**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

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

**ЦЕНТР АВТОРИЗОВАННОГО ОБУЧЕНИЯ ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ**

**ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА**

**Разработка веб-приложения «DocMan» для создания системы электронного документооборота проектной документации с возможностью согласования документов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Автор | | Чижов Эдуард Константинович |  |
|  |  | |  | (Подпись) |
|  | **Центр авторизованного обучения информационным технологиям** | | | |
|  | Наименование программы **«Python-разработчик»** | | | |
|  |  | |  |  |
|  | Руководитель | | Кузьмин Константин Михайлович |  |
|  | |  |  | (Подпись) |

**К защите допустить**

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

Санкт-Петербург, 2023г.

Обучающийся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Группа 124/02

(Фамилия, И. О.)

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

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

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

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

(Фамилия, И. О.) (подпись)

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

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

**Содержание**

[1. Введение 5](#_Toc141199368)

[2. Анализ электронного документооборота проектной документации 6](#_Toc141199369)

[2.1 Пользовательские истории 7](#_Toc141199370)

[2.2 Определение функциональных требований 7](#_Toc141199371)

[2.3 Диаграмма Use Case 8](#_Toc141199372)

[2.4 Модель предметной области 10](#_Toc141199373)

[2.5 Функциональная модель 11](#_Toc141199374)

[2.5.1 Контекстная диаграмма 11](#_Toc141199375)

[2.5.2 Диаграмма декомпозиции 1-го уровня 11](#_Toc141199376)

[2.5.3 Диаграмма декомпозиции 2-го уровня 12](#_Toc141199377)

[3. Описание системы 13](#_Toc141199378)

[3.1 MySQL 13](#_Toc141199379)

[3.2 Python 13](#_Toc141199380)

[3.3 Django 14](#_Toc141199381)

[3.4 HTML и CSS 16](#_Toc141199382)

[4. Структура проекта 16](#_Toc141199383)

[4.1 Приложение «Аутентификация» 16](#_Toc141199384)

[4.2 Приложение «Менеджер документов» 17](#_Toc141199385)

[4.3 Приложение «Согласование» 18](#_Toc141199386)

[5. Функции системы 18](#_Toc141199387)

[5.1 Аутентификация 18](#_Toc141199388)

[5.1.1 Login 18](#_Toc141199389)

[5.1.2 Logout 19](#_Toc141199390)

[5.2 Менеджер документов 19](#_Toc141199391)

[5.2.1 Просмотр всех документов по каждому проекту 19](#_Toc141199392)

[5.2.2 Просмотр карточки документа 19](#_Toc141199393)

[5.2.3 Создание карточки документа 20](#_Toc141199394)

[5.2.4 Работа с прикреплёнными файлами 20](#_Toc141199395)

[5.3 Согласование 22](#_Toc141199396)

[5.3.1 Присоединение процедуры согласования документа 22](#_Toc141199397)

[5.3.2 Просмотр деталей процедуры согласования 23](#_Toc141199398)

[5.3.3 Работа с замечаниями 24](#_Toc141199399)

[5.3.4 Согласование документа 25](#_Toc141199400)

[6. Структура данных 28](#_Toc141199401)

[7. ПРАКТИЧЕСКАЯ ЗНАЧИМОСТЬ РАБОТЫ 29](#_Toc141199402)

[8. Заключение 29](#_Toc141199403)

[9. Сокращения 30](#_Toc141199404)

[10. Терминология 30](#_Toc141199405)

[11. Литература 31](#_Toc141199406)

1. Введение

В современных условиях, когда всё более востребованной становится чистая, недорогая энергетика, значительно увеличивается потребность в проектировании и строительстве атомных электростанций. Ввиду того, что во всём мире существует малое ограниченное количество проектных организаций, способных выполнять такие масштабные и узкоспециализированные проекты, значительно возрастает спрос на их услуги, и как результат возрастает количество разрабатываемых проектов и соответствующей им документации.

Вследствие того, что проект атомной электростанции подразумевает под собой колоссальный объём работы и выпускаемых документов, которые должны пройти все необходимые проверки и согласования, чтобы обеспечить максимальный уровень безопасности станции, особую актуальность приобретает создание системы электронного согласования, управления и хранения документации.

В рамках данной итоговой аттестационной работы разработана информационная система "DocMan" (Document Manager), которая предоставляет функционал для загрузки и просмотра пользователями документации по проектам, её централизованного хранения и прохождения процедуры согласования и утверждения документации в электронном виде.

Применение данной системы позволит не только сократить время, требуемое на согласование документации, но также предоставит постоянный доступ всем сотрудникам организации к актуальным документам, что повысит общую согласованность при проектировании и поможет избежать коллизий и несоответствий между смежными конструкциями и технологическими системами [1].

Целью данной дипломной работы является создание информационной системы, которая позволяет централизованно хранить документацию, соответствующею конкретному объекту проектирования, а также присваивать документу процедуру согласования с возможностью назначения определенных ролей для экспертов, отвечающих за контроль качества.

Для достижения данной цели следует решить следующие задачи:

1. Собрать и проанализировать требования к проектируемому ПО, предъявляемые Заказчиком и потенциальными пользователями;
2. Разработать Базу Данных для хранения информации о пользователях, проектах, документах и процедурах согласования, ролей;
3. Разработать web-приложение, позволяющее пользователям работать с информацией из Базы Данных.
4. Анализ электронного документооборота проектной документации

Одна из проектных организаций госкорпорации «Росатом» занимается проектированием атомных электростанций. Учитывая большое количество выпускаемой проектной документации, а также необходимость согласования выпускаемой документации с большим количеством специалистов, возникла потребность в сокращении временны́х затрат на согласование и выпуск документации. Поскольку в данный момент документооборот происходит с помощью двух отдельных приложений, было принято решение о создании единой системы электронного документооборота, которая позволила бы централизованно хранить всю выпускаемую документацию, а также проходить процедуры согласования и выпуска документации в электронном виде.

Для этого необходимо, чтобы разрабатываемое приложение удовлетворяло требованиям сдачи документации, обусловленным спецификой проектирования объектов ядерной энергетики, и требованиям, выдвигаемым Заказчиком [2].

В настоящий момент есть множество программ, которые отвечают за электронный документооборот. Например, «DocVision», которая по сути является архивом для различных видов документации, «SAP» или «EOS for Microsoft SharePoint», которые предназначены для корпоративного управления. Таких программ достаточно на рынке, но они не учитываю специфику проектирования объектов атомной отрасли и не могут удовлетворить все потребности, которые возникают при сдаче проектной документации, а именно специфический этап согласования документации внутри организации для дальнейшей ее передачи Заказчику, и последующего просмотра документации Заказчиком.

Также стоить отметить, что в организации уже имеются два приложения, которые отвечают за электронный документооборот, но из-за того, что ни одна из них не сочетает в себе весь рабочий функционал, возникает необходимость разработать приложение, которые будет объединять в себе две уже существующих системы по работе с документацией.

Таким образом **бизнес-целью является** снижение временны́х затрат на согласование и выпуск проектной и рабочей документации.

Для сотрудников проектной организации, которым нужно выпускать рабочую и проектную документации, данная Система управления проектной документацией является информационной системой, которая обеспечит единую точку доступа ко всей проектной документации проекта. Система будет хранить комплекты документации по проекту в просматриваемом формате. Эта система сократит временны́е затраты на согласование и выпуск готовой документации, позволив проходить процедуру согласования в электронном виде и передавать согласованный документ в технический архив.

На сегодняшний день в организации-заказчике действует два отдельных приложения: одно приложение отвечает за согласование документации, а другое - за хранение, поэтому сотруднику необходимо дважды проделывать одну и ту же работу: сначала загружать документацию для прохождения процедуры согласования, а уже после прохождения всех этапов согласования и отработки замечаний по ним, загрузить согласованную документацию в отдельную программу для хранения всех документов по объекту проектирования.

В отличие от действующих приложений разрабатываемый проект соединяет в себе эти две функции и предоставляет возможности для электронного прохождения необходимых согласований, а также позволит в режиме реального времени отслеживать текущий статус каждого документа и хранить его в системе.

## 2.1 Пользовательские истории

Роли:

1. сотрудник ГУП (группы управления проектом), повышенные привилегии доступа;
2. инженер (Инженер-разработчик проектной документации);
3. согласующий (Инженер, проверяющий документацию в зоне своей ответственности).

Пользовательские истории:

1. как сотрудник ГУП, я хочу иметь возможность создавать, изменять карточки проектных документов;
2. как инженер, я хочу загружать PDF-файл в карточку документа;
3. как инженер, я хочу запускать процедуру согласования моего документа;
4. как инженер, я хочу выбирать конкретных согласующих из списка сотрудников предприятия;
5. как согласующий, я хочу выставлять замечания к документу;
6. как согласующий, я хочу либо согласовывать документ, либо возвращать его на доработку инженеру с замечаниями.

