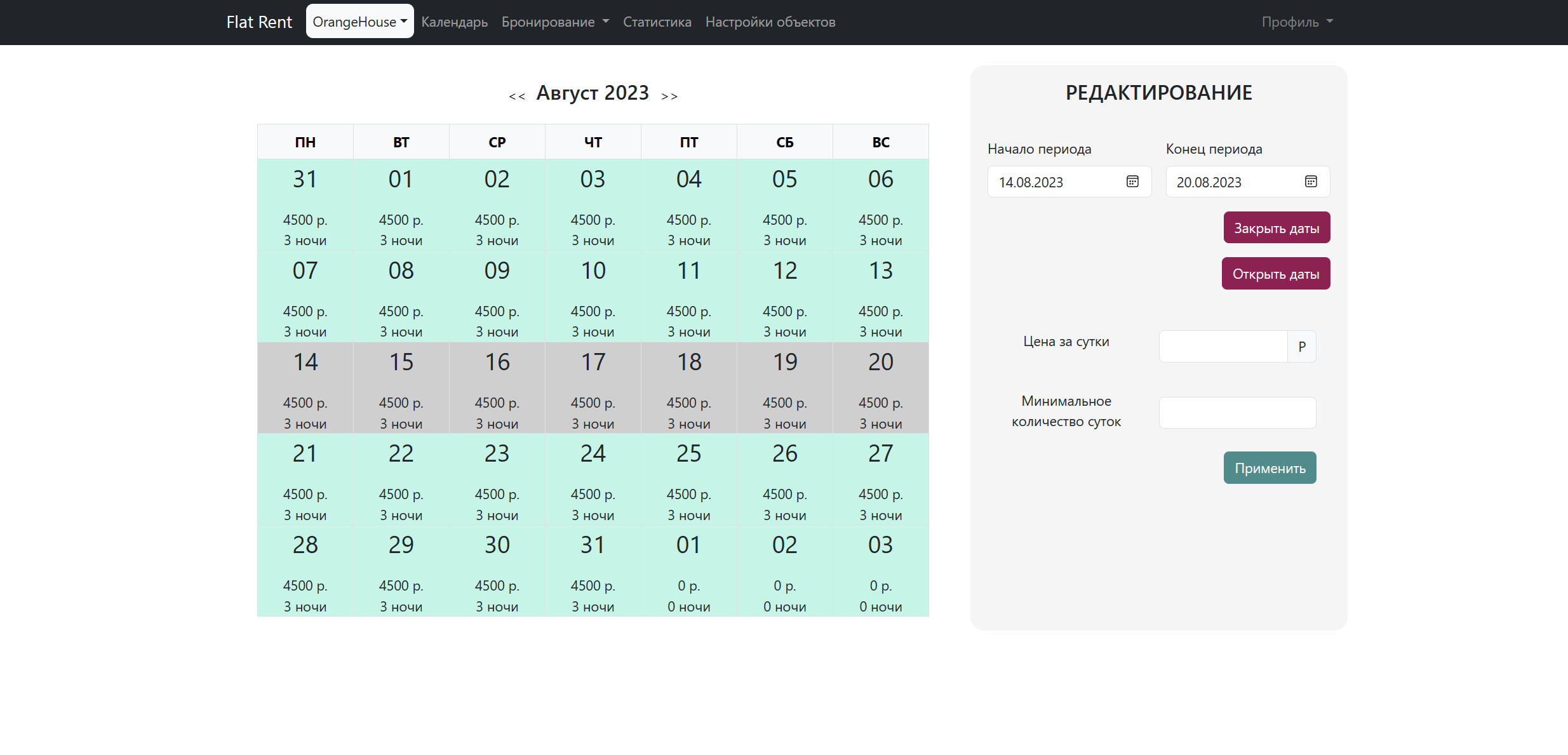


Рисунок . – Страница календаря после закрытия периода

Функция *Открыть даты* позволяет вернуть доступ при бронировании к указанному периоду, после применения даты вновь отображаются на календаре зеленым цветом.

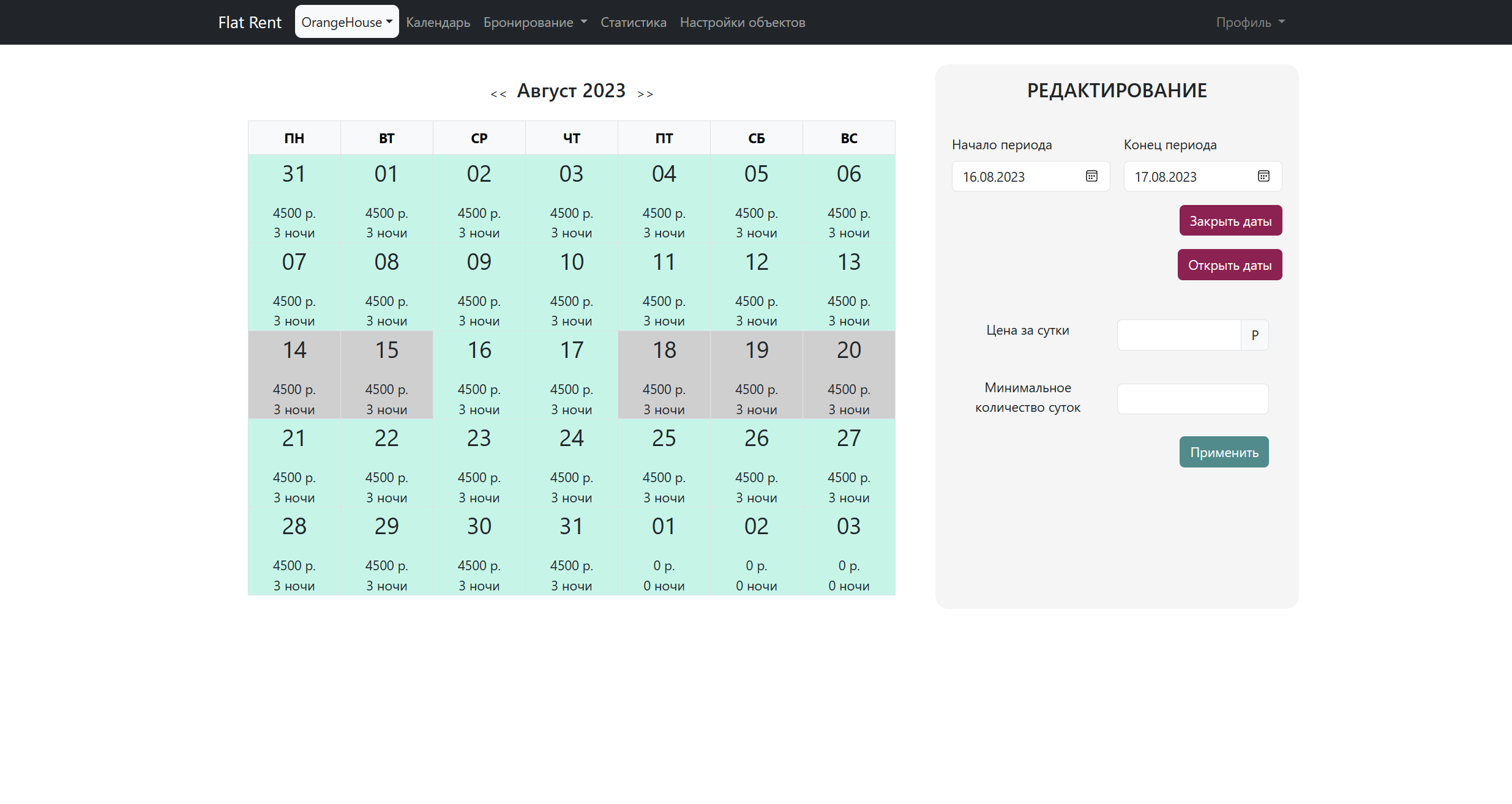


Рисунок . – Страница календаря после открытия периода

Для возможности проверки доступности дат и добавления бронирования перейдите через главное меню в *Бронирование 🡪 Добавить.* На странице бронирований также доступно переключение между объектами в главном меню.

