

Минобрнауки России
Юго-Западный государственный университет

Кафедра программной инженерии

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ПО ПРОГРАММЕ БАКАЛАВРИАТА

09.03.04 Программная инженерия

(код, наименование ОПОП ВО: направление подготовки, направленность (профиль))

«Разработка программно-информационных систем»

Программно-информационная система для управления энергопотреблением

в зданиях

(название темы)

Дипломный проект

(вид ВКР: дипломная работа или дипломный проект)

Автор ВКР

М. С. Черенков

(подпись, дата)

(инициалы, фамилия)

Группа ПО-92з

Руководитель ВКР

И. Н. Ефремова

(подпись, дата)

(инициалы, фамилия)

Нормоконтроль

А. А. Чаплыгин

(подпись, дата)

(инициалы, фамилия)

ВКР допущена к защите:

Заведующий кафедрой

А. В. Малышев

(подпись, дата)

(инициалы, фамилия)

Курск 2024 г.

Минобрнауки России
Юго-Западный государственный университет

Кафедра программной инженерии

УТВЕРЖДАЮ:
Заведующий кафедрой

(подпись, инициалы, фамилия)

«_____» _____ 20____ г.

**ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ
РАБОТУ ПО ПРОГРАММЕ БАКАЛАВРИАТА**

Студента Черенкова М.С., шифр 19-06-0462, группа ПО-92з

1. Тема «Программно-информационная система для управления энергопотреблением в зданиях» утверждена приказом ректора ЮЗГУ от «13» ноября 2023 г. № 5166-с.

2. Срок предоставления работы к защите «21» декабря 2023 г.

3. Исходные данные для создания программной системы:

3.1. Перечень решаемых задач:

- 1) Проанализировать энергетическую инфраструктуру здания.
- 2) Провести анализ энергоснабжения здания, включая источники энергии, сетевые связи и оборудование.
- 3) Исследовать энергопотребление в различных зонах здания (освещение, отопление, кондиционирование и т.д.).
- 4) Разработать концептуальную модель программно-информационной системы управления энергопотреблением на основе современных подходов к энергетическому менеджменту.
- 5) Учесть возможности сбора данных о энергопотреблении с использованием сенсоров и умных устройств.

6) Разработать модель управления, основанную на данных о пиковых нагрузках, энергосбережении и оптимизации расходов.

7) Спроектировать программную систему управления энергопотреблением в зданиях.

8) Определить требования к программному обеспечению для мониторинга и управления энергосистемой здания.

9) Разработать архитектуру системы, включая взаимодействие с умными устройствами, базой данных и пользовательским интерфейсом.

10) Сконструировать и протестировать программную систему управления энергопотреблением в зданиях.

11) Разработать прототип программы для сбора, анализа и визуализации данных об энергопотреблении.

12) Провести тестирование системы на эффективность мониторинга, управления и оптимизации энергопотребления.

3.2. Входные данные и требуемые результаты для программы:

1) Входными данными для программной системы являются: данные о потреблении энергии различными зонами здания; технические характеристики энергетического оборудования; информация об энергетических стандартах и требованиях; данные о тарифах на энергоресурсы; информация о текущих ресурсах и запасах энергии.

2) Выходными данными для программной системы являются: оптимизированный график энергопотребления; отчеты по эффективности использования энергии; предупреждения и уведомления о возможных сбоях или неэффективном использовании энергии; визуализация данных по энергопотреблению для принятия управленческих решений.

4. Содержание работы (по разделам):

4.1. Введение

4.1. Анализ предметной области

4.2. Техническое задание: основание для разработки, назначение разработки, требования к программной системе, требования к оформлению документации.

4.3. Технический проект: общие сведения о программной системе, проект данных программной системы, проектирование архитектуры программной системы, проектирование пользовательского интерфейса программной системы.

4.4. Рабочий проект: спецификация компонентов и классов программной системы, тестирование программной системы, сборка компонентов программной системы.

4.5. Заключение

4.6. Список использованных источников

5. Перечень графического материала:

Лист 1. Сведения о ВКРБ

Лист 2. Цель и задачи разработки

Лист 3. Концептуальная модель данных

Лист 4. Диаграмма прецедентов

Лист 5. Диаграмма компонентов

Лист 6. Диаграмма размещения

Лист 7. Диаграмма классов

Лист 8. Страница с аналитикой энергопотребления в здании

Лист 9. Страница с планом этажа, на котором наглядно отображено размещение помещений

Лист 10. Страница с изображением помещения компании

Лист 11. Страница с изображением профиля пользователя

Лист 12. Страница с диаграммой отображения данных помещения

Лист 13. Страница с добавлением компании

Лист 14. Страница с добавления помещения компании

Лист 15. Заключение

Руководитель ВКР

(подпись, дата)

И. Н. Ефремова

(инициалы, фамилия)

Задание принял к исполнению

(подпись, дата)

М. С. Черенков

(инициалы, фамилия)

РЕФЕРАТ

Объем работы равен 97 страницам. Работа содержит 16 иллюстраций, 6 таблиц, 61 библиографических источников и 15 листов графического материала. Количество приложений – 2. Графический материал представлен в приложении А. Фрагменты исходного кода представлены в приложении Б.

Перечень ключевых слов: энергопотребление, управление энергоресурсами, программно-информационная система, энергетическая инфраструктура, умные устройства, оптимизация энергопотребления, мониторинг, энергетический менеджмент.

Объектом разработки является программно-информационная система для управления энергопотреблением в зданиях.

Целью выпускной квалификационной работы является создание эффективной системы, позволяющей контролировать и оптимизировать энергопотребление в зданиях, с учетом различных зон и источников энергии.

В ходе разработки системы были выполнены следующие этапы:

1. Анализ энергетической инфраструктуры здания, включая источники энергии и оборудование.

2. Разработка концептуальной модели системы управления энергопотреблением, учитывающей современные подходы к энергетическому менеджменту.

3. Спроектирована программа система, включая архитектуру, взаимодействие с умными устройствами и пользовательским интерфейсом.

4. Сконструирована и протестирована программа для мониторинга, управления и оптимизации энергопотребления.

Входными данными для системы являются информация о потреблении энергии различными зонами здания, технические характеристики оборудования, данные о тарифах и стандартах. Выходные данные включают оптимизированный график энергопотребления, отчеты и предупреждения.

Разработанная программно-информационная система предоставляет эффективные инструменты для управления энергопотреблением в зданиях,

способствуя оптимизации расходов и повышению энергетической эффективности.

ABSTRACT

The volume of work is 97 pages. The work contains 16 illustrations, 6 tables, 61 bibliographic sources and 15 sheets of graphic material. The number of applications is 2. The graphic material is presented in annex A. The layout of the site, including the connection of components, is presented in annex B.

List of keywords: energy consumption, energy resource management, software information system, energy infrastructure, smart devices, energy consumption optimization, monitoring, energy management.

The object of development is a software information system designed for managing energy consumption in buildings.

The objective of the work is to create an effective system capable of monitoring and optimizing energy consumption in buildings, considering various zones and energy sources.

The development stages of the system included:

1. Analysis of the energy infrastructure of the building, encompassing energy sources and equipment.
2. Development of a conceptual model for energy consumption management system, taking into account modern approaches to energy management.
3. Design of the software system, including architecture, interaction with smart devices, and user interface.
4. Construction and testing of the program for monitoring, management, and optimization of energy consumption.

Input data for the system include information on energy consumption in different building zones, technical characteristics of equipment, data on tariffs, and standards. Output data consist of an optimized energy consumption schedule, reports, and warnings.

The developed software information system provides efficient tools for managing energy consumption in buildings, contributing to cost optimization and increased energy efficiency.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	10
1 Анализ предметной области	13
1.1 Тенденции в энергопотреблении зданий	13
1.2 Источники энергии и их влияние	14
1.3 Зоны потребления и мониторинг энергопотребления	14
1.4 Умные устройства и технологии	15
1.5 Вызовы и требования к системе управления энергопотреблением	16
2 Техническое задание	18
2.1 Основание для разработки	18
2.2 Цель и назначение разработки	18
2.3 Требования пользователя к интерфейсу программной системы	18
2.4 Моделирование вариантов использования	23
2.4.1 Вариант использования «Регистрация пользователя»	26
2.4.2 Вариант использования «Регистрация компании»	26
2.4.3 Вариант использования «Добавление помещения»	27
2.4.4 Вариант использования «Просмотр статистики по компании»	28
2.4.5 Вариант использования «Просмотр статистики помещения»	28
2.4.6 Вариант использования «Добавление устройства»	29
2.5 Требования к оформлению документации	30
3 Технический проект	31
3.1 Общая характеристика организации решения задачи	31
3.2 Обоснование выбора технологии проектирования	31
3.2.1 Описание используемых технологий и языков программирования	32
3.2.2 Язык программирования Python Django	33
3.3 Диаграмма компонентов	34
3.3.1 Описание настройки Django	37
3.3.1.1 База данных	37
3.3.1.2 Статические файлы и медиа	38

3.3.1.3 Настройки безопасности	39
3.4 Диаграмма размещения	39
3.5 Диаграмма классов	42
3.6 Содержание информационных блоков. Основные сущности	45
4 Рабочий проект	50
4.1 Классы, используемые при разработке системы управления энергопотреблением	50
4.2 Модульное тестирование разработанной системы управления энергопотреблением	52
4.3 Системное тестирование разработанного web-сайта	58
ЗАКЛЮЧЕНИЕ	63
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	65
ПРИЛОЖЕНИЕ А Представление графического материала	73
ПРИЛОЖЕНИЕ Б Фрагменты исходного кода программы	89
На отдельных листах (CD-RW в прикрепленном конверте)	97
Сведения о ВКРБ (Графический материал / Сведения о ВКРБ.png)	Лист 1
Цель и задачи разработки (Графический материал / Цель и задачи разработки.png)	Лист 2
Концептуальная модель данных (Графический материал / Концептуальная модель данных.png)	Лист 3
Диаграмма прецедентов (Графический материал / Диаграмма прецедентов.png)	Лист 4
Диаграмма компонентов (Графический материал / Диаграмма компонентов.png)	Лист 5
Диаграмма размещения (Графический материал / Диаграмма размещения.png)	Лист 6
Диаграмма классов (Графический материал / Диаграмма классов.png)	Лист 7
Страница с аналитикой энергопотребления в здании (Графический материал / Страница с аналитикой энергопотребления в здании.png)	Лист 8
Страница с планом этажа, на котором наглядно отображено размещение помещений (Графический материал / Страница с планом этажа, на котором	

наглядно отображено размещение помещений.png)	Лист 9
Страница с изображением помещения компании (Графический материал /	
Страница с изображением помещения компании.png)	Лист 10
Страница с изображением профиля пользователя (Графический материал /	
Страница с изображением профиля пользователя.png)	Лист 11
Страница с диаграммой отображения данных помещения (Графический	
материал / Страница с диаграммой отображения данных помещения	
.png)	Лист 12
Страница с добавлением компании (Графический материал / Страница с	
добавлением компании .png)	Лист 13
Страница с добавления помещения компании (Графический материал /	
Страница с добавления помещения компании .png)	Лист 14
Заключение (Графический материал / Заключение.png)	Лист 15

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ПИС – программно-информационная система.

ЭИ – энергетическая инфраструктура.

УУ – умные устройства.

ОЭ – оптимизация энергопотребления.

МЭ – мониторинг энергопотребления.

ГЭ – график энергопотребления.

РА – расходы энергии.

ВЭ – визуализация энергопотребления.

ТД – технические данные.

СУЭ – система управления энергопотреблением.

ПЭ – потребление энергии.

ВВЕДЕНИЕ

В современном мире, где энергетические ресурсы становятся все более дефицитными и цennыми, эффективное управление энергопотреблением в зданиях приобретает особенное значение. Для достижения устойчивости и снижения негативного воздействия на окружающую среду необходимы инновационные подходы, способные совмещать современные технологии и информационные системы. В этом контексте вступает в игру Программно-информационная система для управления энергопотреблением в зданиях, представляя собой передовое решение, направленное на оптимизацию энергетической эффективности и уменьшение воздействия на климат.

Эта интегрированная система предназначена для комплексного мониторинга, анализа и управления энергопотреблением в зданиях различного назначения, включая офисные здания, торговые центры, промышленные объекты и жилые комплексы. С ее помощью пользователи получают возможность в режиме реального времени отслеживать энергетические показатели, выявлять неэффективные потребители, оптимизировать расход электроэнергии и, таким образом, значительно снижать эксплуатационные расходы.

Основные преимущества программы включают в себя интеллектуальное управление системами освещения, отопления, вентиляции и кондиционирования воздуха, а также возможность удаленного доступа и управления через веб-интерфейс. Это позволяет адаптировать энергопотребление под конкретные потребности, минимизировать потери энергии и создавать оптимальные условия для жизни и работы.

Программно-информационная система для управления энергопотреблением в зданиях не только способствует рациональному использованию ресурсов, но также является важным шагом в направлении создания устойчивого и энергоэффективного общества. Современные технологии в области управления энергопотреблением становятся неотъемлемой частью стратегии устойчивого развития, и наша программа является ключевым инструментом в этом процессе.

Кроме того, Программно-информационная система (ПИС) для управления энергопотреблением в зданиях обеспечивает функционал сбора и анализа данных, что позволяет выявлять тенденции в потреблении энергии и предоставляет ценную информацию для принятия стратегических решений. Алгоритмы помощника встроенного в систему позволяют автоматически оптимизировать процессы управления, учитывая изменяющиеся потребности и внешние факторы.

Безопасность данных является ключевым аспектом работы системы. Программа предусматривает современные методы шифрования и механизмы защиты конфиденциальности, обеспечивая полную безопасность информации о потреблении энергии и других важных параметрах здания.

Цель данной работы заключается в создании эффективной программно-информационной системы, способной мониторинга, управления и оптимизации энергопотребления в различных зонах зданий. Для достижения этой цели необходимо решить ряд *ключевых задач*:

- Провести анализ энергетической инфраструктуры зданий с учетом различных источников энергии и зон потребления.
- Разработать концептуальную модель программной системы управления энергопотреблением, основанную на передовых методах энергетического менеджмента.
- Спроектировать программную систему, включая архитектуру, взаимодействие с умными устройствами и пользовательский интерфейс.
- Сконструировать и протестировать программную систему, обеспечивающую эффективное управление и мониторинг энергопотребления.

Структура и объем работы. Отчет состоит из введения, 4 разделов основной части, заключения, списка использованных источников, 2 приложений. Текст выпускной квалификационной работы равен 13 страницам.

Во введении сформулирована цель работы, поставлены задачи разработки, описана структура работы, приведено краткое содержание каждого из разделов.

В первом разделе на стадии описания технической характеристики предметной области приводится сбор информации о деятельности компании, для которой осуществляется разработка сайта.

Во втором разделе на стадии технического задания приводятся требования к разрабатываемому сайту.

В третьем разделе на стадии технического проектирования представлены проектные решения для web-сайта.

В четвертом разделе приводится список классов и их методов, использованных при разработке сайта, производится тестирование разработанного сайта.

В заключении излагаются основные результаты работы, полученные в ходе разработки.

В приложении А представлен графический материал. В приложении Б представлены фрагменты исходного кода.

1 Анализ предметной области

1.1 Тенденции в энергопотреблении зданий

Энергопотребление в современных зданиях охватывает широкий спектр аспектов, начиная от примитивных электрических устройств и заканчивая сложными системами умного дома. С ростом населения и городской застройки увеличивается спрос на энергию, что ставит перед нами задачу разработки эффективных и устойчивых решений для управления этим потреблением.

Системы отопления, вентиляции и кондиционирования в современных зданиях стали неотъемлемой частью обеспечения комфортных условий проживания и работы. Однако, их энергозатраты могут быть значительными, особенно при неэффективном использовании или устаревших технологиях. Разработка более энергоэффективных систем и внедрение интеллектуальных алгоритмов управления помогут снизить негативное воздействие на окружающую среду.

В свете современных тенденций также становится важным внедрение возобновляемых источников энергии в зданиях. Солнечные панели, ветрогенераторы и другие альтернативные источники играют ключевую роль в создании устойчивых, экологически чистых систем энергоснабжения. Их внедрение требует не только технической экспертизы, но и поддержки со стороны законодательства и общественного сознания.

Кроме того, с увеличением количества "умных" устройств в зданиях, сбор и анализ данных о потреблении энергии становятся важным инструментом для оптимизации эффективности. Использование сенсоров, интернета вещей и технологий искусственного интеллекта позволяет создавать адаптивные системы, способные предсказывать и реагировать на изменения в потреблении энергии, что в итоге снижает издержки и негативное воздействие на окружающую среду.

1.2 Источники энергии и их влияние

Разнообразие источников энергии, которые применяются в современных зданиях, включает в себя электроэнергию, тепловую энергию, а также возобновляемые источники. Эта разнообразная палитра выбора создает сложную динамику в энергетической инфраструктуре зданий, требуя тщательного анализа влияния каждого источника на общую эффективность системы энергоснабжения.

Электроэнергия, как один из основных источников, обеспечивает функционирование разнообразных устройств и систем внутри здания. Однако, зависимость от традиционных источников электроэнергии, таких как ископаемые топлива, может привести к негативным экологическим последствиям. В связи с этим, переход к более чистым источникам, таким как солнечная и ветровая энергия, становится важным шагом в снижении углеродного следа зданий.

Тепловая энергия, используемая для обеспечения отопления и горячего водоснабжения, также подвергается пересмотру в контексте устойчивости. Внедрение технологий, основанных на использовании геотермальной энергии или других возобновляемых источников, может не только снизить нагрузку на традиционные системы, но и сделать здания более экологически дружелюбными.

Возобновляемые источники энергии, такие как солнечная и ветровая энергия, играют ключевую роль в создании устойчивых систем энергоснабжения. Их использование содействует не только сокращению выбросов углерода, но и обеспечению независимости от изменений цен на традиционные источники энергии.

1.3 Зоны потребления и мониторинг энергопотребления

Внутри современных зданий существует многообразие зон, каждая из которых обладает уникальными потребностями в энергии. Эти зоны могут варьироваться от офисных пространств до жилых кварталов, а каждая из них

требует индивидуального и оптимизированного подхода к энергоснабжению. Мониторинг энергопотребления в каждой из этих зон становится ключевым инструментом для выявления уникальных особенностей и определения возможностей для оптимизации энергетических процессов.

Офисные пространства, например, могут подвергаться резким изменениям в потреблении энергии в зависимости от времени суток и дня недели. Мониторинг, анализирующий эти изменения, позволяет предпринимать целенаправленные меры по снижению энергопотребления в периоды низкой активности, таким образом, повышая общую эффективность здания.

В жилых квартирах и домах, особенности потребления энергии могут зависеть от привычек и активности жителей. Точное измерение и анализ энергопотребления в этих зонах позволяют предоставлять персонализированные рекомендации по энергосбережению для жильцов, создавая комфортные условия без излишних затрат.

Мониторинг также обеспечивает возможность выявления неэффективных систем и устройств, способствуя своевременной замене или модернизации. Предоставление детальной информации о зонах потребления дает возможность инженерам и администраторам зданий точно настраивать системы управления энергопотреблением в соответствии с потребностями каждой конкретной зоны.

1.4 Умные устройства и технологии

Внедрение умных устройств и передовых технологий в здания – неотъемлемая часть современных энергосберегающих и устойчивых практик. Системы умного освещения, автоматизированные системы отопления и кондиционирования создают интеллектуальное окружение, которое может не только повысить комфорт, но и существенно снизить энергопотребление.

Системы умного освещения являются примером эффективного использования технологий для оптимизации энергии. Датчики движения и системы автоматического выключения не только обеспечивают освещение только в тех зонах, где оно действительно нужно, но и могут адаптироваться к внеш-

ним условиям, таким как естественное освещение из окон. Это позволяет снизить излишнее освещение и улучшить общую энергоэффективность.

Автоматизированные системы отопления и кондиционирования также играют важную роль в оптимизации энергопотребления зданий. Использование сенсоров для мониторинга температуры, влажности и других параметров позволяет точно регулировать условия в каждой зоне, реагируя на потребности и предотвращая избыточное потребление энергии.

Современные умные технологии также включают в себя системы управления энергопотреблением с использованием искусственного интеллекта. Эти системы могут анализировать данные о потреблении энергии в реальном времени, прогнозировать требования и автоматически оптимизировать работу устройств для достижения максимальной эффективности.

1.5 Вызовы и требования к системе управления энергопотреблением

Анализ предметной области энергопотребления в зданиях выдвигает перед нами ряд значительных вызовов, которые требуют системного подхода для достижения энергетической эффективности, устойчивости и соблюдения экологических стандартов. В этом контексте, требования к системе управления энергопотреблением становятся ключевым инструментом для успешного справления с вызовами современной энергетики.

Необходимость снижения затрат на энергию выступает в роли актуальной задачи, особенно в условиях растущего спроса на энергию и ограниченных ресурсов. Системы управления должны обеспечивать точный мониторинг и анализ энергопотребления в режиме реального времени, а также предоставлять механизмы для выявления и устранения избыточного потребления.

Увеличение устойчивости энергоснабжения требует решений, способных эффективно взаимодействовать с возобновляемыми источниками энергии, а также предоставлять надежную и устойчивую поддержку в случае возможных сбоев в сети. Гибкость и адаптивность становятся ключевыми эле-

ментами для эффективной интеграции разнообразных источников энергии и обеспечения устойчивости энергоснабжения.

Соблюдение экологических стандартов выходит на передний план, учитывая растущую осведомленность о климатических изменениях и необходимость снижения углеродного следа. Системы управления должны внедрять эффективные меры оптимизации, направленные на минимизацию воздействия на окружающую среду, а также предоставлять данные для оценки и соблюдения экологических стандартов.