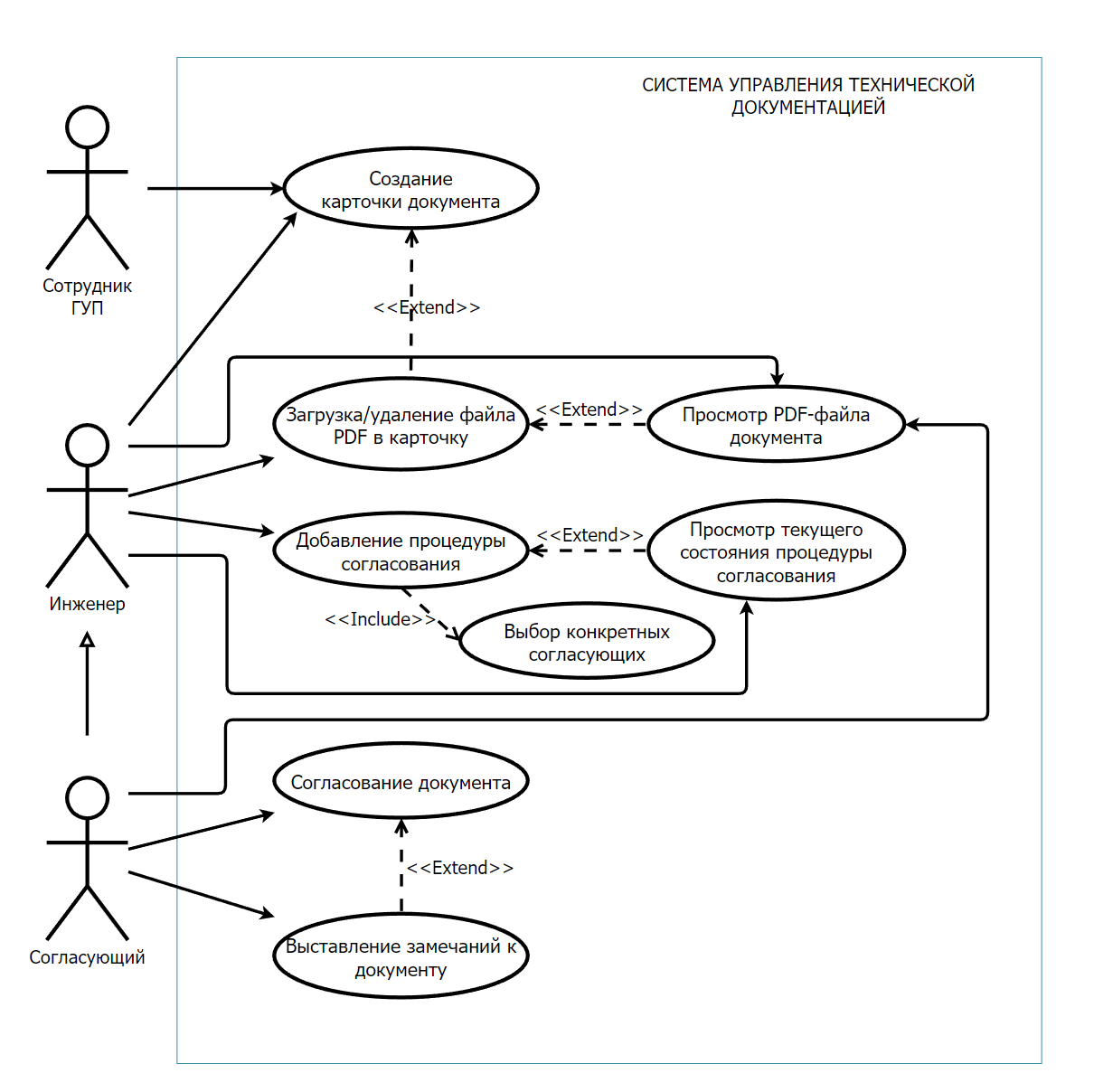
## 2.2 Определение функциональных требований

Основные функциональные требования приведены в таблице 1.

*Таблица 1* – *Функциональные требования*

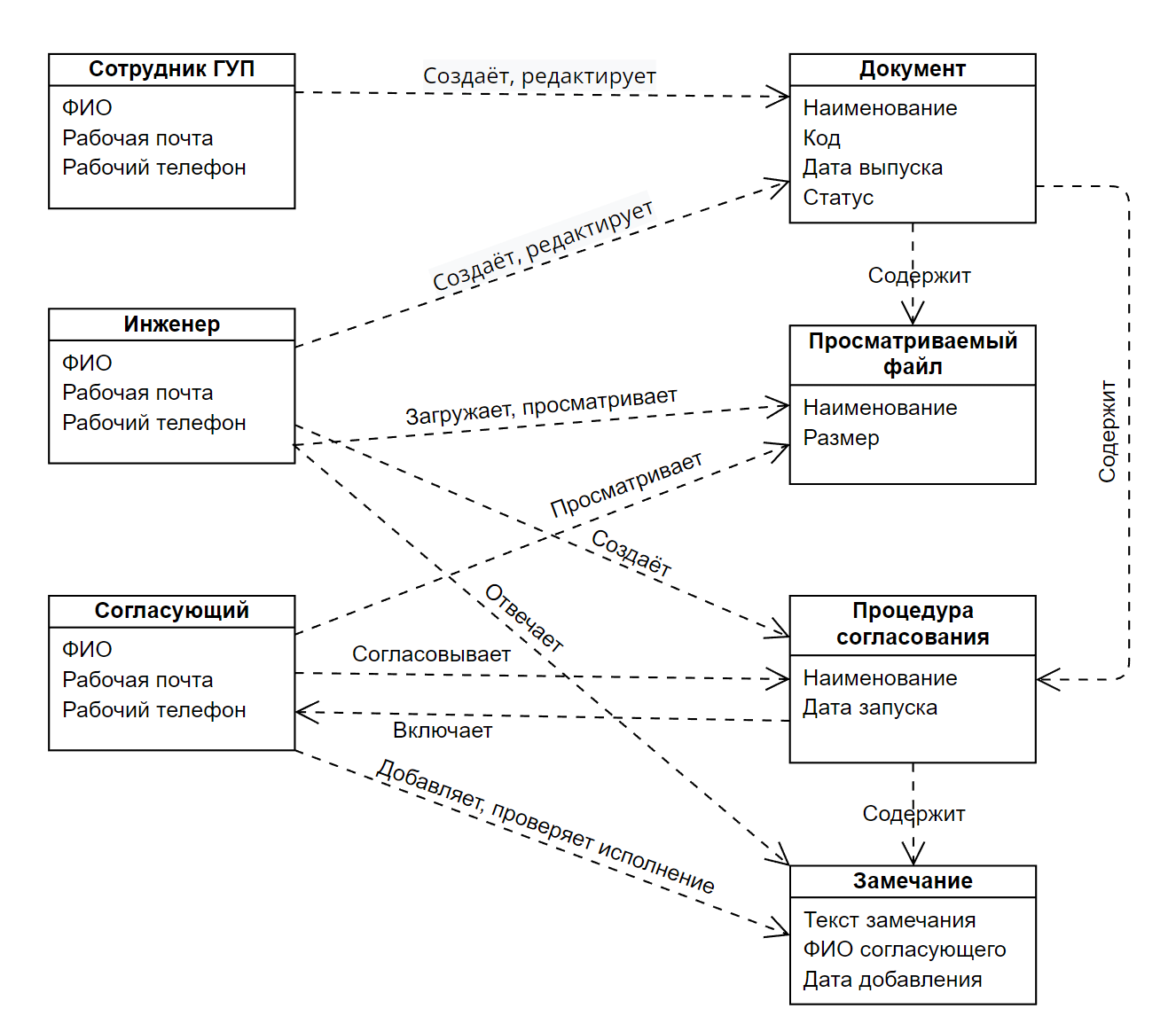
| **ID** | **Описание «рамки решения»** | **ID** | **Описание «рамки проекта»** |
| --- | --- | --- | --- |
| ***Функция: Управление карточками документов*** | | | |
| 1 | Система обеспечивает возможность пользователям создавать, изменять и удалять карточки документов | 101 | Создание меню создания карточки документа с возможностью последующего изменения или удаления |
| ***Функция: Просмотр документов*** | | | |
| 2 | Система обеспечивает возможность поиска документа по уникальному коду | 201 | Создание меню «Поиск» |
| 202 | Реализация просмотра карточки документа |
| 3 | Система позволяет просматривать вложенные в карточку файлы PDF, DOC | 301 | Реализовать открытие вложенных в карточку файлов PDF, DOC |
| ***Функция: Загрузка файлов в карточку документа*** | | | |
| 4 | Система обеспечивает возможность разработчику документа добавлять файлы в карточку документа | 401 | Создание меню загрузки и удаления файла с рабочего ПК в карточку документа |
| ***Функция: Процедура согласования*** | | | |
| 5 | Система обеспечивает возможность пользователю управлять процедурой согласования документа | 501 | Реализовать добавление к документу процедуры согласования |
| 502 | Добавить возможность выбора конкретных сотрудников для конкретного шага процедуры согласования |
| 503 | Реализовать просмотр текущего состояния процедуры согласования |
| 6 | Система обеспечивает возможность согласующему согласовать документ или выставить замечания | 601 | Создание меню согласования документа |
| 602 | Создание меню выставления замечаний |

## 2.3 Диаграмма Use Case

Диаграмма Use Case представлена на рисунке 1.

*Рисунок 1* – *Use Case диаграмма*

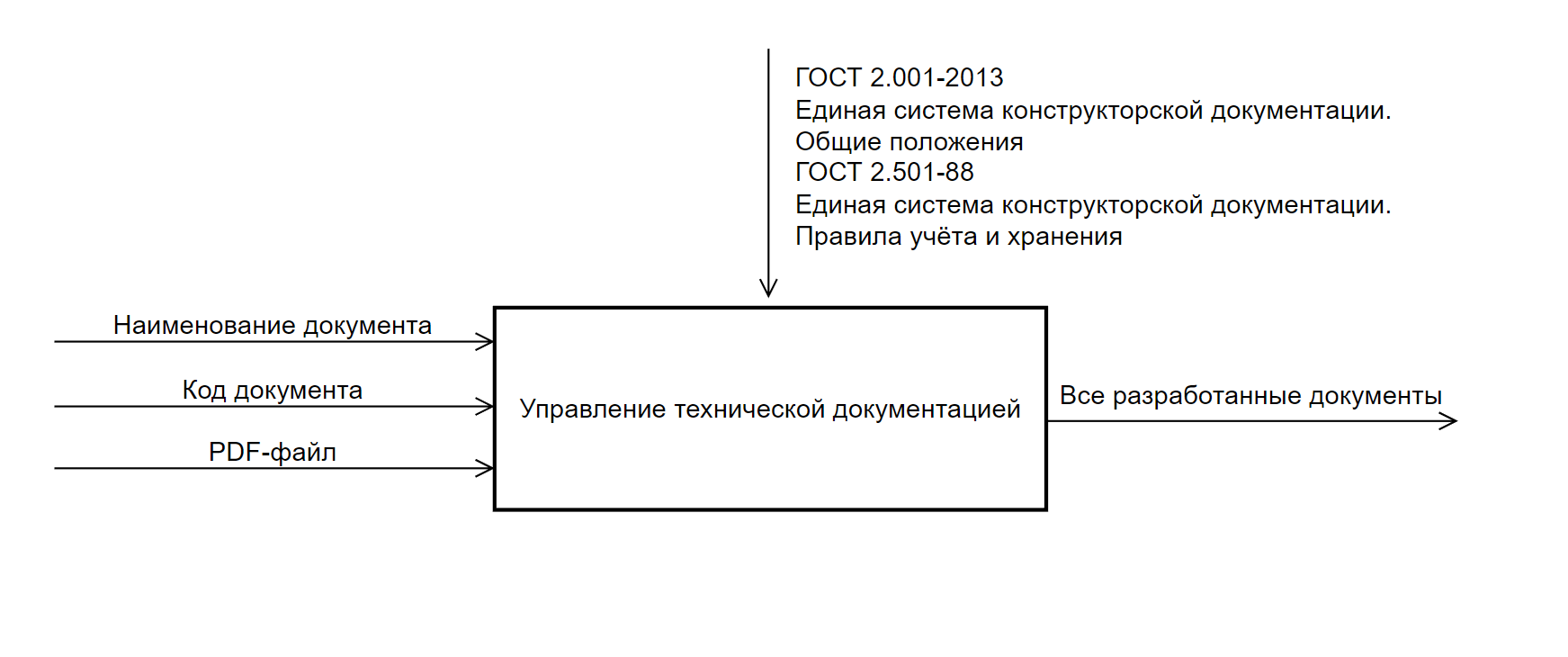
## 2.4 Модель предметной области

Модель предметной области представлена рисунке 2.