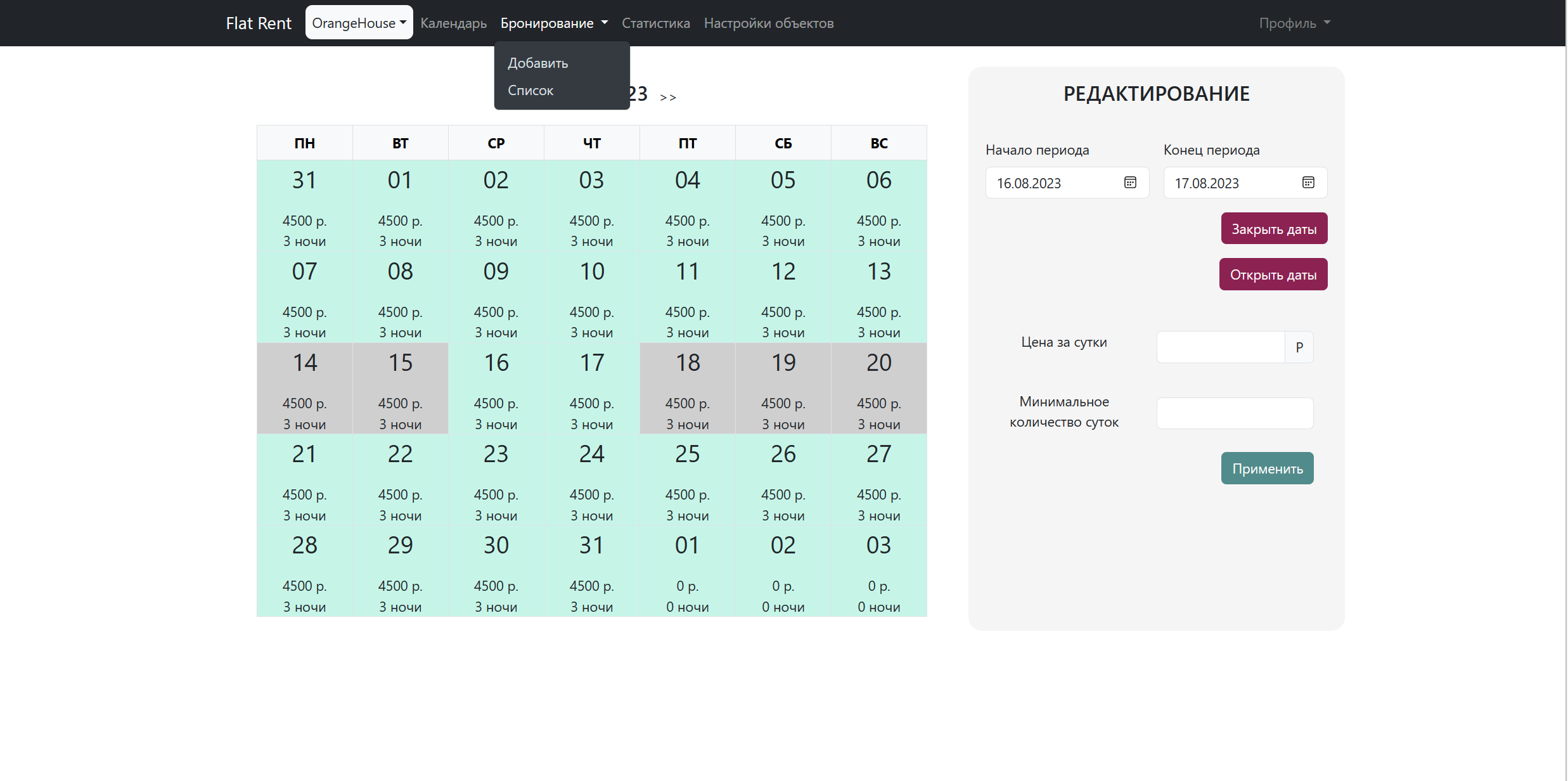


Рисунок . – Переход в окно бронирований

Введите дату заселения и выезда и нажмите *Поиск.* Если даты недоступны или введены некорректно, появится сообщение. При доступности дат появляется информация по стоимости проживания и наличия подходящей скидки. Для бронирования нажмите *Забронировать* или продолжите поиск удобных дат проживания.

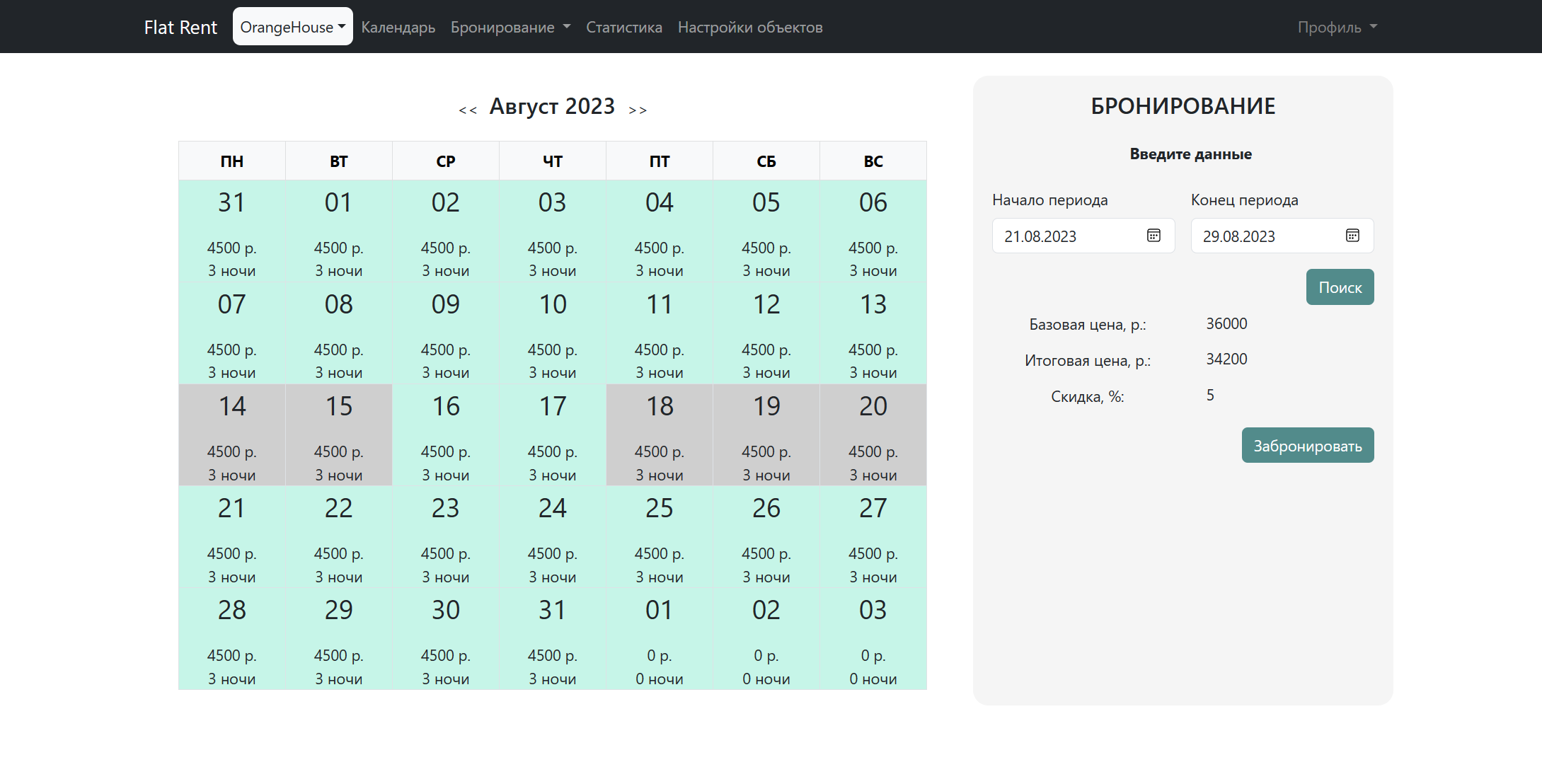


Рисунок . – Страница проверки доступности дат

На странице бронирования заполните данные арендатора и укажите источник, откуда арендатор узнал об объявлении. При необходимости можно изменить цену или скидку, выбрав соответствующий пункт, далее ввести величину выбранного параметра и нажать *Пересчитать цену*. Изменить даты можно через *Новый поиск.* Длясохранения бронирования нажмите *Забронировать.*

