**МиНобрнауки россии**

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ**

**УНИВЕРСИТЕТ им. В.Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Институт *Энергетики, информационных технологий и управляющих систем*

Кафедра *Программного обеспечения вычислительной техники и автоматизированных систем*

Направление подготовки *09.03.04 Программная инженерия*

(шифр, наименование)

Направленность (профиль) образовательной программы *Разработка программно-информационных систем*

(наименование)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

на тему:

***Разработка программной системы автоматизации бюджетных процессов в области энергосбережения***

**Студент (ка) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Харитонов Сергей Дмитриевич

**Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**канд. техн. наук, доц. Поляков В.М.

**Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**ст.пр. Буханов Д. Г.

**К защите допустить**

**Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/** *Поляков В.М .***/**

**«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.**

**Белгород 2020 г.**

**МиНобрнауки россии**

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ**

**УНИВЕРСИТЕТ им. В.Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Институт  *Энергетики, информационных технологий и управляющих систем*

Кафедра *Программного обеспечения вычислительной техники и автоматизированных систем*

Направление подготовки *09.03.04 Программная инженерия*

(шифр, наименование)

Направленность (профиль) образовательной программы *Разработка программно-информационных систем*

(наименование)

|  |  |
| --- | --- |
|  | Утверждаю:  Зав. кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020г. |

**ЗАДАНИЕ**

на выпускную квалификационную работу студента (ки)

*Харитонова Сергея Дмитриевича*

(Фамилия Имя Отчество)

1. Вид выпускной квалификационной работы (ВКР) *бакалаврская работа*

2. Тема ВКР *РАЗРАБОТКА ПРОГРАММНОЙ СИСТЕМЫ АВТОМАТИЗАЦИИ БЮДЖЕТНЫХ ПРОЦЕССОВ В ОБЛАСТИ ЭНЕРГОСБЕРЕЖЕНИЯ*

утверждено приказом по университету от *«27» января 2020 г. № 2/47*

3. Срок сдачи студентом законченной ВКР *22.06.2020*

4. Исходные данные *учёт, автоматизация учёта производства, Java Spring, PostgreSQL, Java Persistence API, Web API, REST, SOAP, Data Transfer Object*

5. Содержание ВКР (перечень подлежащих разработке разделов)

*Производственный учёт кондитерских изделий;*

*Проектирование серверной части программного обеспечения учёта производства*

*кондитерских изделий;*

*Разработка серверной части программного обеспечения учёта производства кондитерских*

*изделий.*

6. Перечень графического материала *Титульный слайд, Цель и задачи, Актуальность работы,*

*Учёт производства на АО «КФ «Белогорье», Программные аналоги, Общая архитектура приложения, Средства разработки, Схема взаимодействия сервисов приложения, База данных «Цех», Диаграмма классов (model) «Цех», Диаграмма классов (controller) «Цех», Основные*

*запросы к системе, Основные запросы к системе, Итоги*

Консультанты по работе с указанием относящихся к ним разделов

|  |  |  |  |
| --- | --- | --- | --- |
| Раздел | Консультант | Задание выдал  (подпись, дата) | Задание принял  (подпись, дата) |

Дата выдачи задания «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Задание принял к исполнению

(подпись студента)

**КАЛЕНДАРНЫЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование этапов работы | Срок выполнения  этапов работы | Примечание |
| **1** | Постановка задачи, описание предметной области, изучение процессов учёта производства АО «Кондитерская фабрика «Белогорье» | *27.01.2020 – 16.03.2020* | *Выполнено* |
| **2** | Прoектирoвaние серверной части программного обеспечения учёта производства кондитерских изделий | *16.03.2020 – 13.04.2020* | *Выполнено* |
| **3** | Разработка и тестирование программного обеспечения учёта производства кондитерских изделий | *14.05.2020 – 15.06.2020* | *Выполнено* |
| **4** | Оформление пояснительной записки, презентации | *16.06.2020 – 19.06.2020* | *Выполнено* |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Харитонов Сергей Дмитриевич

(подпись) (Ф.И.О.)

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Буханов Дмитрий Геннадьевич.

(подпись)

**«Результаты проверки ЭВ ВКР на заимствование»**

Кафедра Программного обеспечения вычислительной техники и автоматизированных систем

Студент Харитонов С.Д. ПВ-42

Фамилия Инициалы Группа Дата

Тема ВКР Разработка серверной части программного обеспечения

учёта производства кондитерских изделий

ВКР прошла проверку на объем заимствований.

Итоговая оценка оригинальности:\_\_\_\_\_\_\_\_\_\_\_

Работу проверил Гончарук А.А.

Подпись Дата Фамилия Инициалы

Руководитель

ВКР доцент, канд. техн. наук Буханов Д.Г.

Должность, уч. степень, звание Подпись Дата Фамилия Инициалы

**OПРЕДЕЛЕНИЯ, СOКРAЩЕНИЯ И OБOЗНAЧЕНИЯ**

ПЭС – Программа ЭнергоСбережения

РСО – РесурсоСнабжающая Компания

СOДЕРЖAние

[ВВЕДЕНИЕ 8](#_Toc98341466)

[1. АНАЛИЗ СОСТОЯНИЯ ИССЛЕДОВАНИЯ В ОБЛАСТИ ИСПОЛЬЗОВАНИЯ ЭНЕРГЕТИЧЕСКИХ РЕСУРСОВ 9](#_Toc98341467)

[1.1. Энергоресурсы и их виды 9](#_Toc98341468)

[1.2. Бюджетные процессы в сфере энергосбережения 10](#_Toc98341469)

[1.3. Программа энергосбережения 10](#_Toc98341470)

[1.4. Анализ теплотехнических характеристик для разработки ПЭС 11](#_Toc98341471)

[1.4.1. Общие данные об организации 12](#_Toc98341472)

[1.4.2. Здания на балансе организации 12](#_Toc98341473)

[1.4.3. Энергопотребление ресурсов 14](#_Toc98341474)

[1.4.4. Сведения об оснащенности приборами учета 15](#_Toc98341475)

[1.5. Анализ существующих подходов подбора энергосберегающих мероприятий 16](#_Toc98341476)

[1.5.1. Анализ методики расчета потенциал энергосбережения и целевые уровни потребления энергоресурсов 16](#_Toc98341477)

[1.5.2. Выбор энергосберегающих мероприятий 18](#_Toc98341478)

[1.6. Постановка задачи 19](#_Toc98341479)

[2. РАЗРАБОТКА СИСТЕМЫ ФОРМИРОВАНИЯ ПЭС 21](#_Toc98341480)

[2.1. Разработка общей структуры ПО 21](#_Toc98341481)

[2.1.1. Разработка общей структуры серверной части 23](#_Toc98341482)

[2.1.2. Разработка общей структуры клиентской части 25](#_Toc98341483)

[2.2. Разработка подсистемы регистрации, авторизации и аутентификации 27](#_Toc98341484)

[2.2.1. Разработка системы ролей 27](#_Toc98341485)

[2.2.2. Разработка API 29](#_Toc98341486)

[2.2.3. Разработка системы разделения доступа к API приложения 31](#_Toc98341487)

[2.3. Разработка подсистемы ввода данных об учреждении 33](#_Toc98341488)

[2.3.1. Разработка структуры базы данных 33](#_Toc98341489)

[2.3.2. Разработка моделей 36](#_Toc98341490)

[2.3.3. Разработка API подсистемы 42](#_Toc98341491)

[2.4. Разработка подсистемы формирования паспорта ПЭС 43](#_Toc98341492)

[2.4.1. Разработка слоя получения данных из подсистемы хранения информации об организации 44](#_Toc98341493)

[2.4.2. Разработка слоя агрегирования данных для разделов 46](#_Toc98341494)

[2.4.3. Разработка слоя заполнения шаблона паспорта ПЭС 46](#_Toc98341495)

[2.4.4. Разработка API подсистемы 47](#_Toc98341496)

[2.5. Разработка подсистемы подбора энергосберегающих мероприятий 48](#_Toc98341497)

[ЗAКЛЮЧЕНИЕ 49](#_Toc98341498)

[СПИСOК ЛИТЕРAТУРЫ 50](#_Toc98341499)

[ПРИЛOЖЕНИЕ А 53](#_Toc98341500)

# ВВЕДЕНИЕ

# АНАЛИЗ СОСТОЯНИЯ ИССЛЕДОВАНИЯ В ОБЛАСТИ ИСПОЛЬЗОВАНИЯ ЭНЕРГЕТИЧЕСКИХ РЕСУРСОВ

В современном мире значительную роль в быту и ежедневной рутине играют энергоресурсы. Трудно представить кухню без газовой плиты, жилую комнату без электрического освещения, а дом без системы отопления. Для доставки энергоресурсов конечным потребителям существует огромная электроэнергетическая система.