*Рисунок 2* – *Модель предметной области*

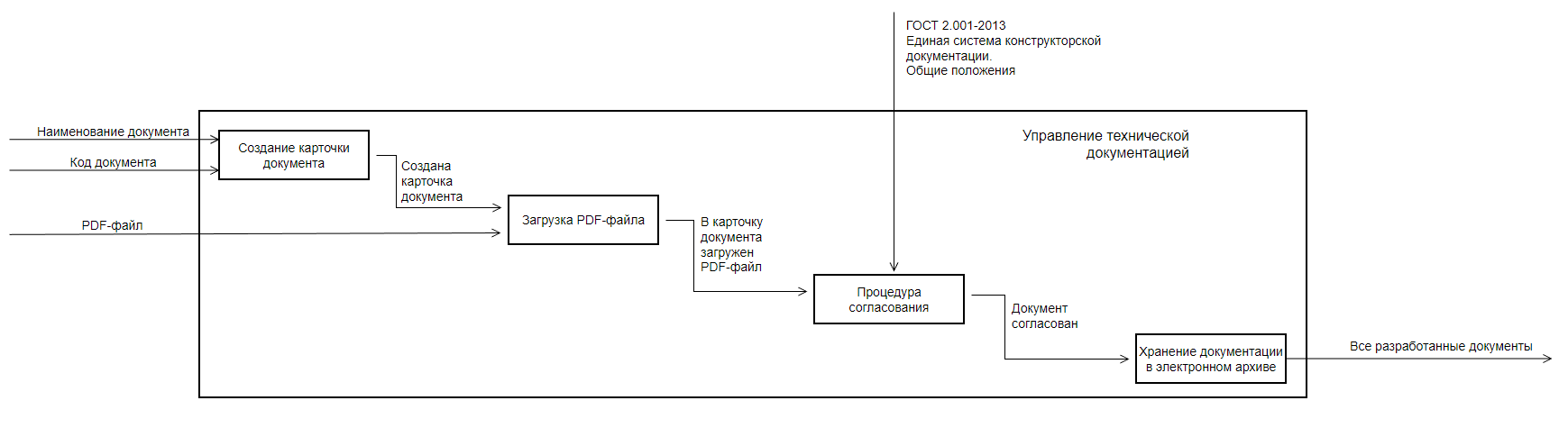
## 2.5 Функциональная модель

### 2.5.1 Контекстная диаграмма

Контекстная диаграмма представлена на рисунке 3.

*Рисунок 3* – *Контекстная диаграмма*

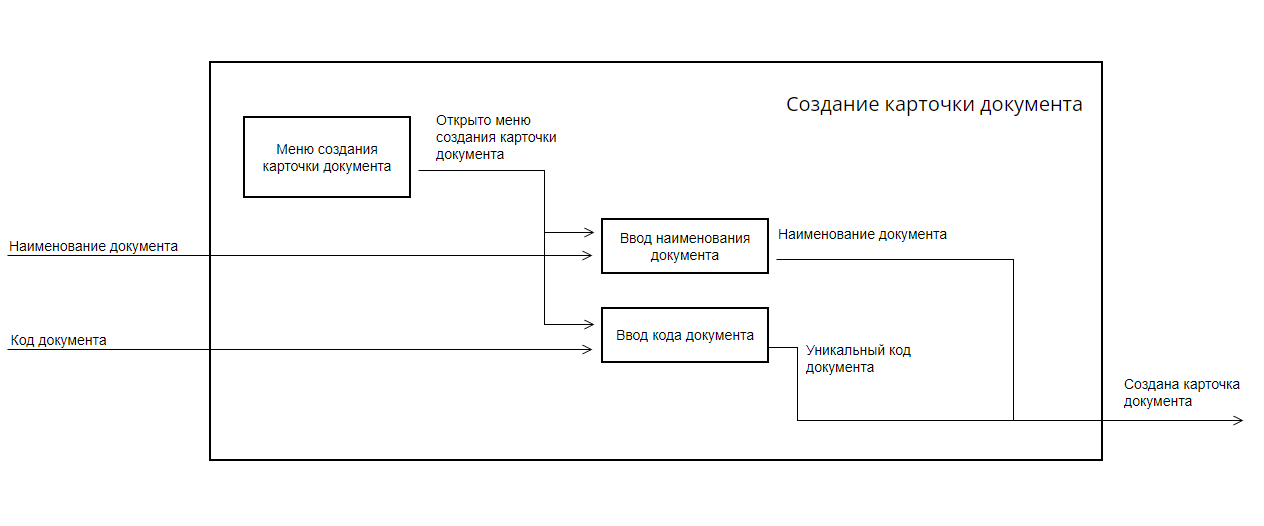
### 2.5.2 Диаграмма декомпозиции 1-го уровня

Диаграмма декомпозиции 1-го уровня «Управление проектной документацией» представлена на рисунке 4.

*Рисунок 4* – *Диаграмма декомпозиции 1-го уровня*

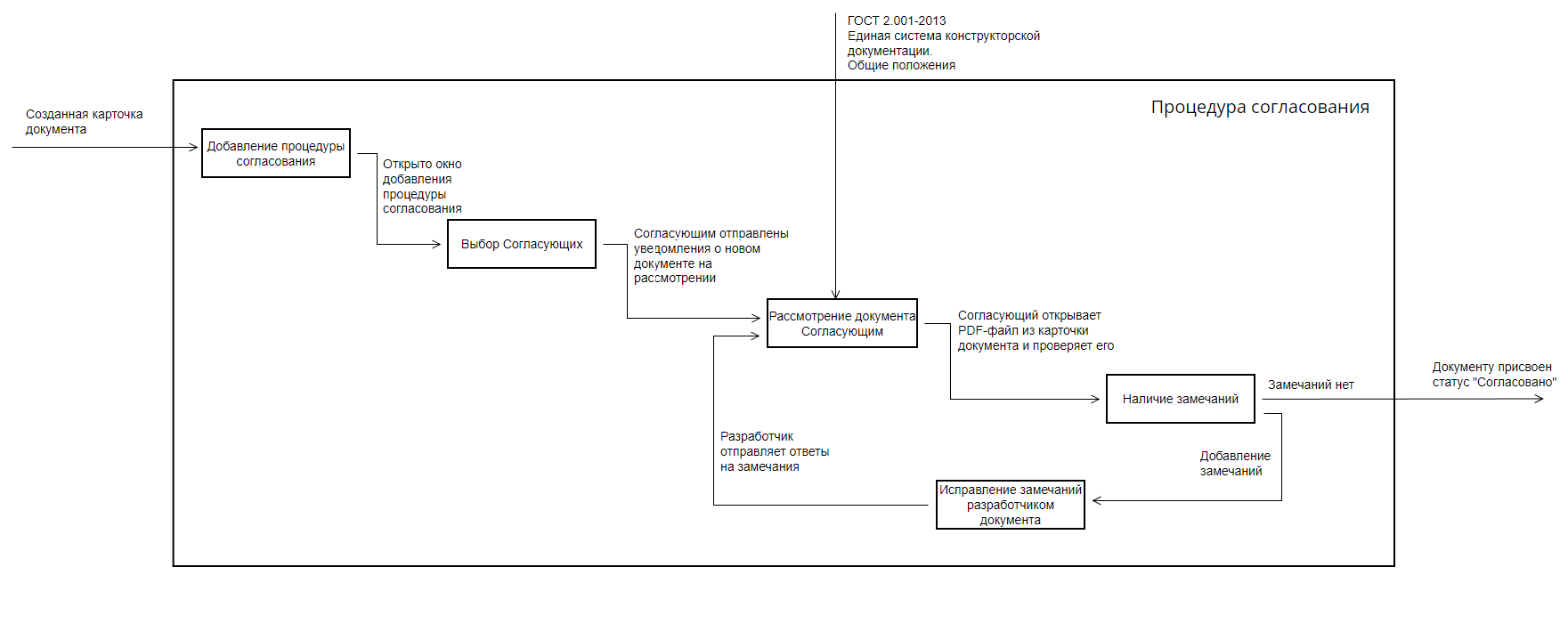
*«Управление проектной документацией»*

### 2.5.3 Диаграмма декомпозиции 2-го уровня

Диаграмма декомпозиции 2-го уровня «Создание карточки документа» представлена на рисунке 5.

*Рисунок 5* – *Диаграмма декомпозиции 2-го уровня*

*«Создание карточки документа»*

Диаграмма декомпозиции 2-го уровня «Процедура согласования» представлена на рисунке 6.

*Рисунок 6* – *Диаграмма декомпозиции 2-го уровня*

*«Процедура согласования»*

# Описание системы

Разработанная в рамках данной аттестационной работы система управления проектной документацией представляет собой веб-приложение, созданное на языке Python [4, 5] с использованием фреймворка Django [3, 6]. Пользовательский веб-интерфейс выполнен с использованием HTML и языка разметки CSS [8, 9, 10]. Хранение данных реализовано при помощи реляционной базы данных MySQL [7].

## MySQL

Для хранения данных о пользователях, проектах, документах и согласованиях в данной работе выбрана одна из самых популярных и широко используемых СУБД - MySQL.

MySQL - это мощная и надежная реляционная СУБД, которая обеспечивает высокую производительность и масштабируемость для решения различных задач, связанных с хранением и обработкой данных, она хорошо подходит для проектов, требующих обработки больших объемов данных. Кроме того, MySQL является бесплатной и открытой СУБД, что делает ее доступной для широкого круга пользователей.

Реляционные базы данных - это тип баз данных, которые используют реляционные модели данных, основанные на отношениях между таблицами. Они широко используются в проектах, так как они обеспечивают высокую гибкость, масштабируемость и надежность хранения и управления данными. Реляционные базы данных могут обрабатывать большие объемы данных быстро и эффективно, что может положительно сказаться на быстродействии системы в целом.