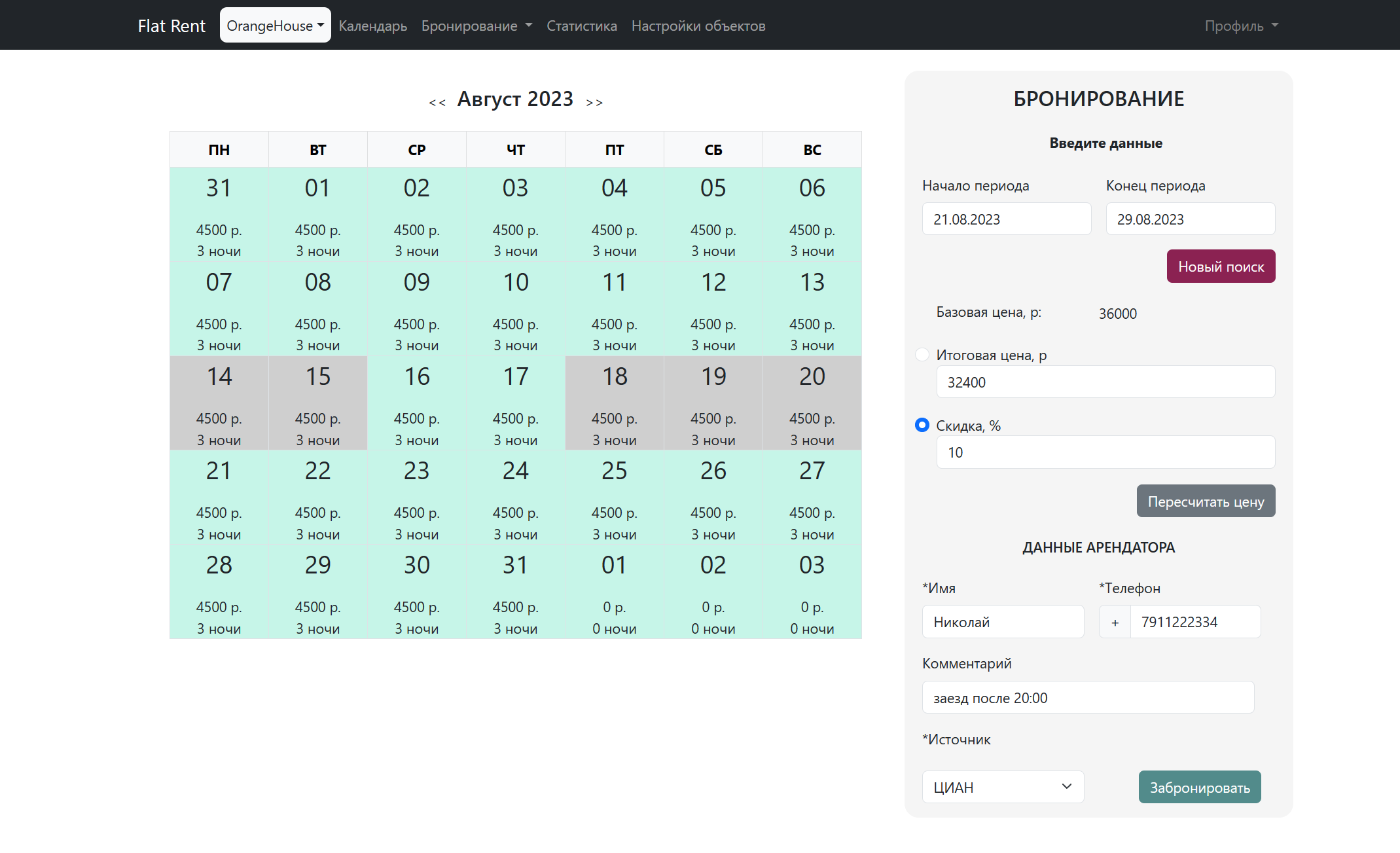


Рисунок . – Заполнение данных бронирования

Сохраненное бронирование отобразится в списке бронирований, а забронированный период будет отмечен красным цветом в календаре, день выезда отмечается всегда доступным.