Бюджетному сектору подвести

## Энергоресурсы и их виды

ДОПИСАТЬ ЭТОТ РАЗДЕЛ ВСЕМИ РЕСУРСАМИ

У уч бюджетного сектора существуют следующие ресурсы:/ они входя в слудеющие системы энергообеспеченияя/

Электроэнергетическая система представляет из себя совокупность объектов электроэнергетики и энергопринимающих установок потребителей̆ электрической̆ энергии, связанных общим режимом работы в едином технологическом процессе производства, передачи и потребления электрической̆ энергии в условиях централизованного оперативно-диспетчерского управления в электроэнергетике. [https://www.elec.ru/viewer?url=/files/2021/09/27/GOST-21027-2021.pdf]

Данная система имеет огромное влияние на работу экономики страны и благосостояние ее граждан, именно поэтому нуждается в государственном регулировании. Согласно энергетической стратегии России одной из главных задач государственного регулирования в сфере энергетики является повышение энергетической и экологической эффективности российской экономики и энергетики, в том числе за счет структурных изменений и активизации технологического энергосбережения.[ <https://centerpolit.org/national-security/energeticheskaya-strategiya-rossii-na-period-do-2030-goda/>]

## Бюджетные процессы в сфере энергосбережения

Особенно пристального внимания заслуживают объекты бюджетного сектора, такие как объекты культуры, медицины, спорта, занятости населения и другие, финансируемые из бюджета страны. Для их контроля в области потребления энергоресурсов используются следующие бюджетные процессы:

* расчет потенциала энергосбережения и целевых уровней снижения потребления энергоресурсов; [https://energy.midural.ru/wp-content/uploads/2020/09/PMER\_15\_07\_2020\_425.pdf]
* представление деклараций о потреблении энергетических ресурсов; [https://set.rk.gov.ru/uploads/txteditor/set/attachments/d4/1d/8c/d98f00b204e9800998ecf8427e/phpjQqoEY\_707.pdf]
* разработка и исполнение программ энергосбережения.[https://www.consultant.ru/document/cons\_doc\_LAW\_93978/]

В данной работе речь пойдет о программах энергосбережения и автоматизации их разработки.

## Программа энергосбережения

ДОПИСАТЬ

Одним из инструментов повышения эффективности использования энергоресурсов является разработка и реализация программ энергосбережения.

Согласно, федеральному закону «Об энергосбережении и о повышении энергетической эффективности и о внесении изменений в отдельные законодательные акты Российской Федерации» каждая бюджетная организация обязана разрабатывать и актуализировать ПЭС. Также помимо бюджетных организаций ПЭС разрабатывают муниципалитеты и субъекты РФ. [https://www.consultant.ru/document/cons\_doc\_LAW\_93978/]

Основной целью ПЭС является сокращение потребления используемых энергоресурсов, а также полное использование потенциала энергосбережения организации. Для этого на основе данных об организации и потребляемых энергоресурсах специалистом в сфере энергосбережения составляется план внедрения мероприятий, по энергосбережению и повышению энергетической эффективности, на ближайшие 3-5 лет. В ходе исполнения плана проводятся различные мероприятия: от сугубо технических (установка приборов учета) до организационных (проведение инструктажей и обучение персонала), нацеленные на более эффективное использование энергоресурсов.

Так же в ходе разработки ПЭС формируется паспорт программы энергосбережения и ежегодная промежуточная отчетность.

Паспорт ПЭС содержит информацию об организации, ее имуществе (здания, транспорт, приборы учета и другое) и план внедрения энергосберегающих мероприятий. Исходя из плана внедрения мероприятий и потребления энергоресурсов за предшествующие года вычисляется плановое ежегодное потребление, которого организация должна достичь в ходе реализации ПЭС.

Ежегодная промежуточная отчетность должна содержать информацию о реализации энергосберегающих мероприятий и сравнение планового потребление энергоресурсов с фактическим. На основании промежуточной отчетности ПЭС может корректироваться для лучшего достижения энергосберегающего эффекта.

## Анализ теплотехнических характеристик для разработки ПЭС

Разработка ПЭС, по сути, представляет из себя объемное исследование организации(энергоаудит) на предмет возможностей улучшения показателей энергоэффективности. Для этого специалисту в сфере энергосбережения предоставляется почти вся информация об организации:

* общая информация об организации;
* информация о транспорте, принадлежащем организации или арендуемом ей;
* технические характеристики зданий и строений организации;
* данные о потреблении энергоресурсов за предшествующие года;
* сведения об оснащенности приборами учета.

Каждый из описанных выше блоков информации по-своему необходим при составлении ПЭС и ее паспорта. Роль каждого блока будет рассмотрена далее.

## Общие данные об организации

Общие данные об организации содержат в большей степени юридическую информацию об организации:

* наименование;
* юридический адрес;
* информация о должностных лицах (директоре, лице ответственном за энергосбережение);
* и другое.

Данный раздел информации не несет практической пользы для разработки ПЭС т. к. не содержит технических характеристик организации. Но, несмотря на это, все равно играет важную роль при разработке. Как было описано ранее одним из основных этапов разработки и реализации ПЭС является составление ее паспорта, а для его заполнения и подписания необходимы описанные выше данные. Поэтому специалист ответственный за энергосбережение в организации обязан собирать и агрегировать их, так же как как технические характеристики.

## Здания на балансе организации

Одним из важнейших входных параметров являются технические характеристики зданий, находящихся на балансе организации. Основной объем потребления энергоресурсов приходится именно на здания, они отапливаются и освещаются, в них же персонал и посетители организации используют горячую и холодную воду, именно поэтому большинство энергосберегающих мероприятий проводится в рамках зданий.

Важным критерием плана внедрения энергосберегающих мероприятий является его реализуемость и целесообразность: не имеет смысла включать в план мероприятия по замене деревянных стеклопакетов, если в здании в принципе нет таковых.

Поэтому для разработки плана внедрения энергосберегающих мероприятий необходимо учитывать технические характеристики зданий, чтобы исключить бессмысленные мероприятия и наоборот понять какие мероприятия принесут максимальную пользу.

Также такие характеристики как отапливая площадь и среднее количество посетителей в сутки используются для расчета удельных показателей потребления энергоресурсов, таких как Гкал/м2 (тепловая энергия), м3/человека(вода) и других. А исходя из удельных показателей рассчитывается потенциал экономии и целевые уровни снижения потребления энергоресурсов, необходимые для разработки ПЭС.

Зачастую используют следующие технические характеристики зданий:

* среднее число посетителей в сутки;
* общая площадь;
* отапливаемая площадь;
* количество этажей;
* материал наружных стен;
* наличие утепления фасада;
* площадь остекления;
* количество вводов энергоресурсов, а также информация о том какое количество вводов оборудовано приборами учета.

## Энергопотребление ресурсов

Как было сказано ранее, основной целью ПЭС является снижение объема потребления энергоресурсов. Для корректной установки данного целевого показателя требуется учитывать множество факторов, в том числе и объем энергоресурсов, потребляемый организацией до внедрения ПЭС.

Плановое потребление энергоресурсов рассчитывается исходя из базового потребления и плана реализации энергосберегающих мероприятий.

За базовое потребление принимают потребление энергоресурсов в базовом году (предшествующий первому году начала реализации ПЭС).

Для расчета планового потребления предполагается, что без внедрения мероприятий по энергосбережению объем потребляемых энергоресурсов был бы не изменен и равнялся базовому потреблению. Исходя из этого плановое потребление принимается как разница потребления в базовом году и экономии, полученной вследствие реализации энергосберегающих мероприятий.

Так же в паспорте ПЭС для отслеживания динамики изменения потребления энергоресурсов указывается потребление за два года предшествующие базовому.

Исходя из выше сказанного следует что для разработки ПЭС необходимо знать потребление энергоресурсов за базовый год и 2 года предшествующих ему.

Чаще всего объем потребления энергоресурсов указывается для следующих видов энергоресурсов:

* тепловая энергия;
* природный газ;
* электрическая энергия;
* холодное водоснабжение;
* горячее водоснабжение.

## Сведения об оснащенности приборами учета

Для составления ПЭС необходима информация об объеме потребления энергоресурсов и чем точнее эта информация, тем качестве будут определены плановые показатели.

Для точной оценки объема потребления используются приборы коммерческого учета. Прибор (или узел) учета тепловой энергии — это комплекс приборов и устройств, обеспечивающих учет тепловой энергии, массы (объема) теплоносителя, а также контроль и регистрацию его параметров. Как правило они устанавливаются на границу ответственности между РСО и абонентом. [<http://www.energosovet.ru/bul_stat.php?idd=118>]

Одним из наиболее эффективных энергосберегающих мероприятий является установка прибора учета, которое в среднем позволяет сэкономить до 15% потребляемого объема. Сама установка прибора учета не уменьшает объем потребление, но позволяет отказаться от расчета по средним показателям, который зачастую превышает реальное потребление.

Так же в паспорт ПЭС вносится информация не только о количестве вводов, оборудованных приборами учета, но и информация по каждому конкретному прибору учета. Поэтому для разработки ПЭС необходима следующая информация:

* место установки;
* тип энергоресурса, подлежащий учету;
* класс точности;
* заводской номер;
* год ввода в эксплуатацию;
* дата последней поверки;
* меж поверочный интервал.

## Анализ существующих подходов подбора энергосберегающих мероприятий

План по внедрению энергосберегающих мероприятий является ядром ПЭС и представляет из себя список энергосберегающих мероприятий, сроки их проведения, а также количество и источники бюджетных средств, выделяемых на их реализацию.

План составляется лицом, имеющим высокую квалификацию в сфере энергосбережения, которое использует технические характеристики организации, целевые уровни снижения потребления энергоресурсов и потенциал энергосбережения, для выбора максимально эффективных и реализуемых в рамках организации мероприятий.

Данный процесс представляет из широкое поле для внедрения автоматизации, ведь именно он является наиболее затратным этапом разработки ПЭС, как в материальном, так и во временном плане.

Для автоматизации и контроля данного процесса необходимо реализовать два этапа:

* расчет целевых уровней снижения и потенциала энергосбережения;
* подбор энергосберегающих мероприятий на основе входных данных.

## Анализ методики расчета потенциал энергосбережения и целевые уровни потребления энергоресурсов

При составлении плана внедрения энергосберегающих мероприятий необходимо соблюсти баланс между размером предполагаемой экономии и ее целесообразностью. Для того что бы подобрать оптимальное количество мероприятий, проводится анализ организации и рассчитываются целевые уровни потребления энергоресурсов и потенциал энергосбережения.