Реляционные базы данных также обеспечивают хорошую поддержку целостности данных и их безопасности. Это важно для проектов, где необходимо обеспечить точность и достоверность данных. Кроме того, реляционные базы данных легко интегрируются с другими системами и приложениями, что позволяет легко обмениваться данными и интегрировать различные функции.

## Python

Python является высокоуровневым языком программирования, поддерживающим принципы императивного, функционального и объектно-ориентированного программирования. На сегодняшний день он является одним из самых популярных языков программирования, который широко используется в различных областях.

Среди основных преимуществ языка Python можно выделить:

* Относительно простой, понятный и лаконичный синтаксис, что делает язык доступным для начинающих программистов и позволяет решать нетривиальные задачи, не имея большого опыта программирования;
* Обширная библиотека пакетов, которые предоставляют доступ к различным инструментам и функциям;
* Возможность как свободно использовать большое количество существующих модулей, так и создавать собственные модули и использовать их в разных проектах;
* Кроссплатформенность. Python может быть использован на различных операционных системах, таких как Windows, macOS и Linux.
* Язык с открытым исходным кодом, что позволяет разработчикам свободно использовать и расширять его возможности;
* Все сущности в языке Python является объектами, однако объектно-ориентированных подход не навязывается разработчику;
* Наличие инструментов для простой и функциональной веб-разработки.

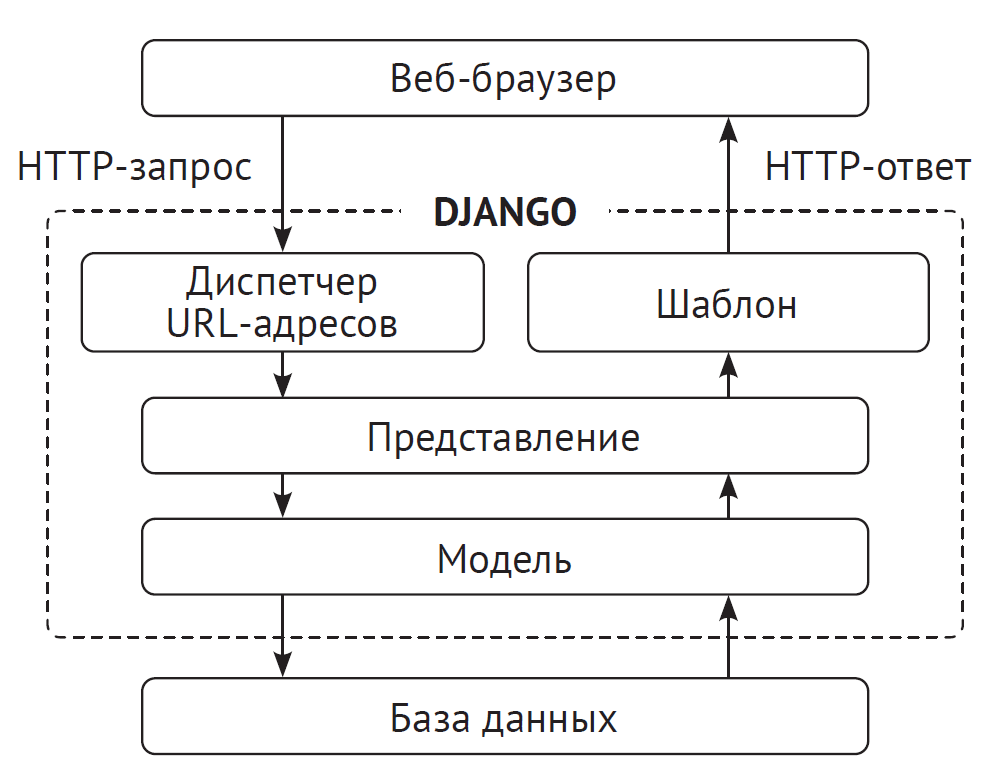
Учитывая описанные выше преимущества, язык программирования Python подходит для выполнения поставленной задачи

Особенно интересен объектно-ориентированный подход при работе с сущностями базы данных, в приложениях, спроектированных с использованием фреймворка Django.

## Django

Django – один из самых популярных фреймворков для создания приложений с веб-интерфейсом на языке Python.

Проектирование системы с использованием шаблона Django подразумевает реализацию архитектурного паттерна MVT – Model-View-Template, который представляет собой модификацию популярного в проектировании веб-ресурсов паттерна MVC – Model-View-Controller. Схема паттерна MVT представлена на рисунке 7 [3].



*Рисунок 7. Архитектура шаблона Model-View-Template*

Подобная архитектура позволяет обеспечить низкую связанность (coupling) между разными элементами приложения. Этим обуславливается основное преимущество паттерна MVT(MVC) – чёткое разделение работы с моделями данных, основной бизнес-логики приложения и отображения пользовательского интерфейса. Данный подход позволяет изменять один структурный элемент проекта, не затрагивая или незначительно затрагивая смежные с ним структурные элементы.

Разрабатываемый в рамках данной работы проект состоит из трёх приложений: основного – «Менеджер документов» (doc\_manager) и двух приложений, расширяющих функциональные возможности проекта, - приложения «Аутентификация» (authentication) и «Согласование» (approvement). Подробнее структура проекта описана в главе 4 «Структура проекта».

Подобное разделение проекта на приложения обеспечивают необходимый уровень независимости каждого приложения, а также позволяет использовать каждое приложение повторно в других проектах, что полностью удовлетворяет одному из основных принципов Django – DRY (Don’t repeat yourself / Не повторяй себя). Кроме того, дальнейшее расширение функциональных возможностей проекта может быть выполнено за счёт добавления новых и минимального изменения уже существующих приложений.

Ещё одним преимуществом Django является собственный встроенный ORM (Object-Relational Mapping / Объектно-реляционное отображение), позволяющий работать с сущностями базы данных в объектно-ориентированной парадигме посредством установления связи между базой данных и классами в модели разрабатываемого приложения. Модели в Django по сути являются описанием таблиц базы данных и связей между ними, выполненным в виде кода на языке Python.

## HTML и CSS

Шаблоны в разрабатываемом приложении, ответственные за пользовательский веб-интерфейс, выполняются с использованием языков HTML и CSS.

HTML (HyperText Markup Language) - это язык разметки, который используется для создания и оформления веб-страниц. HTML состоит из тегов, которые определяют структуру и содержимое страницы, которая будет отображена в браузере. Теги могут содержать текст, изображения, ссылки, формы и другие элементы.

Одной из главных особенностей HTML является возможность создания ссылок на другие страницы или файлы на сервере. Это позволяет создавать навигацию по сайту и облегчать его использование.

Для обеспечения динамической работы представлений Django с веб-интерфейсом, при разработке HTML-шаблонов активно использовались теги шаблонов Django, которые могут использоваться для вывода данных из базы данных, для форматирования текста и для создания динамических элементов страницы. Также, используя теги шаблонов, можно расширять базовый HTML-шаблон вспомогательными шаблонами.

В целом, HTML является важным инструментом для создания красивых и функциональных веб-сайтов. Он предоставляет разработчикам широкие возможности для создания динамических и интерактивных веб-приложений.

CSS (Cascading Style Sheets) - это язык разметки, который используется для создания стилей веб-страниц. Он позволяет задавать цвет, шрифт, размер шрифта, выравнивание, отступы, границы, тени и другие свойства элементов HTML.

CSS является одним из основных инструментов веб-разработки и используется для оформления сайтов и приложений. Он также позволяет создавать таблицы и формы, а также управлять поведением элементов на странице.

# Структура проекта

Проект состоит из трёх основных приложений:

* Аутентификация;
* Менеджер документов;
* Согласование.

## Приложение «Аутентификация»

Приложение «Аутентификация» - это часть проекта, которая предоставляет возможность для пользователей входить в систему и выходить из нее. Оно использует встроенные в Django формы аутентификации, позволяющие зарегистрированным пользователям войти в систему, введя свои логин и пароль. После входа пользователь может получить доступ к различным функциональным возможностям проекта. Например, он может создавать и просматривать карточки документов, загружать в систему, просматривать и скачивать файлы, относящиеся к документу, участвовать в согласовании документов.

Самостоятельная регистрация пользователей в проекте не предусмотрена, поскольку доступ к документам должен быть предоставлен только верифицированным пользователям. В связи с этим регистрация новых учётных записей пользователей возможна исключительно централизованно через администратора.

Выход из системы осуществляется нажатием на соответствующую кнопку. После этого пользователь будет автоматически перенаправлен на страницу авторизации, а доступ к функциональным возможностям системы будет заблокирован.

## Приложение «Менеджер документов»

Данное приложение является основным приложением проекта. «Менеджер документов» содержит в себе весь базовый функционал системы по работе с проектной документацией и вложенными файлами.

Это приложение позволяет аутентифицированным пользователям выбирать интересующий проект, создавать и просматривать карточки документов, загружать в систему файлы, относящиеся к документу (например, финальный PDF-файл проектного документа, DOC-файлы текстовой части и чертежи в редактируемом формате DWG), просматривать и скачивать вложенные файлы.

Кроме того, в заголовке сайта реализована строка поиска, позволяющая производить поиск документов по уникальному KKS коду среди всех разрабатываемых проектов.

Хранение всех вложенных файлов в проекте происходит в файловой системе сервера, структура хранения имеет чёткую строгую нисходящую иерархию типа «Проект-Здание-Документ-Вложенные файлы документа». Подобная иерархическая структура позволяет избежать хранения большого количества файлов в одной директории, что могло бы снизить быстродействие системы. Взаимодействие с файлами в данном проекте организовано с использованием специального поля модели Django FileField, который позволяет загружать файлы на сервер и хранить информацию о них в базе данных.

## Приложение «Согласование»

Данное приложение является расширением основного приложения «Менеджер документов». Интегрирование данного приложения в проект позволяет добавить функционал электронного согласования документа.

Инженер-разработчик документа получает возможность прикреплять к своему документу процедуру согласования, лично выбирая пользователя на каждый шаг процедуры. Указанные в процедуре в роли Согласующего пользователи могут как согласовать документ, так и отправить предложения по улучшению в виде выставления замечаний. Разработчик документа в свою очередь может оставлять ответ на полученные замечания, которые будут закрыты Согласующим после устранения всех недостатков. После того, как все шаги процедуры будут согласованы ответственными пользователями, документ переходит в статус «Согласован».