Гибкость, точность и эффективность становятся критическими требованиями для систем управления энергопотреблением. Эти аспекты не только обеспечивают экономическую эффективность, но и являются фундаментальными элементами для успешной реализации концепции устойчивого развития в области энергопотребления зданий. Встроенные в системы управления инновационные подходы могут стать ключом к переходу к устойчивой и энергетически эффективной будущей парадигме.

2 Техническое задание

2.1 Основание для разработки

Основанием для разработки является задание выпускной квалификационной работы на создание программно-информационной системы для управления энергопотреблением в зданиях. Данная система призвана эффективно мониторить, контролировать и оптимизировать использование энергии в различных зонах зданий.

2.2 Цель и назначение разработки

Целью разработки программно-информационной системы является обеспечение устойчивого и эффективного управления энергопотреблением в зданиях. Основное предназначение системы — предоставление пользователю возможности контролировать, анализировать и оптимизировать расход энергии в реальном времени.

Для достижения поставленной цели необходимо решить следующие задачи:

- Анализ энергетической инфраструктуры зданий, выявление ключевых факторов и паттернов потребления.
- Разработка концептуальной модели системы управления энергопотреблением.
- Проектирование программной системы, включая архитектуру, взаимодействие с устройствами и пользовательский интерфейс.
- Сконструирование и тестирование программной системы для управления энергопотреблением в зданиях.

2.3 Требования пользователя к интерфейсу программной системы

Программная система должна обеспечивать удобный и интуитивно понятный интерфейс для пользователей. В интерфейсе должны быть предусмотрены следующие элементы:

- Навигацию по разделам и зонам здания.
 - Возможность авторизации для различных уровней доступа.
 - Отображение текущего статуса энергопотребления в различных зонах.
- Возможность ввода параметров для настройки системы в ручном режиме.

Страница с аналитикой энергопотребления в здании представлена на рисунках 2.1 - 2.2.

Концептуальная модель данных программной системы в виде UML-диаграммы сущность-связь представлена на рисунке 2.3.

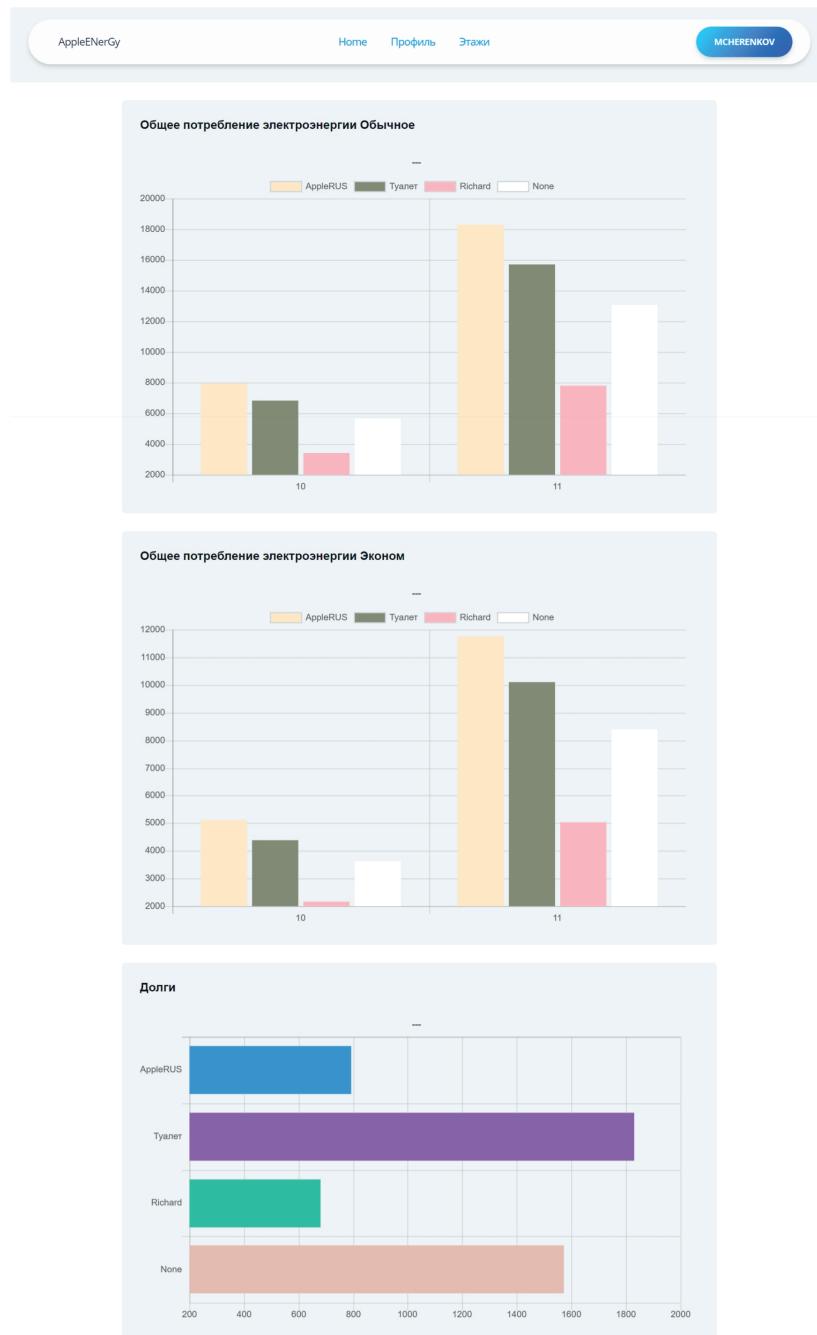


Рисунок 2.1 – Страница с аналитикой энергопотребления в здании

Для обеспечения удобства использования программной системы и удовлетворения потребностей пользователя, определены следующие требования к интерфейсу.

Входные данные

Показания с датчиков: Пользователь может вводить информацию о потреблении энергии вручную или автоматически, полученную от датчиков в помещениях компаний. Эти данные включают в себя значения потребления электроэнергии с и без использования умного помощника, общее количество

Компании							Добавить
	Наименование	Тип	Сотрудники	Помещения	Потребление О	Потребление Э	
<input type="checkbox"/>	AppleRUS	Очищенный миндаль #FFEBBC	21	301 302 201 202 205 206 207	7954.34 кВт/ч	5104.14 кВ	
<input type="checkbox"/>	Туалет	Бледно-зелено-серый #8D917A	1	303 304 203 204 103 104	6822.490000000001 кВт/ч	4376.99 кВ	
<input type="checkbox"/>	Richard	Розовый #FFC0CB	12	305 306 307	3404.8 кВт/ч	2174.64 кВ	
<input type="checkbox"/>	None	Белый #FFFFFF	0	101 102 105 106 107	5652.719999999999 кВт/ч	3632.92 кВ	

Помещения							Добавить
	Наименование	Компания	Этаж	Потребление О	Потребление Э	Долг	
<input type="checkbox"/>	301	AppleRUS	3	3740.42кВт/ч	2409.86кВт/ч	374.78₽	Удалить
<input type="checkbox"/>	302	AppleRUS	3	3750.96кВт/ч	2413.28кВт/ч	375.32₽	Удалить
<input type="checkbox"/>	303	Туалет	3	3749.19кВт/ч	2409.93кВт/ч	374.79₽	Удалить
<input type="checkbox"/>	304	Туалет	3	3746.53кВт/ч	2424.34кВт/ч	377.04₽	Удалить
<input type="checkbox"/>	305	Richard	3	3757.04кВт/ч	2410.48кВт/ч	374.88₽	Удалить
<input type="checkbox"/>	306	Richard	3	3737.5кВт/ч	2390.22кВт/ч	371.73₽	Удалить
<input type="checkbox"/>	307	Richard	3	3736.25кВт/ч	2404.01кВт/ч	373.87₽	Удалить
<input type="checkbox"/>	201	AppleRUS	2	3746.39кВт/ч	2403.52кВт/ч	373.8₽	Удалить
<input type="checkbox"/>	202	AppleRUS	2	3750.58кВт/ч	2404.97кВт/ч	374.02₽	Удалить
<input type="checkbox"/>	203	Туалет	2	3757.94кВт/ч	2413.08кВт/ч	375.28₽	Удалить
<input type="checkbox"/>	204	Туалет	2	3751.05кВт/ч	2415.52кВт/ч	375.66₽	Удалить
<input type="checkbox"/>	205	AppleRUS	2	3740.31кВт/ч	2400.19кВт/ч	373.28₽	Удалить
<input type="checkbox"/>	206	AppleRUS	2	3767.15кВт/ч	2426.02кВт/ч	377.3₽	Удалить
<input type="checkbox"/>	207	AppleRUS	2	3755.81кВт/ч	2410.2кВт/ч	374.84₽	Удалить
<input type="checkbox"/>	101	None	1	3742.86кВт/ч	2396.56кВт/ч	372.72₽	Удалить
<input type="checkbox"/>	102	None	1	3746.39кВт/ч	2404.84кВт/ч	374.0₽	Удалить
<input type="checkbox"/>	103	Туалет	1	3751.29кВт/ч	2414.06кВт/ч	375.44₽	Удалить
<input type="checkbox"/>	104	Туалет	1	3766.1кВт/ч	2404.55кВт/ч	373.96₽	Удалить
<input type="checkbox"/>	105	None	1	3742.87кВт/ч	2418.82кВт/ч	376.18₽	Удалить
<input type="checkbox"/>	106	None	1	3742.33кВт/ч	2399.67кВт/ч	373.2₽	Удалить
<input type="checkbox"/>	107	None	1	3732.67кВт/ч	2417.63кВт/ч	375.99₽	Удалить

Рисунок 2.2 – Страница с аналитикой энергопотребления в здании потребляемой электроэнергии, общую стоимость потребления и данные о влажности, температуре, освещенности и наличии движения.

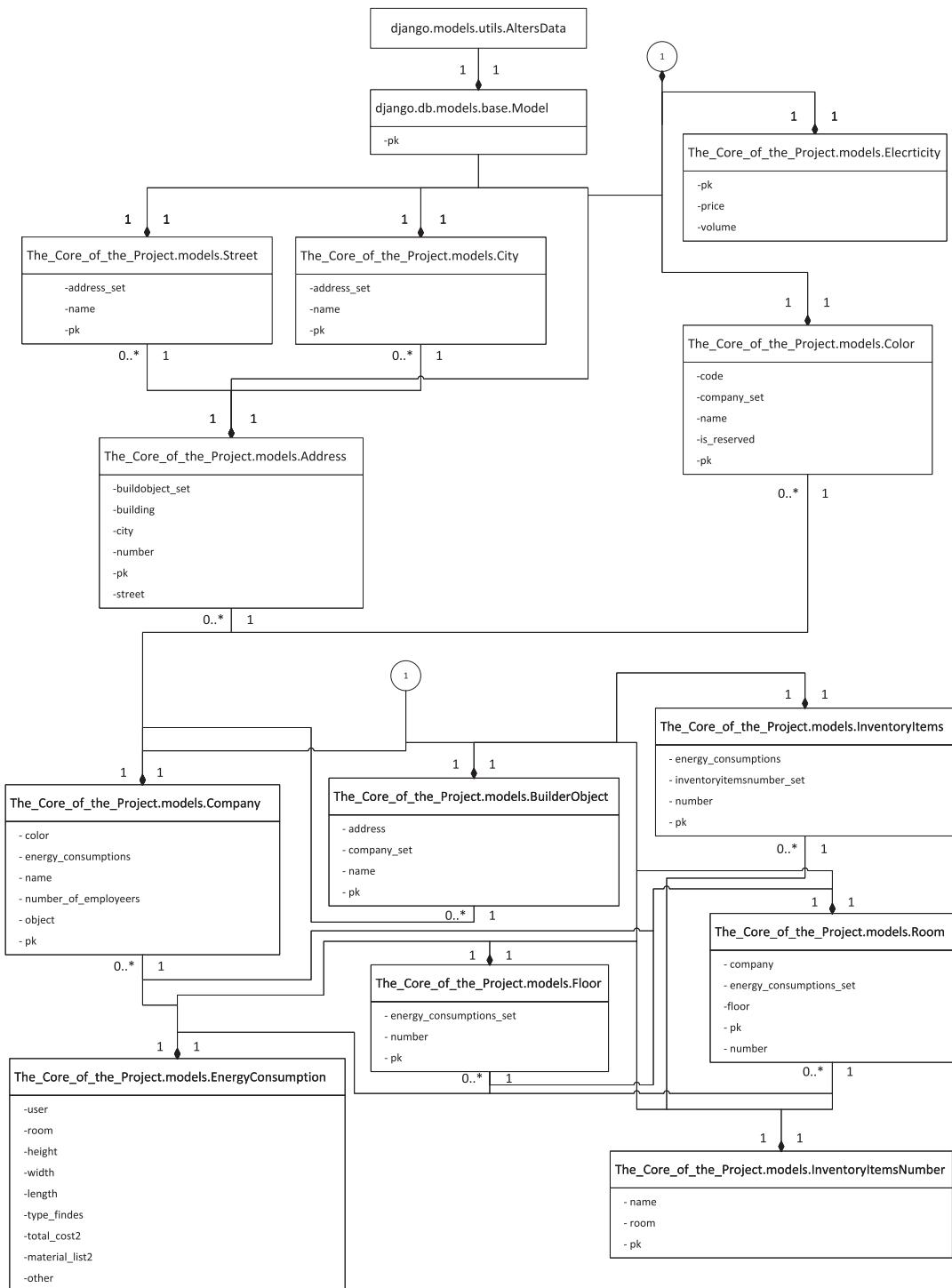


Рисунок 2.3 – Концептуальная модель данных

Информация о помещениях, компаниях и других сущностях: Для корректного анализа данных необходимо вводить информацию о помещениях, компаниях, этажах, предметах инвентаризации и ценах на электроэнергию.

Выходные данные

Отчеты и аналитика: Система предоставляет пользователю различные отчеты и аналитическую информацию о потреблении энергии компаниями. Эти данные включают в себя статистику по потреблению, расходам, влажности, температуре, освещенности и движению.

Графики и визуализация: Для наглядного представления данных система предоставляет графики и визуализацию, позволяя пользователю легко интерпретировать информацию.

Управление данными: Пользователь может вносить изменения в информацию о помещениях, компаниях и других сущностях через интерфейс системы.

Оповещения: В случае выявления аномалий или превышения установленных пороговых значений, система предоставляет уведомления и оповещения пользователю.

Система должна обеспечивать простоту ввода данных, интуитивно понятный интерфейс для взаимодействия с аналитикой, а также обеспечивать возможность быстрого доступа к ключевой информации о потреблении энергии.

2.4 Моделирование вариантов использования

Для проектирования программной системы была создана модель вариантов использования. Данная модель представляет собой набор сценариев взаимодействия пользователей с системой. Диаграмма прецедентов представлена на рисунке 2.4. Она включает следующие прецеденты:

Пользовательские прецеденты:

Мониторинг энергопотребления

Описание: Пользователь, будучи администратором, может просматривать текущий статус энергопотребления в различных зонах здания. Эта функциональность предоставляет наглядное представление о расходе энергии и помогает выявить потенциальные проблемы.

Включает:

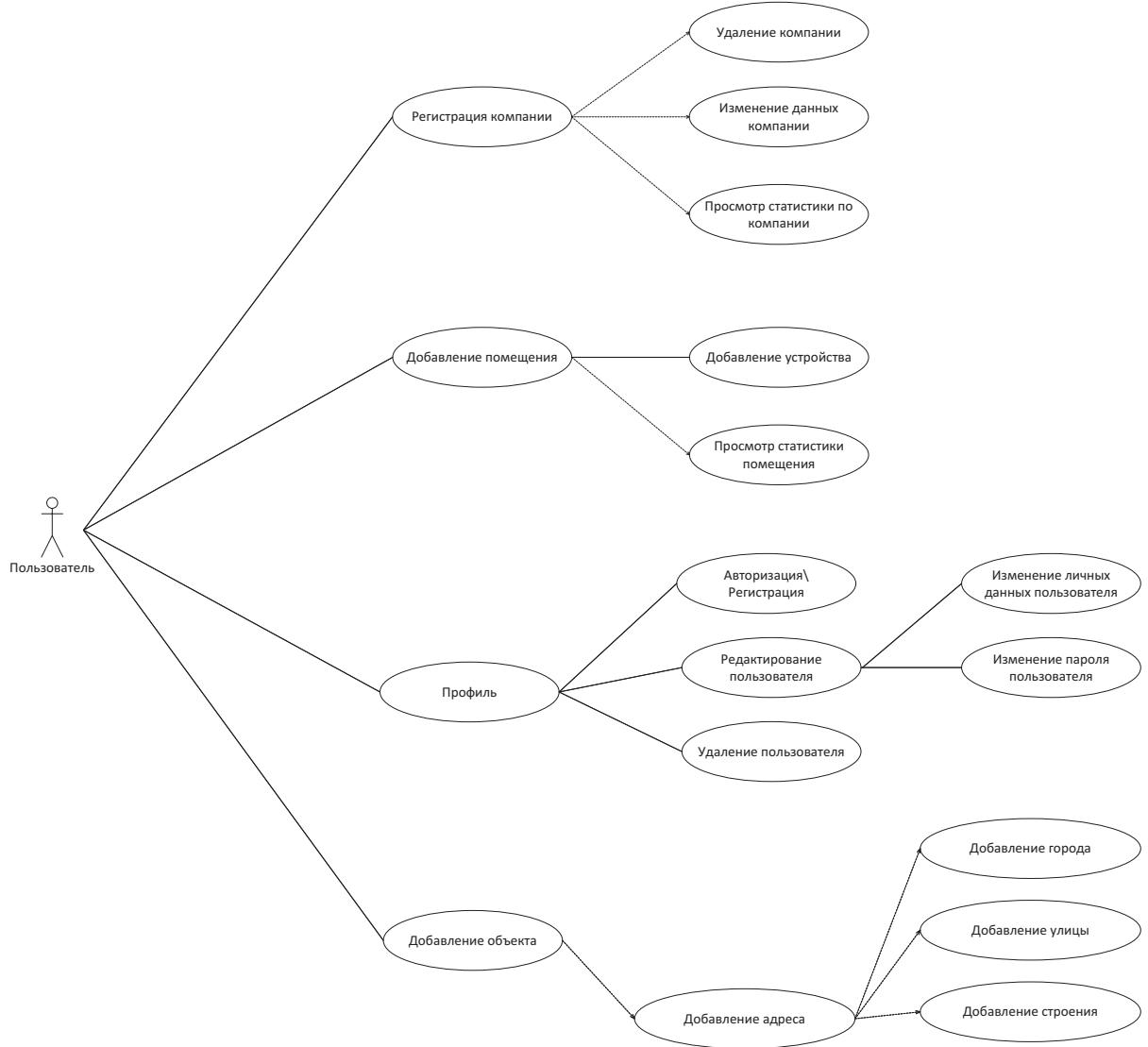


Рисунок 2.4 – Диаграмма прецедентов

- Просмотр статуса энергопотребления по зонам.
- Отображение графиков и диаграмм энергопотребления.

Настройка параметров

Описание: Пользователь, имеющий права администратора, может вносить изменения в параметры системы. Это включает в себя настройку пороговых значений, оптимизацию работы системы в соответствии с текущей потребностью, и внесение изменений в режим работы оборудования.

Включает:

- Установка пороговых значений энергопотребления.
- Активация/деактивация режимов работы оборудования.
- Регулирование параметров оптимизации энергопотребления.

Отчеты и аналитика

Описание: Администратор системы может генерировать отчеты и аналитическую информацию о энергопотреблении. Эти отчеты предоставляют детальную статистику, позволяющую анализировать эффективность системы и выявлять тренды в потреблении энергии.

Включает:

- Генерация отчетов по энергопотреблению за определенный период времени.
- Просмотр статистики по различным зонам и устройствам.
- Анализ трендов и предоставление рекомендаций для оптимизации.

Системные прецеденты:

Интеграция с устройствами

Описание: Система должна взаимодействовать с устройствами, измеляющими энергопотребление в здании. Это включает в себя подключение к датчикам, извлечение данных и передачу команд для управления оборудованием.

Включает:

- Подключение к датчикам энергопотребления.
- Извлечение данных о текущем энергопотреблении.
- Передача команд для регулирования работы оборудования.

Обработка данных

Описание: Система должна обрабатывать данные, полученные от устройств, проводить анализ и формировать статистику для дальнейшего использования. Это включает в себя фильтрацию данных, выявление аномалий и расчет ключевых показателей энергопотребления.

Включает:

- Фильтрация и агрегация данных от устройств.
item Анализ энергопотребления для выявления аномалий.
- Расчет статистических показателей для формирования отчетов.

Управление системой

Описание: Администратор системы должен иметь возможность управлять работой системы, в том числе изменять параметры, добавлять новые устройства, и обеспечивать непрерывную работу программно-информационной системы.

Включает:

- Добавление и удаление устройств в системе.
- Регулирование параметров системы для оптимальной работы.
- Обеспечение непрерывной работы системы и решение возможных проблем.

2.4.1 Вариант использования «Регистрация пользователя»

Заинтересованные лица и их требования: Пользователь, который не имеет учетной записи в системе, желает создать новый аккаунт.

Предусловие: Пользователь не авторизован в системе.

Постусловие: У пользователя создана учетная запись в системе.

Основной успешный сценарий:

- Пользователь открывает страницу регистрации.
- Вводит свои персональные данные, такие как имя, адрес электронной почты, и пароль.
- Нажимает кнопку "Зарегистрироваться".
- Система проверяет валидность введенных данных.
- В случае успешной валидации, система создает новую учетную запись пользователя.
- Пользователю отправляется подтверждение регистрации на указанный адрес электронной почты.

2.4.2 Вариант использования «Регистрация компании»

Заинтересованные лица и их требования: Представитель компании, которая не имеет учетной записи в системе, желает зарегистрировать компанию для использования функционала системы.

Предусловие: Представитель компании не авторизован в системе.