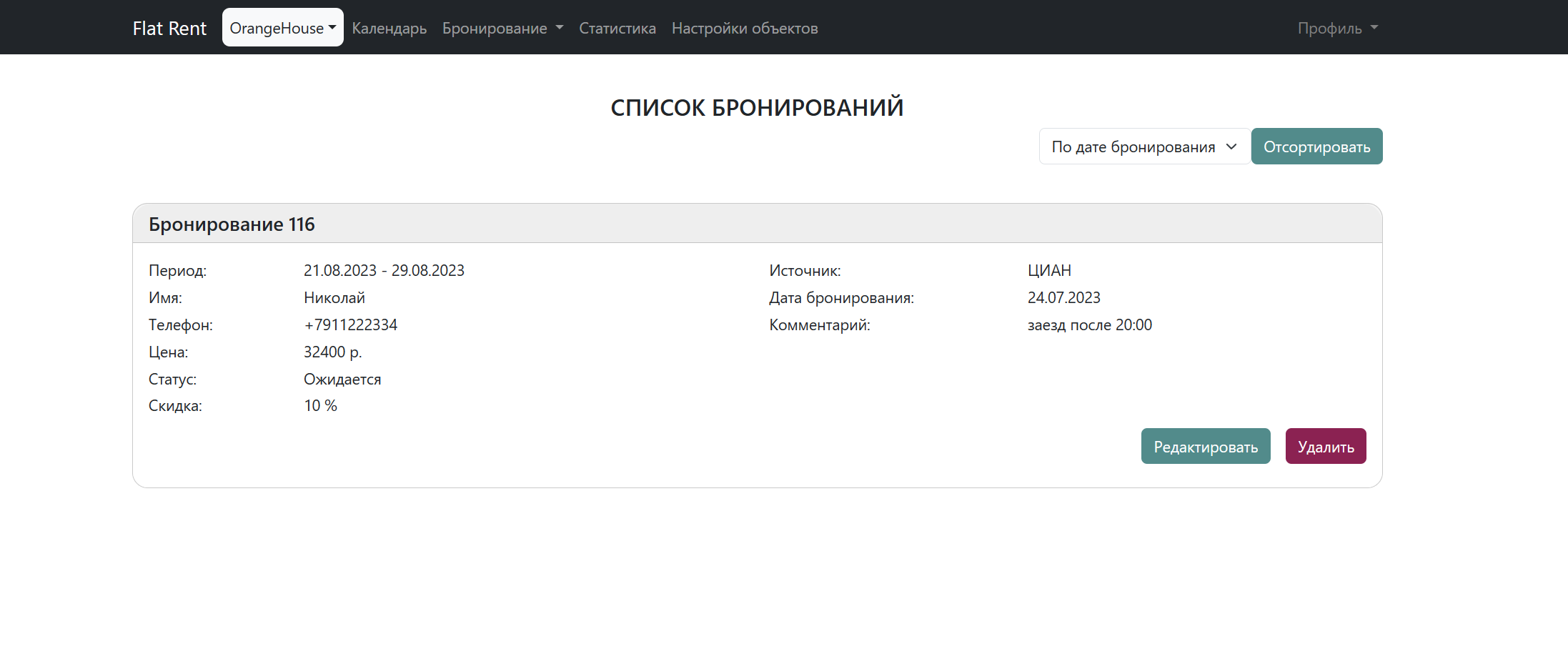


Рисунок . – Список бронирований

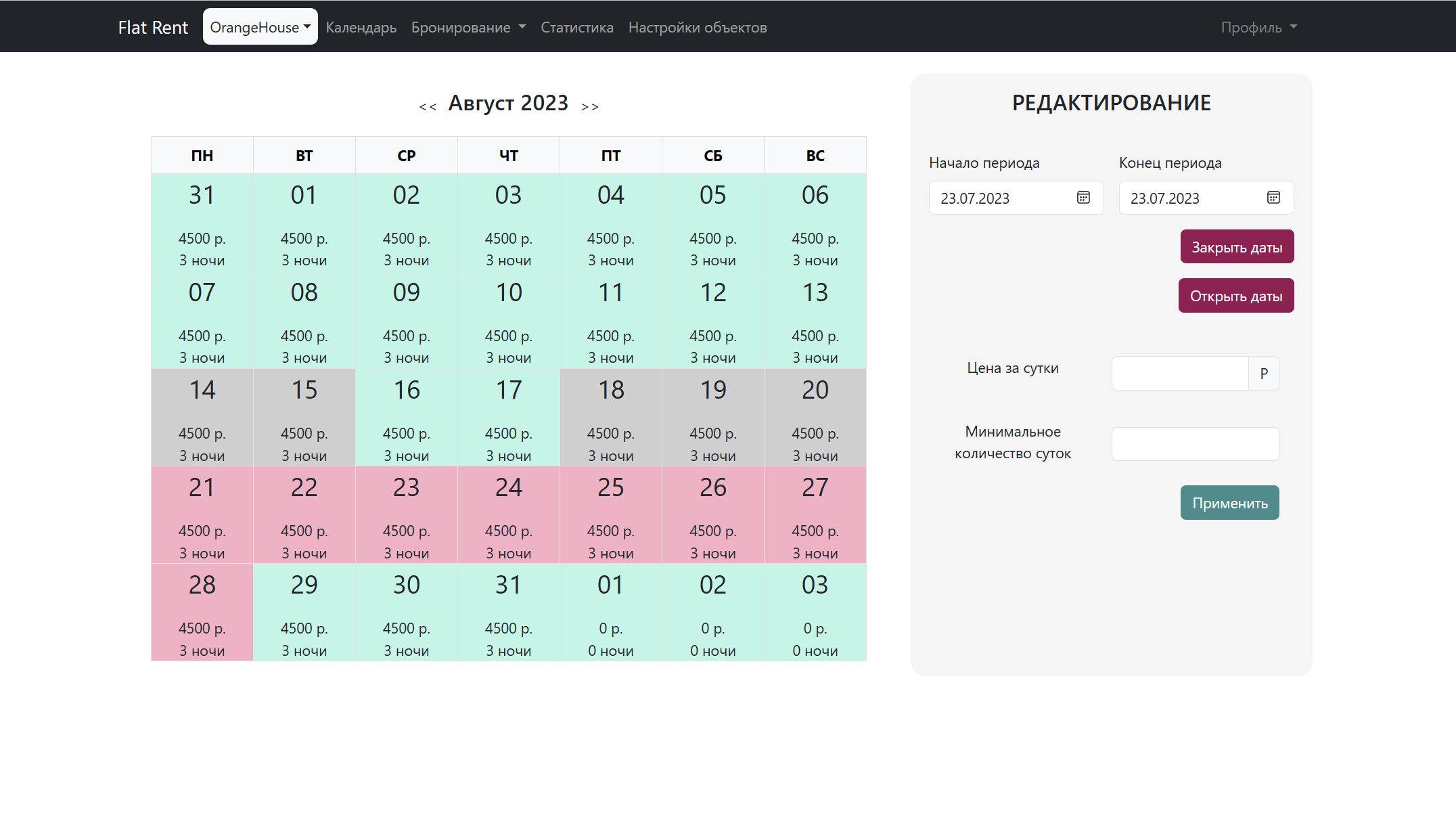


Рисунок . – Календарь с отображением закрытого и забронированного периодов

Вернуться в список бронирования можно через главное меню *Бронирование 🡪 Список*. При наличии нескольких бронирований в списке, можно проводить сортировку, выбрав соответствующую категорию и нажав *Отсортировать.*

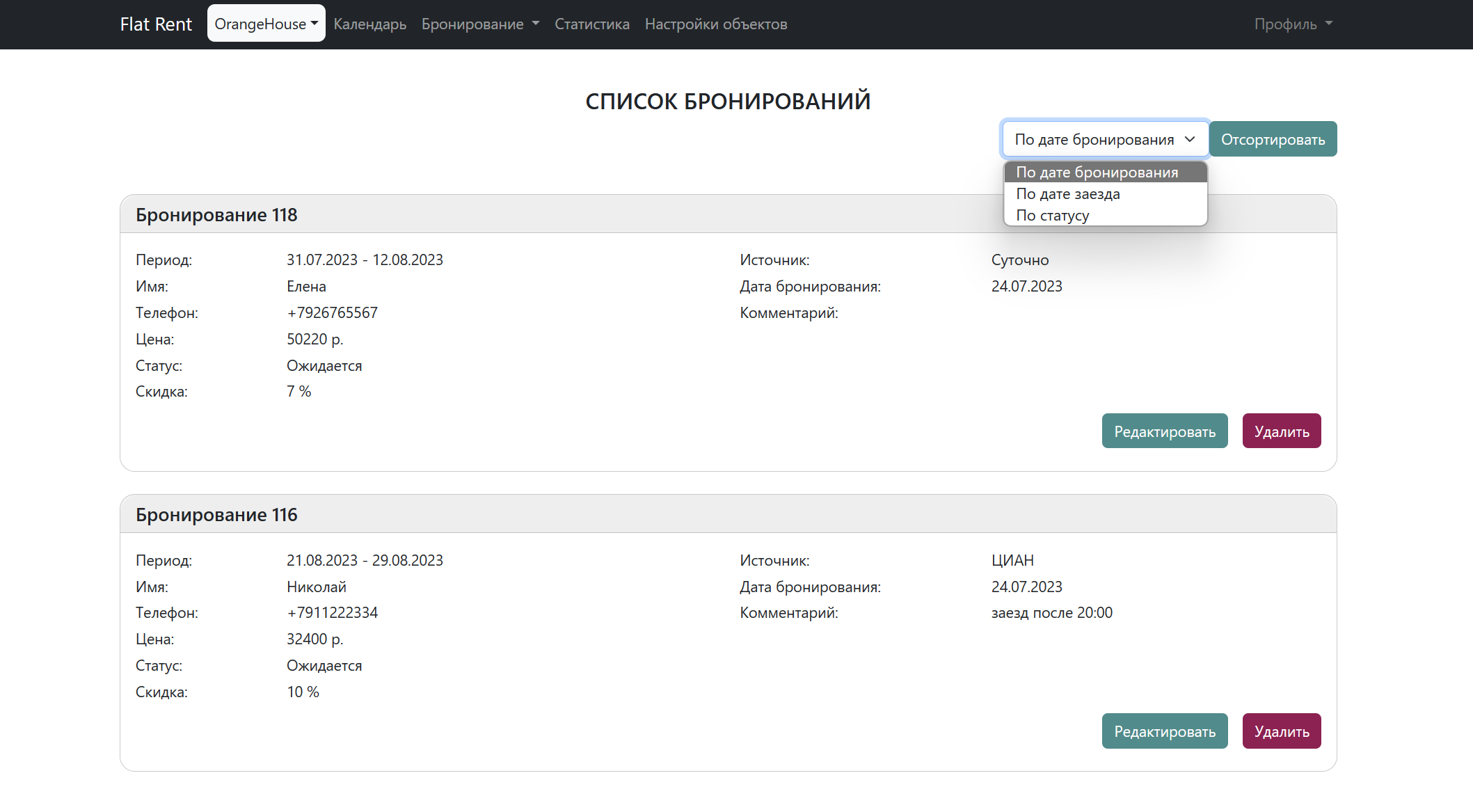


Рисунок . – Список с несколькими бронированиями

Для редактирования бронирования нажмите *Редактировать,* после чего откроется форма с редактируемыми полями. При изменении дат проживания проводится проверка их корректности и доступности. Есть возможность изменения статуса бронирования. После отмены бронирование не будет отображаться на календаре. Внесите необходимые изменения и нажмите *Сохранить.* Удаление бронирования возможно из списка бронирований или через карточку редактирования.