Данные характеристики рассчитываются на основании методики из приказа 425 Министерства Энергетики.

Целевой уровень снижения потребления ресурса - плановый̆ удельный̆ годовой̆ расход ресурса, до которого государственное (муниципальное) учреждение обязано снизить свой фактический удельный годовой расход данного ресурса после его приведения к сопоставимым условиям.

Потенциал снижения потребления ресурсов рекомендуется определять как разницу между текущим уровнем потребления ресурсов государственным (муниципальным) учреждением и уровнем, при котором потребление ресурсов осуществляется максимально эффективно и соответствует высокому классу энергетической эффективности. [https://energy.midural.ru/wp-content/uploads/2020/09/PMER\_15\_07\_2020\_425.pdf]

Для определения данных показателей потребление организации за базовый год переводится в удельные показатели, так же потребление тепловой энергии дополнительно приводится к сопоставимым условиям исходя из длительности отопительного периода (формула 1.3.1) и этажности зданий (формула 1.3.2).

ЗАМЕНИТЬ НА НОРМАЛЬНУЮ ФОРМУЛУ И ПОДПИСАТЬ НОМЕР

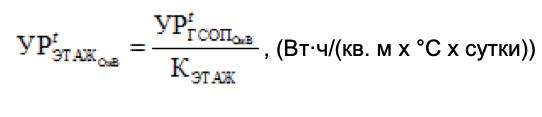
Изображение выглядит как текст

Автоматически созданное описание(1.3.1)

УРоив – удельный годовой расход тепловой энергии на нужды отопления и вентиляции в календарном году (Гкал/кв.м)

ГСОП – число градус суток отопительного периода за этот же календарный год

ЗАМЕНИТЬ НА НОРМАЛЬНУЮ ФОРМУЛУ И ПОДПИСАТЬ НОМЕР

(1.3.2)

УРгсоп – удельный годовой расход тепловой энергии на нужды отопления и вентиляции в году приведенный к сопостовимым климатическим условиям (Вт\*ч/(кв.м х С х сутки)

Кэтаж – корректировочный коэффициент на этажность и режим работы.

На основе удельных показателей и типа организации в соответствии со справочными таблицами из методики находится потенциал энергосбережения и целевой уровень потребления энергоресурсов.

Исходя из рассчитанных показателей мероприятия подбираются таким образом чтобы достичь целевого уровня снижения потребления энергоресурсов, а также полностью использовать потенциал энергосбережения в ходе реализации ПЭС.

Обычно ПЭС реализуется в течение трех лет и, согласно методике, рекомендуется разделять рассчитанное снижение потребления энергоресурсов по 50%, 25% и 25% на каждый год соответственно.

## Выбор энергосберегающих мероприятий

Для достижения целевых показателей ПЭС у лица ответственного за энергосбережение в инструментарии есть множество различных мероприятий, нацеленных на сокращение потребления энергоресурсов.

Условно мероприятия можно поделить на две группы:

* Нацеленные на сокращение потребления конкретных ресурсов (установка приборов учета, утепление фасадов)
* Организационные мероприятия применяемые к персоналу организации (разработка Положения о порядке стимулирования работников за экономию энергии и энергоресурсов)

После расчета целевых уровней потребления энергоресурсов, а также потенциала энергосбережения, лицо разрабатывающее ПЭС выбирает энергосберегающие мероприятия так, чтобы максимально приблизиться к целевым показателям и при этом сохранить реализуемость плана по внедрению энергосберегающих мероприятий.

По сути, это является творческим процессом, но можно выделить некоторые закономерности, которые будут использоваться для автоматизации составления плана:

* все вводы энергоресурсов, не оборудованные приборами учета, оборудуются ими;
* в случае наличия не энергоэффективных приборов освещения, они заменяются на энергоэффективные;
* не утепленные помещения, в которых работает отопление, утепляются;
* деревянные стеклопакеты заменяются на энергоэффективные;
* и так далее, все не энергоэффективное оборудование заменяется на энергоэффективное.

## Постановка задачи

Проведенный анализ существующего процесса составления ПЭС позволяет выявить следующие недостатки:

разработчик ПЭС должен иметь высокую квалификацию в сфере энергосбережения;

разработка и контроль ПЭС производится раз в три года, что снижает оперативность реагирования;

изменения в законодательстве либо других нормативных документах в среде бюджетных учреждений распространяются не равномерно;

необходим обширный объем информации (технические характеристики зданий и приборов учета, информация об объемах потребления энергоресурсов и т.д.), указанной пользователем;

отсутствие методик подбора мероприятий.

Таким образом, *целью* выпускной квалификационной работы является повышение энергоэффективности бюджетных организаций, за счет предоставления автоматизации бюджетных процессов и разработки автоматизированной системы создания и контроля программ энергосбережения и их паспортов.

**Вывод**

Подводя итоги текущей главы, можно отметить, что разработка ПЭС и ее паспорта является сложным бюджетным процессом, таящим в себе массу нюансов, расчетов и решений, принимаемых специалистом в сфере энергосбережения. Так же были выделены основные этапы разработки ПЭС и необходимые входные данные, на основе которых можно сделать вывод о наличии возможности внедрения автоматизации и ее выгоды в данном бюджетном процессе.

Самым удобным для пользователя способом автоматизации было принято использование WEB-приложения, позволяющего разработать паспорт ПЭС.

## РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ СИСТЕМЫ ФОРМИРОВАНИЯ ПЭС

Для разработки серверной части системы требуется решить ряд задач:

* разработка общей структуры ПО;
* разработка общей структуры серверной части ПО;
* разработка подсистемы регистрации, авторизации и аутентификации;
* разработка подсистемы ввода данных об учреждении;
* разработка подсистемы формирования паспорта;
* разработка подбора энергосберегающих мероприятия;
* разработка интеллектуальной системы валидации ввода данных пользователей.

## Разработка общей структуры ПО

В качестве основы общей структуры ПО была выбрана клиент-серверная архитектура.

Главным принципом клиент-серверной архитектуры является то, что ПО разделяется на 3 слоя (указанные на рисунке 2.1) взаимодействующие между собой через некий API.

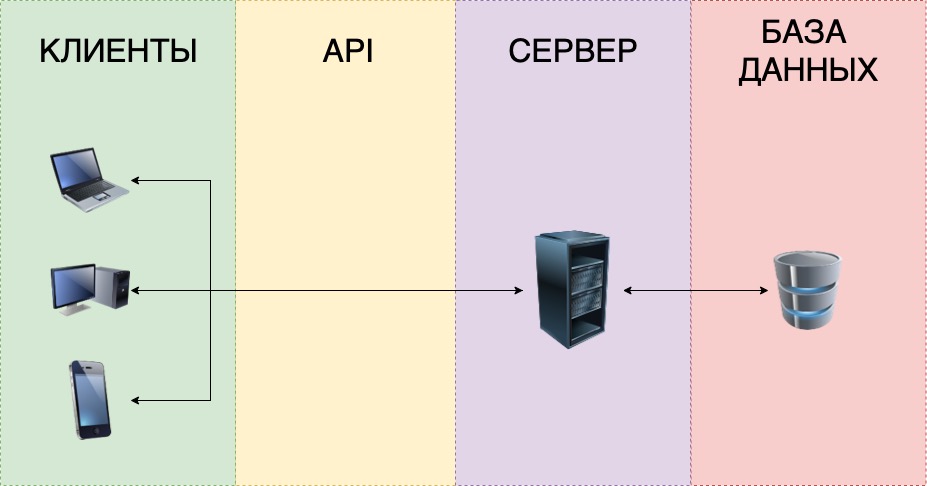


Рисунок 2.1 – клиент-серверная архитектура

API (от англ. Application Programming Interface – программный интерфейс приложения) — это набор определений, протоколов и инструментов для разработки ПО и приложений. API-интерфейс разрабатывается для упрощения создания программ, путем предоставления всех необходимых функциональных блоков.

В клиент-серверной архитектуре выделяют следующие слои:

* клиент;
* сервер;
* база данных.

Клиент (пользовательский интерфейс), основной обязанностью данного слоя является взаимодействие с пользованием. Это может быть веб-браузер, отображающий HTML-страницы, или графическое приложение. На данном слое хранится минимум логики, для отображения данных или их изменения клиент обращается к серверу.

Сервер. На данном слое хранится основная бизнес-логика, описывающая процессы необходимые пользователю для выполнения определенных задач. При этом сервер является прослойкой между клиентом (отображающим данные) и сервером (хранящим данные). Он обрабатывает запросы и ответы клиента.

Сервер базы данных отвечает за корректное хранение и консистентность данных, тем самым снимая данную обязанность с сервера, позволяя ему фокусироваться на бизнес-логике приложения.

## Разработка общей структуры серверной части

В рамках серверной части выделим подсистемы, отвечающие за блоки функциональности приложения:

* подсистема регистрации, авторизации и аутентификации;
* подсистема ввода данных об учреждении;
* подсистема формирования паспорта ПЭС;
* подсистема подбора энергосберегающих мероприятий;
* подсистема интеллектуальной валидации ввода данных пользователей.

В зависимости от необходимости подсистемы серверной части могут взаимодействовать:

* между собой;
* с клиентской частью;
* с сервером базы данных.

На рисунке 2.2 изображена схема взаимодействия между подсистемами серверной части.