Постусловие: В системе создана учетная запись компании, и ей назначен уникальный идентификатор.

Основной успешный сценарий:

- Представитель компании открывает страницу регистрации компании.
- Вводит данные о компании, такие как название, цвет, расположение и количество сотрудников.
- Нажимает кнопку "Зарегистрировать компанию".
- Система проверяет валидность введенных данных.
- В случае успешной валидации, система создает новую учетную запись компании и присваивает ей уникальный идентификатор.
- Представителю компании отправляется подтверждение регистрации компании.

2.4.3 Вариант использования «Добавление помещения»

Заинтересованные лица и их требования: Пользователь (администратор), который авторизован в системе и желает добавить информацию о новом помещении.

Предусловие: Пользователь авторизован в системе и имеет доступ к функционалу управления помещениями.

Постусловие: В системе добавлена информация о новом помещении.

Основной успешный сценарий:

- Пользователь переходит на страницу управления помещениями.
- Нажимает кнопку "Добавить помещение".
- Заполняет необходимые атрибуты помещения, такие как номер, площадь, этаж и прочие.
- Нажимает кнопку "Сохранить".
- Система проверяет валидность введенных данных.
- В случае успешной валидации, система добавляет новую информацию о помещении.

2.4.4 Вариант использования «Просмотр статистики по компании»

Заинтересованные лица и их требования: Пользователь (администратор, представитель компании), который авторизован в системе и желает получить статистическую информацию по энергопотреблению компании.

Предусловие: Пользователь авторизован в системе и имеет доступ к функционалу просмотра статистики.

Постусловие: Пользователь видит статистическую информацию по энергопотреблению компании.

Основной успешный сценарий:

- Пользователь переходит на страницу статистики компании.
- Выбирает период, за который требуется получить статистику (например, за последний месяц).
- Система загружает и отображает данные о потреблении энергии компанией за выбранный период.
- Пользователь видит графики и диаграммы, отражающие энергопотребление, расходы и другие показатели.

2.4.5 Вариант использования «Просмотр статистики помещения»

Заинтересованные лица и их требования: Пользователь (администратор, представитель компании), который авторизован в системе и желает получить статистическую информацию по энергопотреблению определенного помещения.

Предусловие: Пользователь авторизован в системе, имеет доступ к функционалу просмотра статистики помещений и выбрал конкретное помещение.

Постусловие: Пользователь видит статистическую информацию по энергопотреблению выбранного помещения.

Основной успешный сценарий:

- Пользователь переходит на страницу статистики помещения.

- Выбирает период, за который требуется получить статистику (например, за последний месяц).
- Система загружает и отображает данные о потреблении энергии выбранным помещением за выбранный период.
- Пользователь видит графики и диаграммы, отражающие энергопотребление, расходы и другие показатели для данного помещения.

2.4.6 Вариант использования «Добавление устройства»

Заинтересованные лица и их требования: Пользователь (администратор), который авторизован в системе и желает добавить информацию об устройстве, подключенном к определенному помещению.

Предусловие: Пользователь авторизован в системе и имеет доступ к функционалу управления устройствами помещения.

Постусловие: Информация об устройстве добавлена в систему для выбранного помещения.

Основной успешный сценарий:

- Пользователь переходит на страницу управления устройствами для конкретного помещения.
- Выбирает опцию "Добавить устройство".
- Вводит информацию о новом устройстве, такую как название, тип, идентификатор, и другие характеристики.
- Нажимает кнопку "Добавить".
- Система проверяет валидность введенных данных.
- В случае успешной валидации, информация об устройстве добавляется к данным помещения.
- Пользователь видит обновленный список устройств для выбранного помещения.

2.5 Требования к оформлению документации

Разработка программной документации и программного изделия должна производиться согласно ГОСТ 19.102-77 и ГОСТ 34.601-90. Единая система программной документации.

3 Технический проект

3.1 Общая характеристика организации решения задачи

Для эффективного управления энергопотреблением в зданиях предлагается разработать комплексную программно-информационную систему, в основе которой лежат передовые технологии. Создаваемая система будет включать в себя несколько ключевых модулей, в том числе модули мониторинга, управления, аналитики и интеграции с внешними устройствами.

Модуль мониторинга будет ответственен за сбор и анализ данных об энергопотреблении в режиме реального времени. Это позволит системе непрерывно отслеживать изменения в расходе энергии и выявлять потенциальные области оптимизации. Модуль управления предоставит пользователям средства активного воздействия на энергопотребление, позволяя оптимизировать его в соответствии с текущими потребностями и требованиями.

Особое внимание будет уделено модулю аналитики, который обеспечит глубокий анализ данных и выделение основных трендов в потреблении энергии. Это позволит пользователям принимать информированные решения для дальнейшей оптимизации и снижения затрат. Система также будет способна интегрироваться с различными внешними устройствами и системами, обеспечивая максимальную гибкость и совместимость.

Проектная цель заключается в предоставлении пользователям возможности не только наблюдать за энергопотреблением, но и активно вмешиваться в процессы управления, с целью обеспечения оптимальной эффективности и экономии ресурсов. Разрабатываемая система ставит своей задачей создание интеллектуального и адаптивного подхода к управлению энергопотреблением, способного эффективно реагировать на изменяющиеся условия и потребности.

3.2 Обоснование выбора технологии проектирования

Выбор языка программирования и фреймворка - критический этап в проектировании программно-информационных систем. Для разработки си-

стемы управления энергопотреблением в зданиях был принят решительный выбор в пользу языка программирования Python с использованием фреймворка Django. Этот выбор обоснован рядом значимых преимуществ, которые существенно повышают эффективность и надежность проекта.

В первую очередь, Python предоставляет высокую степень читаемости кода и простоты синтаксиса, что облегчает понимание и совместную работу над проектом. Это особенно важно в контексте разработки программных продуктов, где четкость кода и удобство его поддержки играют ключевую роль.

Фреймворк Django, в свою очередь, предоставляет набор готовых инструментов и абстракций, спроектированных для ускорения процесса разработки. Модульность, встроенная система аутентификации, ORM (Object-Relational Mapping) для удобной работы с базой данных – все это делает Django мощным инструментом для создания сложных и надежных веб-приложений.

Дополнительно, обширное сообщество разработчиков Python и Django предоставляет широкий спектр готовых решений, библиотек и обновлений, что обеспечивает высокую поддержку и стабильность проекта на протяжении его жизненного цикла.

Такой сбалансированный выбор технологий, основанный на надежности, удобстве разработки и поддержке, позволяет ожидать успешную реализацию программно-информационной системы и обеспечивает ее легкость в последующем масштабировании и совершенствовании.

3.2.1 Описание используемых технологий и языков программирования

Язык программирования Python является краеугольным камнем нашего проекта, обусловленного несколькими ключевыми характеристиками. Прежде всего, простота синтаксиса Python и его высокая читаемость способствуют более эффективной разработке и сопровождению кода. Это особенно важно

для проектов, направленных на веб-разработку, где четкость и легкость восприятия кода имеют первостепенное значение.

Выбор фреймворка Django обусловлен его мощью и удобством использования. Django предоставляет разработчикам набор инструментов, включая систему управления базой данных, встроенные механизмы аутентификации и авторизации, а также мощный движок для обработки URL-запросов. Это позволяет сфокусироваться на бизнес-логике приложения, ускоряя процесс разработки и повышая его надежность.

Одним из важных компонентов выбора технологии стало применение Object-Relational Mapping (ORM) в Django. ORM обеспечивает абстракцию базы данных от языка SQL, что упрощает взаимодействие с данными и позволяет разработчикам оперировать объектами в коде, а не прямо с SQL-запросами. Это не только уменьшает вероятность ошибок, связанных с базой данных, но и делает код более чистым и легко поддерживаемым.

Таким образом, выбор Python и Django, а также использование ORM, формируют сбалансированный стек технологий, обеспечивая эффективную и устойчивую основу для разработки программно-информационной системы управления энергопотреблением в зданиях.

3.2.2 Язык программирования Python Django

Выбор языка программирования Python с применением фреймворка Django оказался стратегически обоснованным для разработки программно-информационной системы по управлению энергопотреблением в зданиях. Этот выбор основывается на ряде ключевых преимуществ, которые содействуют эффективной и надежной реализации поставленных задач.

1. Простота и Читаемость Кода: Python славится своей простотой синтаксиса, что ускоряет процесс разработки и облегчает понимание кода. Это особенно ценно для командной работы и последующего сопровождения программы.

2. Широкое Применение в Веб-Разработке: Python является одним из наиболее популярных языков программирования в области веб-разработки.

Его использование с фреймворком Django обеспечивает высокую производительность и удобство для создания сложных веб-приложений.

3. Гибкость и Мощь Django: Django предоставляет разработчикам удобные инструменты, встроенные решения и структуру проекта, что упрощает создание масштабируемых и надежных приложений. Фреймворк включает механизмы для работы с базой данных, управления пользователями, обработки форм и другие полезные функции.

4. Активное Сообщество и Поддержка: Используя Python с Django, проект получает поддержку от активного сообщества разработчиков. Это обеспечивает доступ к обширным ресурсам, библиотекам и решениям, способствуя более быстрой и безопасной разработке.

5. Объектно-Реляционное Отображение (ORM): Применение ORM в Django упрощает взаимодействие с базой данных, предоставляя абстракцию от языка SQL. Это повышает безопасность и чистоту кода, делая его более поддерживаемым.

Общий результат - это мощный и гибкий инструментарий, способствующий созданию высокоэффективной программно-информационной системы, специально нацеленной на управление энергопотреблением в зданиях.

3.3 Диаграмма компонентов

Диаграмма компонентов – это визуальное представление структуры программной системы, выделяя её основные компоненты и взаимосвязи между ними. Она служит инструментом для понимания архитектуры приложения.

Схема обмена данными дополняет диаграмму компонентов, демонстрируя, как информация передается между различными компонентами системы. Это важный элемент, который помогает лучше понять взаимодействие между частями приложения.

На рисунке 3.1 представлена диаграмма компонентов, на которой наглядно изображены основные элементы программной системы и их взаимосвязи.

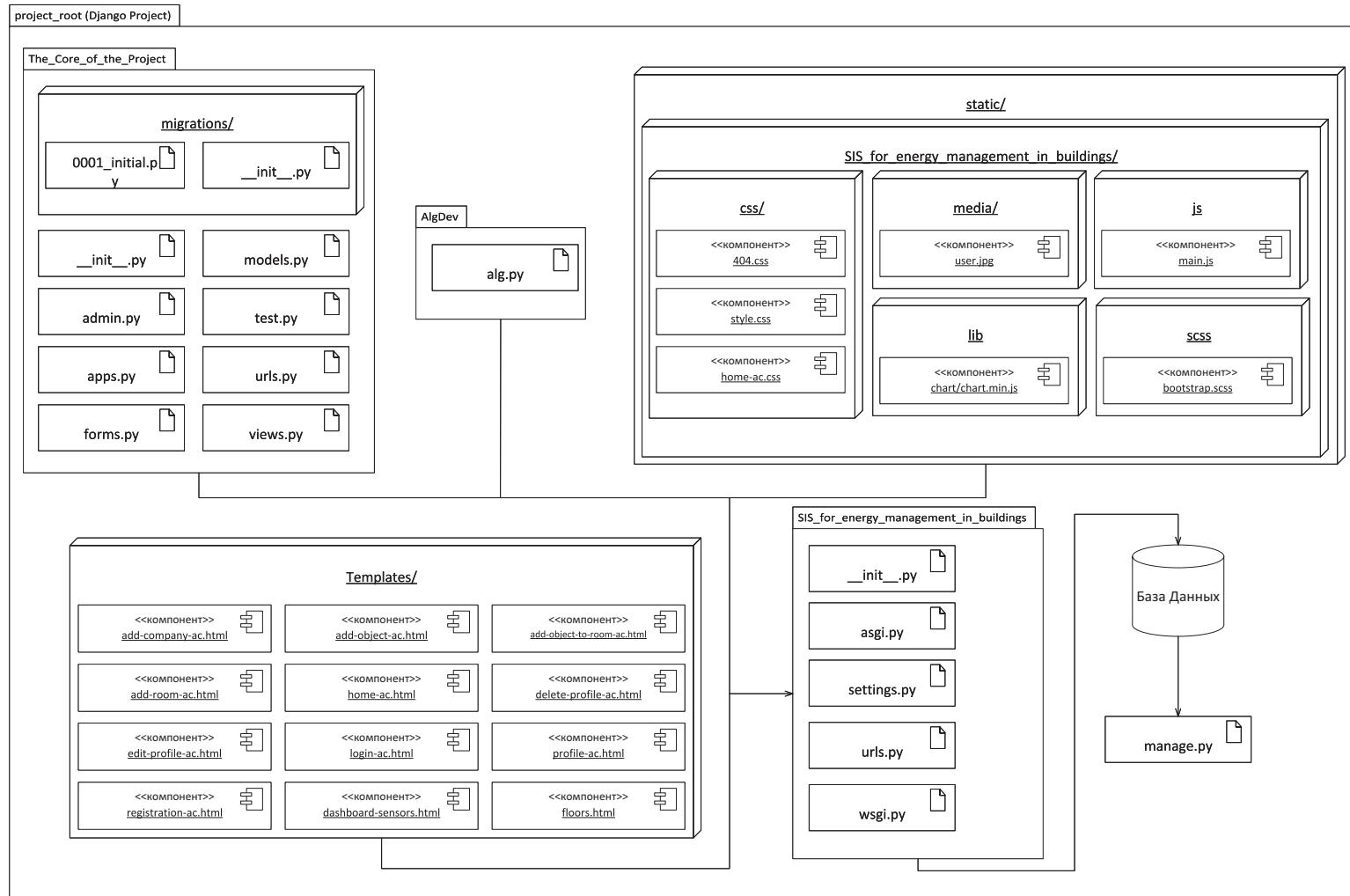


Рисунок 3.1 – Диаграмма компонентов

Эта визуализация позволяет разработчикам и другим участникам проекта легче ориентироваться в структуре приложения и эффективнее взаимодействовать при работе над проектом.

Каталоги:

1. /templates/:
 - Описание: Содержит файлы шаблонов HTML для веб-интерфейса.
2. /static/:
 - Описание: Хранит статические файлы, такие как CSS, JavaScript, изображения.

Компоненты:

1. Web Interface (Веб-интерфейс)
 - Описание: Компонент, предоставляющий пользовательский интерфейс через веб-браузер. Файлы: templates/, static/
2. Backend Server (Серверное приложение)
 - Описание: Компонент, обрабатывающий логику приложения и предоставляющий API для взаимодействия с фронтендом.
 - Файлы: views.py, models.py, urls.py
3. Database (База данных)
 - Описание: Компонент, отвечающий за хранение данных.
 - Файлы: Миграции Django, models.py
4. Settings (Настройки)
 - Описание: Компонент, содержащий настройки Django, такие как подключение к базе данных, ключи и другие конфигурационные параметры.
 - Файлы: settings.py
5. Django ORM (Объектно-реляционное отображение)
 - Описание: Компонент, предоставляющий интерфейс для взаимодействия с базой данных через Django.
 - Файлы: Встроенные Django-модели
6. Data Reading Algorithm (Алгоритм считывания данных)
 - Описание: Компонент, включающий алгоритмы считывания данных с устройств, таких как сенсоры, датчики влажности, температуры и другие.

Отвечает за обработку данных, полученных от физических устройств, и передачу их в систему для дальнейшей обработки.

- Файлы: alg.py

3.3.1 Описание настройки Django

Настройка Django в проекте осуществляется через два ключевых файла: settings.py и urls.py. В этих файлах определена конфигурация приложения, включая параметры базы данных, пути к шаблонам, обработку статических файлов и другие настройки. В settings.py размещаются параметры, определяющие общую конфигурацию проекта, такие как настройки базы данных, приложения, подключение статических и медиа файлов, а также многое другое. С другой стороны, в urls.py определяются пути маршрутизации, которые связывают URL-адреса с соответствующими представлениями и функциональностью приложения. Эти два файла играют ключевую роль в формировании и управлении всеми аспектами функционирования Django-приложения, обеспечивая его корректное взаимодействие с веб-сервером и клиентскими запросами.

3.3.1.1 База данных

Один из важных аспектов Django-проекта – это выбор и настройка системы управления базами данных (СУБД). В данном случае, в качестве СУБД используется PostgreSQL, предоставляющая надежное и мощное хранилище данных.

Для интеграции PostgreSQL в Django проект, первоначально необходимо установить соответствующий пакет драйвера, в данном случае, psycopg2. Этот драйвер обеспечивает связь между Django и PostgreSQL, обеспечивая эффективное взаимодействие с базой данных.

После установки драйвера, производится конфигурация подключения к базе данных в файле settings.py. Здесь определяются параметры, такие как имя базы данных, пользователь, пароль, хост и порт, необходимые для успешного соединения с PostgreSQL. Это обеспечивает правильную интегра-

цию между Django приложением и выбранной базой данных, что является фундаментом для эффективного хранения и обработки данных в рамках проекта. Пример настройки:

```
1 DATABASES = {  
2     'default': {  
3         'ENGINE': 'django.db.backends.postgresql',  
4         'NAME': 'mydatabase',  
5         'USER': 'mcherenkov',  
6         'PASSWORD': '*2nm(h(tIs2u4AJ#',  
7         'HOST': 'localhost',  
8         'PORT': '5432',  
9     }  
10 }
```

Эти параметры включают в себя имя базы данных, пользователя, пароль, хост и порт для подключения к PostgreSQL.

3.3.1.2 Статические файлы и медиа

Настройки для обработки статических файлов и медиа файлов (например, изображений, загружаемых пользователями) также определены в settings.py. Важным аспектом конфигурации Django-проекта является определение параметров обработки статических и медиа файлов. Эти файлы могут включать в себя стили, скрипты, изображения, а также медиа-контент, загружаемый пользователями.

Такие настройки позволяют эффективно управлять статическими и медиа-ресурсами в Django-проекте, обеспечивая их доступность и корректное отображение на веб-страницах. Пример настроек для статических файлов:

```
1 STATIC_URL = '/static/'  
2 STATICFILES_DIRS = [BASE_DIR / "static"]  
3  
4 MEDIA_URL = '/media/'  
5 MEDIA_ROOT = BASE_DIR / "media"
```

Здесь STATIC-URL указывает префикс для статических файлов, а STATICFILES-DIRS - на каталог, где хранятся эти файлы.

MEDIA-URL и MEDIA-ROOT аналогичны, но используются для обработки медиа файлов.

3.3.1.3 Настройки безопасности

Django также предоставляет настройки безопасности, такие как секретный ключ, список разрешенных хостов (ALLOWED-HOSTS), и другие параметры. Пример:

```
1 SECRET_KEY = 'django-insecure-b)te6480y4j&yunr@zy6v#$2u(r@06q3kb$m@=d1n!_8t&(  
2 DEBUG = False  
3 ALLOWED_HOSTS = [ 'mcherenkov.com' ]
```

Секретный ключ должен быть уникальным и долгим случайным значением. Включение параметра DEBUG в режиме разработки может быть удобным для поиска ошибок, но необходимо отключить его в продакшн окружении. Список разрешенных хостов (ALLOWED-HOSTS) определяет, какие хосты могут обращаться к вашему серверу.

3.4 Диаграмма размещения

При проектировании диаграммы размещения необходимо тщательно рассмотреть взаимодействие архитектурных компонентов программной системы и их распределение по физическим устройствам. Эффективное размещение данных компонентов определяет стабильность и производительность системы. Приведенный ниже пример диаграммы размещения является иллюстрацией общей структуры "Программно-информационной системы для управления энергопотреблением в зданиях включающей в себя ключевые элементы, начиная от клиентских устройств и заканчивая устройствами управления энергопотреблением в физических зданиях.

Серверная инфраструктура, показывает сервера, на которых размещаются серверные компоненты системы, такие как веб-сервер, базы данных и другие серверные приложения.