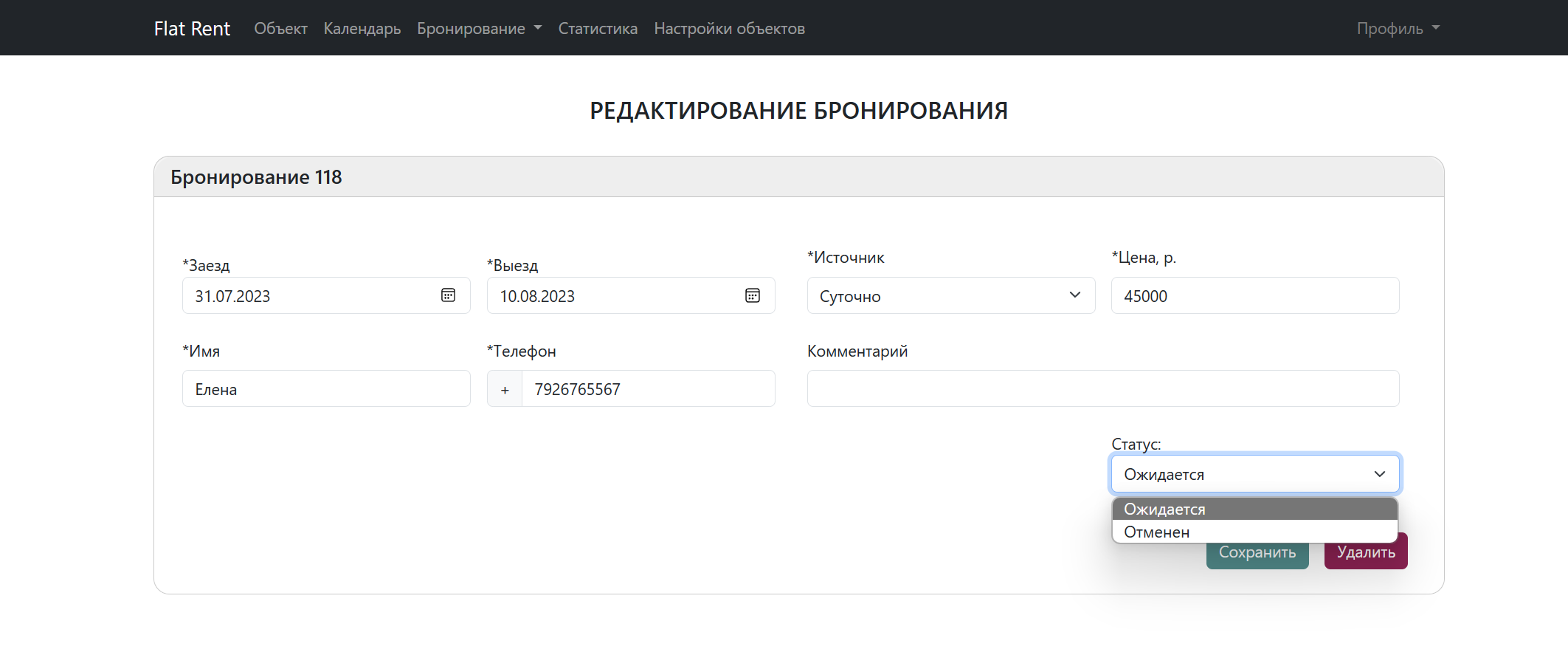


Рисунок . – Редактирование бронирования

Для анализа статистики перейдите через главное меню в *Статистика*. В данном окне доступен выбор объекта и выбор года, за который будет выведена статистика. Зеленым цветом отобразятся прошедшие месяца, красным – текущие или будущие.

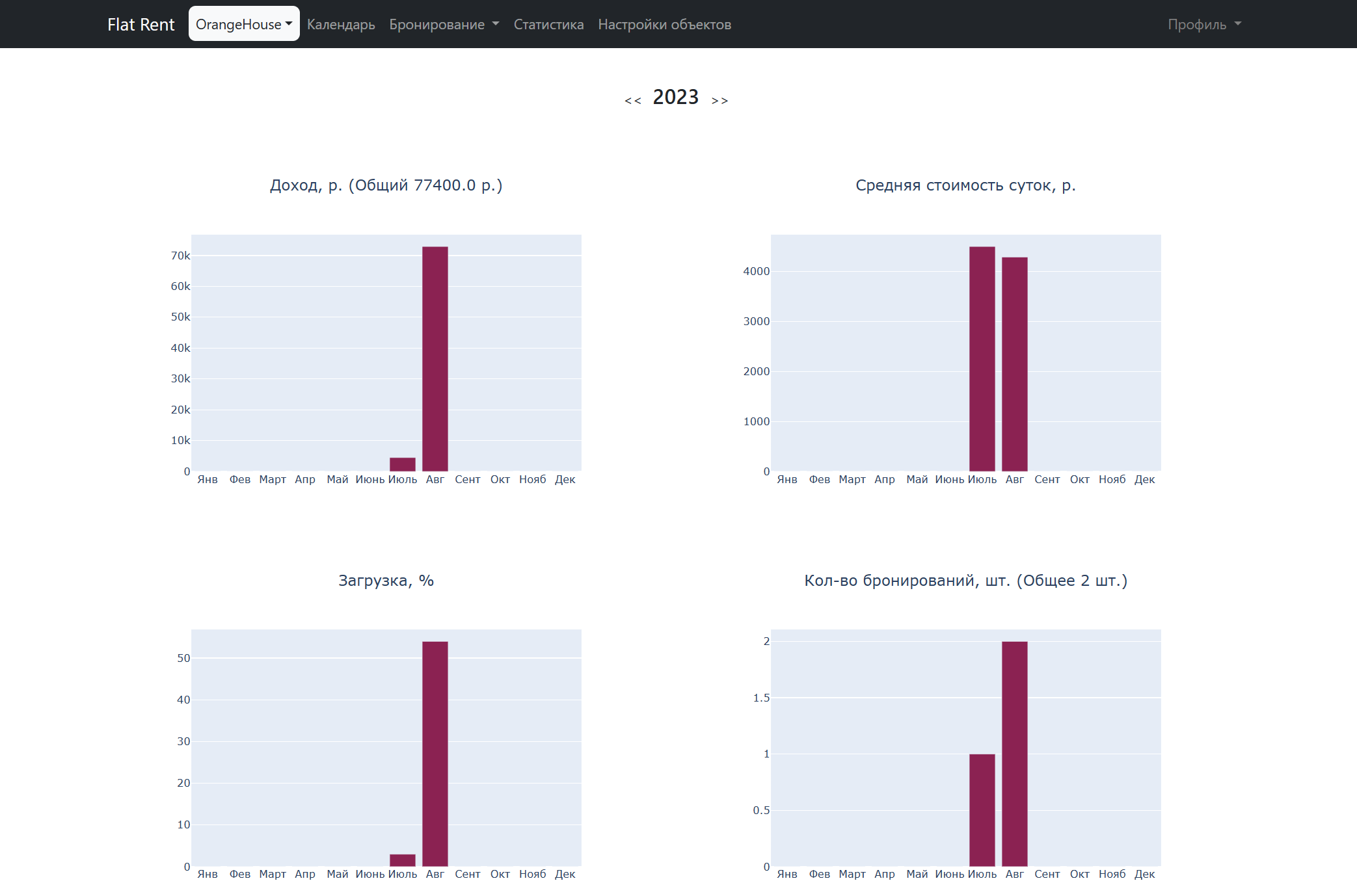


Рисунок . – Страница статистики

Для редактирования профиля перейдите в главном меню в *Профиль🡪 Редактировать.* Измените имя или пароль пользователя при необходимостии нажмите *Сохранить.*