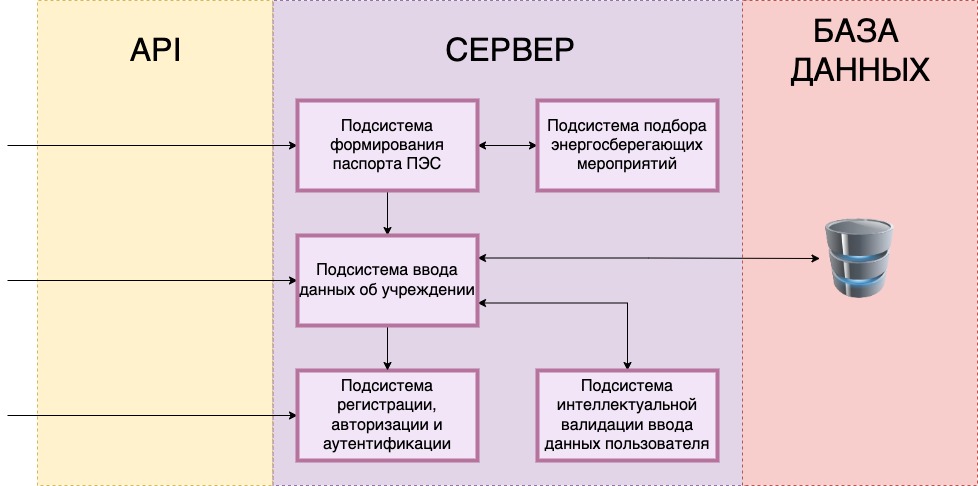


Рисунок 2.2 – Схема взаимодействия подсистем серверной части

Подсистема регистрации, авторизации и аутентификации – данный модуль отвечает за идентификацию пользователя и разграничение прав. Пользователями системы являются бюджетные организации вводящие свои данные, поэтому важно ограничивать доступ к ней и предоставлять его только владельцам данных.

Подсистема ввода данных об учреждении. Для формирования паспорта ПЭС и подбора списка энергосберегающих ресурсов необходим значительный объем данных. Для того что бы не вносить его каждый раз пользователю необходимо предоставить возможность сохранять их на сервере и получать к ним доступ, это и является обязанностью подсистемы ввода данных об учреждении. Для выполнения своей функции ей так же необходимо взаимодействовать с другими подсистемами и сервером базы данных.

Подсистема интеллектуальной валидации пользовательских данных. Поскольку, вводимые пользователем, данные могут быть некачественные или не консистентные, необходимо проверять их на достоверность и соответствие действительности. Ввиду сложности, поставленной задачи из-за разнородности данных, было принято решение интеллектуализировать данную подсистему.

Подсистема формирования паспорта ПЭС – отвечает за формирования .docx файла содержащего паспорт ПЭС, для этого подсистема берет данные из модуля хранения информации об организации.

Подсистема подбора энергосберегающих мероприятий – облегчает работу пользователя за счет рекомендации энергосберегающих мероприятий при помощи выведенных эвристик. Данная подсистема используется подсистемой формирования паспорта ПЭС.

## Разработка подсистемы регистрации, авторизации и аутентификации

Разработанная система аутентификации пользователей базируется на встроенной во фреймворк Django авторизации с помощью sessionID и базовом классе для хранения данных о пользователе User.

Для того что бы определить от чьего имени осуществляется запрос, используется идентификатор сессии, хранящийся в cookie браузера.

При входе в систему сервер генерирует идентификатор и сохраняет его в базе данных, затем отправляет его на клиент в виде ответа, а клиент получив идентификатор сохраняет его в cookie.

При отправке запроса клиент добавляет идентификатор в заголовки запроса, он в свою очередь проверяется на соответствие с хранящимся в базе данных.

При выходе из системы идентификатор, хранящийся в базе данных, удаляется.

## Разработка системы ролей

Для реализации системы ролей было принято расширить существующую структуру.

Для хранения информации о пользователе, которая не предусмотрена в базовом классе User, создана промежуточная модель Employer, хранящая ссылку на класс User и содержащая дополнительные поля:

* first\_login – был ли осуществлен первый вход в систему;
* last\_change\_password – когда последний раз был изменен пароль.

class Employer(models.Model):

user = models.OneToOneField(User, on\_delete=models.CASCADE)

first\_login = models.BooleanField(default=True)

last\_change\_password = models.DateField(default=None, blank=True, null=True)

Для представления информации о каждой конкретной роли, создаются модели наследующие свойства базовой модели Role, которая ссылается на Employer. RoleUser хранит внешний ключ на конкретную организацию, существование RoleUser между пользователем и организацией предоставляет пользователю доступ к ней.

class Role(models.Model):

employer = models.ForeignKey(Employer, on\_delete=models.SET\_NULL, blank=True, null=True)

class RoleUser(Role):

eobject = models.ForeignKey(EObject, on\_delete=models.CASCADE)

Реализованная система ролей предоставляет большую гибкость в выдаче прав, т. к. позволяет как нескольким пользователем иметь доступ к одной и той же организации, так и одному пользователю иметь доступ к множеству организаций.

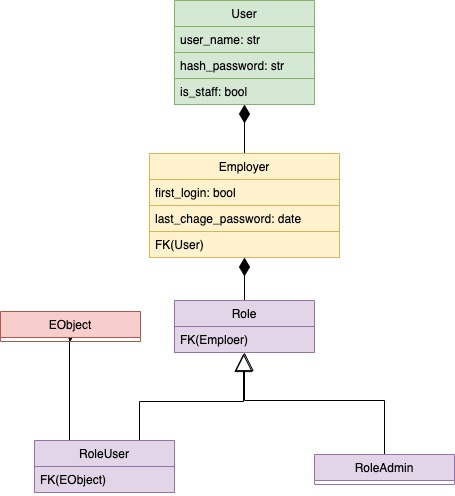


Рисунок 2.3 – Диаграмма классов системы ролей

## Разработка API

Для реализаций функций авторизации и регистрации было разработано следующее API.

POST: /accounts/login

В качестве параметров принимает две строки username(логин) и password(пароль). Для авторизации и входа используются стандартные функции login и authenticate.

Функция authenticate принимает в качестве аргументов логин и пароль и возвращает объект User в случае успешной аутентификации или возвращает ошибку.

Функция login принимает объект запроса и объект User, после чего добавляет идентификатор сессии и отправляет ответ для его сохранения на стороне клиента.

def login(self, request):

try:

username = request.data["username"]

password = request.data["password"]

user = authenticate(username=username,

password=password)

except:

user = None

if user is not None:

login(request, user)

return Response(username + " authorized", status=status.HTTP\_200\_OK)

else:

return Response(

"Проверьте правильность ввода ИНН и пароля. Пароль был выслан Вам на указанный e-mail при регистрации.",

status=status.HTTP\_400\_BAD\_REQUEST)

POST: /accounts/logout

Отправляется без параметров и выполняет выход из системы для пользователя с помощью встроенной функции logout.

def logout(self, request):

logout(request)

return Response("Logout", status=status.HTTP\_200\_OK)

POST: /accounts/register

Принимает в качестве параметров логин и электронную почту пользователя. В качестве результата создает для пользователя аккаунт и объект организации с пустыми данными. После этого отправляет пользователю на почту письмо с информацией для входа в систему.

def register(self, request):

try:

serializer = UserSerializerCreate(data=request.data)

if serializer.is\_valid():

user = serializer.save()

employer = Employer.objects.create(user=user,

first\_login=False)

login(request, user)

return EObjectController.create\_first\_eobject(employer)

except BadRequest400 as br:

return Response({"errors": {"username": br.detail}}, 400)

return Response({"errors": {"username": "Введите ИНН и E-mail"}}, 400)

## Разработка системы разделения доступа к API приложения

Разработанная система ролей используется для разграничения доступа к данным. В классе ModelViewSet, используемом для представления группы API, предусмотрено поле permission\_classes, в которое необходимо поместить список классов наследуемых от BasePermission.

При получении запроса фреймворк вызывает у них методы has\_permission и has\_object\_permission принимающий в качестве аргумента объект запроса, хранящий все необходимые данные для предоставления доступа:

- тип запроса;

- данные, содержащиеся в запросе;

- информация о пользователе, отправившем запрос.

Пример класса, разграничивающего доступ к объектам, хранящим информацию об организации:

class EObjectPartPermission(BasePermission):

def has\_permission(self, request, view):

if (request.method == "POST") or \

((request.method == "PATCH") and ("eobject" in request.data.keys())):

controller = UserPermissionController(request.user)

eobject = EObject.objects.get(pk=request.data["eobject"])

return eobject in controller.get\_avalible\_eobjects()

return True

def has\_object\_permission(self, request, view, obj):

controller = UserPermissionController(request.user)

return obj.eobject in controller.get\_avalible\_eobjects()

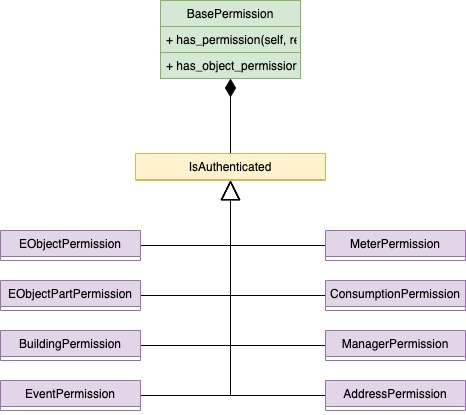


Рисунок 2.4 – Диаграмма классов, разграничивающих доступ к API подсистемы хранения данных об организации

Все классы в данной иерархии наследуются от IsAuthenticated, данный класс определяет, что для доступа к подсистеме пользователь обязан осуществить вход и авторизоваться в системе.