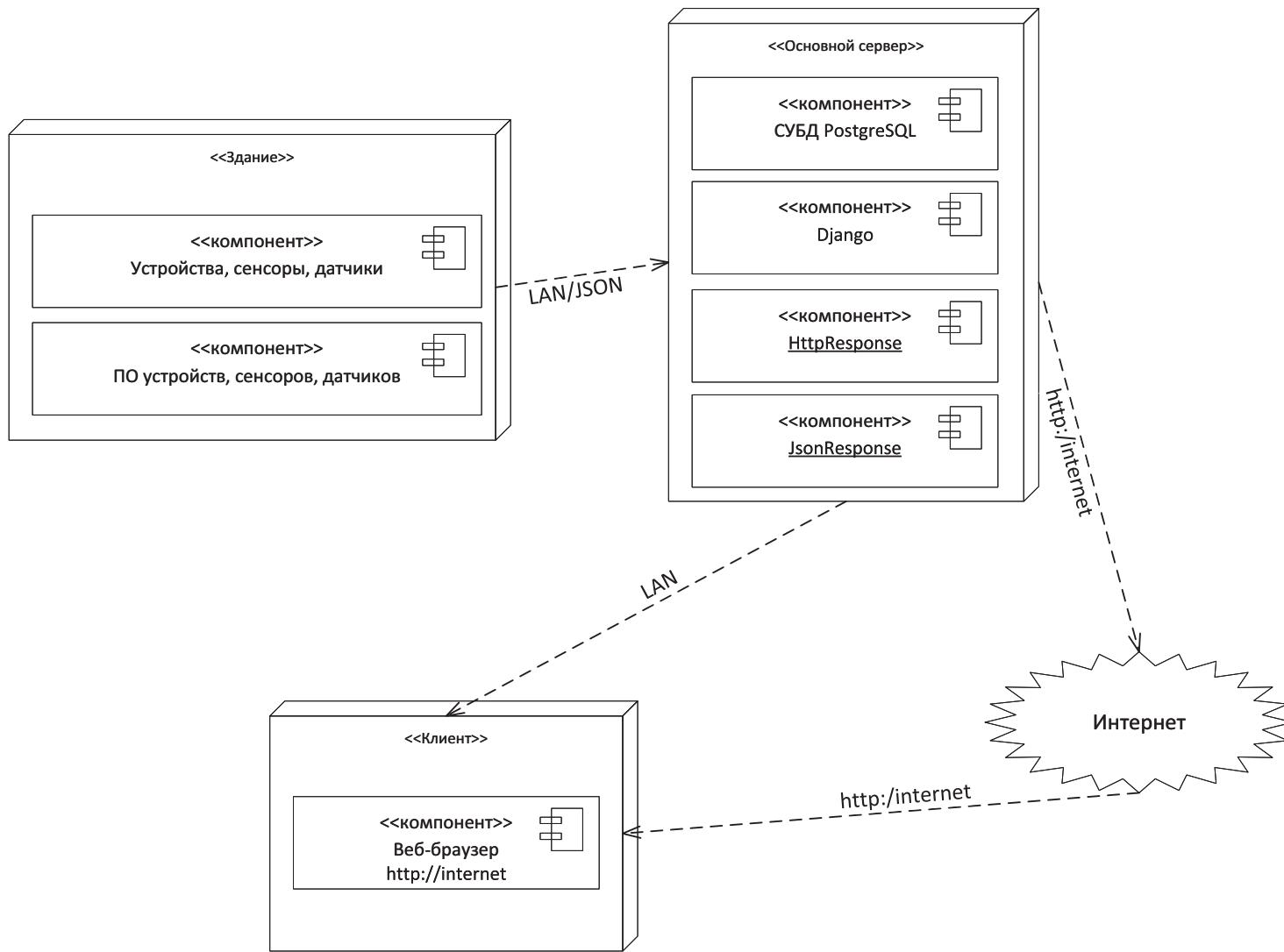


Рисунок 3.2 – Диаграмма размещения

1. Сервер:

- Описание: Физический или виртуальный сервер, на котором размещена программно-информационная система. Сервер выполняет обработку запросов, взаимодействует с базой данных, и управляет основной логикой системы.
- Размещение: Сервер расположен в data-центре компании и подключен к высокоскоростной сети интернета.
- Компоненты: Django приложение, бизнес-логика, веб-сервер.

2. База данных:

- Описание: Хранилище данных, где сохраняются все необходимые для системы информационные записи. В данной системе используется реляционная база данных для хранения данных о компаниях, потреблении энергии, настройках и других сущностях.

- Размещение: Физически база данных размещена на сервере в data-центре, обеспечивая быстрый доступ к данным.

- Компоненты: PostgreSQL (или другая СУБД), таблицы с данными.

3. Устройства сбора данных:

- Описание: Физические устройства, установленные в зданиях, ответственные за сбор данных о потреблении энергии и внесение корректива в управление энергопотреблением в реальном времени. Компоненты включают в себя сенсоры, датчики влажности, температуры, освещенности и другие.

- Размещение: Сенсоры и датчики подключены к локальной сети в каждом здании компании, обеспечивая сбор данных в реальном времени.

- Компоненты: Сенсоры, датчики, средства сбора и передачи данных.

- Сеть: Устройства сбора данных подключены к локальной сети здания, обеспечивая передачу данных на сервер через сетевой протокол (например, TCP/IP). Данные могут быть переданы в формате JSON для удобства обработки на сервере.

4. Пользовательские устройства():

- Описание: Пользовательские интерфейсы представлены на клиентских устройствах, таких как компьютеры, планшеты или мобильные устройства.

ства. Здесь пользователи могут взаимодействовать с системой, вводя данные или получая информацию.

- Размещение: Устройства клиентов находятся в офисах компаний, а также могут использоваться удаленные рабочие места.
- Компоненты: Браузеры (Google Chrome, Mozilla Firefox, Safari и др.), интерфейс пользователя.

5. Сетевые соединения, стрелки, обозначающие сетевые соединения между устройствами, чтобы показать, как данные передаются между ними.

Диаграмма размещения демонстрирует (рис. 3.2) , как каждый компонент системы взаимодействует друг с другом и как данные передаются между устройствами.

3.5 Диаграмма классов

Диаграмма классов представляет собой визуализацию структуры классов программной системы и их взаимосвязей. Ниже приведено описание основных классов системы ”Программно-информационная система для управления энергопотреблением в зданиях”.

Классы, представление классов с их атрибутами и методами:

1. Класс «EnergyConsumption»: Модель для учета потребленной энергии.

- Атрибуты: company, room, floor, date, consumption_without_an_assistant, consumption_with_an_assistant, total_amount_of_electricity_consumed_without_an_assistant, total_amount_of_electricity_consumed_with_the_assistant, total_cost_without_an_assistant, total_cost_with_an_assistant, humidity, temperature, illumination, motion.

- Методы: `__str__`.

2. Класс «InventoryItemsNumber»: Модель для представления предметов в помещении.

- Атрибуты: name, room.
- Методы: `__str__`.

3. Класс «Company»: Модель для представления компаний.

- Атрибуты: name, color, object, number-of-employees.
- Методы: `__str__`.

4. Класс «Floor»: Модель для представления этажей.

- Атрибуты: number.
- Методы: `__str__`.

5. Класс «Electricity»: Модель для представления электроэнергии.

- Атрибуты: price, volume.
- Методы: `__str__`.

Объяснение основных классов:

- EnergyConsumption: Этот класс предназначен для учета и хранения данных о потреблении энергии в зданиях. Атрибуты класса включают в себя информацию о компании, помещении, этаже, дате, количестве потребленной энергии с и без использования системы управления, общей стоимости, а также дополнительные параметры, такие как влажность, температура, освещенность и обнаружение движения.

- Класс «InventoryItemsNumber»: Данный класс предназначен для представления предметов, находящихся в конкретном помещении. Атрибуты класса включают в себя название предмета и ссылку на помещение, где он находится.

- Класс «Company»: Этот класс используется для представления компаний, использующих систему управления энергопотреблением. Атрибуты класса включают в себя название компании, цвет для визуализации, тип объекта компании и количество сотрудников.

- Класс «Floor»: Класс предназначен для хранения информации об этажах здания. Атрибуты класса включают в себя номер этажа.

- Класс «Electricity»: Данный класс используется для представления электроэнергии, собираемой и потребляемой системой. Атрибуты класса включают в себя цену за единицу электроэнергии и объем потребленной энергии.

Диаграмма классов представлена на рисунке 3.3, представляющее основные модели данных в вашем проекте на Django.

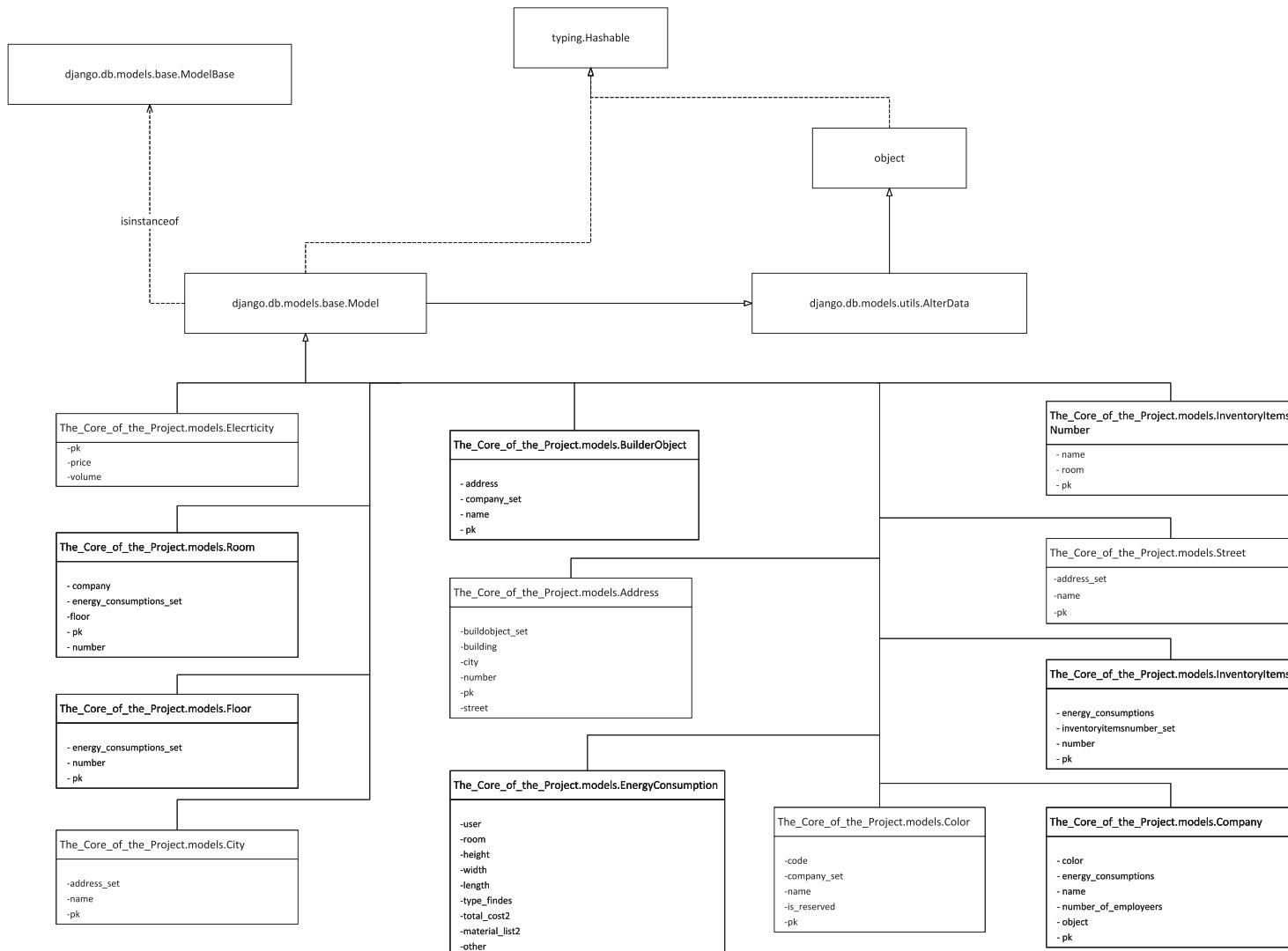


Рисунок 3.3 – Диаграмма классов

3.6 Содержание информационных блоков. Основные сущности

Проанализировав требования, можно выделить несколько основных сущностей:

- «Потребление энергии»;
- «Помещения»;
- «Компании»;
- «Этажи»;
- «Предметы инвентаризации»;
- «Электроэнергия».

Сущность, отражающая данные о потреблении энергии компаниями на различных этажах и в разных помещениях. В состав сущности «Потребление энергии» можно включить атрибуты, представленные в таблице 3.1.

Таблица 3.1 – Атрибуты сущности «Потребление энергии»

Поле	Тип	Обязательное	Описание
company	ForeignKey(Company)	Да	Компания, потребляющая энергию.
room	ForeignKey(Room)	Нет	Помещение, в котором измеряется потребление.
floor	ForeignKey(Floor)	Нет	Этаж, на котором расположено помещение.
date	DateField	Да	Дата, когда было измерено потребление.

Продолжение таблицы 3.1

Поле	Тип	Обязательное	Описание
consumption-without-an-assistant	JSONField	Да	Потребление электроэнергии без использования умного помощника.
consumption-with-an-assistant	JSONField	Да	Потребление электроэнергии с использованием умного помощника.
total-amount-without-an-assistant	FloatField	Да	Общее количество потребляемой электричества без использования помощника.
total-amount-with-the-assistant	FloatField	Да	Общее количество электричества, потребляемого с использованием помощника.
total-cost-without-an-assistant	JSONField	Да	Общая стоимость потребления без использования помощника.
total-cost-with-an-assistant	JSONField	Да	Общая стоимость потребления с использованием помощника.

Продолжение таблицы 3.1

Поле	Тип	Обязательное	Описание
humidity	JSONField	Да	Значение влажности.
temperature	JSONField	Да	Значение температуры.
illumination	JSONField	Да	Значение освещенности.
motion	JSONField	Да	Наличие движения.

Сущность, описывающая различные помещения на этажах компаний. В состав сущности «Помещения» можно включить атрибуты, представленные в таблице 3.3.

Таблица 3.3 – Атрибуты сущности «Помещения»

Поле	Тип	Обязательное	Описание
name	CharField	Да	Название помещения.
floor	ForeignKey(Floor)	Да	Этаж, на котором расположено помещение.

Сущность, представляющая информацию о компаниях, потребляющих энергию. В состав сущности «Компании» можно включить атрибуты, представленные в таблице 3.5.

Таблица 3.5 – Атрибуты сущности «Компании»

Поле	Тип	Обязательное	Описание
name	CharField	Да	Название компании.
color	ForeignKey(Color)	Нет	Цвет, ассоциированный с компанией.
object	ForeignKey(BuilderObject)	Да	Расположение компании.
number-of-employees	CharField	Да	Количество сотрудников компании.

Сущность, описывающая различные этажи в здании. В состав сущности «Этажи» можно включить атрибуты, представленные в таблице 3.7.

Таблица 3.7 – Атрибуты сущности «Этажи»

Поле	Тип	Обязательное	Описание
number	CharField	Да	Номер этажа.

Сущность, отражающая предметы в помещении. В состав сущности «Предметы инвентаризации» можно включить атрибуты, представленные в таблице 3.9.

Таблица 3.9 – Атрибуты сущности «Предметы инвентаризации»

Поле	Тип	Обязательное	Описание
name	ForeignKey(InventoryItems)	Да	Предмет инвентаризации.
room	ForeignKey(Room)	Да	Помещение, в котором размещается предмет.

Эти сущности предоставляют функционал для отслеживания и анализа потребления энергии, структуры помещений, информации о компаниях, этажах, предметах инвентаризации и стоимости электроэнергии.

4 Рабочий проект

4.1 Классы, используемые при разработке системы управления энергопотреблением

В процессе разработки системы управления энергопотреблением были использованы различные классы для обеспечения функциональности и взаимодействия компонентов. Ниже представлен список ключевых классов и их методов, применяемых в разрабатываемой программной системе (таблица 4.1).

Таблица 4.1 – Описание классов системы управления энергопотреблением

Название класса	Модуль, к которому относится класс	Описание класса	Методы
Energy Consumption	Models	Класс, отвечающий за учет и хранение данных о потреблении энергии в зданиях. Атрибуты класса включают в себя информацию о компании, помещении, этаже, дате, количестве потребленной энергии с и без использования системы управления, общей стоимости, а также дополнительные параметры, такие как влажность, температура, освещенность и обнаружение движения.	Метод <code>__str__</code> обеспечивает читаемое представление объекта этого класса.

Продолжение таблицы 4.1

Название класса	Модуль, к которому относится класс	Описание класса	Методы
Inventory Items Number	Models	Класс, предназначенный для представления предметов, находящихся в конкретном помещении. Атрибуты класса включают в себя название предмета и ссылку на помещение, где он находится.	Метод <code>__str__</code> обеспечивает читаемое представление объекта этого класса.
Company	Models	Класс, используемый для представления компаний, использующих систему управления энергопотреблением. Атрибуты класса включают в себя название компании, цвет для визуализации, тип объекта компании и количество сотрудников.	Метод <code>__str__</code> обеспечивает читаемое представление объекта этого класса.
Floor	Models	Класс, предназначенный для представления этажей.	Метод <code>__str__</code> обеспечивает читаемое представление объекта этого класса.
Electricity	Models	Класс, используемый для представления электроэнергии, собираемой и потребляемой системой. Атрибуты класса включают в себя цену за единицу электроэнергии и объем потребленной энергии.	Метод <code>__str__</code> обеспечивает читаемое представление объекта этого класса.

Продолжение таблицы 4.1

Название класса	Модуль, к которому относится класс	Описание класса	Методы
Inventory Items	Models	Класс, представляющий предметы инвентаря. Атрибуты включают в себя наименование предмета и потребление энергии в кВт·ч.	Метод <code>__str__</code> обеспечивает читаемое представление объекта этого класса.
Address	Models	Класс, представляющий адрес объекта. Атрибуты включают в себя город, улицу, номер дома и строение.	Метод <code>__str__</code> обеспечивает читаемое представление объекта этого класса.
Street	Models	Класс, представляющий улицы. Атрибут включает в себя название улицы.	Метод <code>__str__</code> обеспечивает читаемое представление объекта этого класса.
City	Models	Класс, представляющий города. Атрибут включает в себя название города.	Метод <code>__str__</code> обеспечивает читаемое представление объекта этого класса.

4.2 Модульное тестирование разработанной системы управления энергопотреблением

Для обеспечения надежности и корректности работы класса `EnergyConsumption` в приложении, был разработан и реализован модульный тест. Этот тест написан с использованием фреймворка тестирования `Django` и охватывает различные аспекты функционала класса.

Описание теста:

1. Настройка среды тестирования: - Создание тестового объекта `EnergyConsumption`. - Установка значений полей объекта для проведения тестов.

2. Проверка метода `__str__`: - Вызов метода `__str__` для объекта `EnergyConsumption`. - Сравнение возвращаемой строки с ожидаемым результатом. - Ожидается, что строка содержит информацию о компании, комнате и дате.

3. Ожидаемые результаты: - Успешное создание и сохранение объекта `EnergyConsumption`. - Корректное функционирование метода `__str__` с возвращением ожидаемой строки. - Этот модульный тест помогает обеспечить стабильность и правильную работу класса `EnergyConsumption`, а также может быть использован при внесении изменений в код для подтверждения его надежности.

Модульное тестирование класса `EnergyConsumption` представлено на рисунке 4.1.

```
1  from django.test import TestCase
2  from .models import EnergyConsumption
3
4  class EnergyConsumptionTestCase(TestCase):
5      def setUp(self):
6          self.energy_consumption = EnergyConsumption(
7              company_name='Test Company',
8              room='Test Room',
9              floor='Test Floor',
10             date='2023-01-01',
11             consumption_without_an_assistant={'value': 100, 'unit': 'kWh'},
12             consumption_with_an_assistant={'value': 80, 'unit': 'kWh'},
13             total_amount_of_electricity_consumed_without_an_assistant=100,
14             total_amount_of_electricity_consumed_with_the_assistant=80,
15             total_cost_without_an_assistant={'value': 100, 'currency': 'USD'},
16             total_cost_with_an_assistant={'value': 80, 'currency': 'USD'},
17             humidity={'value': 50, 'unit': '%'},
18             temperature={'value': 25, 'unit': 'C'},
19             illumination={'value': 500, 'unit': 'lux'},
20             motion={'value': True}
21         )
22         self.energy_consumption.save()
23
24     def test_energy_consumption_str_method(self):
25         # Проверяем, что метод __str__ возвращает ожидаемую строку
26         expected_str = 'Компания: Test Company|Test Room, Дата: 2023-01-01'
27         self.assertEqual(str(self.energy_consumption), expected_str)
```

Рисунок 4.1 – Модульный тест класса `EnergyConsumption`

Для обеспечения корректной работы и надежности функционала класса Company, был разработан модульный тест, использующий фреймворк тестирования Django.

Описание теста:

1. Настройка среды тестирования: - Создание тестового объекта Company. - Установка значений полей объекта для проведения тестов.

2. Проверка метода `__str__`: - Вызов метода `__str__` для объекта Company. - Сравнение возвращаемой строки с ожидаемым результатом. - Ожидается, что строка содержит информацию о названии компании.

3. Проверка метода `save`: - Сохранение объекта Company в базе данных. - Попытка извлечения сохраненного объекта из базы данных. - Сравнение извлеченного объекта с исходным для подтверждения сохранения в базе.

Ожидаемые результаты: - Успешное создание и сохранение объекта Company. - Корректное функционирование метода `__str__` с возвращением ожидаемой строки. - Успешное сохранение объекта в базе данных и его последующее извлечение.

Модульное тестирование класса Company представлено на рисунке 4.2.

```
1 from django.test import TestCase
2 from .models import Company
3
4 class CompanyTestCase(TestCase):
5
6     def setUp(self):
7         # Создаем тестовый объект Company
8         self.company = Company(
9             name='Test Company',
10            color='Test Color',
11            object='Test Object',
12            number_of_employees='50'
13        )
14        self.company.save()
15
16    def test_company_str_method(self):
17        # Проверяем, что метод __str__ возвращает ожидаемую строку
18        expected_str = 'Test Company'
19        self.assertEqual(str(self.company), expected_str)
```

Рисунок 4.2 – Модульный тест класса EnergyConsumption

Для обеспечения корректной работы и надежности функционала класса InventoryItemsNumber, был разработан модульный тест.

Описание теста:

1. Настройка среды тестирования: - Создание тестового объекта InventoryItemsNumber. - Установка значений полей объекта для проведения тестов.

2. Проверка метода `__str__`: - Вызов метода `__str__` для объекта InventoryItemsNumber. - Сравнение возвращаемой строки с ожидаемым результатом. - Ожидается, что строка содержит информацию о номере инвентаря.

3. Проверка метода `save`: - Сохранение объекта InventoryItemsNumber в базе данных. - Попытка извлечения сохраненного объекта из базы данных. - Сравнение извлеченного объекта с исходным для сохранения в базе.

Ожидаемые результаты: - Успешное создание и сохранение объекта InventoryItemsNumber. - Корректное функционирование метода `__str__` с возвращением ожидаемой строки. - Успешное сохранение объекта в базе данных и его последующее извлечение.