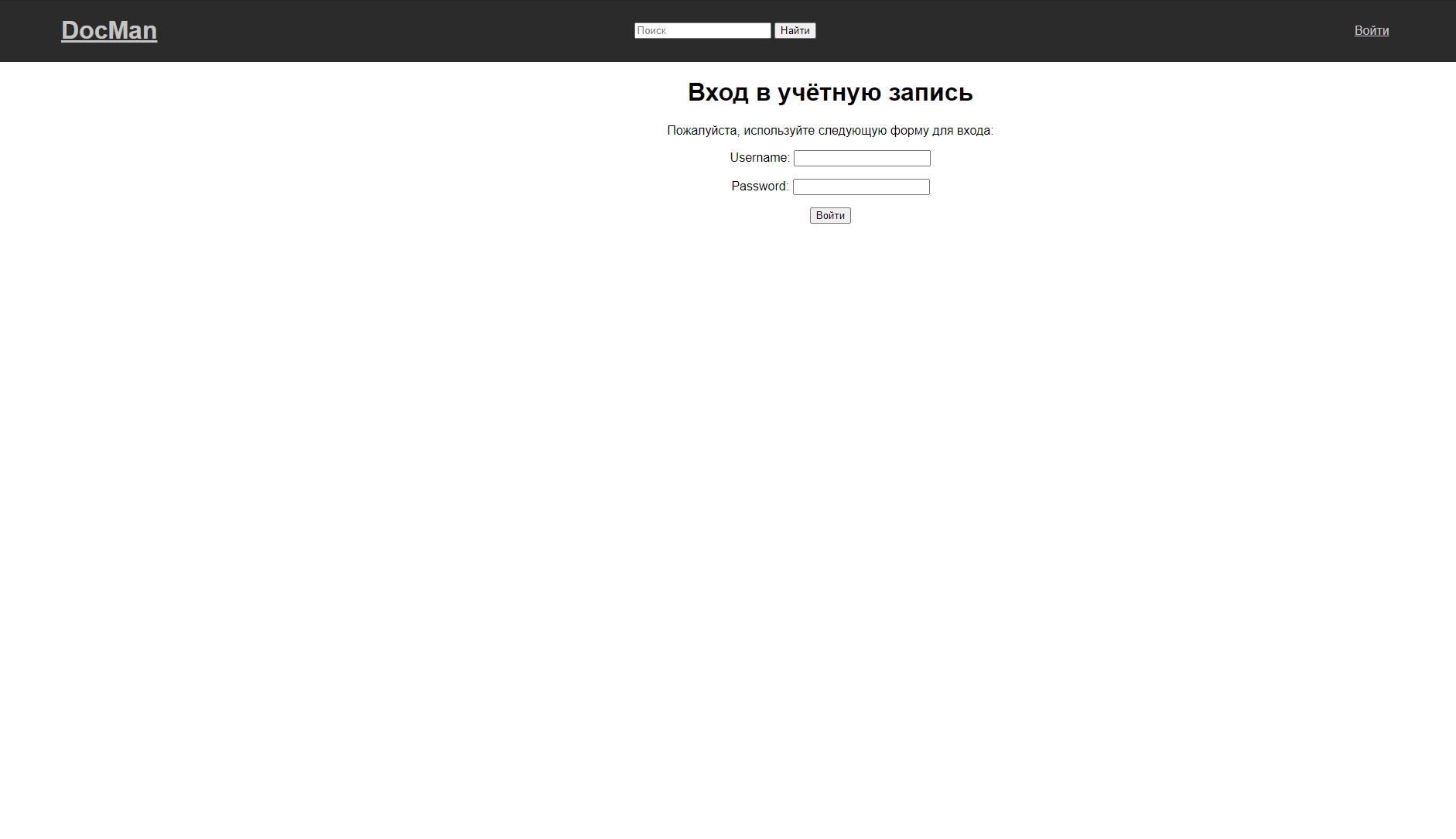
# Функции системы

## Аутентификация

Функция предназначена для аутентификации пользователей и предоставления аутентифицированным пользователям доступа к остальным функциям приложения.

### Login

Вход в учётную запись осуществляется по имени пользователя и паролю, выданным пользователю администратором системы. Поскольку система является закрытой и доступ к хранящимся в ней данным должен быть предоставлен только верифицированным пользователям, самостоятельная регистрация пользователей не предусмотрена, учётные записи создаются администратором централизованно.



*Рисунок 8. Форма аутентификации пользователя*

### Logout

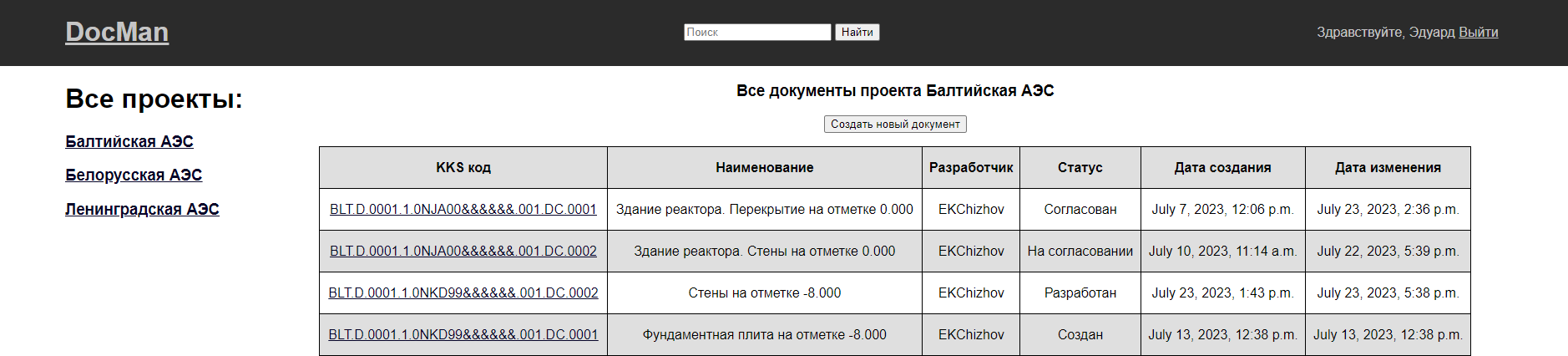
Выход из учётной записи осуществляется при нажатии на кнопку «Выйти», после чего происходит переадресация на страницу входа в учётную запись

Рисунок 9. Кнопка выхода из учётной записи

## Менеджер документов

В данном разделе описывается основной функционал по работе с документами и вложенными в них файлами. Структурирование документов происходит по проектам, список которых представлен в левой боковой панели веб-страницы. Для начала работы с менеджером документов необходимо выбрать интересующий проект.

### Просмотр всех документов по каждому проекту

После выбора объекта открывается список всех документов по объекту. Отсюда можно войти в карточку какого-то конкретного документа или создать новый документ.

*Рисунок 10. Просмотр перечня всех документов по проекту «Балтийская АЭС»*

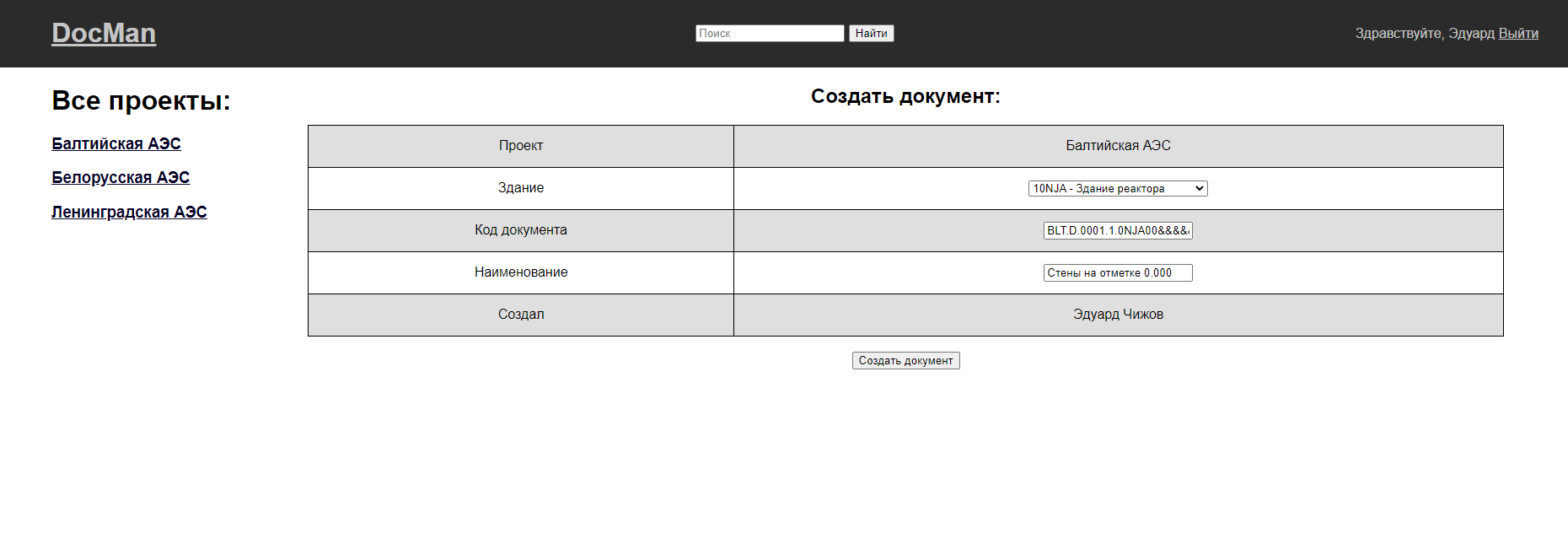
### Просмотр карточки документа

Страница для просмотра основной информации о документе, загрузки и просмотра вложенных файлов, работой с процедурой согласования документа.

 *Рисунок 11. Просмотр карточки документа*

### Создание карточки документа

Для создания нового документа необходимо выбрать здание из списка зданий, принадлежащих выбранному ранее проекту, ввести уникальный KKS код документа и его название. После создания документу присваивается статус «Создан».