## Разработка подсистемы ввода данных об учреждении

## Разработка структуры базы данных

В результате анализа предметной области были выделены следующие сущности:

Организация – организация, для которой необходимо составить ПЭС у организации есть тип организации, уровень бюджета, управляющий состав (менеджеры) директор, директор в сфере энергосбережения, лицо ответственное за энергосбережение, адрес, ИНН, ОКВЭД, ОГРН, номер телефона, полное название организации, веб-сайт, ОКОГУ, номер энергетического паспорта, сведение о наличие утвержденной ПЭС, организация использует ресурсы, так же на его балансе состоят здания.

Адрес - адрес состоит из почтового индекса, региона, района, населенного пункта, улицы, дома, корпуса.

Менеджер – представляет из себя человека, у него есть фамилия, имя, отчество, должность, номер телефона, электронная почта.

Здание - у здания есть название, описание, местонахождение, площадь, количество этажей, объем. Так же остекление, отопление, информация о праве собственности, стены, чердак и подвал определенного типа, в здании могут быть установлены приборы учета.

Энергетический ресурс – энергетический ресурс, используемый организацией, для его учета может быть установлен прибор учета, он имеет определенный тип, так же хранится информация о том какая РСО его поставляет и о контракте на его поставку.

Тип ресурса. Энергетический ресурс может быть пяти видов Тепло, Электричество, Холодная вода, Горячая вода, Природный газ

Потребление энергетического ресурса - потребление энергетического ресурса как фактическое, так и материальное за определенный год

Прибор учета, устанавливаемый в здании для учета энергетического ресурса, у него есть марка, заводской номер, класс точности, год ввода в эксплуатацию, год последней поверки.

Энергосберегающее мероприятие. Для определенной организации проводятся мероприятия у них есть тип, год выполнения, затраты на внедрение и процентная экономия.

Тип энергосберегающего мероприятия - содержит описание мероприятия, стандартную экономию в процентах и по какому типу ресурса будет экономия.

Транспорт на балансе организации. Организации может принадлежать автотранспорт для него хранится информация о его названии, количестве, грузоподъемности, пробеге, удельный расход топлива норматив, удельный расход топлива факт.

Тип организации - содержит описания типов бюджетной организации.

Тип уровня бюджета - содержит описания уровней финансирования бюджета организации.

Тип владения зданием - содержит описания типов владения, в котором находится здание.

Тип подключения системы теплоснабжения - содержит описания типов системы подключения системы теплоснабжения.

Тип жидкого топлива - содержит описания типов используемого для отопления топлива.

Тип моторного топлива - содержит описания типов используемого моторного топлива.

Тип регуляции отопления - содержит описания типов регуляции отопления.

Тип материала стен - содержит описания типов материала стен здания.

Тип деревянных оконных рам - содержит описания типов деревянных оконных рам, если они установлены.

Тип пластиковых оконных рам - содержит описания типов пластиковых оконных рам, если они установлены.

Тип чердака - содержит описания типов чердака здания.

Тип отопления чердака - содержит описания типов отопления чердака здания.

Тип подвала - содержит описания типов подвала здания

Марка автомобиля - содержит описания марок автотранспорта.

Тип энергоресурса - содержит описания типов энергоресурсов, используемых организацией.

Тип учета ГВС - содержит описания типов учета водоснабжения горячей водой.

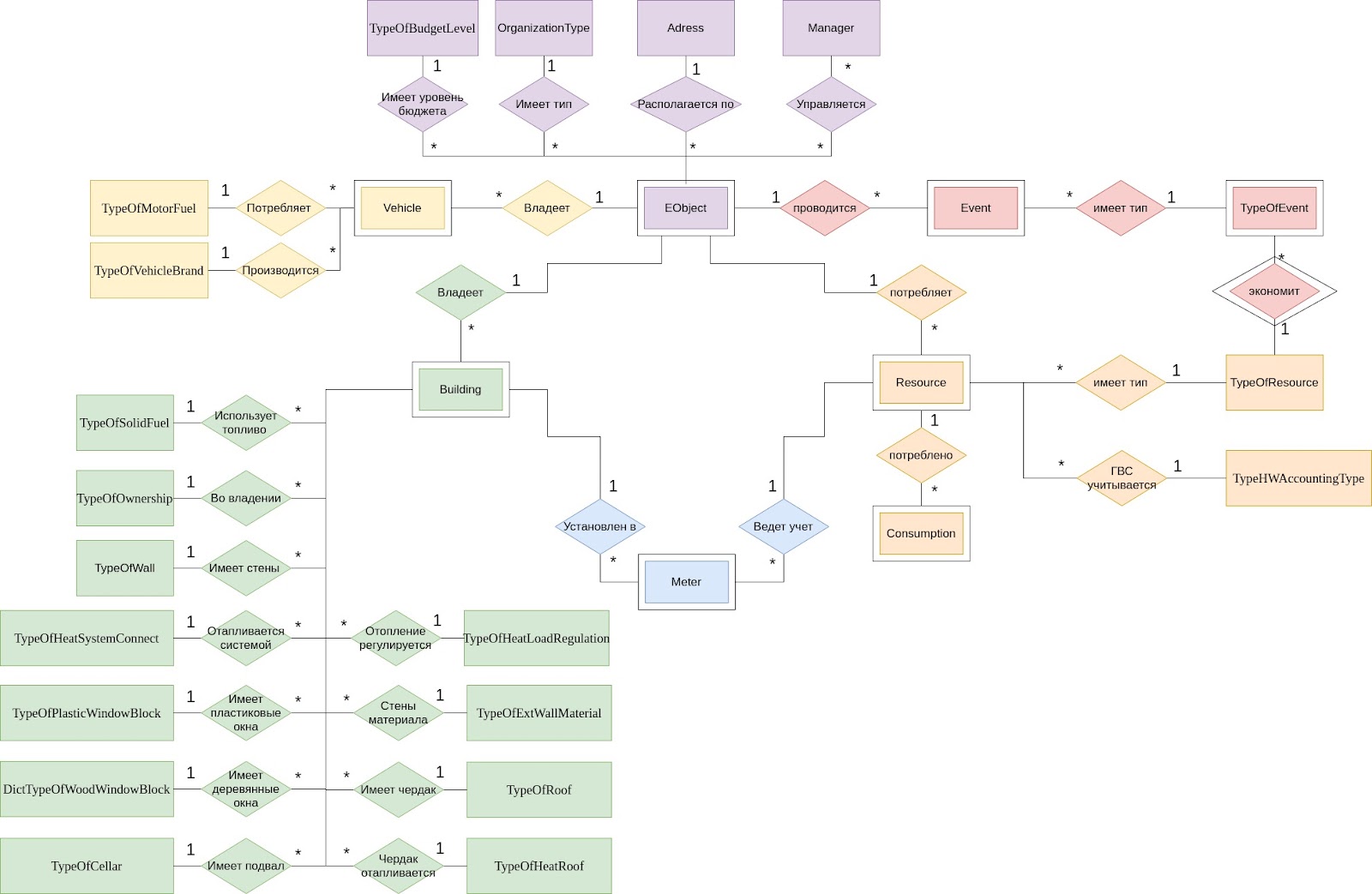


Рисунок 2.5 – Диаграмма сущность связь структуры базы данных

БД соответствует первой нормальной форме, так как все входящие в отношение атрибуты являются атомарными (неделимыми).

БД соответствует второй нормальной форме, так как находится в первой нормальной форме и каждый не ключевой атрибут отношения функционально зависит от первичного ключа.

БД соответствует третьей нормальной форме, так как находится во второй нормальной форме, и каждый не ключевой атрибут не транзитивно зависит от первичного ключа.

## Разработка моделей

Для каждой сущности были разработаны соответствующие модели, описанные на языке Python для встроенное во фреймворк Django ORM. ORM позволит обращаться к сущностям базы данных и строить запросы с помощью Python, тем самым создавая простую абстракцию для работы с базой данных.

Так же это позволит отделить хранение данных от их использования и менять сервер базы данных при необходимости, без изменения кодовой базы.

EObject:

* ID: целочисленный, первичный ключ;
* type\_org: целочисленный, внешний ключ, ID типа организации;
* place\_adress: целочисленный, внешний ключ, ID адреса;
* director: целочисленный, внешний ключ, ID менеджера директора;
* inn: строка, ИНН;
* ogrn: строка, ОГРН;
* okved: строка, ОКВЭД;
* full\_name: строка, полное название организации;
* phone\_number: строка, номер телефона;
* pes\_having: булевой тип, наличие утвержденной программы энергосбережения и повышения энергетической эффективности.

Adress:

* ID: целочисленный, первичный ключ;
* index: строка, почтовый индекс;
* region: строка, регион;
* rajon: строка, район;
* town: строка, населенный пункт;
* building: строка, дом;
* street: строка, улица;
* corpus: строка, корпус.

Manager:

* ID: целочисленный, первичный ключ;
* last\_name: строка, фамилия;
* first\_name: строка, имя;
* position: строка, должность;
* phone: строка, номер телефона;
* email: строка, электронная почта.

Vehicle:

* ID: целочисленный, первичный ключ;
* eobject: целочисленный, внешний ключ, ID организации;
* name: строка, название;
* type\_motor\_fuel: целочисленный, внешний ключ, ID типа топлива;
* type\_brand: целочисленный, внешний ключ, ID производителя;
* count: целочисленный, количество ТС;
* carrying: число с плавающей запятой, объем грузоперевозок (тыс. т-км или пасс-км);
* milleage: число с плавающей запятой, пробег (тыс.км);
* fuel\_rate: число с плавающей запятой, количество потраченного топлива (тыс.л).

Event:

* ID: целочисленный, первичный ключ;
* eobject: целочисленный, внешний ключ, ID организации;
* type\_event: целочисленный, внешний ключ, ID типа мероприятия;
* percentage\_save: число с плавающей запятой, экономия от мероприятия (%);
* year: целочисленный, год проведения;
* cost: целочисленный, фактическая стоимость за год (тыс.руб).