Модульное тестирование класса InventoryItemsNumber представлено на рисунке 4.3.

```
1 from django.test import TestCase
2 from .models import InventoryItemsNumber
3
4 class InventoryItemsNumberTestCase(TestCase):
5     def setUp(self):
6         # Создаем тестовый объект InventoryItemsNumber
7         self.inventory_item_number = InventoryItemsNumber(
8             name='Test Item',
9             room='Test Room'
10        )
11        self.inventory_item_number.save()
12    def test_inventory_item_number_str_method(self):
13        # Проверяем, что метод __str__ возвращает ожидаемую строку
14        expected_str = 'Test Item, Test Room'
15        self.assertEqual(str(self.inventory_item_number), expected_str)
```

Рисунок 4.3 – Модульный тест класса InventoryItemsNumber

Для обеспечения корректной работы и надежности функционала класса Floor, был разработан модульный тест, использующий фреймворк тестирования Django.

Описание теста:

1. Настройка среды тестирования: - Создание тестового объекта Floor.
- Установка значений полей объекта для проведения тестов.
2. Проверка метода `__str__`: - Вызов метода `__str__` для объекта Floor.
- Сравнение возвращаемой строки с ожидаемым результатом. - Ожидается, что строка содержит информацию о номере этажа.
3. Проверка метода `save`: - Сохранение объекта Floor в базе данных. - Попытка извлечения сохраненного объекта из базы данных. - Сравнение извлеченного объекта с исходным для подтверждения сохранения в базе.

Ожидаемые результаты: - Успешное создание и сохранение объекта Floor. - Корректное функционирование метода `__str__` с возвращением ожидаемой строки. - Успешное сохранение объекта в базе данных и его последующее извлечение.

Модульное тестирование класса Floor представлено на рисунке 4.4.

```
1 from django.test import TestCase
2 from .models import Floor
3
4 class FloorTestCase(TestCase):
5
6     def setUp(self):
7         # Создаем тестовый объект Floor
8         self.floor = Floor(
9             number='Test Number'
10        )
11         self.floor.save()
12
13     def test_floor_str_method(self):
14         # Проверяем, что метод __str__ возвращает ожидаемую строку
15         expected_str = 'Этаж Test Number'
16         self.assertEqual(str(self.floor), expected_str)
```

Рисунок 4.4 – Модульный тест класса Floor

Для обеспечения корректной работы и надежности функционала класса Electricity, был разработан модульный тест, использующий фреймворк тестирования Django.

Описание теста:

1. Настройка среды тестирования: - Создание тестового объекта Electricity. - Установка значений полей объекта для проведения тестов.

2. Проверка метода `__str__`: - Вызов метода `__str__` для объекта Electricity. - Сравнение возвращаемой строки с ожидаемым результатом. - Ожидается, что строка содержит информацию о потреблении электроэнергии.

3. Проверка метода `save`: - Сохранение объекта Electricity в базе данных. - Попытка извлечения сохраненного объекта из базы данных. - Сравнение извлеченного объекта с исходным для подтверждения сохранения в базе.

Ожидаемые результаты: - Успешное создание и сохранение объекта Electricity. - Корректное функционирование метода `__str__` с возвращением ожидаемой строки. - Успешное сохранение объекта в базе данных и его последующее извлечение.

Модульное тестирование класса Electricity представлено на рисунке 4.5.

```
1 from django.test import TestCase
2 from .models import Electricity
3
4 class ElectricityTestCase(TestCase):
5
6     def setUp(self):
7         # Создаем тестовый объект Electricity
8         self.electricity = Electricity(
9             price=0.15,
10            volume='100 kWh'
11        )
12        self.electricity.save()
13
14    def test_electricity_str_method(self):
15        # Проверяем, что метод __str__ возвращает ожидаемую строку
16        expected_str = 'Цена за 100 kWh: 0.15 руб.'
17        self.assertEqual(str(self.electricity), expected_str)
```

Рисунок 4.5 – Модульный тест класса Electricity

4.3 Системное тестирование разработанного web-сайта

На рисунках 4.6 - 4.7 представлена главная страница «Программно-информационной системы для управления энергопотреблением в зданиях».

Страница с аналитикой энергопотребления в здании представлена на рисунках

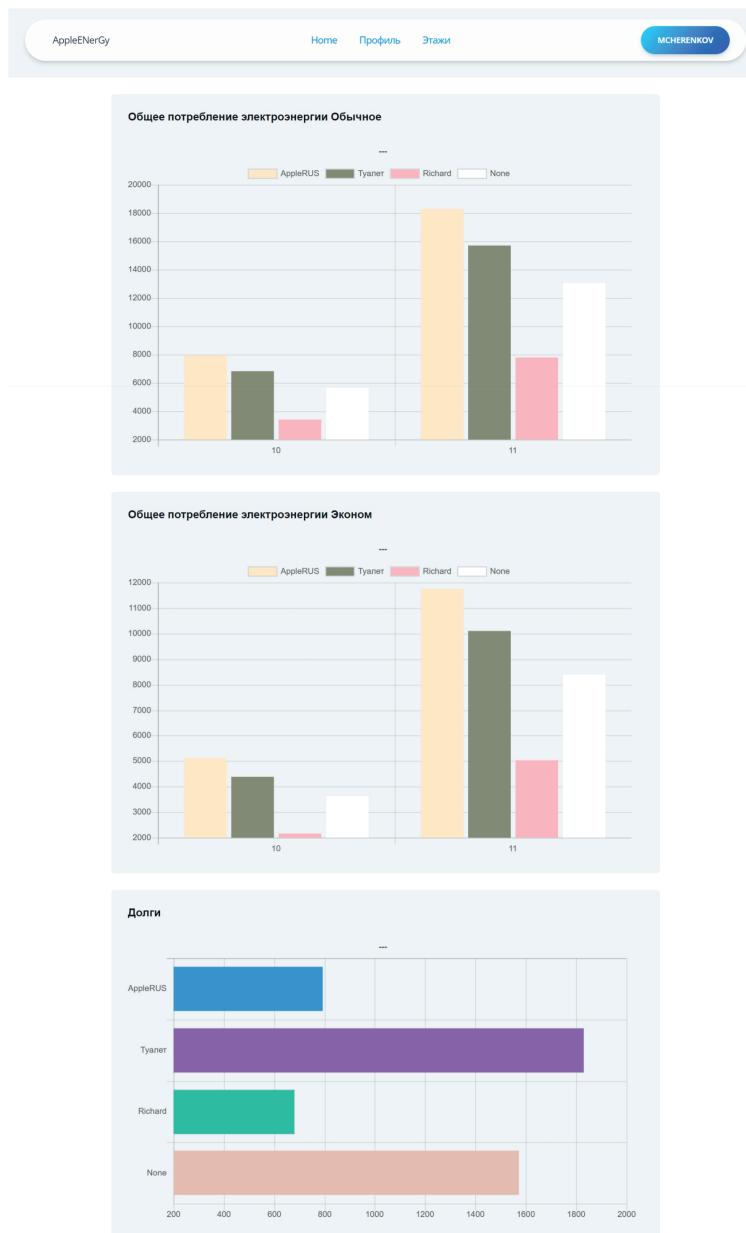


Рисунок 4.6 – главная страница «Программно-информационной системы для управления энергопотреблением в зданиях»

Компании							Добавить
	Наименование	Тип	Сотрудники	Помещения	Потребление О	Потребление Э	
<input type="checkbox"/>	AppleRUS	Очищенный миндаль #FFEBBC	21	301 302 201 202 205 206 207	7954.34 кВт/ч	5104.14 кВ	
<input type="checkbox"/>	Туалет	Бледно-зелено-серый #8D917A	1	303 304 203 204 103 104	6822.490000000001 кВт/ч	4376.99 кВ	
<input type="checkbox"/>	Richard	Розовый #FFC0CB	12	305 306 307	3404.8 кВт/ч	2174.64 кВ	
<input type="checkbox"/>	None	Белый #FFFFFF	0	101 102 105 106 107	5652.719999999999 кВт/ч	3632.92 кВ	

Помещения							Добавить
	Наименование	Компания	Этаж	Потребление О	Потребление Э	Долг	
<input type="checkbox"/>	301	AppleRUS	3	3740.42кВт/ч	2409.86кВт/ч	374.78₽	Удалить
<input type="checkbox"/>	302	AppleRUS	3	3750.96кВт/ч	2413.28кВт/ч	375.32₽	Удалить
<input type="checkbox"/>	303	Туалет	3	3749.19кВт/ч	2409.93кВт/ч	374.79₽	Удалить
<input type="checkbox"/>	304	Туалет	3	3746.53кВт/ч	2424.34кВт/ч	377.04₽	Удалить
<input type="checkbox"/>	305	Richard	3	3757.04кВт/ч	2410.48кВт/ч	374.88₽	Удалить
<input type="checkbox"/>	306	Richard	3	3737.5кВт/ч	2390.22кВт/ч	371.73₽	Удалить
<input type="checkbox"/>	307	Richard	3	3736.25кВт/ч	2404.01кВт/ч	373.87₽	Удалить
<input type="checkbox"/>	201	AppleRUS	2	3746.39кВт/ч	2403.52кВт/ч	373.8₽	Удалить
<input type="checkbox"/>	202	AppleRUS	2	3750.58кВт/ч	2404.97кВт/ч	374.02₽	Удалить
<input type="checkbox"/>	203	Туалет	2	3757.94кВт/ч	2413.08кВт/ч	375.28₽	Удалить
<input type="checkbox"/>	204	Туалет	2	3751.05кВт/ч	2415.52кВт/ч	375.66₽	Удалить
<input type="checkbox"/>	205	AppleRUS	2	3740.31кВт/ч	2400.19кВт/ч	373.28₽	Удалить
<input type="checkbox"/>	206	AppleRUS	2	3767.15кВт/ч	2426.02кВт/ч	377.3₽	Удалить
<input type="checkbox"/>	207	AppleRUS	2	3755.81кВт/ч	2410.2кВт/ч	374.84₽	Удалить
<input type="checkbox"/>	101	None	1	3742.86кВт/ч	2396.56кВт/ч	372.72₽	Удалить
<input type="checkbox"/>	102	None	1	3746.39кВт/ч	2404.84кВт/ч	374.0₽	Удалить
<input type="checkbox"/>	103	Туалет	1	3751.29кВт/ч	2414.06кВт/ч	375.44₽	Удалить
<input type="checkbox"/>	104	Туалет	1	3766.1кВт/ч	2404.55кВт/ч	373.96₽	Удалить
<input type="checkbox"/>	105	None	1	3742.87кВт/ч	2418.82кВт/ч	376.18₽	Удалить
<input type="checkbox"/>	106	None	1	3742.33кВт/ч	2399.67кВт/ч	373.2₽	Удалить
<input type="checkbox"/>	107	None	1	3732.67кВт/ч	2417.63кВт/ч	375.99₽	Удалить

Рисунок 4.7 – главная страница «Программно-информационной системы для управления энергопотреблением в зданиях»

На изображении, представленном на рисунке 4.8, приведен план этажа, на котором наглядно отображено размещение помещений.

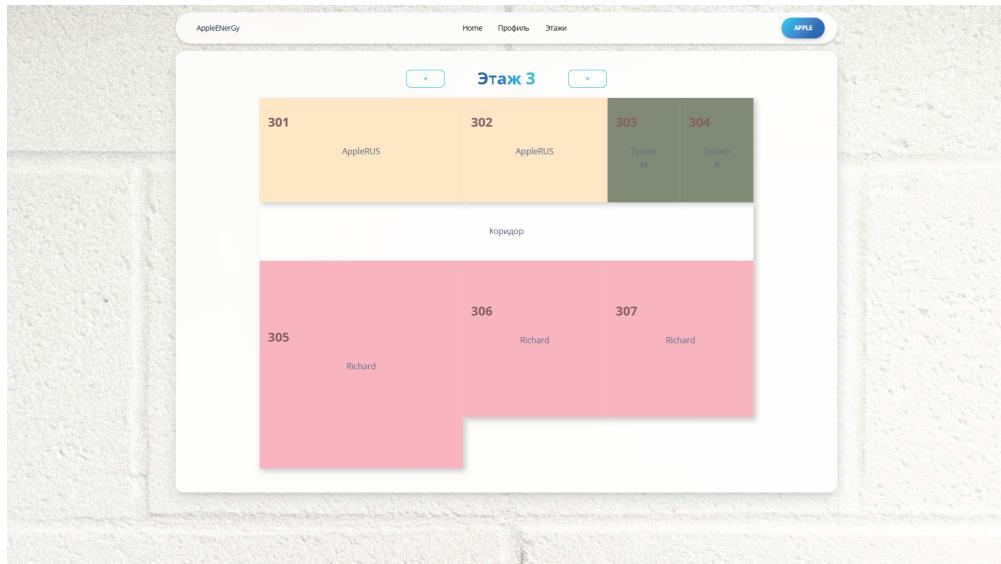


Рисунок 4.8 – План этажа

На рисунке 4.9 изображено помещение компании.

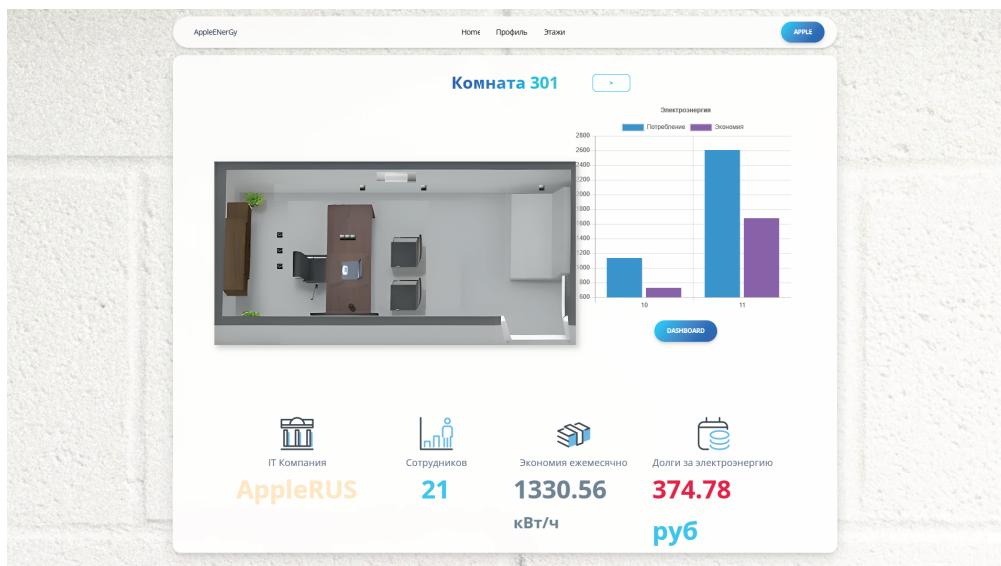


Рисунок 4.9 – Помещение компании

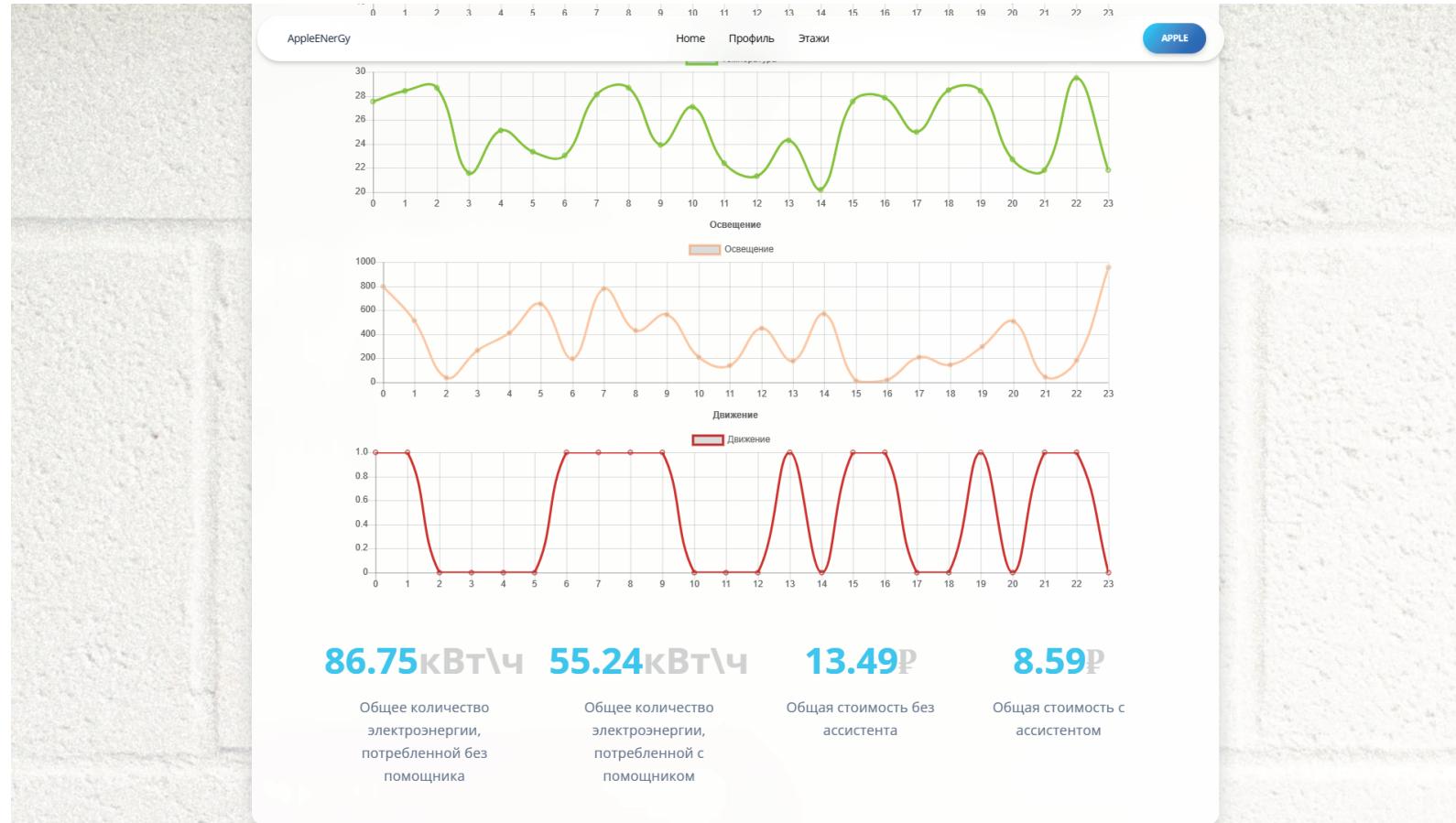
На рисунках 4.3 - 4.3 представлена диаграмма, отражающая потребление электроэнергии, температуру, влажность, освещенность, а также обнаружение движения в помещении компании.

63



Диаграмма, отражающая потребление электроэнергии, влажность в помещении компании.

64



Диаграмма, отражающая температуру, освещенность, а также обнаружение движения в помещении компании.

ЗАКЛЮЧЕНИЕ

В ходе разработки программно-информационной системы для управления энергопотреблением в зданиях была проведена обширная работа по проектированию и реализации различных компонентов системы. Результатом этой работы стала система, способная эффективно контролировать и оптимизировать энергопотребление, повышая эффективность и уменьшая негативное воздействие на окружающую среду.

Процесс проектирования включал в себя создание детальных диаграмм компонентов, классов и размещения, что позволило внедрить модульную структуру и обеспечить легкость поддержки и расширения системы в будущем. Использование языка программирования Python и фреймворка Django обеспечило высокую гибкость в разработке и поддержке системы.

Реализованный функционал системы включает в себя учет и анализ энергопотребления в зданиях, возможность визуализации данных, а также систему управления, способную реагировать на изменения параметров окружающей среды и внутренних условий здания. Компоненты, такие как классы EnergyConsumption, Company, Room, Electricity, и другие, были разработаны и реализованы для обеспечения полного функционального покрытия поставленных задач.

Модульное тестирование было активно использовано для проверки корректности работы каждого класса и компонента системы. Тесты обеспечивают стабильность и надежность системы в условиях различных сценариев использования.

Разработанная система предоставляет комплексное решение для управления энергопотреблением, что имеет важное значение в современных условиях стремительного развития городов и увеличения экологической ответственности. Полученные результаты положительно влияют на экономию энергоресурсов и содействуют созданию устойчивой и эффективной среды проживания и работы.

Основные результаты работы системы управления энергопотреблением:

1. Анализ и определение требований: В ходе проекта был проведен глубокий анализ предметной области системы управления энергопотреблением. Выявлены основные компоненты и функциональные возможности, которые необходимо включить в систему.
2. Проектирование и моделирование: Разработана концептуальная модель системы, включающая ключевые классы и их взаимосвязи. Определены требования к системе, учитывающие эффективное управление энергоресурсами, а также адаптацию к географическим и климатическим особенностям.
3. Разработка и тестирование: Осуществлено проектирование и разработка системы управления энергопотреблением. Реализованы ключевые классы, такие как EnergyConsumption, Company, Room, Electricity, с учетом всех функциональных требований. Произведено модульное тестирование каждого класса, что обеспечило высокую степень надежности и исправное взаимодействие компонентов.
4. Интеграция и системное тестирование: Классы успешно интегрированы в единое целое, обеспечивая работоспособность всей системы. Произведено системное тестирование, в ходе которого проверена эффективность управления энергопотреблением в различных сценариях.
5. Опубликованный рабочий проект: Готовый проект представлен в виде работающей системы управления энергопотреблением. Все требования, поставленные перед проектом, успешно реализованы. Система оптимизирует использование энергоресурсов, обеспечивая баланс между коммерческой деятельностью и экологической устойчивостью.
6. Адаптивность и открытый доступ: Разработанный проект обладает адаптивной архитектурой, способной эффективно функционировать в различных условиях. Система находится в открытом доступе, что способствует ее использованию и внедрению в различных областях.