*Рисунок 12. Форма создания нового документа по проекту «Балтийская АЭС»*

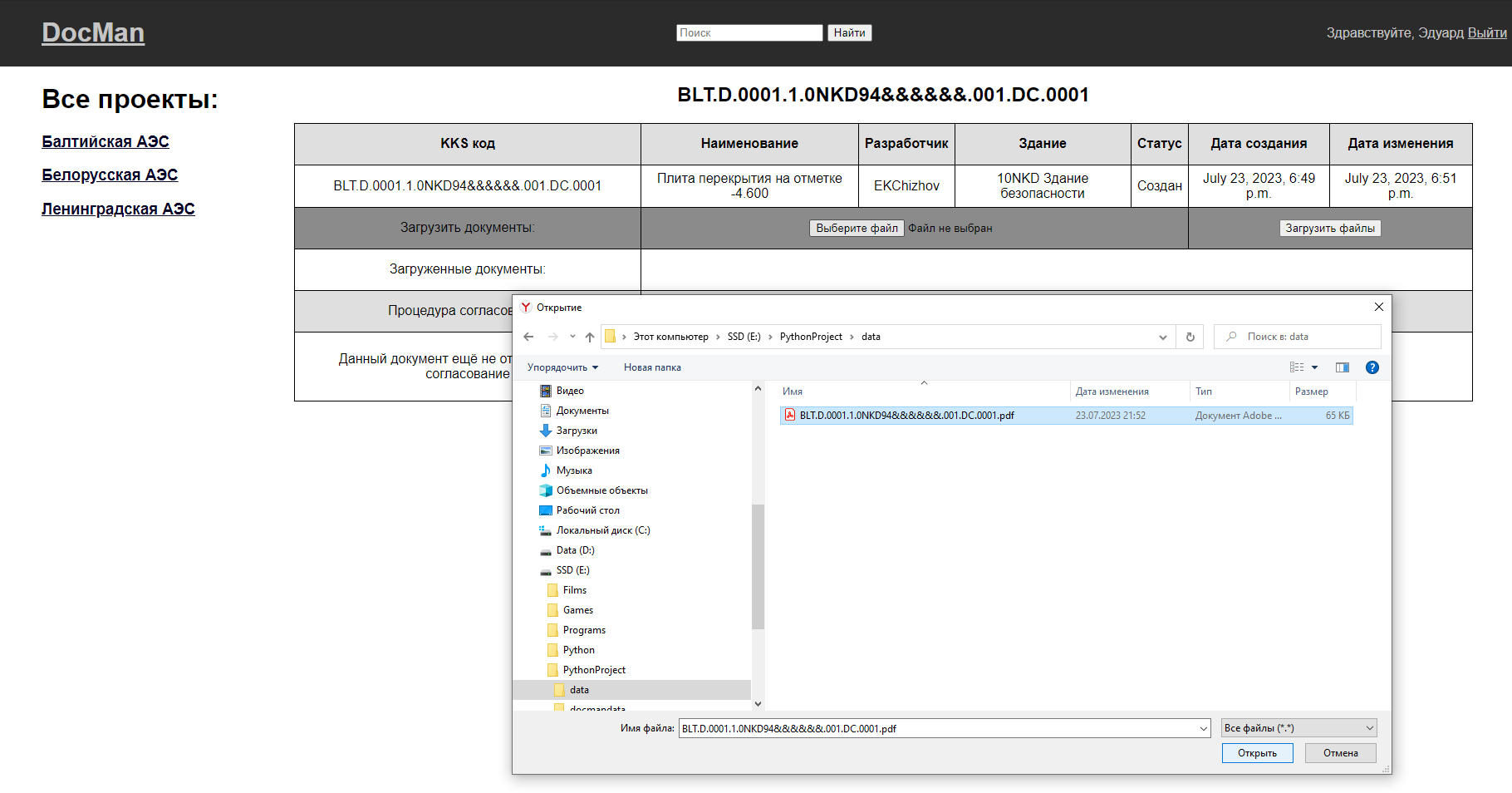
### Работа с прикреплёнными файлами

Функционал приложения предусматривает возможность загрузки и просмотра файлов, относящихся к выбранному документу.

#### Загрузка просматриваемых файлов в карточку документа

Загрузка файлов осуществляется при выборе соответствующей кнопки в карточке документа. Файлы загружаются в файловую систему сервера в директорию, определённую в файле settings.py текущего Django-проекта.

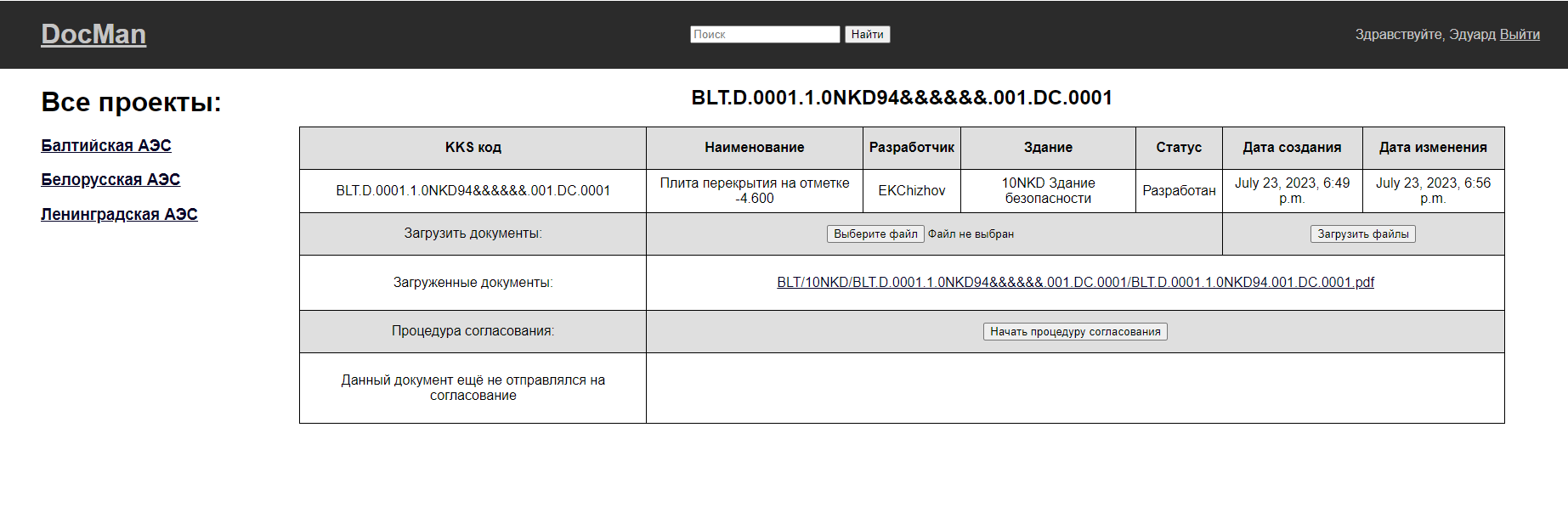
После загрузки файлов в карточку документа, статус документа меняется на «Разработан».



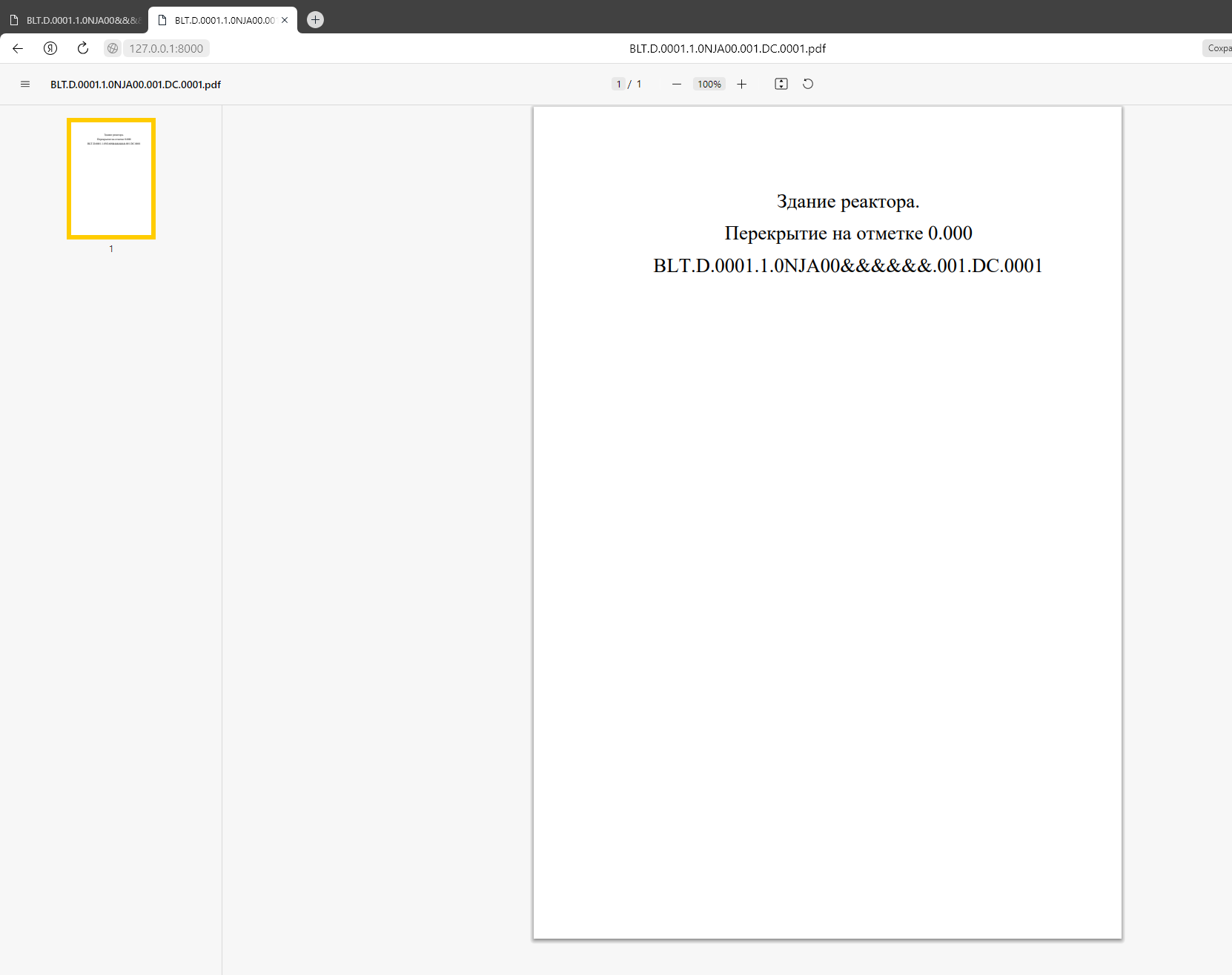
*Рисунок 13. Прикрепление pdf-файла к карточке документа*

#### Просмотр и скачивание прикреплённых файлов

Просмотр прикреплённых документов производится встроенными в браузер инструментами отображения. Просматривание доступно только для распространённых просматриваемых форматов (.pdf, .doc и др.), скачивание доступно для всех типов файлов (в том числе редактируемых файлов чертежей в форматах .dwg, .dgn, .ifc и др.).