TypeOfEvent:

* ID: целочисленный, первичный ключ;
* type\_resource: целочисленный, внешний ключ, ID вида энергоресурса. Может быть null если мероприятие организационного характера;
* description: строка, описание;
* default\_saved: число с плавающей запятой, базовая экономия.

Building:

* ID: целочисленный, первичный ключ;
* eobject: целочисленный, внешний ключ, ID организации;
* type\_wall: целочисленный, внешний ключ, ID типа стен;
* type\_solid\_fuel: целочисленный, внешний ключ, ID типа жидкого топлива;
* type\_wonership: целочисленный, внешний ключ, ID типа владения;
* type\_heat\_connect: целочисленный, внешний ключ, ID типа подключения отопления;
* type\_plastic\_window: целочисленный, внешний ключ, ID типа пластиковых окон;
* type\_wood\_window: целочисленный, внешний ключ, ID типа деревянных окон;
* type\_cellar: целочисленный, внешний ключ, ID типа чердака;
* type\_heat\_regulation: целочисленный, внешний ключ, ID типа управления отоплением;
* type\_ext\_wall: целочисленный, внешний ключ, ID типа материала стен;
* type\_roof: целочисленный, внешний ключ, ID типа чердака;
* type\_heat\_roof: целочисленный, внешний ключ, ID типа отопления чердака;
* name: строка, наименование здания;
* description: строка, описание;
* place: строка, местонахождение;
* square: число с плавающей запятой, общая площадь;
* space: число с плавающей запятой, общий объем;
* level\_count: целочисленный, количество этажей.

Meter:

* ID: целочисленный, первичный ключ;
* building: целочисленный, внешний ключ, ID здания;
* resource: целочисленный, внешний ключ, ID энергетического ресурса;
* meter\_model: строка, тип и марка прибора;
* accuracy\_class: строка, класс точности";
* date\_commissioning: дата, дата ввода в эксплуатацию;
* factory\_number: строка, заводской номер;
* date\_verification: дата, дата последней поверки.

Resource:

* ID: целочисленный, первичный ключ;
* eobject: целочисленный, внешний ключ, ID организации;
* type\_resource: целочисленный, внешний ключ, ID вида энергоресурса;
* name\_rso: строка, наименование ресурс снабжающей организации;
* contract\_number: строка, номер контракта;
* contract\_date: дата, дата подписания контракта.

Consumption:

* ID: целочисленный, первичный ключ;
* resource: целочисленный, внешний ключ, ID энергоресурса;
* year: целочисленный, год потребления;
* annual\_rate: число с плавающей точкой, фактическое потребление за год (тыс.л);
* annual\_cost: число с плавающей точкой, фактическая стоимость за год (тыс.руб).

DictOrganizationType:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа организации.

DictTypeOfBudgetLevel:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа уровня бюджета.

DictTypeOfOwnership:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа владения.

DictTypeOfHeatSystemConnect:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа подключения системы отопления.

DictTypeOfSolidFuel:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа используемого жидкого топлива.

DictTypeOfMotorFuel:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа использования моторного топлива.

DictTypeOfHeatLoadRegulation:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа регуляции отопления.

DictTypeOfExtWallMaterial:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа материала стен.

DictTypeOfWoodWindowBlock:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа деревянных окон.

DictTypeOfPlasticWindowBlock:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа пластиковых окон.

DictTypeOfRoof:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа чердака.

DictTypeOfHeatRoof:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа отопления чердака.

DictTypeOfCellar:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа подвала.

DictTypeOfVehicleBrand:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа производителя транспорта.

DictTypeOfResource:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа энергоресурсов.

DictTypeOfEvents:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа мероприятий.

DictTypeHWAccountingType:

* ID: целочисленный, первичный ключ;
* description: строка, описание типа учета ГВС.

## Разработка API подсистемы

Для описанных моделей было реализовано CRUD API:

* create (создание);
* read (чтение);
* update (обновление);
* delete (удаление).

Для моделей описывающие различные типы (окон, утепления, материала стен и т. д.) предусмотрено только API для чтения, т. к. они общие для всех пользователей и создаются, а также редактируются и обновляются, администратором системы.

Для упрощения построения API использовался базовый класс фреймворка Django Rest Framework – ModelViewSet. Он позволяет в декларативном стиле описать реализуемое API с помощью следующих полей:

* queryset – список объектов доступных в рамках API;
* http\_method\_names – список http запросов, которые может принимать данный класс;
* serializer\_class – класс используемый для сериализации и десерализации объектов в формат json и обратно соответственно;
* permission\_classes – cписок классов разграничивающих доступ к API.

Список разработанных API:

* /dict/\* - GET запросы используемые для получения списков типов;
* /user\_lk/address – управление адресами;
* /user\_lk/available\_events/get\_available\_event\_for\_eobject – GET запрос для получения списка энергосберегающих мероприятий, которые можно провести в рамках объекта;
* /user\_lk/building – управление информацией о зданиях;
* /user\_lk/eobject – управление информацией об организации;
* /user\_lk/events – управление информацией об энергосберегающих мероприятиях;
* /user\_lk/manager – управление информацией о директорах (менеджерах) организации;
* /user\_lk/meter – управление информацией о приборах учета;
* /user\_lk/resource/consumption – управление информацией о потреблениях энергетических ресурсов;
* /user\_lk/vehicle – управление информацией о транспртных средствах;

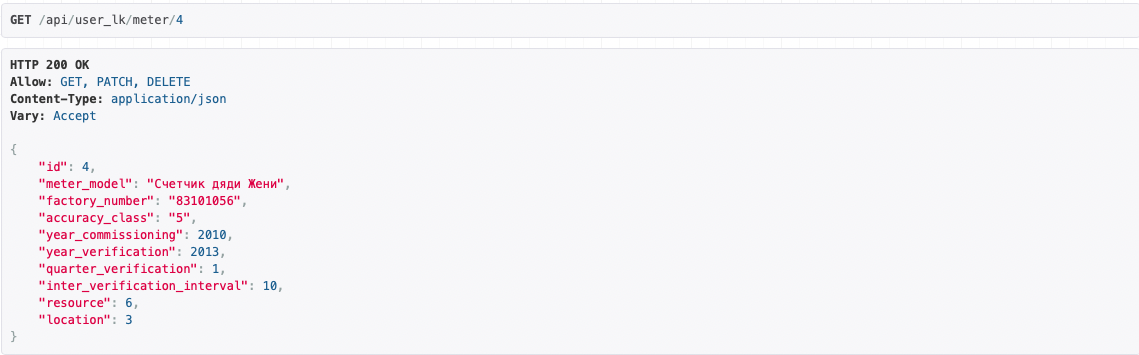


Рисунок 2.6 – Пример получения информации о приборе учета.

## Разработка подсистемы формирования паспорта ПЭС

Подсистема формирования паспорта ПЭС используя данные из подсистемы хранения информации об организации генерирует .docx документ. Для выполнения задачи и сохранения максимальной гибкости подсистемы вся логика поделена на следующие слои:

* получение данных из подсистемы хранения информации об организации;
* агрегирование полученной информации для удобства заполнения шаблона документа;
* заполнение шаблона документа.



Рисунок 2.7 – Общая структура подсистемы.

## Разработка слоя получения данных из подсистемы хранения информации об организации

Для каждой модели, хранящей информацию об организации, были разработаны промежуточные классы (Loader), в конструкторе они получают объект модели и затем предоставляют интерфейс, предоставляющий всю необходимую информацию для агрегатов.

Данный подход используется для разделения формата хранения информации и ее использования, в результате данного решения можно изменять подсистему хранения информации об организации без значительных изменений в подсистеме формирования паспорта ПЭС.

Ниже представлен пример одного из Loader класса предназначенного для получения информации о приборе учета.

class MeterLoader:

def \_\_init\_\_(self, meter: MeterPES):

self.meter = meter

@property

def place(self) -> str:

if self.meter.building is None:

return "Неизвестно"

info\_loader = BuildingLoader(self.meter.building)

return info\_loader.name

@property

def accuracy\_class(self) -> str:

return self.meter.accuracy\_class

@property

def verification\_info(self) -> str:

return ", ".join([str(self.meter.year\_commissioning),

str(self.meter.year\_verification),

str(self.meter.quarter\_verification)])

@property

def verification\_interval(self) -> str:

return str(self.meter.inter\_verification\_interval) + " лет"

Предлагаемый подход так же полезен тем, что не вся необходимая информация хранится в базе данных в требуемом виде и для ее использования производится модификация, как в методах verification\_info, place и interval.

Все методы классов типа Loader отмечены декоратором @property позволяющем обращаться к методу как к полю класса без использования конструкции для вызова функции.

## Разработка слоя агрегирования данных для разделов

В ходе анализа предметной области было выяснено, что паспорт ПЭС состоит из пяти частей:

* заголовка;
* трех приложения;
* дополнительной и справочной информации.

Для удобства заполнения шаблона и генерации документа было принято решение выделить классы агрегирующую информацию, полученную на предыдущем слое. По своей структуре они очень похожи на обычные классы типа Loader, но на выходе мы получаем информацию, структурированную в более крупные объекты, которые и будут использоваться при генерации документа.

Использование классов для агрегирования информации позволяет соединить большое количество разнородной информации в цельные и консистентные объекты более подходящие для использования в языке разметки шаблонизатора jinja2 используемом на следующем этапе. [ВСТАВИТЬ ССЫЛКУ НА Jinja]

## Разработка слоя заполнения шаблона паспорта ПЭС

На данном этапе необходимо внести полученную из агрегаторов информацию в .docx документ.