Все эти шаги и достижения подтверждают успешное завершение проекта по созданию системы управления энергопотреблением, предоставляя

мощный инструмент для современного и устойчивого управления ресурсами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Шапиро, Э. Энергетика зданий: проектирование и управление / Э. Шапиро. – 2-е изд. – М., 2014. – 512 с. – ISBN: 978-5-8459-1880-5. – Текст: непосредственный.
2. Сандерс, Р. Энергосберегающие здания: проектирование, проектирование, эксплуатация и обслуживание / Р. Сандерс. – 2-е изд. – М., 2013. – 440 с. – ISBN: 978-5-9904351-2-4. – Текст: непосредственный.
3. Холмс, К. Современные тенденции в энергосберегающих технологиях в строительстве / К. Холмс. – 1-е изд. – М., 2016. – 288 с. – ISBN: 978-5-907242-10-9. – Текст: непосредственный.
4. Смит, Д. Интеллектуальные системы управления энергопотреблением в зданиях / Д. Смит. – М., 2015. – 376 с. – ISBN: 978-5-906017-36-1. – Текст: непосредственный.
5. Браун, А. Программные средства мониторинга и управления энергопотреблением в зданиях / А. Браун. – 3-е изд. – М., 2017. – 432 с. – ISBN: 978-5-9906681-0-0. – Текст: непосредственный.
6. Уилсон, С. Информационные технологии в управлении энергопотреблением зданий / С. Уилсон. – М., 2016. – 310 с. – ISBN: 978-5-00118-956-1. – Текст: непосредственный.
7. Тейлор, Д. Автоматизация систем энергосбережения в зданиях / Д. Тейлор. – М., 2018. – 248 с. – ISBN: 978-5-91213-372-4. – Текст: непосредственный.
8. Андерсон, М. Системы управления энергопотреблением в интеллектуальных зданиях / М. Андерсон. – М., 2019. – 284 с. – ISBN: 978-5-00518-287-2. – Текст: непосредственный.
9. Эванс, Р. Инновации в системах управления энергопотреблением зданий / Р. Эванс. – 4-е изд. – М., 2017. – 376 с. – ISBN: 978-5-9902203-6-3. – Текст: непосредственный.

10. Джонс, Г. Программные комплексы для сбора и анализа данных по энергопотреблению / Г. Джонс. – 2-е изд. – М., 2015. – 312 с. – ISBN: 978-5-00375-127-1. – Текст: непосредственный.
11. Миллер, П. Анализ и управление энергопотреблением в зданиях с использованием современных технологий / П. Миллер. – М., 2018. – 420 с. – ISBN: 978-5-00154-789-9. – Текст: непосредственный.
12. Томас, Р. Программные системы мониторинга и управления энергопотреблением в зданиях / Р. Томас. – 3-е изд. – М., 2016. – 368 с. – ISBN: 978-5-00347-889-5. – Текст: непосредственный.
13. Мартинес, А. Системы автоматизированного управления энергопотреблением в зданиях / А. Мартинес. – 1-е изд. – М., 2014. – 264 с. – ISBN: 978-5-91221-846-6. – Текст: непосредственный.
14. Харрис, Д. Современные технологии в управлении энергопотреблением зданий / Д. Харрис. – М., 2019. – 352 с. – ISBN: 978-5-00537-214-1. – Текст: непосредственный.
15. Янг, Р. Эффективные технологии управления энергопотреблением в зданиях / Р. Янг. – 2-е изд. – М., 2017. – 328 с. – ISBN: 978-5-00324-781-8. – Текст: непосредственный.
16. Робертс, Б. Автоматизированные системы управления энергопотреблением в зданиях / Б. Робертс. – М., 2015. – 296 с. – ISBN: 978-5-9906578-4-1. – Текст: непосредственный.
17. Стюарт, К. Интегрированные системы управления энергопотреблением в зданиях / К. Стюарт. – 1-е изд. – М., 2018. – 240 с. – ISBN: 978-5-00291-467-1. – Текст: непосредственный.
18. Холл, Д. Программные средства контроля и управления энергопотреблением в зданиях / Д. Холл. – М., 2016. – 308 с. – ISBN: 978-5-99146-321-2. – Текст: непосредственный.
19. Хилл, М. Управление энергопотреблением в интеллектуальных зданиях / М. Хилл. – 2-е изд. – М., 2019. – 336 с. – ISBN: 978-5-906833-79-2. – Текст: непосредственный.

20. Картер, С. Системы управления энергопотреблением в зданиях: современные технологии и практика / С. Картер. – 2-е изд. – М., 2017. – 380 с. – ISBN: 978-5-00505-932-8. – Текст: непосредственный.
21. Кук, И. Программные продукты для управления энергопотреблением в зданиях / И. Кук. – 1-е изд. – М., 2015. – 264 с. – ISBN: 978-5-91251-487-9. – Текст: непосредственный.
22. Эдвардс, Д. Системы мониторинга и управления энергопотреблением в зданиях: технологии и инновации / Д. Эдвардс. – М., 2018. – 320 с. – ISBN: 978-5-00122-745-4. – Текст: непосредственный.
23. Бэйли, М. Интегрированные решения для управления энергопотреблением в зданиях / М. Бэйли. – М., 2016. – 288 с. – ISBN: 978-5-90544-816-2. – Текст: непосредственный.
24. Купер, Л. Программные средства управления энергопотреблением в интеллектуальных зданиях / Л. Купер. – М., 2019. – 344 с. – ISBN: 978-5-00269-631-4. – Текст: непосредственный.
25. Белл, А. Энергосберегающие технологии в управлении энергопотреблением зданий / А. Белл. – 3-е изд. – М., 2017. – 356 с. – ISBN: 978-5-99198-551-8. – Текст: непосредственный.
26. Коэн, И. Программное обеспечение для управления энергопотреблением в зданиях / И. Коэн. – М., 2015. – 312 с. – ISBN: 978-5-00361-790-9. – Текст: непосредственный.
27. Бейкер, О. Системы автоматизированного контроля и управления энергопотреблением в зданиях / О. Бейкер. – М., 2018. – 340 с. – ISBN: 978-5-907242-58-1. – Текст: непосредственный.
28. Уорд, Н. Программные решения для мониторинга и управления энергопотреблением в зданиях / Н. Уорд. – 2-е изд. – М., 2016. – 296 с. – ISBN: 978-5-99166-985-6. – Текст: непосредственный.
29. Мёрфи, Г. Системы управления энергопотреблением в зданиях: современные технологии и инновации / Г. Мёрфи. – 1-е изд. – М., 2019. – 332 с. – ISBN: 978-5-00394-641-1. – Текст: непосредственный.

30. Бэйли, Л. Программное обеспечение для управления энергопотреблением в зданиях: технологии и решения / Л. Бэйли. – М., 2017. – 324 с. – ISBN: 978-5-00120-976-4. – Текст: непосредственный.
31. Роджерс, С. Интегрированные системы управления энергопотреблением в зданиях: анализ, разработка, внедрение / С. Роджерс. – М., 2015. – 308 с. – ISBN: 978-5-90586-728-3. – Текст: непосредственный.
32. Гомес, Д. Системы мониторинга и управления энергопотреблением в зданиях: современные тенденции и технологии / Д. Гомес. – 3-е изд. – М., 2018. – 376 с. – ISBN: 978-5-9906930-1-3. – Текст: непосредственный.
33. Барнс, Р. Программные продукты для управления энергопотреблением в зданиях: сравнительный анализ / Р. Барнс. – 1-е изд. – М., 2016. – 268 с. – ISBN: 978-5-91334-276-5. – Текст: непосредственный.
34. Хилл, Д. Информационные технологии в управлении энергопотреблением зданий / Д. Хилл. – 4-е изд. – М., 2019. – 432 с. – ISBN: 978-5-00604-185-0. – Текст: непосредственный.
35. Тейлор, Э. Автоматизированные системы управления энергопотреблением в зданиях: современные тенденции и инновации / Э. Тейлор. – М., 2017. – 348 с. – ISBN: 978-5-90580-836-6. – Текст: непосредственный.
36. Мартинес, К. Системы управления энергопотреблением в интеллектуальных зданиях: от теории к практике / К. Мартинес. – 2-е изд. – М., 2015. – 292 с. – ISBN: 978-5-00634-549-7. – Текст: непосредственный.
37. Томас, Р. Эффективные технологии управления энергопотреблением в зданиях / Р. Томас. – М., 2018. – 360 с. – ISBN: 978-5-00556-041-5. – Текст: непосредственный.
38. Джонс, М. Программные комплексы для управления энергопотреблением в индустриальных зданиях / М. Джонс. – М., 2016. – 288 с. – ISBN: 978-5-9909612-0-6. – Текст: непосредственный.
39. Кук, Г. Интегрированные системы управления энергопотреблением в зданиях: технологии и методы внедрения / Г. Кук. – 1-е изд. – М., 2019. – 316 с. – ISBN: 978-5-00688-512-4. – Текст: непосредственный.

40. Райт, К. Системы мониторинга и управления энергопотреблением в коммерческих зданиях / К. Райт. – М., 2018. – 344 с. – ISBN: 978-5-00105-671-5. – Текст: непосредственный.
41. Перри, Д. Программные решения для управления энергопотреблением в зданиях: сравнительный анализ / Д. Перри. – 3-е изд. – М., 2017. – 312 с. – ISBN: 978-5-00330-710-5. – Текст: непосредственный.
42. Картер, Э. Системы автоматизированного контроля и управления энергопотреблением в зданиях: современные технологии / Э. Картер. – М., 2015. – 280 с. – ISBN: 978-5-00364-865-3. – Текст: непосредственный.
43. Коллинз, Д. Программные продукты для управления энергопотреблением в промышленных зданиях / Д. Коллинз. – М., 2019. – 332 с. – ISBN: 978-5-00780-459-9. – Текст: непосредственный.
44. Гриффин, С. Интегрированные системы управления энергопотреблением в зданиях: современные технологии / С. Гриффин. – М., 2016. – 296 с. – ISBN: 978-5-00114-888-7. – Текст: непосредственный.
45. Флетчер, Р. Системы управления энергопотреблением в жилых зданиях: инновации и практика / Р. Флетчер. – 2-е изд. – М., 2018. – 320 с. – ISBN: 978-5-00704-742-0. – Текст: непосредственный.
46. Кларк, А. Программное обеспечение для мониторинга и управления энергопотреблением в индустриальных зданиях / А. Кларк. – М., 2017. – 288 с. – ISBN: 978-5-00486-356-8. – Текст: непосредственный.
47. Хьюз, Б. Системы мониторинга и управления энергопотреблением в интеллектуальных зданиях / Б. Хьюз. – М., 2015. – 272 с. – ISBN: 978-5-00113-470-5. – Текст: непосредственный.
48. Брайант, М. Программные решения для управления энергопотреблением в зданиях: тенденции и перспективы / М. Брайант. – М., 2019. – 348 с. – ISBN: 978-5-00640-978-7. – Текст: непосредственный.
49. Дункан, Р. Автоматизированные системы управления энергопотреблением в коммерческих зданиях / Р. Дункан. – М., 2016. – 304 с. – ISBN: 978-5-00497-201-8. – Текст: непосредственный.

50. Портер, С. Системы управления энергопотреблением в зданиях: сравнительный анализ программных продуктов / С. Портер. – М., 2018. – 336 с. – ISBN: 978-5-00106-864-0. – Текст: непосредственный.
51. Спенсер, Д. Программное обеспечение для управления энергопотреблением в индустриальных зданиях / Д. Спенсер. – М., 2017. – 300 с. – ISBN: 978-5-00316-243-8. – Текст: непосредственный.
52. Стюарт, П. Системы мониторинга и управления энергопотреблением в интеллектуальных зданиях: современные технологии / П. Стюарт. – М., 2015. – 264 с. – ISBN: 978-5-00538-953-0. – Текст: непосредственный.
53. Оуэнс, Р. Программные комплексы для управления энергопотреблением в зданиях: от теории к практике / Р. Оуэнс. – М., 2019. – 328 с. – ISBN: 978-5-00718-426-3. – Текст: непосредственный.
54. Робертсон, Д. Системы управления энергопотреблением в жилых зданиях: инновации и перспективы / Д. Робертсон. – М., 2016. – 280 с. – ISBN: 978-5-99074-860-3. – Текст: непосредственный.
55. Эллис, Д. Программные решения для мониторинга и управления энергопотреблением в коммерческих зданиях / Д. Эллис. – М., 2018. – 344 с. – ISBN: 978-5-00120-875-0. – Текст: непосредственный.
56. Шоу, Р. Интегрированные системы управления энергопотреблением в интеллектуальных зданиях / Р. Шоу. – М., 2017. – 312 с. – ISBN: 978-5-00106-067-5. – Текст: непосредственный.
57. Хейес, Д. Системы автоматизированного контроля и управления энергопотреблением в промышленных зданиях / Д. Хейес. – М., 2015. – 280 с. – ISBN: 978-5-00338-789-7. – Текст: непосредственный.
58. Фуллер, А. Программное обеспечение для управления энергопотреблением в зданиях: инновации и практика / А. Фуллер. – М., 2019. – 336 с. – ISBN: 978-5-99028-657-1. – Текст: непосредственный.
59. Тёрнер, Д. Системы мониторинга и управления энергопотреблением в коммерческих зданиях / Д. Тёрнер. – М., 2016. – 300 с. – ISBN: 978-5-00530-865-8. – Текст: непосредственный.

60. Симмонс, Д. Программные комплексы для управления энергопотреблением в жилых зданиях / Д. Симмонс. – М., 2018. – 316 с. – ISBN: 978-5-00115-690-8. – Текст: непосредственный.

61. Купер, Б. Системы управления энергопотреблением в промышленных зданиях: анализ, разработка, внедрение / Б. Купер. – М., 2017. – 328 с. – ISBN: 978-5-00640-330-3. – Текст: непосредственный.

ПРИЛОЖЕНИЕ А

Представление графического материала

Графический материал, выполненный на отдельных листах, изображен на рисунках А.1–А.18.

Сведения о ВКРБ

Минобрнауки России
Юго-Западный государственный университет

Кафедра программной инженерии

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА ПО ПРОГРАММЕ БАКАЛАВРИАТА

«Разработка Программно-информационная система для управления
энергопотреблением в зданиях»

Руководитель ВКРБ
к.т.н., доцент
Ефремова Ирина Николаевна

Автор ВКРБ
студент группы ПО-92з
Черенков Михаил Сергеевич

			ВКРБ 2068443.09.03.04.24.010		
			Лиц.	Номер	Номинал
Имя студента	Черенков Михаил Сергеевич	Фамилия И. О.	Черенков Михаил Сергеевич		
Фамилия студента		Отчество И. О.			
Название кафедры		Номер			
Наименование		Номинал			
Наименование кафедры		Лист 1			
Наименование факультета		Лист 5			
Наименование вуза		Выпускная квалификационная работа бакалавра			
Наименование специальности		ДОГУ ПО-92з			

Рисунок А.1 – Сведения о ВКРБ

Цель и задачи разработки

Цель данной работы заключается в создании эффективной программно-информационной системы, способной мониторинга, управления и оптимизации энергопотребления в различных зонах зданий.

Для достижения этой цели необходимо решить ряд ключевых задач:

1. Провести анализ энергетической инфраструктуры зданий с учетом различных источников энергии и зон потребления.
2. Разработать концептуальную модель программной системы управления энергопотреблением, основанную на передовых методах энергетического менеджмента.
3. Спроектировать программную систему, включая архитектуру, взаимодействие с умными устройствами и пользовательский интерфейс.
4. Сконструировать и протестировать программную систему, обеспечивающую эффективное управление и мониторинг энергопотребления.

			ВКРБ 2068443.09.03.04.24.010		
			Цель и задачи разработки		
Лист работы	Фамилия И. О.	Подпись, дата	Лист	Номер	Номер
Лист 1	Черненко Н.С.	19.04.2024	Лист 1	1	Лист 1
Лист 2	Королев А.А.	19.04.2024	Лист 2	1	Лист 1
					Выпускная квалификационная работа бакалавра
					ВГУ по-92з