*Рисунок 14. Отображение прикреплённых файлов в карточке документа*



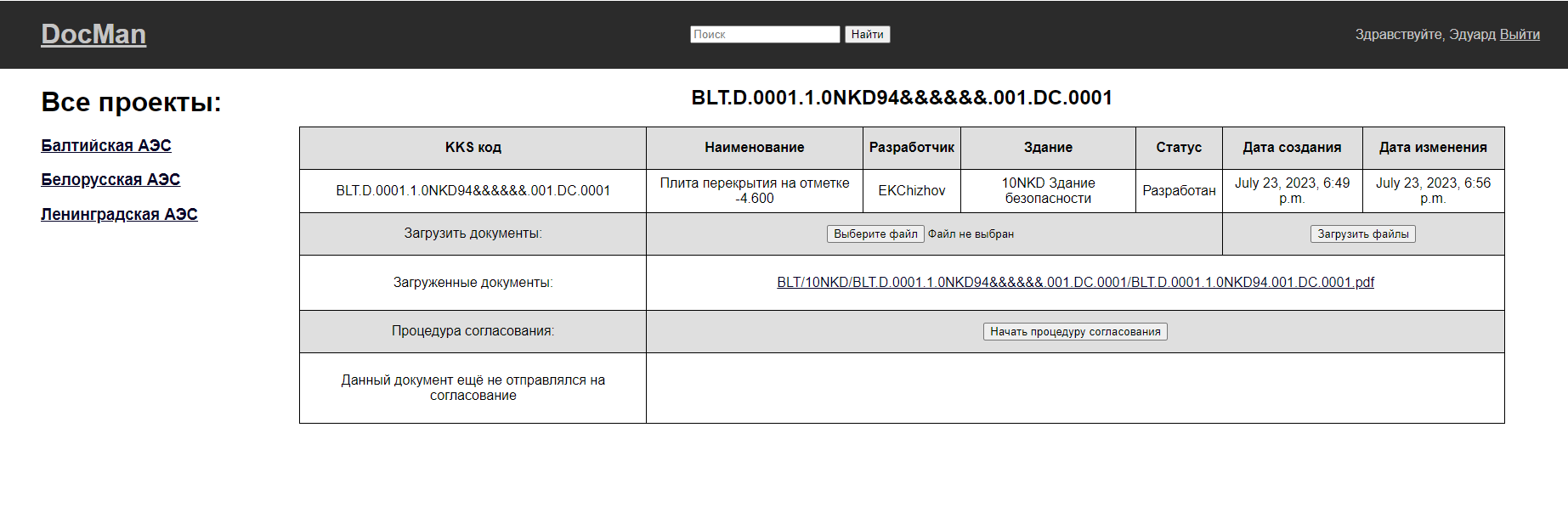
*Рисунок 15. Просмотр прикреплённого файла в новом окне браузера*

## Согласование

Приложение «Согласование» в проекте отвечает за запуск и прохождение процедуры согласования документов, а также выставление замечаний и работу с ними.

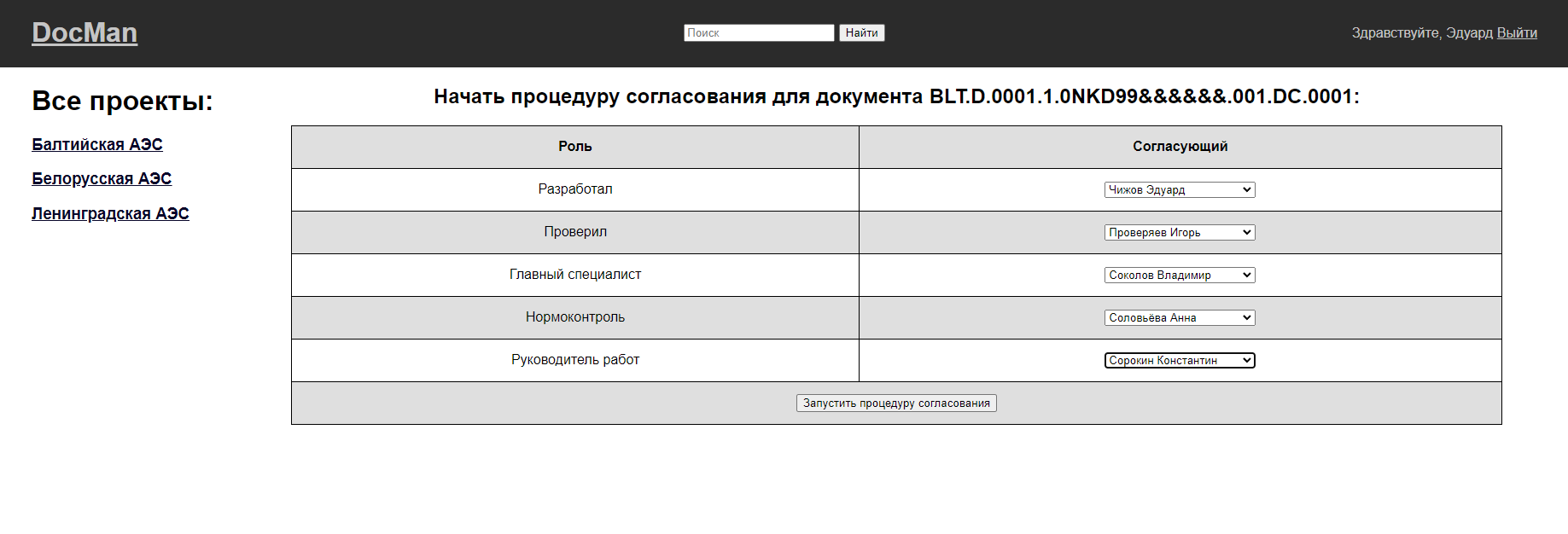
### Присоединение процедуры согласования документа

Начать процедуру согласования документа можно, нажав соответствующую кнопку в карточке документа, подлежащего согласованию.



*Рисунок 16. Кнопка начала процедуры согласования в карточке документа*

Для начала процедуры согласования требуется выбрать ответственных сотрудников на каждый шаг процедуры и нажать соответствующую кнопку. После прикрепления процедуры статус документа изменится на «На согласовании».



*Рисунок 17. Форма выбора согласующих для процедуры согласования*

### Просмотр деталей процедуры согласования

Если на документ запущена процедура согласования, появляется возможность посмотреть её детали и увидеть, на какой стадии находится каждый шаг, а также просмотреть, есть ли замечания по конкретному шагу. Каждый шаг процедуры согласования может иметь один из следующих статусов:

* На согласовании
* Получены замечания
* Замечания исправлены
* Согласовано

Если аутентифицированный пользователь является ответственным согласующим на каком-либо шаге, помимо кнопки просмотра (и добавления) замечаний и него также появляется кнопка согласования документа.

После создания процедуры все шаги имеют статус «На согласовании».

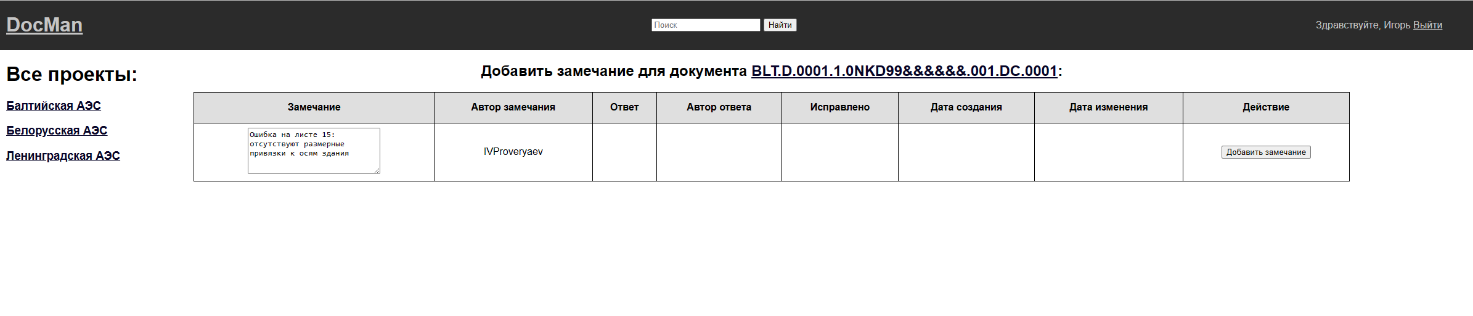


*Рисунок 18. Просмотр деталей действующей процедуры согласования*

### Работа с замечаниями

#### Выставление замечаний Согласующим

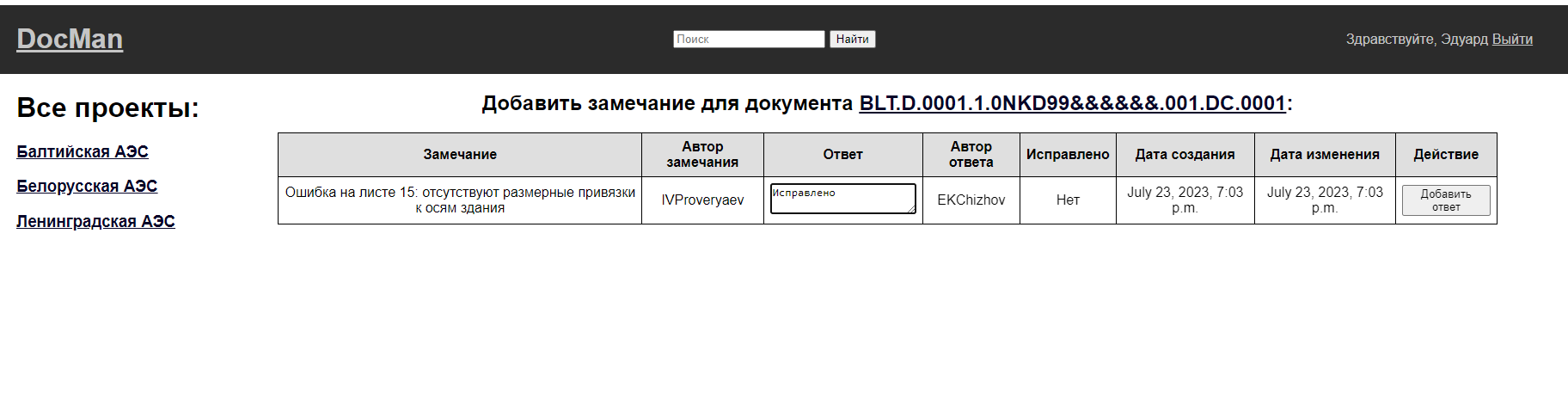
Если Согласующий по каким-то причинам не согласовывает документ, он может добавить замечание в окне просмотра замечаний своего шага процедуры согласования. При добавлении замечаний шаг процедуры согласования приобретает статус «Получены замечания».