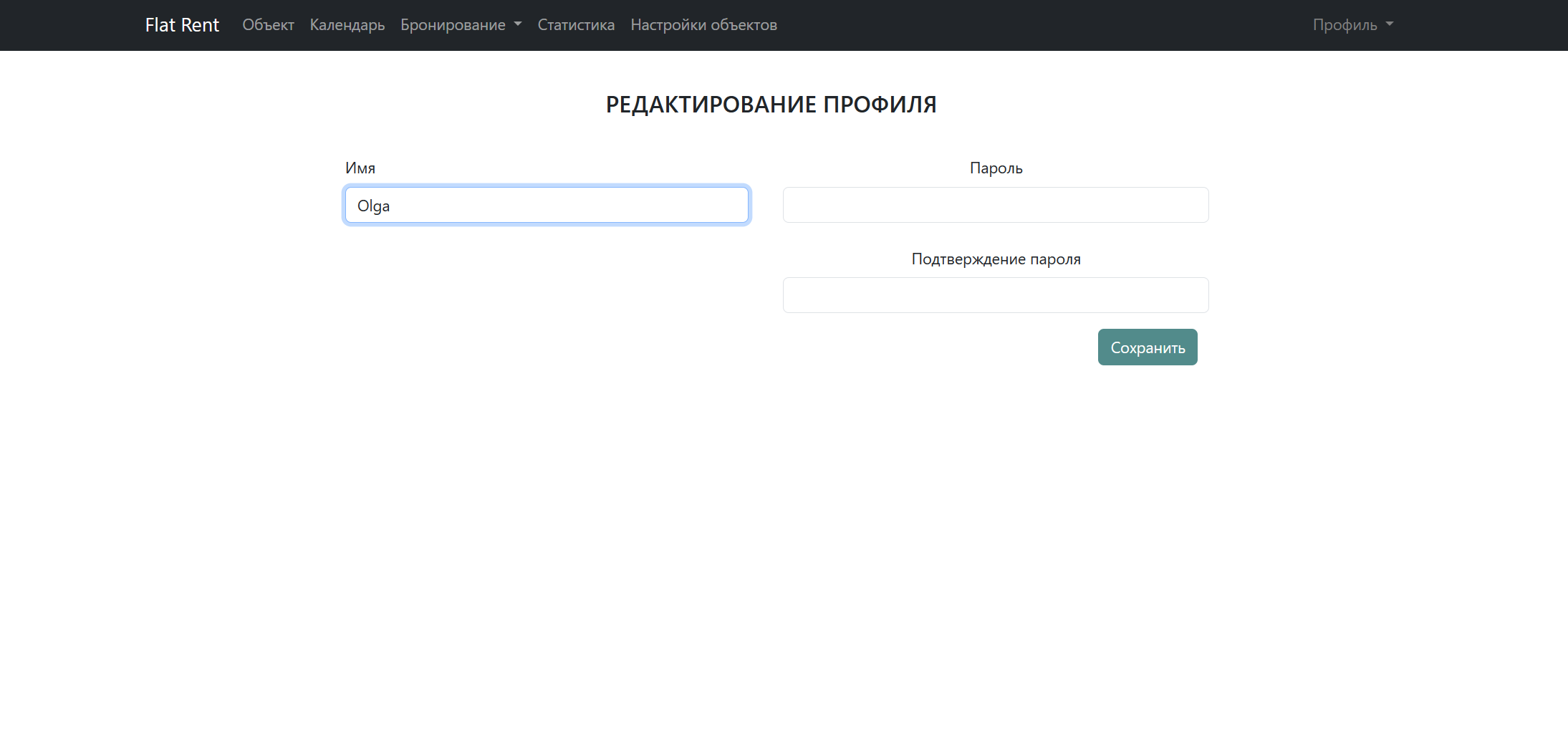


Рисунок . – Редактирование профиля

Для редактирования профиля перейдите в главном меню в *Профиль🡪 Редактировать.* Изменить имя или пароль пользователя при необходимостии *Сохранить.*

**Руководство для арендатора**

Перейдите по ссылке, переданной арендодателем, в открывшемся окне отобразится календарь квартиры, зеленым цветом отмечены доступные даты, красным – забронированные, серым – недоступные к бронированию даты. Введите интересующий период и нажмите *Поиск,* в случае доступности дат появится информация о стоимости и возможной скидке.

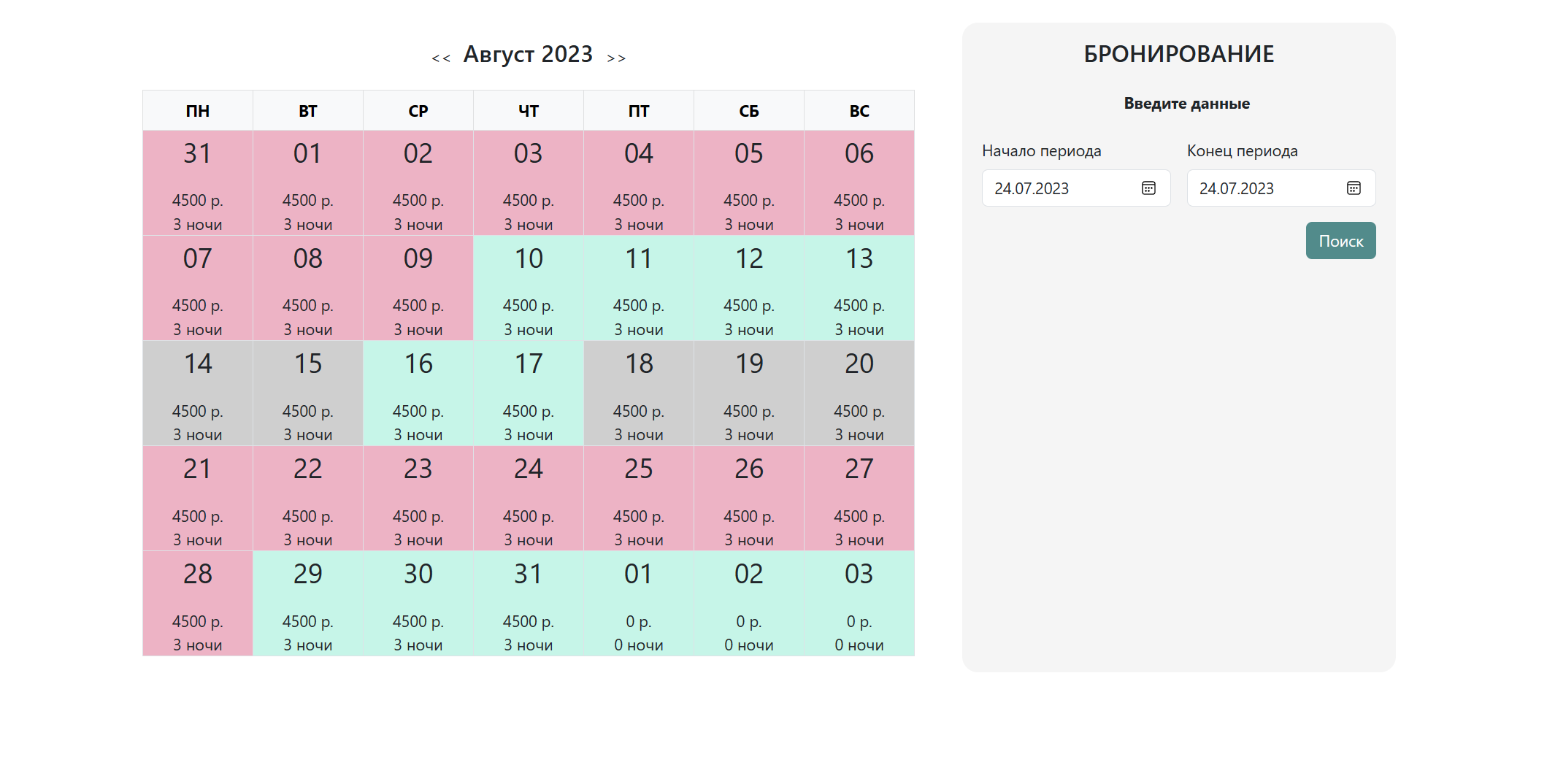


Рисунок . – Страница календаря для арендатора

## Руководство по развертыванию приложения

Ниже приводится руководство по развертыванию приложения на базе сервиса pythonanywhere [12].

1. Создайте аккаунт на сервисе pythonanywhere.com.
2. Через меню Files создайте каталог и загрузите в него предоставляемый по запросу дамп базы данных “**Dump20230721.sql**”, в котором содержится структура БД, наполнение таблиц и запускаемые события для регулярной обработки данных в таблице.
3. Во вкладке Databases создайте базу данных MySQL и укажите её имя, в данном примере база имеет название “olgamarakueva$flatrent”.
4. Задайте MySQL пароль, который будет использоваться для подключения к БД.
5. На странице отображаются данные для подключения к БД, которые затем будут использоваться при настройке проекта.

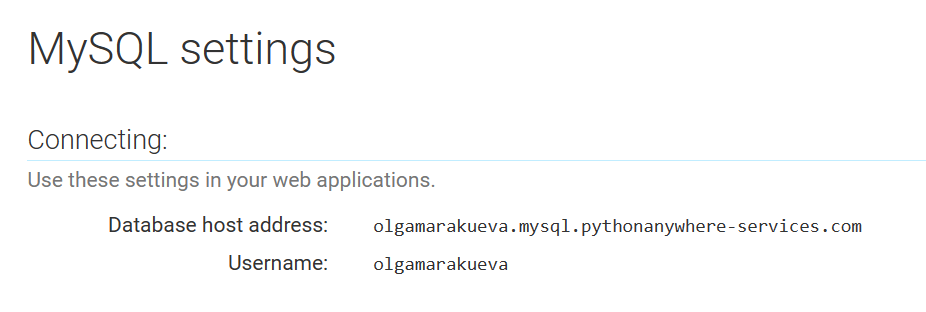


Рисунок . – Данные для подключения к БД

1. Запустите консоль для базы данных, нажав на её имя. В консоли выполните команду: SOURCE path\_to\_dump