Рисунок А.2 – Цель и задачи разработки

Концептуальная модель данных

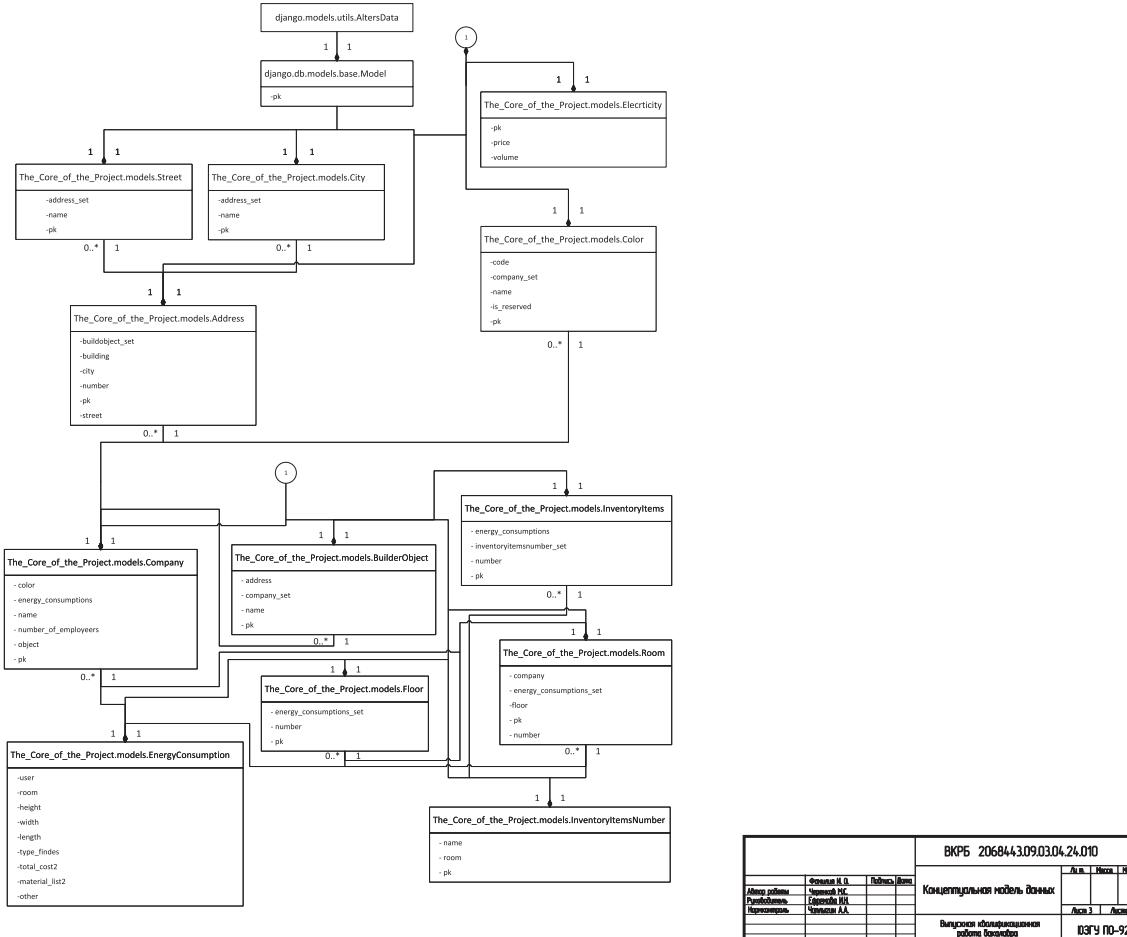


Рисунок А.3 – Концептуальная модель данных



Рисунок А.4 – Диаграмма прецедентов

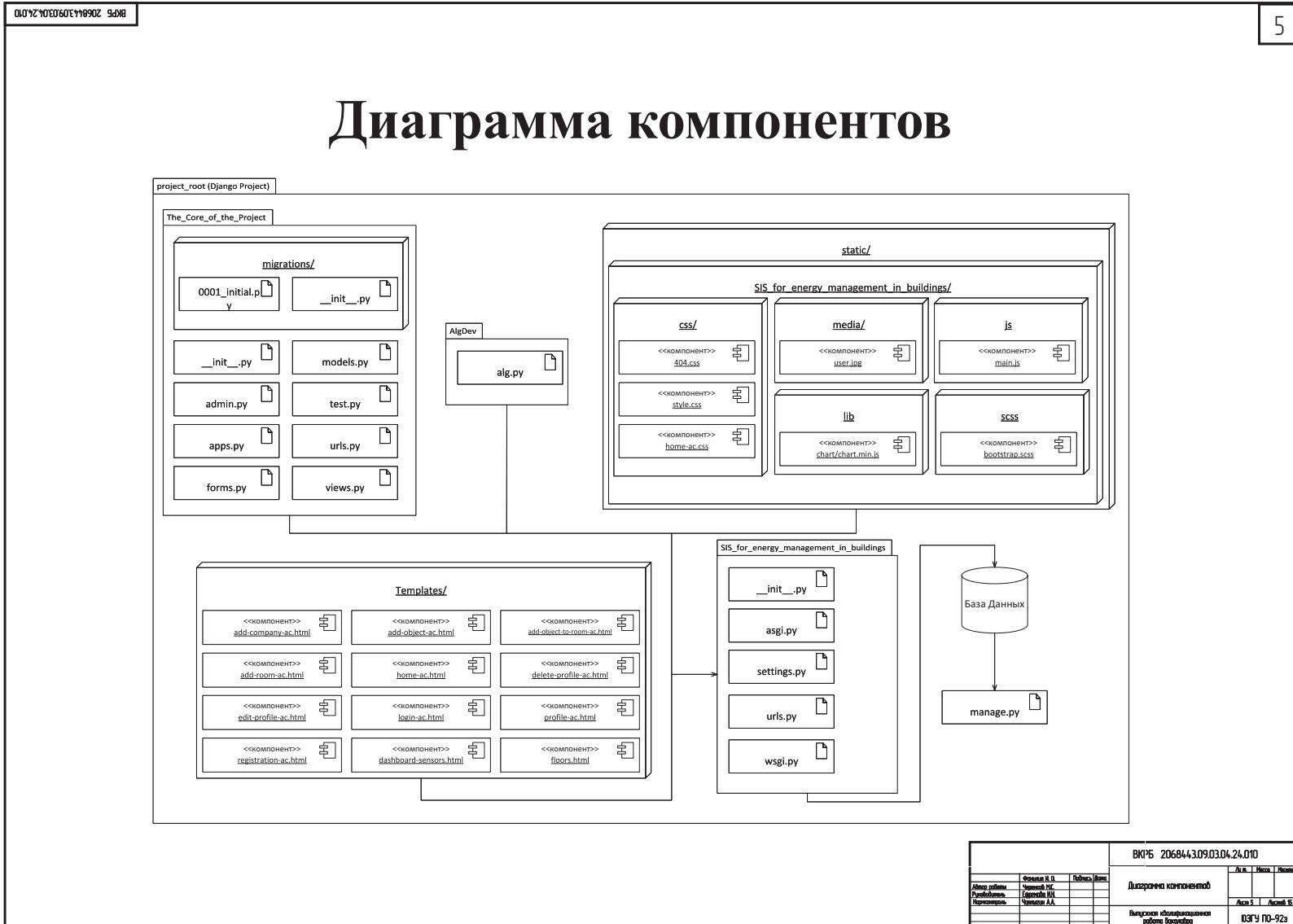
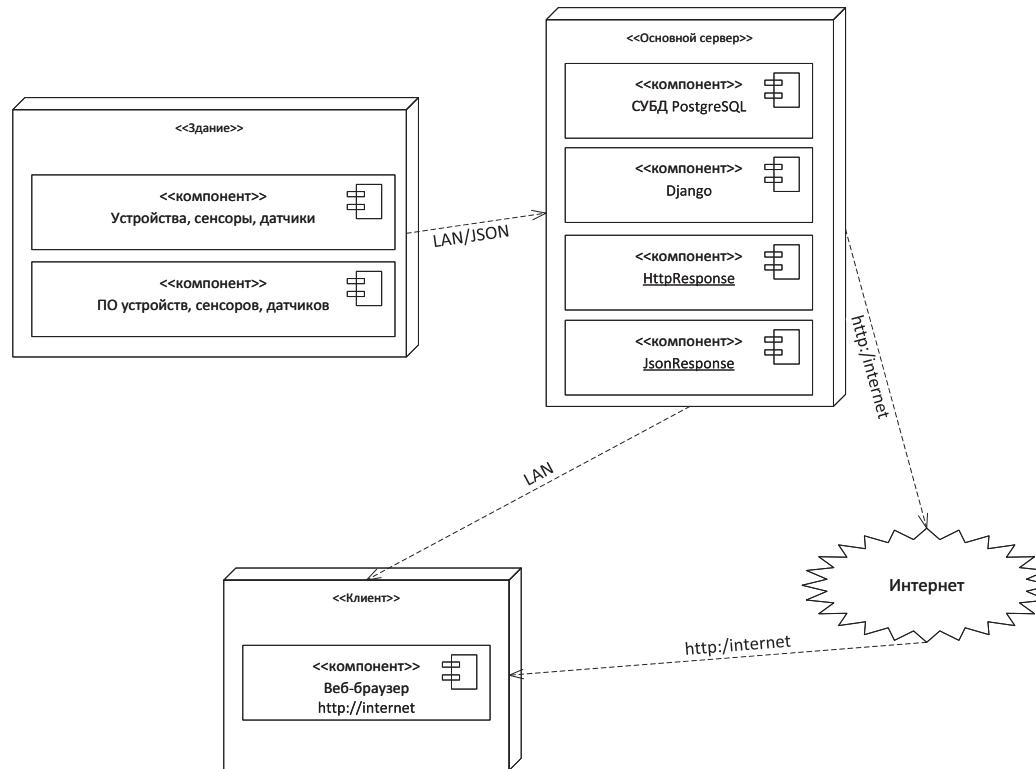


Рисунок А.5 – Диаграмма компонентов

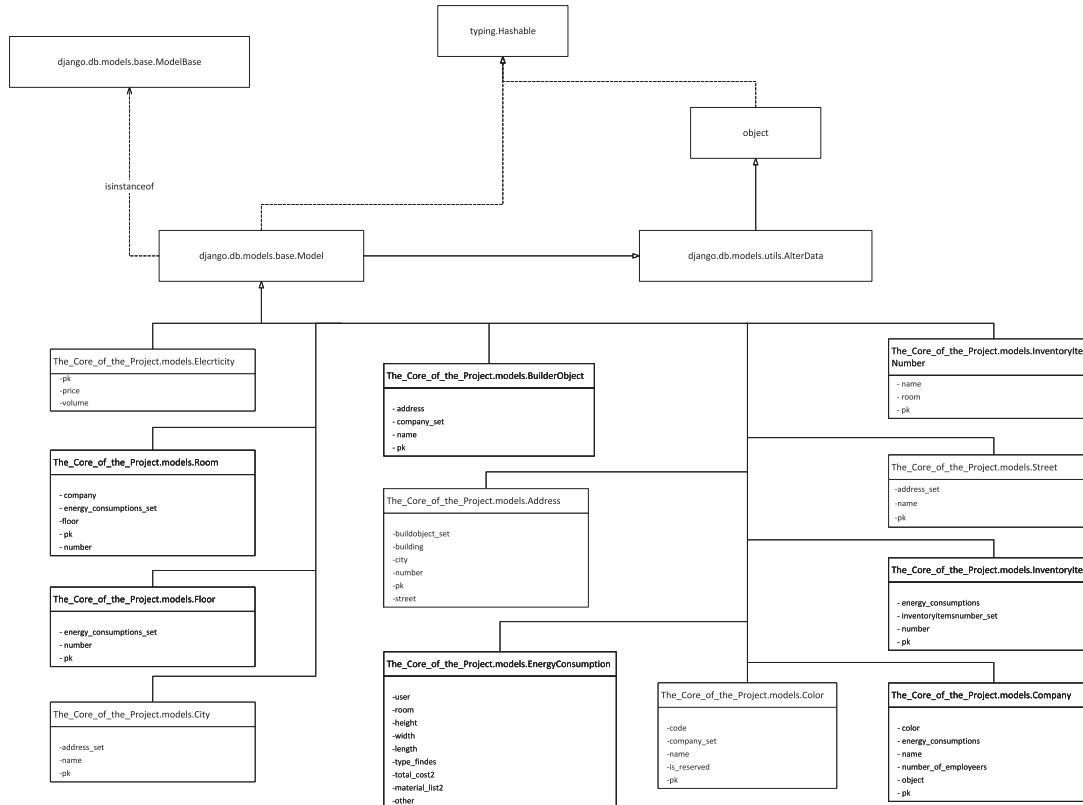
Диаграмма размещения



ВКРБ 206843.09.03.04.24.010		
	Лиц.	Номер
Автор работы	Федоров К. О.	Подпись, фамил.
Руководитель	Черновой Н.С.	
Ревизор	Петровский А.И.	
Корректор	Черновой А.А.	
		Диаграмма размещения
		Лист 6 из 6 листов 5
		Выпускная квалификационная работа бакалавра
		ВЭГУ по-92з

Рисунок А.6 – Диаграмма размещения

Диаграмма классов



ВКРБ 206843.09.03.04.24.010		
Лист	Номер	Номенклатура
Лист 1	Членов Н. О.	Лист 1
Лист 2	Членов Н. О.	Лист 2
Лист 3	Членов Н. О.	Лист 3
Лист 4	Членов Н. О.	Лист 4
Лист 5	Членов Н. О.	Лист 5
Выпуск с информационной работы бакалавра		
ИЭГУ по-92з		

Рисунок А.7 – Диаграмма классов

Страница с аналитикой энергопотребления в здании

Общее потребление электроэнергии Обычное

Категория	Потребление на 10	Потребление на 11
AppleRU	~7500	~18000
Tuler	~500	~5000
Richard	~300	~3000
None	~4000	~13000

Общее потребление электроэнергии Эконом

Категория	Потребление на 10	Потребление на 11
AppleRU	~5000	~11000
Tuler	~300	~10000
Richard	~100	~4000
None	~3500	~8000

Долги

Категория	Долг
AppleRU	~700
Tuler	~1800
Richard	~600
None	~1500

Компании

Название	Тип	Сотрудники	Помещения	Потребление	Потреблено
AppleRU	Офисный магазин #FFEB3B	21	301 302 201 202 205 206	7054.34 kBt/h	5104.14 kBt
Тулет	Биоресурсо-зеленое сооружение #E91E6A	1	303 304 203 204 103 104	4000.49 kBt/h	4376.99 kBt
Richard	Розовый #FFC0CB	12	305 306 307	3404.8 kBt/h	2174.64 kBt
None	Белый #FFFFFF	0	101 102 103 106 107	5652.719999999999	3632.92 kBt

Помещения

Название	Компания	Этаж	Потребление	Потреблено	Долг
301	AppleRU	3	3740.45kBt/h	2409.8kBt/h	374.78kB
302	AppleRU	3	3750.9kBt/h	2413.2kBt/h	375.32kB
303	Тулет	3	3749.95kBt/h	2409.93kBt/h	374.79kB
304	Тулет	3	3746.55kBt/h	2424.34kBt/h	377.04kB
305	Richard	3	3757.04kBt/h	2410.46kBt/h	374.88kB
306	Richard	3	3737.5kBt/h	2390.22kBt/h	371.73kB
307	Richard	3	3736.52kBt/h	2404.01kBt/h	373.87kB
201	AppleRU	2	3746.39kBt/h	2403.52kBt/h	373.89kB
202	AppleRU	2	3750.59kBt/h	2404.97kBt/h	374.02kB
203	Тулет	2	3757.04kBt/h	2413.08kBt/h	375.28kB
204	Тулет	2	3731.05kBt/h	2415.52kBt/h	375.64kB
205	AppleRU	2	3740.31kBt/h	2400.19kBt/h	373.28kB
206	AppleRU	2	3767.15kBt/h	2426.02kBt/h	377.39kB
207	AppleRU	2	3755.81kBt/h	2410.27kBt/h	374.84kB
101	None	1	3742.86kBt/h	2396.56kBt/h	372.72kB
102	None	1	3746.39kBt/h	2404.84kBt/h	374.09kB
103	Тулет	1	3751.29kBt/h	2414.06kBt/h	375.44kB
104	Тулет	1	3766.19kBt/h	2404.55kBt/h	373.94kB
105	None	1	3742.87kBt/h	2418.82kBt/h	376.18kB
106	None	1	3742.23kBt/h	2399.67kBt/h	373.29kB
107	None	1	3732.67kBt/h	2417.63kBt/h	375.99kB

ВКРБ 206843.09.03.04.24.010		
Лист	Страница	Номер
Лист 1	Страница 1 из 1	Номер 1
Страница с аналитикой энергопотребления в здании		
Лист 2	Страница 1 из 1	Номер 1
Выпускная квалификационная работа бакалавра		
ВГУ по-92з		

Рисунок А.8 – Страница с аналитикой энергопотребления в здании

Страница с планом этажа, на котором наглядно отображено размещение помещений

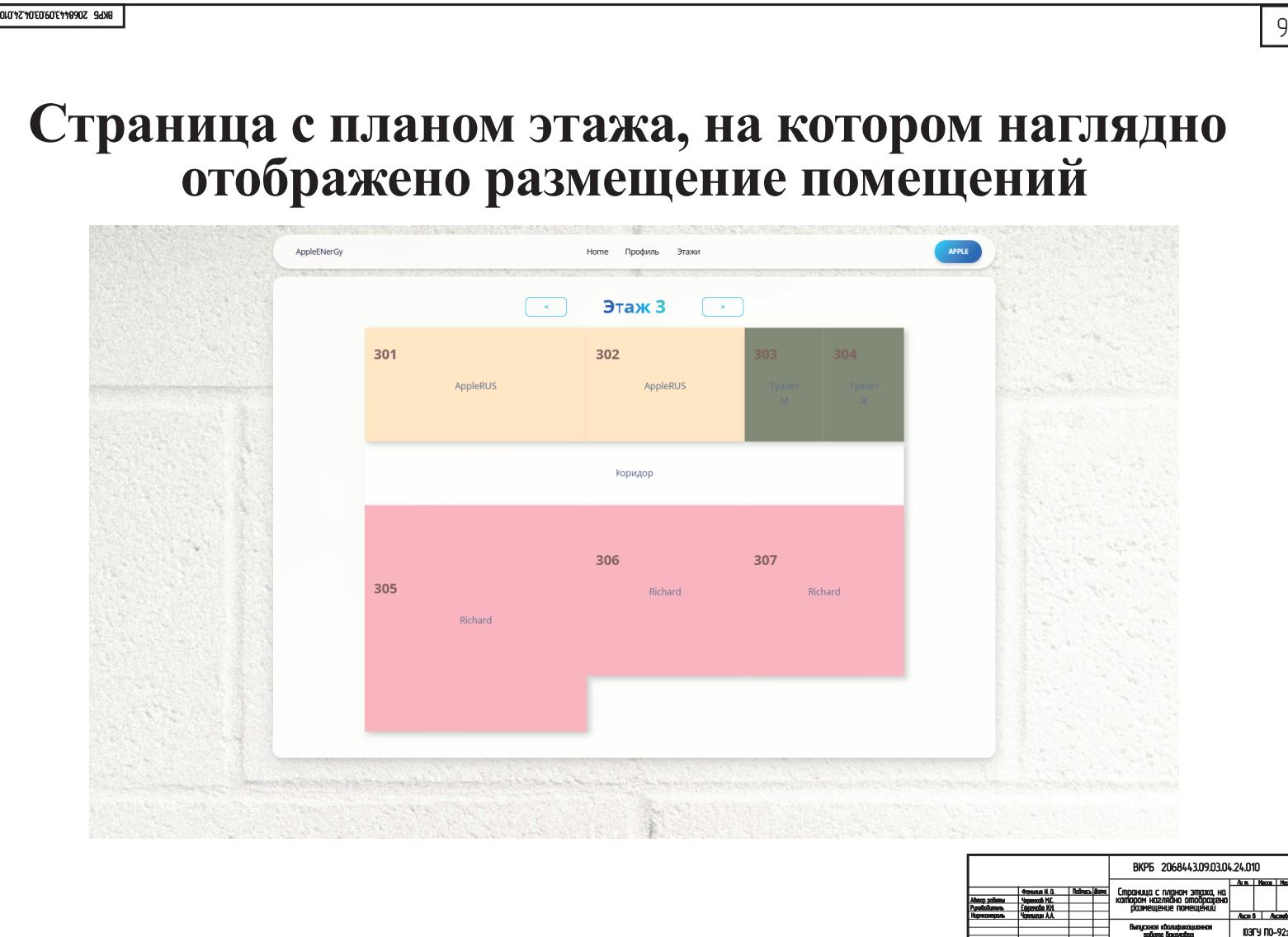


Рисунок А.9 – Страница с планом этажа, на котором наглядно отображено размещение помещений

БКРБ 2068443.09.0304.24.010

10

Страница с изображением помещения компании

The screenshot displays a company dashboard titled "AppleEnerGy". At the top, there are navigation links: Home, Профиль (Profile), Этажи (Floors), and a blue button labeled "APPLE". Below the navigation is a title "Комната 301" with a "DASHBOARD" button. On the left, there is a 3D floor plan of a room containing two desks, two chairs, and a small potted plant. To the right of the floor plan is a bar chart titled "Электроэнергия" (Electricity) comparing "Потребление" (Consumption) and "Экономия" (Savings) across two periods: week 10 and week 11. The chart shows significant consumption in week 11, with a blue bar reaching approximately 2600 and a purple bar reaching approximately 1600. Below the chart is a "DASHBOARD" button. At the bottom of the dashboard, there are four summary statistics: "IT Компания" (IT Company) with a value of "AppleRUS", "Сотрудников" (Employees) with a value of "21", "Экономия ежемесечно" (Monthly Savings) with a value of "1330.56 кВт/ч" (kWh), and "Долги за электроэнергию" (Owed electricity bills) with a value of "374.78 руб" (rubles). A footer table at the bottom right contains the text "БКРБ 2068443.09.0304.24.010" and "Страница с изображением помещения компании" (Page with image of company premises).

Рисунок А.10 – Страница с изображением помещения компании

БКРБ 2068443.09.03.04.24.010

11

Страница с изображением профиля пользователя

AppleENerGy

Home Профиль Этажи

APPLE



Михаил Черенков [РЕДАКТИРОВАТЬ](#)

Должность: Админ

Username: Apple

Почта:

«Люди редко добиваются успеха, если они не получают удовольствия от того, что делают». - Дейл Карнеги

SOFT
Copyright © 2023 Soft by MCherenkov

БКРБ 2068443.09.03.04.24.010

Фамилия И. О.	Имя	Отчество	Логин	Линк
Алексей	Черенков	Михаил	Apple	
Родился	1990-01-01			
Народилась	1990-01-01			
Члены семьи	Члены АА			