Для упрощения задачи используется библиотека языка Python3 – docxtpl. Данная библиотека представляет из себя обертку над шаблонизатором jinja2 и позволяет, разметив docx файл, легко заполнить его данными. [ВСТАВИТЬ ССЫЛКУ НА docxtpl]

Для заполнения шаблона данными происходит в четыре этапа:

* создание объекта типа DocxTemplate из библиотеки docxtpl (конструктор принимает путь к файлу, содержащему шаблон);
* получение данных из агрегаторов в виде объекта типа dict языка Python;
* передача объекта в метод render объекта DocxTemplate;
* сохранение файла с помощью метода save.

Перед тем как выполнять заполнение шаблона его так же необходимо создать и разметить. Формат разметки напоминает любой другой язык разметки типа html. Внутри файла имеется доступ к объекту, который был передан в функцию render.

Обращение к полям объекта аналогично обращению в языке python, за исключением того, что его необходимо обрамлять в двойные фигурные скобки.

Так же язык разметки поддерживает и другие стандартные конструкции языков программирования такие как if else, for и т. д., их для использования необходимо обрамлять в конструкцию {% %}.

## Разработка API подсистемы

В качестве API данная подсистема предоставляет один GET HTTP запрос принимающий на вход идентификатор организации и отправляющий в ответ файл составленного паспорта ПЭС. В случае нехватки каких-либо данных API возвращает код ошибки 400 и описание информации, которую необходимо заполнить для корректной генерации.



Рисунок 2.8 – Вид запроса в swagger.

## Разработка подсистемы подбора энергосберегающих мероприятий

Для упрощения разработки плана внедрения энергосберегающих мероприятий была разработана подсистема, позволяющая на основе данных об организации порекомендовать типы энергосберегающих мероприятий, которые принесут максимальную пользу в рамках организации.

В качестве входных параметров подсистема получает данные об организации, а результатом работы подсистемы являются созданные объекты Event (Энергосберегающие мероприятия). В силу того, что расчет фактической стоимости и выбор года внедрения невозможны исходя из имеющихся в системе данных эти поля остаются пустыми.

Для рекомендации энергосберегающих мероприятий используются эвристики, выведенные в ходе анализа предметной области и уже разработанных ПЭС.

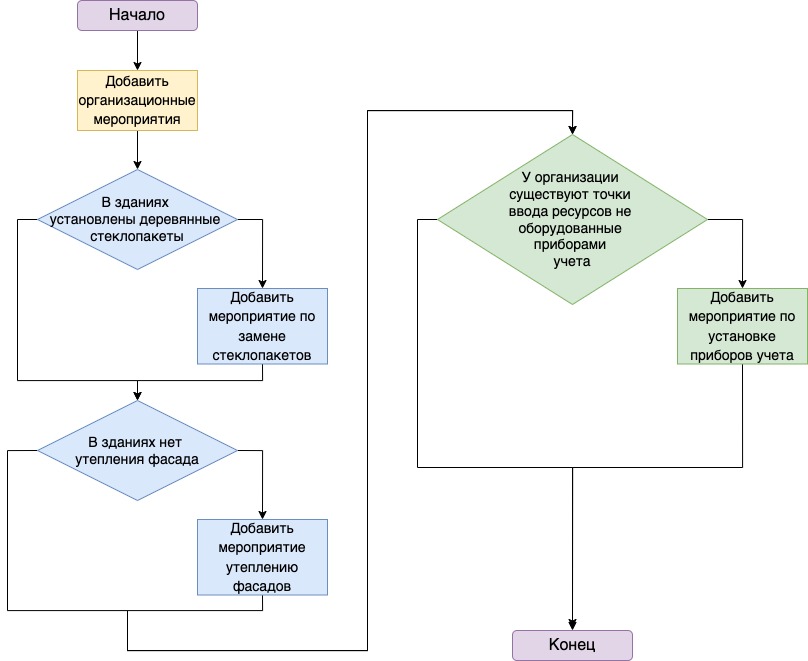


Рисунок 2.9 – Пример некоторых эвристик для рекомендации энергосберегающих мероприятий.

## Разработка подсистемы интеллектуальной валидации ввода данных

Основной сложностью при разработке ПЭС является получение актуальных и верных сведений об учреждении. Современные способы решения данной проблемы предполагают привлечение специалиста в области энергетики и энергосбережения для проверки, введенных пользователем, данных. Для полной автоматизации процесса было решено заменить человека на интеллектуальные средства валидации полученных данных.

Наиболее развитыми методами современного интеллектуального анализа являются методы, основанные на:

ДОБАВИТЬ ССЫЛКИ ИЗ СТАТЬИ ЗА МЕСТО ТЕХ ЧТО В СПИСКЕ

* нечеткой логике [4-6];
* деревьях решений [7-9];
* кластеризации [10, 11];
* искусственных нейронных сетях (ИНС) [8, 12-14].

Для решения поставленной задачи было принято решение использовать ИНС.

## Разработка структуры ИНС

В качестве модели ИНС была создана модель многослойного персептрона. Код модели на языке Python с использованием библиотеки Keras [15, 16] ВСТАВИТЬ ССЫЛКУ НА КЕРАС представлен ниже:

model = Sequential()

model.add(Dense(56, input\_dim=28, activation='elu'))

model.add(Dense(112, input\_dim=56, activation='elu'))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary\_crossentropy', metrics=['accuracy'])

Разработанная модель многослойного персептрона состоит из плотных слоев (Dense):

1. Входного слоя (dense\_1\_input) с 28-ью нейронами.

2. Слоя dense\_1, содержащего функцию активации нейронов elu (28 входов, 56 выходов).

3. Слоя dense\_2, содержащего функцию активации нейронов elu (56 входов, 112 выходов).

4. Выходного слоя dense\_3 с функцией активации sigmoid, содержащего 112 входов и 1 выход.

Сигмовидная функцию активации в слое dense\_3 используется для бинарной классификации, для малых значений функция возвращает значение, близкое к нулю, а для больших значений результат функции приближается к 1.

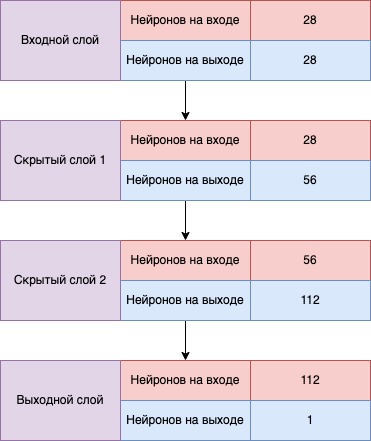


Рисунок 2.10 – Модель многослойного персептрона, выполняющего бинарную классификацию.

## Обучающая выборка и ее параметры

В качестве обучающей выборки, подаваемой на вход в разработанный классификатор, были выбраны данные организаций, которые были вручную проверены специалистом в области энергосбережения. Размер обучающей выборки составил 1912 образов, проверенных и утвержденных как корректные и 300 образов, которые были отмечены как некорректные. В качестве полей для валидации были выделены поля из таблицы 2.1.

Таблица 2.1 – Параметры набора данных.

|  |  |  |
| --- | --- | --- |
| **№** | **Тип данных** | **Смысловое содержание** |
| 1 | Целое без знаковое | Количество вводов по холодной воде |
| 2 | Целое без знаковое | Количество вводов по холодной воде, оборудованных узлами коммерческого учета |
| 3 | Целое без знаковое | Количество вводов по горячей воде |
| 4 | Целое без знаковое | Количество вводов по горячей воде, оборудованных узлами коммерческого учета |
| 5 | Целое без знаковое | Количество вводов тепловой энергии |
| 6 | Целое без знаковое | Количество вводов тепловой энергии, оборудованных узлами коммерческого учета |
| 7 | Целое беззнаковое | Количество вводов электроэнергии |
| 8 | Целое беззнаковое | Количество вводов электрической энергии, оборудованных узлами коммерческого учета |
| 9 | Целое беззнаковое | Количество вводов по газу |
| 10 | Целое беззнаковое | Количество вводов по газу, оборудованных узлами коммерческого учета |
| 11 | Целое беззнаковое | Количество всех сотрудников |
| 12 | Целое беззнаковое | Общая отапливаемая площадь |
| 13 | Вещественное | Общий отапливаемый объем |
| 14 | Вещественное | Полезная площадь |
| 15 | Вещественное | Общая площадь ограждающих конструкций, включая окна |
| 16 | Вещественное | Площадь чердачного перекрытия |
| 17 | Вещественное | Площадь подвальных перекрытий |
| 18 | Целое беззнаковое | Среднее количество людей |
| 19 | Вещественное | Общая площадь |
| 20 | Вещественное | Длина здания |
| 21 | Вещественное | Ширина здания |
| 22 | Вещественное | Высота здания |
| 23 | Вещественное | Периметр здания |
| 24 | Целое беззнаковое | Этажность |
| 25 | Вещественное | Общий объем |
| 26 | Вещественное | Площадь остекления |
| 27 | Вещественное | Остекление энергосберегающими стеклопакетами (% от общего остекления) |
| 28 | Целое беззнаковое | Количество окон |

## Результаты работы подсистемы

На рисунке 2.11 изображены результаты обучения модели на выборке из 1912 верифицированных образов.