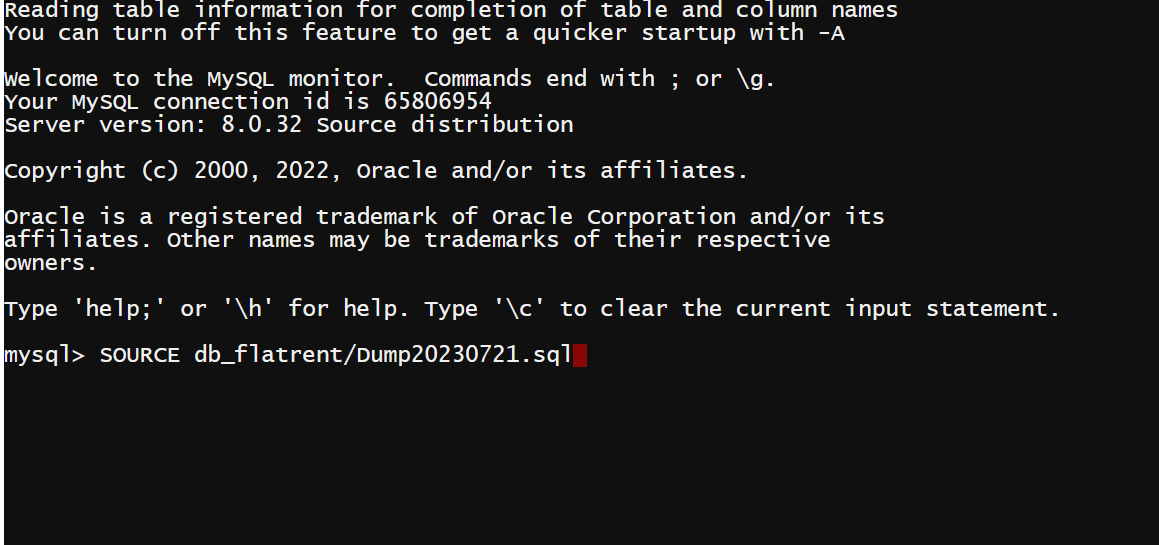


Рисунок . – Загрузка данных в БД

1. Для проверки выполненных команд запустите команды SHOW TABLES и SHOW EVENTS, результатами которых будут загруженные таблицы и события. Если отображаются все данные, база готова к работе.

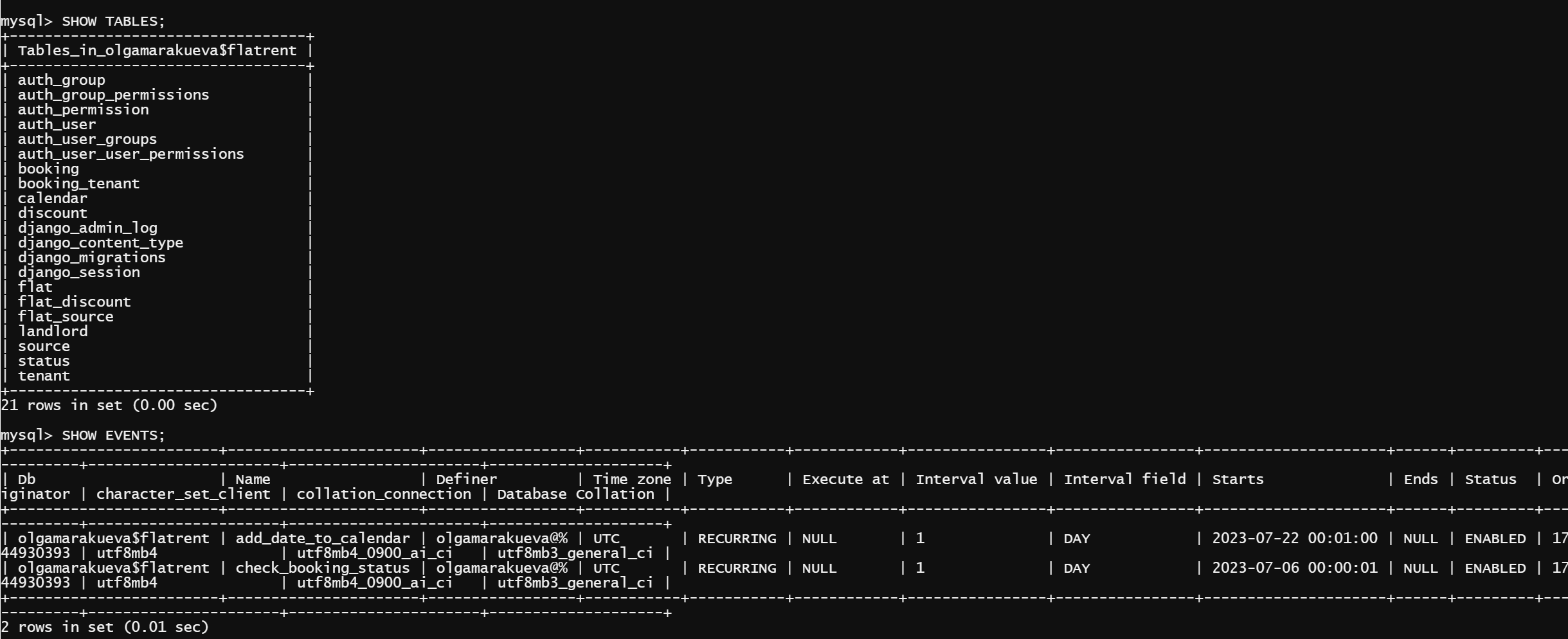


Рисунок . – Проверка выполненного импорта данных в БД

1. Запустите консоль с главной страницы pythonanywhere. И склониируйте содержимое репозитория **https://github.com/OlgaMarakueva/Flat\_Rent.git**



Рисунок . – Клонирование репозитория

1. Создайте виртуальное окружение, запустив следующую команду, где venv – имя окружения



Рисунок . – Создание виртуального окружения

1. Из директории, в которой находится файл requirements.txt, выполните следующую команду для установки всех необходимых библиотек



Рисунок . – Установка библиотек и пакетов

1. Перейдите на главную страницу в Web apps и создайте Web-приложение, выбрав ручную конфигурацию

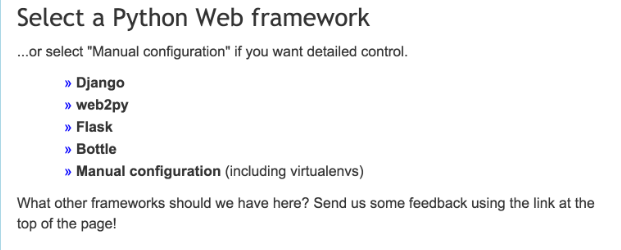


Рисунок . – Выбор конфигурации фреймворка

1. На странице с настройками приложения появится выделенный для него URL



Рисунок . – URL приложения

1. В настройках укажите виртуальную среду, настроенную ранее (рисунок Рисунок 3.30)

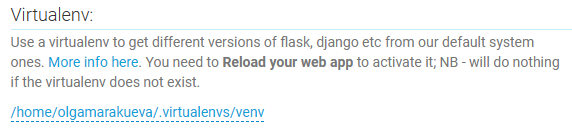


Рисунок . – Настройка виртуальной среды проекта

1. Перейдите к настройкам WSGI конфигурации, открыв соответствующий файл .py



Рисунок . – Настройка WSGI конфигурации

1. В настройках WSGI конфигурации необходимо активировать строки, как на рисунке ниже. Указать путь к проекту (21-я строка) – к каталогу с файлом manage.py. Также необходимо изменить имя модуля в 25-ой строке на имя каталога, в котором содержится файл с настройками. Сохраните файл.



Рисунок . – Файл с настройками WSGI конфигурации

1. Откройте файл settings.py, который находится в каталоге “Flatrent\_website”. В файле отключите режим откладки и в разрешенных хостах укажите ранее выделенное приложению доменное имя (рисунок Рисунок 3.33).