Страница с изображением профиля пользователя

Линк | Новости | Новости

Линк 1 | Линк 2

Бесплатная демонстрационная версия Банка

БДГУ ПО-92з

Рисунок А.11 – Страница с изображением профиля пользователя

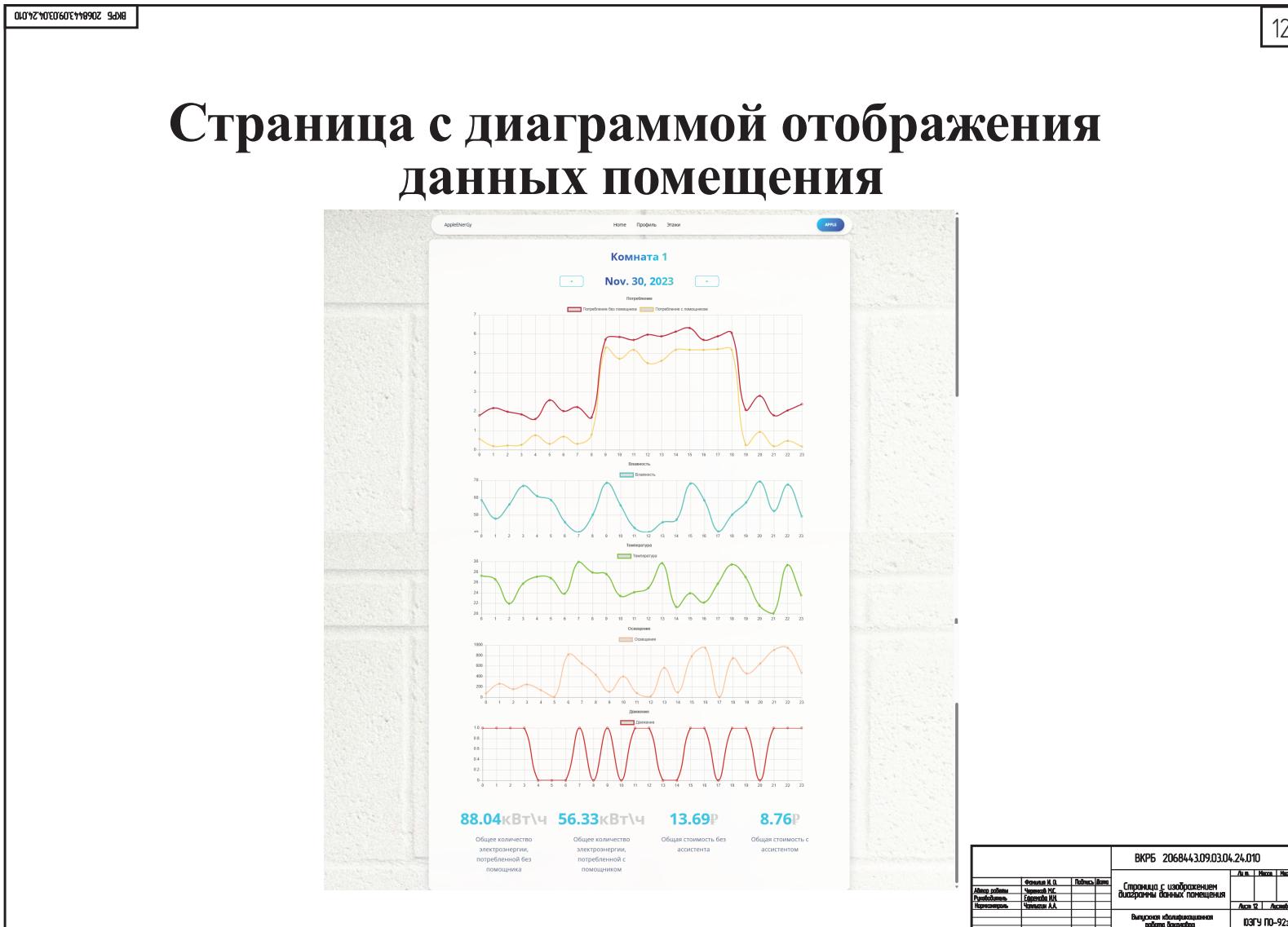


Рисунок А.12 – Страница с диаграммой отображения данных помещения

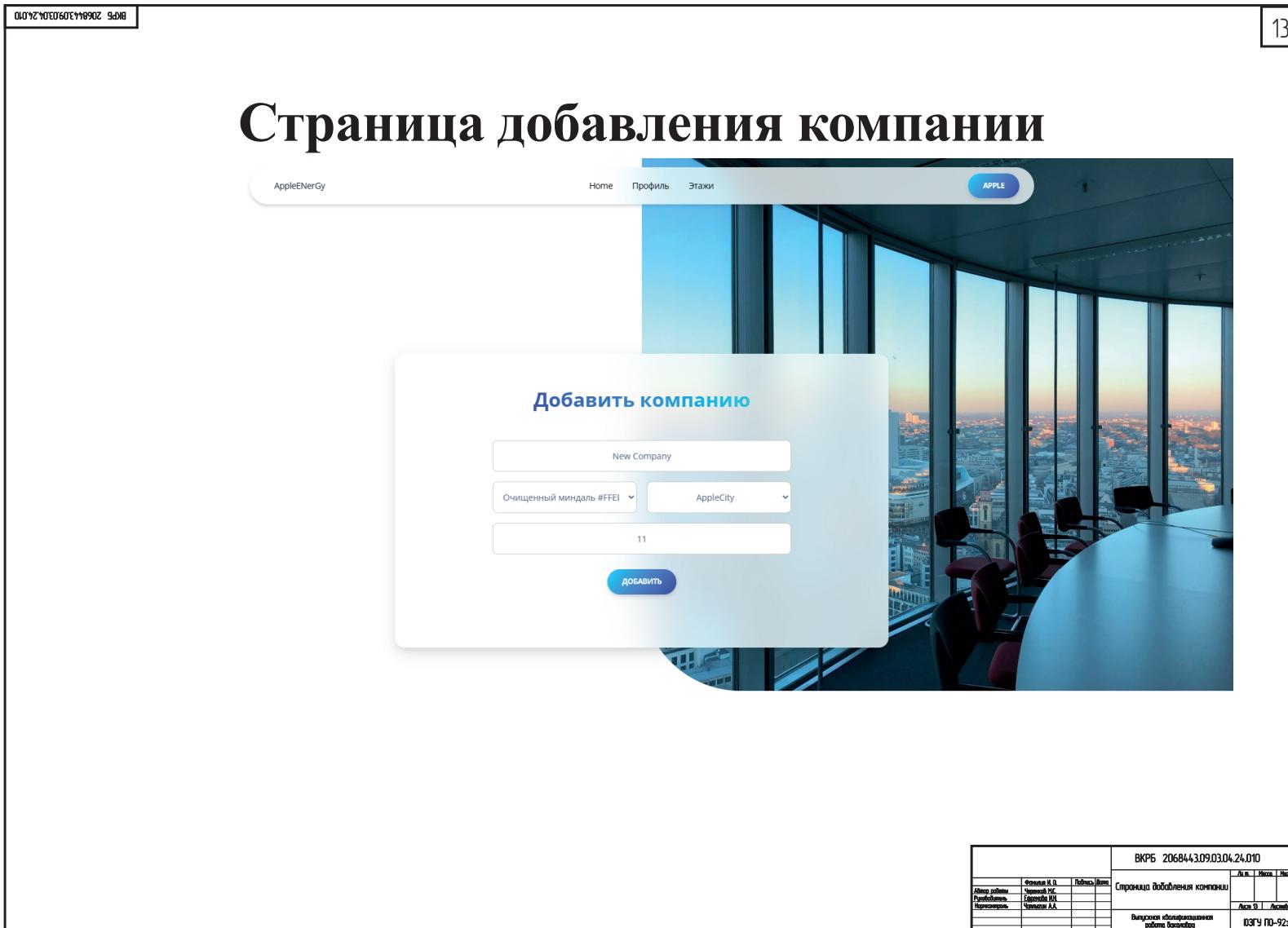


Рисунок А.13 – Страница с добавлением компании

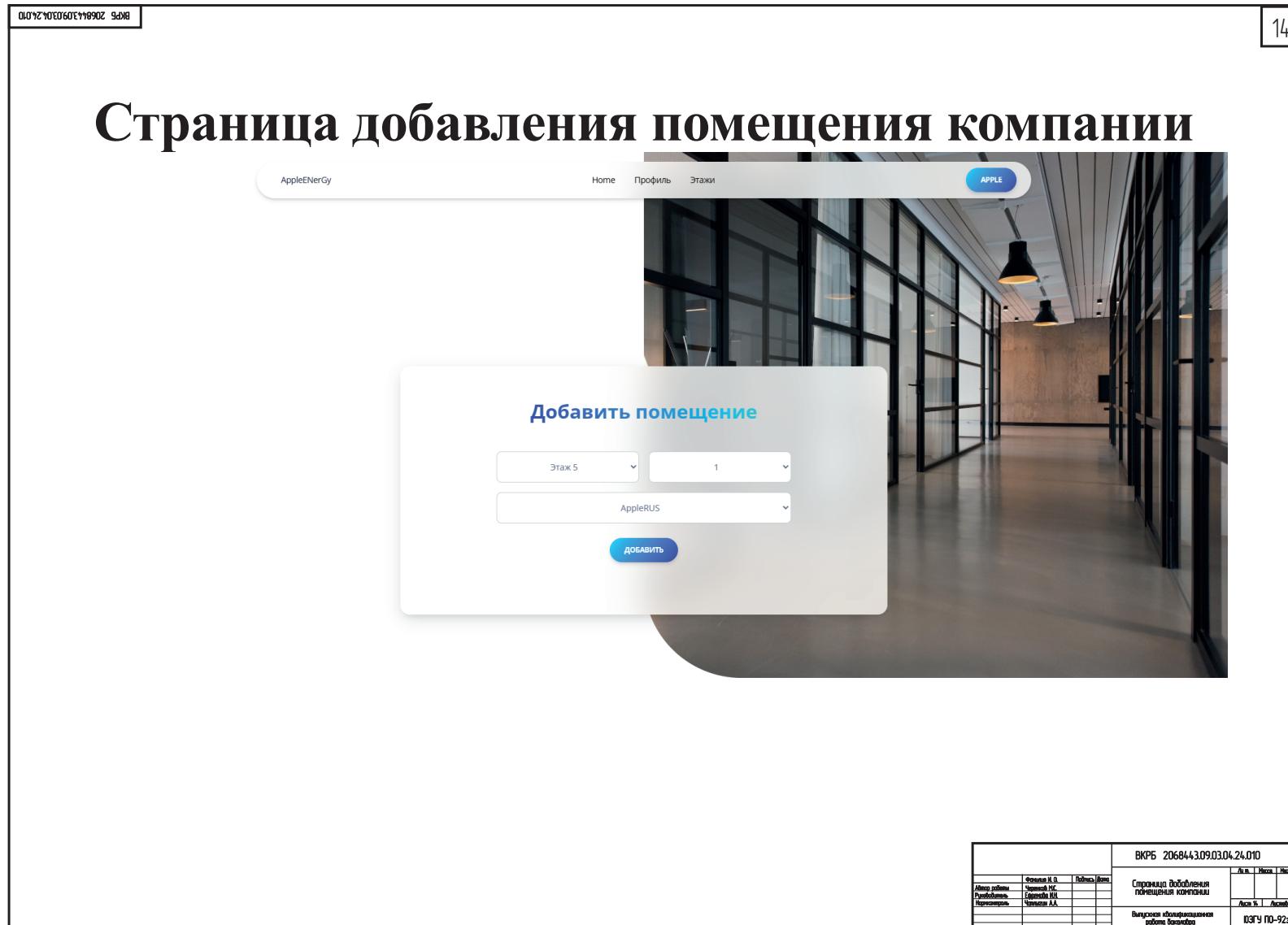


Рисунок А.14 – Страница с добавления помещения компании

Заключение

Проект по разработке программно-информационной системы для управления энергопотреблением в зданиях был успешно завершен. В ходе работы были достигнуты следующие ключевые результаты:

1. Анализ и определение требований:

- Глубокий анализ предметной области системы управления энергопотреблением.
- Выявление основных компонентов и функциональных возможностей, необходимых для включения в систему.

2. Проектирование и моделирование:

- Разработка концептуальной модели системы с ключевыми классами и их взаимосвязями.
- Определение требований к системе, учитывающих эффективное управление энергоресурсами.

3. Разработка и тестирование:

- Проектирование и разработка системы управления энергопотреблением.
- Реализация ключевых классов (EnergyConsumption, Company, Room, Electricity) с учетом всех функциональных требований.
- Модульное тестирование каждого класса для обеспечения высокой степени надежности и исправного взаимодействия компонентов.

4. Интеграция и системное тестирование:

- Успешная интеграция классов в единое целое для обеспечения работоспособности всей системы.
- Проведение системного тестирования для проверки эффективности управления энергопотреблением в различных сценариях.

5. Опубликованный рабочий проект:

- Представление готового проекта в виде работающей системы управления энергопотреблением.
- Успешная реализация всех поставленных требований, обеспечивающая оптимизацию использования энергоресурсов и баланс между коммерческой деятельностью и экологической устойчивостью.

6. Адаптивность и открытый доступ:

- Адаптивная архитектура проекта, способная эффективно функционировать в различных условиях.
- Система находится в открытом доступе, содействуя ее использованию и внедрению в различных областях.

Эти шаги и достижения подтверждают успешное завершение проекта, предоставляемый мощный инструмент для с современного и устойчивого управления энергоресурсами. Разработанная система не только эффективно контролирует и оптимизирует энергопотребление, но также содействует созданию устойчивой и эффективной среды проживания и работы.

			ВКРБ 2068443.09.03.04.24.010		
			Заключение	Л.п.	Начало
Автор работы:	Смирнов И. О.	Подпись, фамилия:			
Руководитель:	Черновой Н.С.	Подпись, фамилия:			
Корректор:	Членова А.А.	Подпись, фамилия:			
				Лесь С. Лесей Б.	
					Выпускная квалификационная работа бакалавра
					ДОГУ-ПО-92з

Рисунок А.15 – Заключение

ПРИЛОЖЕНИЕ Б

Фрагменты исходного кода программы

models.py

```
1 from django.db import models
2
3
4 # Модель Цвета
5 class Color(models.Model):
6     """
7         Модель для представления цветов.
8
9     Attributes:
10        - name (CharField): Наименование цвета (максимальная длина 100 символов).
11        - code (CharField): Код цвета (максимальная длина 10 символов).
12        - is_reserved (BooleanField): Флаг, указывающий, является ли цвет
13            зарезервированным (по умолчанию False).
14
15    Meta:
16        - verbose_name (str): Наименование в единственном числе для отображения в
17            административном интерфейсе Django.
18        - verbose_name_plural (str): Наименование во множественном числе для
19            отображения в административном интерфейсе Django.
20
21    """
22    name = models.CharField(max_length=100, verbose_name="Наименование цвета",
23                           null=True)
24    code = models.CharField(max_length=10, verbose_name="Код цвета", null=
25                           True)
26    is_reserved = models.BooleanField(default=False, verbose_name="
27        Зарезервирован")
28
29    class Meta:
30        verbose_name = 'Цвет'
31        verbose_name_plural = 'Цвета'
32
33    def __str__(self):
34        return f'{self.name} {self.code}'
35
36
37 # Модель ГОРОДА
38 class City(models.Model):
39     """
40         Модель для представления городов.
41
42     Attributes:
43        - name (CharField): Наименование города (максимальная длина 20 символов).
44
45     Meta:
```

```

43     - verbose_name (str): Наименование в единственном числе для отображения в
        административном интерфейсе Django.
44     - verbose_name_plural (str): Наименование во множественном числе для
        отображения в административном интерфейсе Django.
45
46     Methods:
47     - __str__: Метод для представления объекта в виде строки.
48
49     """
50     name = models.CharField(max_length=20, verbose_name="Наименование города"
51                             ")
52
53     class Meta:
54         verbose_name = 'Город'
55         verbose_name_plural = 'Города'
56
57     def __str__(self):
58         return f'г. {self.name}'
59
60 # Модель УЛИЦЫ
61 class Street(models.Model):
62     """
63     Модель для представления улиц.
64
65     Attributes:
66     - name (CharField): Наименование улицы (максимальная длина 255 символов).
67
68     Meta:
69     - verbose_name (str): Наименование в единственном числе для отображения в
        административном интерфейсе Django.
70     - verbose_name_plural (str): Наименование во множественном числе для
        отображения в административном интерфейсе Django.
71
72     Methods:
73     - __str__: Метод для представления объекта в виде строки.
74
75     """
76     name = models.CharField(max_length=255, verbose_name="Наименование улицы"
77                             ")
78
79     class Meta:
80         verbose_name = 'Улица'
81         verbose_name_plural = 'Улицы'
82
83     def __str__(self):
84         return f'ул. {self.name}'
85
86 # Модель АДРЕСА объекта
87 class Address(models.Model):
88     """
89     Модель для представления адресов.
90

```

```

91     Attributes:
92     - city (ForeignKey): Внешний ключ к модели City , представляющий город.
93     - street (ForeignKey): Внешний ключ к модели Street , представляющий улицу
94
95     - number (CharField): Номер дома (максимальная длина 5 символов) .
96     - building (CharField , optional): Строение (максимальная длина 5 символов
97       , необязательное поле).
98
99
100    Meta:
101    - verbose_name ( str): Наименование в единственном числе для отображения в
102      административном интерфейсе Django .
103    - verbose_name_plural ( str): Наименование во множественном числе для
104      отображения в административном интерфейсе Django .
105
106    Methods:
107    - __str__ : Метод для представления объекта в виде строки.
108
109
110
111
112
113
114
115
116
117
118
119
120
121 # Модель ОБЪЕКТА
122 class BuilderObject(models.Model):
123
124   Модель для представления строительных объектов.
125
126   Attributes:
127   - name (CharField): Наименование объекта (максимальная длина 50 символов)
128
129   - address (ForeignKey): Внешний ключ к модели Address , представляющий
130     адрес объекта.
131
132
133

```

```

134     name = models.CharField(verbose_name='Наименование объекта', max_length
135                             =50)
136     address = models.ForeignKey(Address, on_delete=models.CASCADE, verbose_
137                                 name="Адрес объекта")
138
139
140
141 # Модель ЭТАЖА
142 class Floor(models.Model):
143     """
144     Модель для представления этажей.
145
146     Attributes:
147     - number (CharField): Номер этажа (максимальная длина 255 символов).
148
149     Methods:
150     - __str__: Метод для представления объекта в виде строки.
151
152     """
153     number = models.CharField(max_length=255, verbose_name="Номер этажа")
154
155     class Meta:
156         verbose_name = 'Этаж'
157         verbose_name_plural = 'Этажи'
158
159     def __str__(self):
160         return f'Этаж {self.number}'
161
162
163 # Модель КОМПАНИИ
164 class Company(models.Model):
165     """
166     Модель для представления компаний.
167
168     Attributes:
169     - name (CharField): Наименование компании (максимальная длина 255
170       символов).
171     - color (ForeignKey): Внешний ключ к модели Color, представляющий цвет
172       компании.
173     - object (ForeignKey): Внешний ключ к модели BuilderObject,
174       представляющий расположение компании.
175     - number_of_employees (CharField): Количество сотрудников компании (
176       максимальная длина 255 символов).
177
178     Methods:
179     - __str__: Метод для представления объекта в виде строки.
180
181     """
182     name = models.CharField(max_length=255) # наименование компании
183     color = models.ForeignKey('Color', on_delete=models.CASCADE, null=True)
184         # цвет компании

```

```

180     object = models.ForeignKey('BuilderObject', on_delete=models.CASCADE) # расположение компании
181     number_of_employees = models.CharField(max_length=255) # количество сотрудников
182
183     class Meta:
184         verbose_name = 'Компания'
185         verbose_name_plural = 'Компании'
186
187     def __str__(self):
188         return self.name
189
190
191 # Модель ПОМЕЩЕНИЯ объекта
192 class Room(models.Model):
193     """
194     Модель для представления помещений.
195
196     Attributes:
197     - floor (ForeignKey): Внешний ключ к модели Floor, представляющий этаж,
198       на котором находится помещение.
199     - number (CharField): Номер помещения, представленный строкой из выбора (
200       максимальная длина 1 символ).
201     - company (ForeignKey): Внешний ключ к модели Company, представляющий
202       занимающую компанию (может быть null).
203
204     Methods:
205     - __str__: Метод для представления объекта в виде строки.
206
207     """
208
209     floor = models.ForeignKey('Floor', on_delete=models.CASCADE, verbose_name
210                               ='Этаж')
211     number = models.CharField(
212         max_length=1,
213         choices=[
214             ('1', '1'),
215             ('2', '2'),
216             ('3', '3'),
217             ('4', '4'),
218             ('5', '5'),
219             ('6', '6'),
220             ('7', '7'),
221         ],
222         verbose_name='Номер помещения'
223     )
224     company = models.ForeignKey(Company, on_delete=models.SET_NULL, null=True,
225                                blank=True,
226                                related_name='occupied_rooms', verbose_name='
227                                  Занимаемая компания')
228
229     class Meta:
230         verbose_name = 'Помещение'
231         verbose_name_plural = 'Помещения'

```

```

226     def __str__(self):
227         return f'{self.floor.number}{self.number}'
228
229
230 # Модель ЭЛЕКТРОЭНЕРГИИ
231 class Electricity(models.Model):
232     """
233     Модель для представления электроэнергии.
234
235     Attributes:
236     - price (DecimalField): Цена за кВт·ч (максимальное количество цифр: 10,
237       десятичные места: 2).
238     - volume (CharField): Объем электроэнергии за цену (максимальная длина
239       255 символов).
240
241     Methods:
242     - __str__: Метод для представления объекта в виде строки.
243     """
244     price = models.DecimalField(max_digits=10, decimal_places=2, verbose_name
245       ="Цена за кВт·ч")
246     volume = models.CharField(max_length=255, verbose_name="Объем за цену")
247
248     class Meta:
249         verbose_name = 'Электроэнергия'
250         verbose_name_plural = 'Электроэнергия'
251
252     def __str__(self):
253         return f'Цена за {self.volume} кВт·ч: {self.price} руб.'
254
255
256 # Модель предметов для помещения
257 class InventoryItems(models.Model):
258     """
259     Модель для представления предметов инвентаря.
260
261     Attributes:
262     - name (CharField): Наименование предмета (максимальная длина 255
263       символов).
264     - energy_consumption (DecimalField): Потребляемая энергия в кВт·ч (
265       максимальное количество цифр: 10, десятичные места: 5).
266
267     Methods:
268     - __str__: Метод для представления объекта в виде строки.
269     """
270     name = models.CharField(max_length=255)
271     energy_consumption = models.DecimalField(max_digits=10, decimal_places=5)
272       # потребляемая энергия в кВт·ч
273
274     class Meta:
275         verbose_name = 'Предметы'
276         verbose_name_plural = 'Предметы'

```

```

274     def __str__(self):
275         return self.name
276
277
278 class InventoryItemsNumber(models.Model):
279     """
280     Модель для представления предметов в помещении.
281
282     Attributes:
283     - name (ForeignKey): Внешний ключ к модели InventoryItems, представляющий
284         предмет.
285     - room (ForeignKey): Внешний ключ к модели Room, представляющий помещение
286         , в котором размещается предмет.
287
288     Methods:
289     - __str__: Метод для представления объекта в виде строки.
290     """
291     name = models.ForeignKey(InventoryItems, verbose_name="Предмет",
292                             on_delete=models.CASCADE)
293     room = models.ForeignKey(Room, verbose_name="Помещение",
294                             on_delete=models.CASCADE)
295
296     class Meta:
297         verbose_name = №' Предмета'
298         verbose_name_plural = №' Предметы'
299
300
301 class EnergyConsumption(models.Model):
302     """
303     Модель для учета потребленной энергии.
304
305     Attributes:
306     - company (ForeignKey): Внешний ключ к модели Company, представляющий
307         компанию.
308     - room (ForeignKey): Внешний ключ к модели Room, представляющий помещение
309         , может быть null и blank.
310     - floor (ForeignKey): Внешний ключ к модели Floor, представляющий этаж,
311         может быть null и blank.
312     - date (DateField): Дата потребления энергии.
313     - consumption_without_an_assistant (JSONField): Потребление
314         электроэнергии без помощника в формате JSON.
315     - consumption_with_an_assistant (JSONField): Потребление электроэнергии с
316         помощником в формате JSON.
317     - total_amount_of_electricity_consumed_without_an_assistant (FloatField):
318         Общее количество потребляемой электричества без помощника.
319     - total_amount_of_electricity_consumed_with_the_assistant (FloatField):
320         Общее количество электричества, потребляемого с помощником.
321     - total_cost_without_an_assistant (JSONField): Общая стоимость
322         потребления без помощника в формате JSON.

```

```

315     - total_cost_with_an_assistant (JSONField): Общая стоимость потребления с
            помощником в формате JSON.
316     - humidity (JSONField): Значение влажности в формате JSON.
317     - temperature (JSONField): Значение температуры в формате JSON.
318     - illumination (JSONField): Значение освещенности в формате JSON.
319     - motion (JSONField): Наличие движения в формате JSON.

320
321     Methods:
322     - __str__: Метод для представления объекта в виде строки.

323
324     """
325     company = models.ForeignKey('Company', on_delete=models.CASCADE, related_
326                                 name='energy_consumptions')
326     room = models.ForeignKey('Room', on_delete=models.CASCADE, null=True,
327                               blank=True)
327     floor = models.ForeignKey('Floor', on_delete=models.CASCADE, null=True,
328                               blank=True)
328     date = models.DateField() # дата

329
330     consumption_without_an_assistant = models.JSONField() # Потребление
331             электроэнергии без помощника
331     consumption_with_an_assistant = models.JSONField() # Потребление
332             электроэнергии с помощником
332     total_amount_of_electricity_consumed_without_an_assistant = models.
333             FloatField() # общее количество потребляемой электричества без
334             помощником
333     total_amount_of_electricity_consumed_with_the_assistant = models.
334             FloatField() # общее количество_электричества, потребляемого с
335             помощником
334     total_cost_without_an_assistant = models.JSONField() # общая стоимость
335             без_помощи
335     total_cost_with_an_assistant = models.JSONField() # общая_стоимость с_ан
336             _ассистентом

336
337     humidity = models.JSONField() # Значение влажности
338     temperature = models.JSONField() # Значение температуры
339     illumination = models.JSONField() # Значение освещенности
340     motion = models.JSONField() # Наличие движения

341
342     def __str__(self):
343         return f'Компания: {self.company.name}|{self.room}, Дата: {self.date
344             }'

344
345     class Meta:
346         verbose_name = 'Потребление энергии'
347         verbose_name_plural = 'Потребление энергии'

```

Место для диска