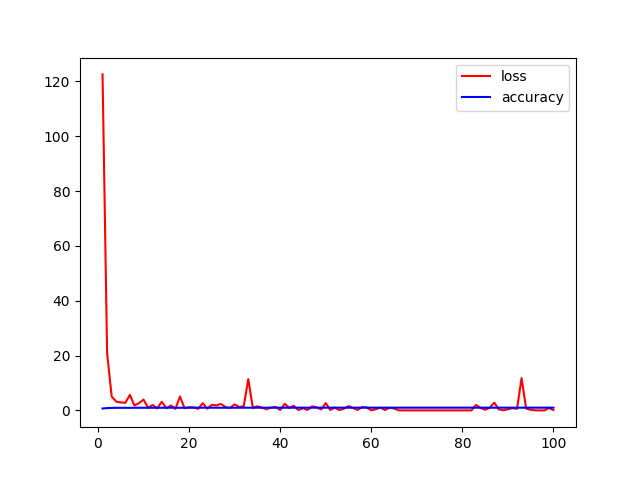


Рисунок 2.11 – Модель многослойного персептрона, выполняющего бинарную классификацию.

Обучение проходило в 100 эпох и размером пакета данных, равным 239. Также на рисунке отражаются такие характеристики ИНС, как точность (accuracy) и потери (loss) для каждой эпохи.

В качестве тестовых данных было взято 4000 образов, хранящих информацию о бюджетных организациях.

Процент ложноположительного срабатывания (0.28%) и ложноотрицательного (0.5%) относительно мал. Точность классификации равна 0.997.

Приведенные результаты точности распознавания (99.7%) разработанным классификатором на основе многослойного персептрона говорят о том, что данный подход может нести только рекомендательный характер, следовательно пользователь в любом случае должен перепроверять вводимые данные, но разработанная подсистема поможет обнаруживать ошибки и оперативно их исправлять.

**Вывод**

В данной главе были определены основные задачи разработки серверной части ПО. Так же были реализованы все ключевые подсистемы серверной части, разработана структура базы данных и налажено взаимодействие между подсистемами.

Подсистемы реализовывались с помощью языка программирования Python и фреймворка Django Rest Framework. В качестве системы управления базой данных был использован PostgreSQL.

В результате работы, проделанной в данной главе, была реализована серверная часть ПО, а так же REST API используемый клиентской частью ПО, описанной в следующей главе.

## РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ СИСТЕМЫ ФОРМИРОВАНИЯ ПЭС

Для разработки клиентской части системы требуется решить ряд задач:

* разработка общей структуры клиентской части ПО;
* разработка подсистемы взаимодействия с API;
* разработка подсистемы роутинга;
* разработка подсистемы для пред показа паспорта ПЭС;
* разработка контейнеров;
* разработка компонентов.

## Разработка общей структуры клиентской части

В рамках клиентской части выделим подсистемы:

* + подсистема взаимодействия с API;
  + подсистема роутинга;
  + подсистема для пред показа паспорта ПЭС;
  + контейнеры;
  + компоненты.

В зависимости от необходимости модули клиентской части могут взаимодействовать:

* между собой;
* с пользователем;
* с серверной частью по средствам API.

На рисунке 2.3 изображена схема взаимодействия между компонентами клиентской части.

****

Рисунок 3.1 – Схема взаимодействия подсистем клиентской части

При открытии WEB-приложения в браузере пользователь использует URL ресурса, по которому осуществляется http запрос. В зависимости от адреса по которому переходит пользователь WEB-приложение должно отображать различные экранные формы и страницы, для сопоставления URL и контента, предоставляемого пользователю, используется подсистема роутинга.

При проектировании клиентской части используется компонентно-ориентированный подход, при котором пользовательский интерфейс делится на небольшие пере используемые компоненты.

Для объединения компонентов в страницы используется контейнеры, так же в контейнеры выносится вся логика управления данными для упрощения проектирования и использования компонентов.

Паспорт ПЭС представляет из себя .docx файл и для его просмотра необходимо свернуть браузер и открыть скачанный файл. Для того что бы упростить процесс и сделать его более бесшовным используется подсистема пред просмотра паспорта ПЭС, позволяющая ознакомится с содержанием сгенерированного файла внутри браузера.

Для разделения отображения данных в компонентах и манипуляций с ними используется подсистема взаимодействия с API, представляющая из себя использования паттерна адаптер и позволяющая в любой момент изменить API без изменения остальной кодовой базы.

# ЗAКЛЮЧЕНИЕ

# СПИСOК ЛИТЕРAТУРЫ

1. Экономическая теория / ред. Видяпин В.И. Добрынин А.И., Журавлёва Г.П., Тарасевич Л.С. – Москва : ИНФРА-М, 2019. – 714 с. – ISBN 978-5-16-003507-9
2. Бурмистров А.Н. Формирование модернизационной стратегии развития предприятий пищевой промышленности / авт. Бурмистров А.Н. – Саратов, 2011. – 162 с.
3. ГОСТ Р 53041-2008. Изделия кондитерские и полуфабрикаты кондитерского производства [Текст]. – Москва: Стандартинформ, 2019. – 16 с.
4. Кондитерская промышленность. [Электронный ресурс]. Кондитерская промышленность на территории России. – Электрон, дан., 2019. – Режим доступа: <https://fabricators.ru/article/konditerskaya-promyshlennost>
5. Бутейкис Н. Г. Технология приготовления мучных кондитерских изделий / Н. Г. Бутейкис, А.А. Жукова – 3-е изд., стереотипное – Москва: ACADEMA, 2015. – 302 с. – ISBN 978-5-7695-6733-9
6. Толкаченко, А. А. Правовая (судебная) бухгалтерия: курс лекций / А. А. Толкаченко, К. В. Харабет; ред. Н. А. Петухов. – 3-е изд. – Москва: Омега-Л, 2004. – 216 с. – (Библиотека высшей школы). – ISBN 5-98119-281-0
7. Оперативный учёт. [Электронный ресурс]. Библиотека банковских документов. – Электрон, дан., 2020. – Режим доступа: http://www.bbdoc.ru
8. Хозяйственный учёт. [Электронный ресурс] Хозяйственный учёт, его сущность и значение. – Электрон, дан., 2017. – Режим доступа: https://buhbook.net/buhgalterskij-uchet/teoriya-buhgalterskogo-ucheta/hozyaystvennyiy-uchet-ego-sushhnost-i-znachenie/
9. Оперативный учёт: цели, особенности, измерители. [Электронный ресурс] Оперативный учёт. – Электрон, дан., 2017. – Режим доступа: <https://www.audit-it.ru/terms/accounting/operativnyy_uchet.html>
10. Оперативный учёт. [Электронный ресурс]. Оперативный учёт. Оперативный счёт. – Электрон, дан., 2020. ­ – Режим доступа: <https://bankfs.ru/deposits/operativnyi-schet-operativnyi-uch-t-buhgalterskii-uchet-v.html>
11. Кузнецова Л. С Технология приготовления мучных кондитерских изделий: учебн. для студ. учреждений сред. проф. образования. / Л.С. Кузнецова, М. Ю Сиданова – Москва: ACADEMIA, 2013. – 320 с. – ISBN 978-5-7695-9525-7
12. ERP-система. [Электронный ресурс]. Что такое ERP-система. – Электрон, дан., 2017. ­ – Режим доступа: <https://trinion.org/articles/chto-takoe-erp-sistema>
13. Корякин А. О. Разработка клиентской части программного обеспечения учёта производства кондитерских изделий [Отчет]. – Белгород, 2020.
14. Фаулер М. Архитектура коропративных программных приложений / М. Фаулер – Москва : Издательский дом «Вильяме», 2018. – 540 с. – ISBN 978-5-6041394-0-0
15. Язык программирования Java SE 8. Подробное описание / Д. Гослинг, Б. Джой, Г. Стил и др. – 5-е изд. – Москва: «Вильямс», 2015. – 672 с. – ISBN 978-5-8459-1875-8
16. Маншин Т. С. Web-сервисы Java / Т. С. Маншин – Санкт-Петербург: БХВ-Петербург, 2012. – 560 с. – ISBN 978-5-9775-0778-3
17. Webber J. REST in Practice / J. Webber, S. Parastatidis S., I. Robinson – O'Reilly Media, Inc., 2010. – First Edition – 448 c. – ISBN: 978-0-596-80582-1
18. Коннолли Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг / ред. Н. С. Тригуб – 3-е изд., – Москва: Вильямс, 2017. – 1440 с. – ISBN 978-5-8459-2020-1
19. Дейт К. Введение в системы баз данных / К. Дейт / ред. К.А. Птицына / перев. К.А. Птицына. – Москва: Диалектика, 2019. – 1328 c. – ISBN 978-5-8459-0788-2
20. PostgreSQL. [Электронный ресурс]. Manual Archive. – Электрон, дан. 2020. ­ – Режим доступа: <https://www.postgresql.org/docs/manuals/archive/>
21. Национальная библиотека им. Н. Э. Баумана. [Электронный ресурс]. PostgreSQL. – Электрон, дан., 2019.  – Режим доступа: <https://ru.bmstu.wiki/PostgreSQL>
22. Уоллс К. Spring в действии / К. Уоллс / ред. Д.А. Мовчан. – Москва: ДМК Пресс, 2015. – 752 c. – ISBN 978-5-97060-171-6
23. Consul.io. [Электронный ресурс]. Habr: Consul.io, часть 1. — Электрон, дан., 2016. — Режим доступа:: <https://habr.com/en/post/278085>
24. DTO. [Электронный ресурс]. DTO vs POCO и Value Object . – Электрон, дан., 2016.  – Режим доступа: <https://habr.com/ru/post/268371/>
25. MapStruct. [Электронный ресурс]. MapStruct: Frequently Asked Questions – Электрон, дан., 2020.  – Режим доступа: <https://mapstruct.org/>
26. The Postman API Platform. [Электронный ресурс]. Create API Documentation with Postman. – Электрон, дан., 2020.  – Режим доступа: <https://postman.com/>

# ПРИЛOЖЕНИЕ А

Основные запросы к системе

1. Результат запроса на получение данных о сырье