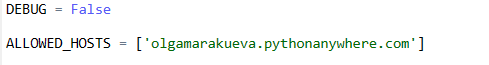


Рисунок . – Настройка домена в settings.py

1. Укажите параметры подключения к БД (рисунок Рисунок 3.26)

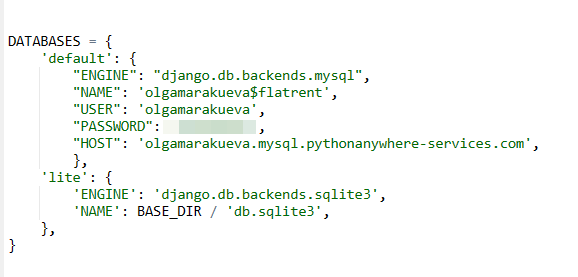


Рисунок . – Настройки БД в settings.py

1. Укажите абсолютный путь к папке STATIC\_ROOT, в которой лежат все статические файлы проекта

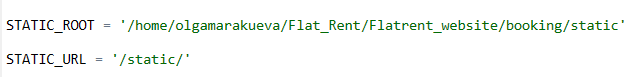


Рисунок . – Настройки путей к статическим файлам в settings.py

1. Сохраните файл и перейдите во вкладку настроек веб-приложения. В параметрах статических файлов задайте URL и директорию согласно файлу settings.py

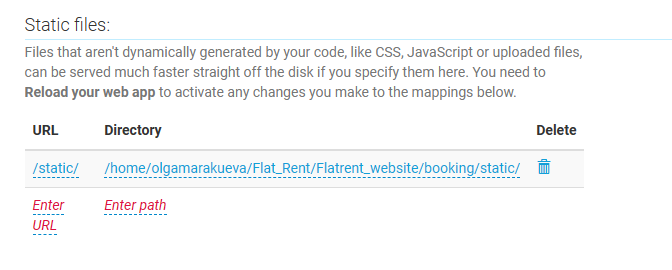


Рисунок . – Настройки путей к статическим файлам

1. Нажмите Reload и перейдите по URL приложения, будет загружена страница входа и регистрации пользователей.

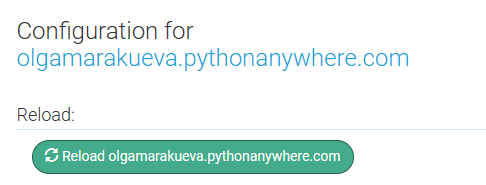


Рисунок . – Конфигурация приложения

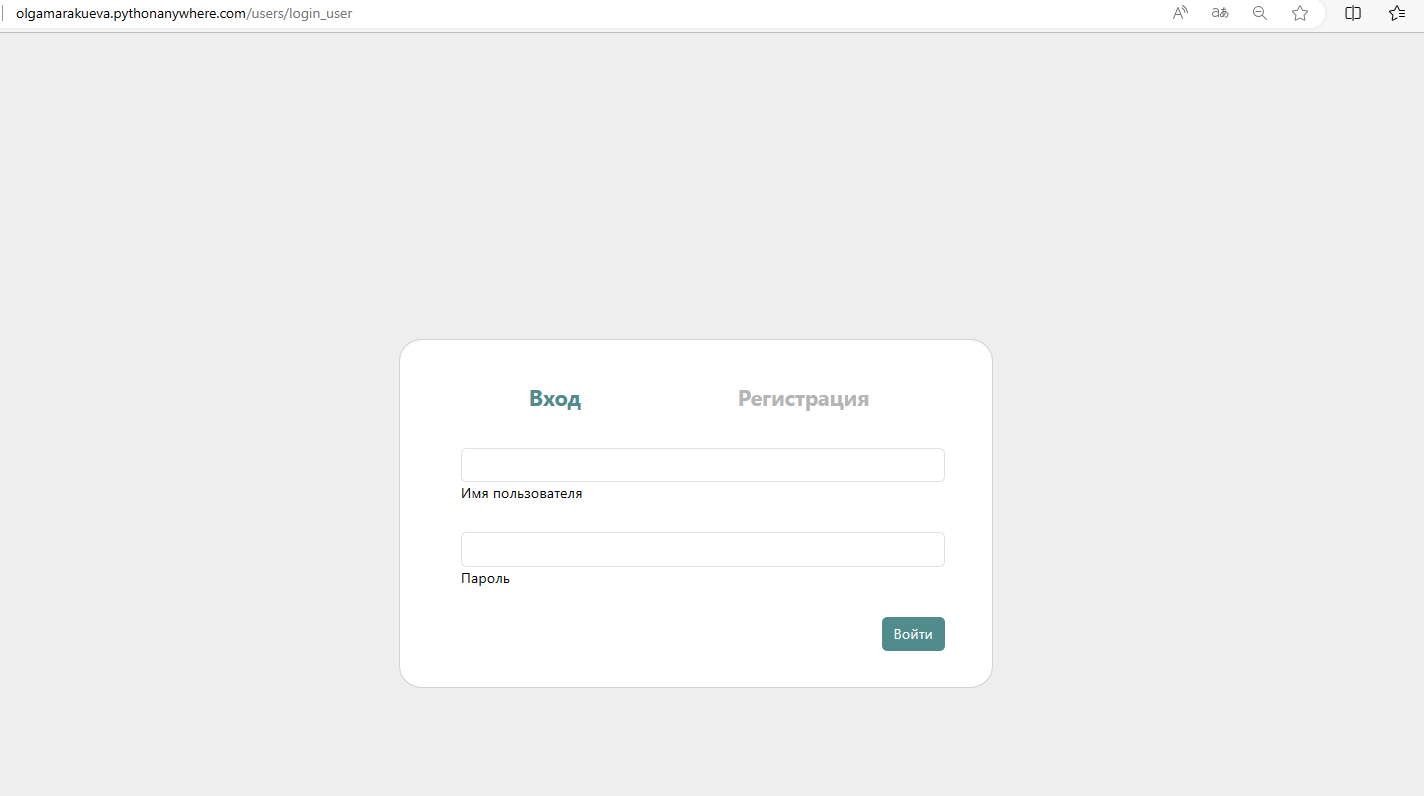


Рисунок . – Страница входа и регистрации

# Заключение

В рамках работы с помощью веб-фреймворка Django выполнена разработка приложения, позволяющего автоматизировать процесс управления посуточной арендой жилья. Проведен анализ предметной области, определены основные понятия и проблемы, сформулированы требования к системе. Спроектирована база данных с применением MySQL. В процессе разработки приложения реализовано большинство функциональных возможностей, определенных на этапе постановки задачи. Приложение размещено на бесплатном облачном сервере pythonanywhere.

На базе разработанного приложения в дальнейшем можно создать инструментарий, анализирующий уровень цен в аналогичных сдающихся объектах, что позволит оптимизировать настройку цен в календаре. Также есть возможность введения личного кабинета для арендаторов, что позволит им отслеживать историю своих бронирований.

# Использованные источники

1. Вигерс К. Разработка требований к программному обеспечению/Пер. с англ. — М.: Издательство-торговый дом «Русская Редакция», 2014. — 736 с.

2. Что такое RealtyCalendar и чем он может помочь посуточнику [Электронный ресурс]: <https://dzen.ru/a/Y8eW8vO9yyUNzeX2>

3. Кара-Ушанов, В.Ю. Модель «Сущность-связь»: учебное электронное текстовое издание / ред. А.В. Ерофеева. – Информационный портал УрФу. [Электронный ресурс]: <https://urfu.ru/ru/>

4. Entity-Relationship (ER) Models [Электронный ресурс]: [https://www.cs.rpi.edu/academics/courses/summer23/csci4380/summer2023/course\_notes/er.html#id5](https://www.cs.rpi.edu/academics/courses/summer23/csci4380/summer2023/course_notes/er.html" \l "id5)

5. Michael Dinder. Becoming an Enterprise Django Developer: Discover best practices, tooling, and solutions for writing and organizing Django applications in production – Packt Publishing Ltd., Birmingham, UK, 2022 – 506p.

6. Антонио Меле. Django 4 в примерах. Разрабатывайте мощные и надежные приложения на Python с нуля – ДМК, Москва, 2023 – 799 стр.

7. Анатолий Постолит. Python Django и PyCharm для начинающих – СПб: «БХВ-Петербург», 2021. — 464 с.

8. Документация Django 4 [Электронный ресурс]: <https://docs.djangoproject.com/en/4.2/>

9. Плюсы и минусы Django [Электронный ресурс]: <https://habr.com/ru/articles/473042/>

10. Bootstrap [Электронный ресурс]: <https://getbootstrap.com>

11. Usage statistics and market share of Bootstrap for websites [Электронный ресурс]: <https://w3techs.com/technologies/details/cs-bootstrap>

12. Deploying an existing Django project on PythonAnywhere [Электронный ресурс]: <https://help.pythonanywhere.com/pages/DeployExistingDjangoProject/>