*Рисунок 19. Добавление замечания Согласующим*

#### Ответ на замечания разработчиком документа

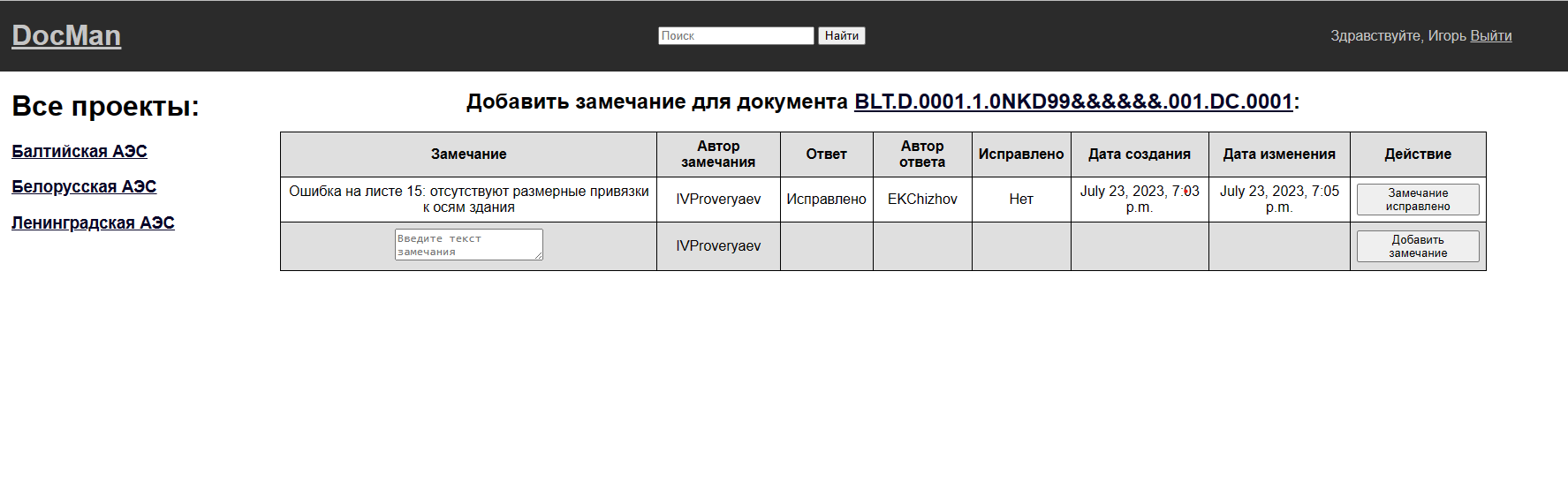
После получения замечаний разработчик должен дать ответ: либо отчитаться об исправлении, либо дать аргументированный ответ, почему данное замечание устранено не будет.



*Рисунок 20. Добавление ответа на замечание разработчиком документа*

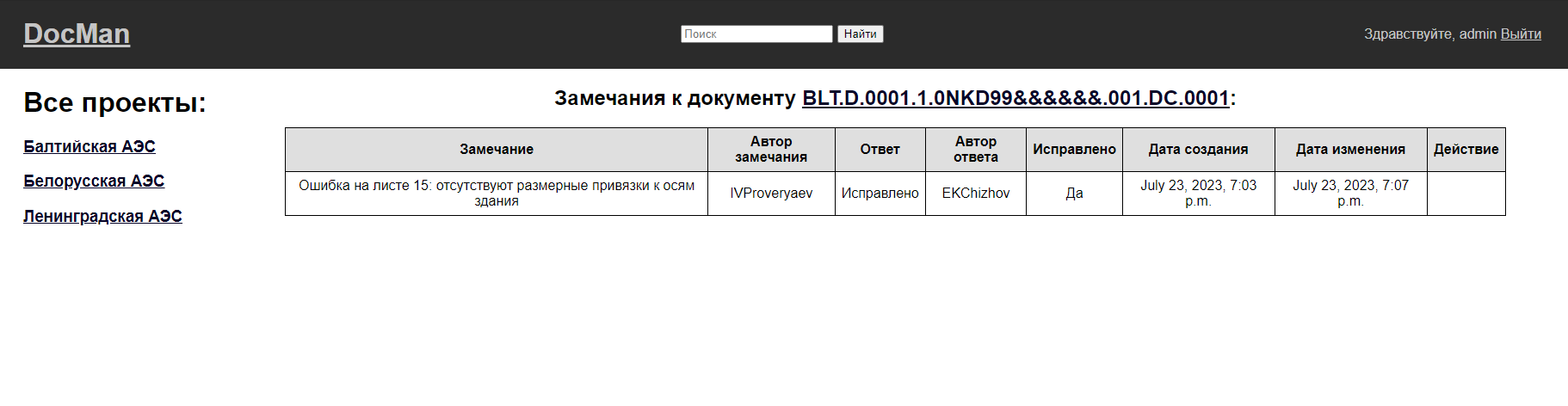
#### Подтверждение исправления замечаний Согласующим

После получения ответа от разработчика документа и устранения всех замечаний Согласующий получает возможность подтвердить, что замечание исправлено, после чего значение статуса замечания «Исправлено» изменится с «Нет» на «Да».

 *Рисунок 21. Подтверждение Согласующим, что замечание исправлено*

#### Просмотр замечаний всеми авторизованными пользователями

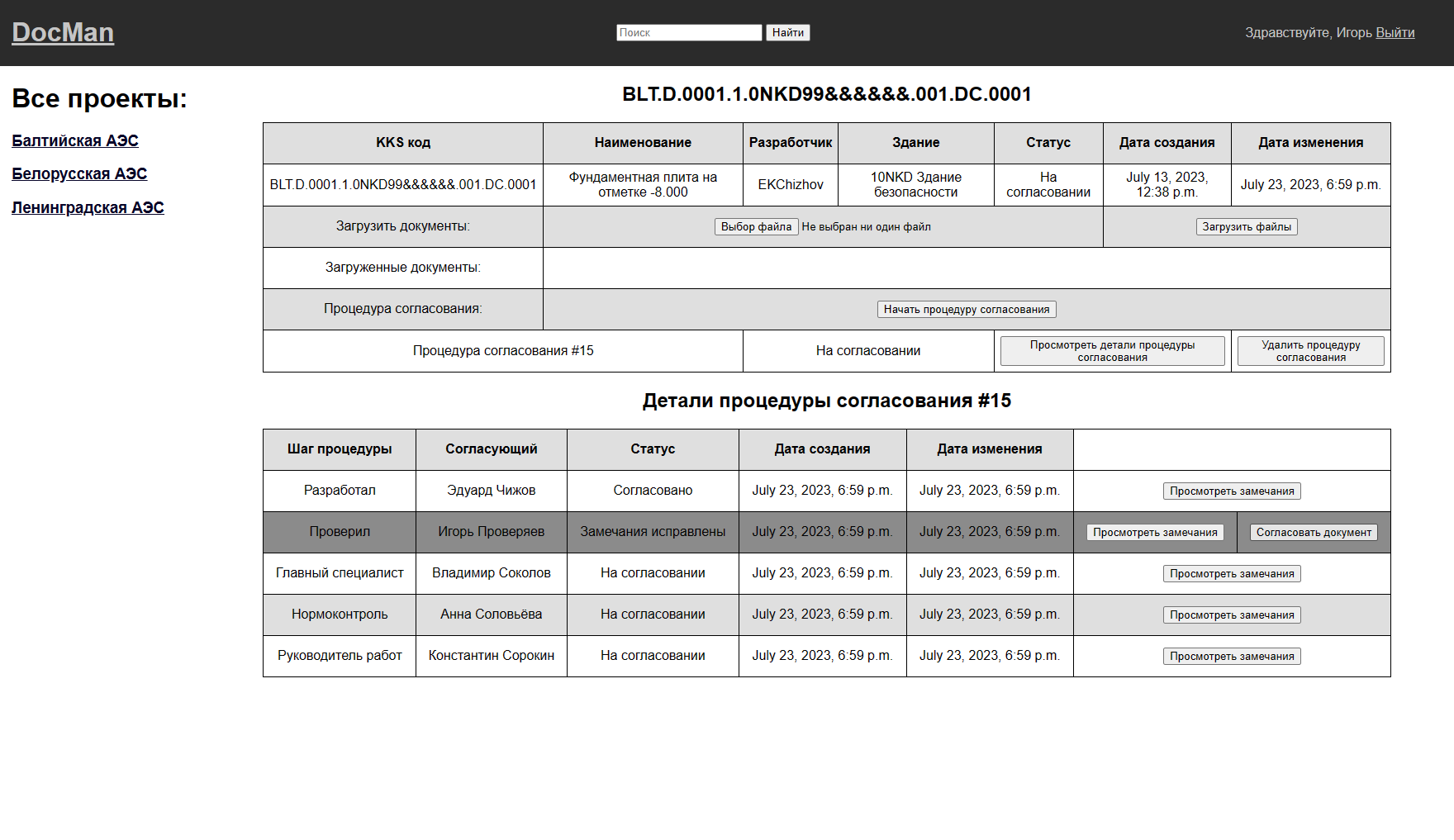
Для всех аутентифицированных пользователей, не являющихся разработчиком документа или Согласующим, также доступна функция просмотра замечаний, но без возможности их добавления или ответа.



*Рисунок 22. Просмотр выставленных замечаний всеми зарегистрированными пользователями*

### Согласование документа

После того, как все замечания конкретного шага устранены, и Согласующий подтвердил их исправление, шаг процедуры получает статус «Замечания исправлены», а Согласующий получает возможность согласовать документ. После согласования шаг приобретает статус «Согласовано».



*Рисунок 23. Кнопка согласования документа конкретным Согласующим   
(доступна, если нет не исправленных замечаний)*

После того, как все шаги процедуры согласования приобретают статус «Согласовано», документ автоматически приобретает «Согласован», после чего считается утверждённым всеми ответственными и может быть взят в работу.

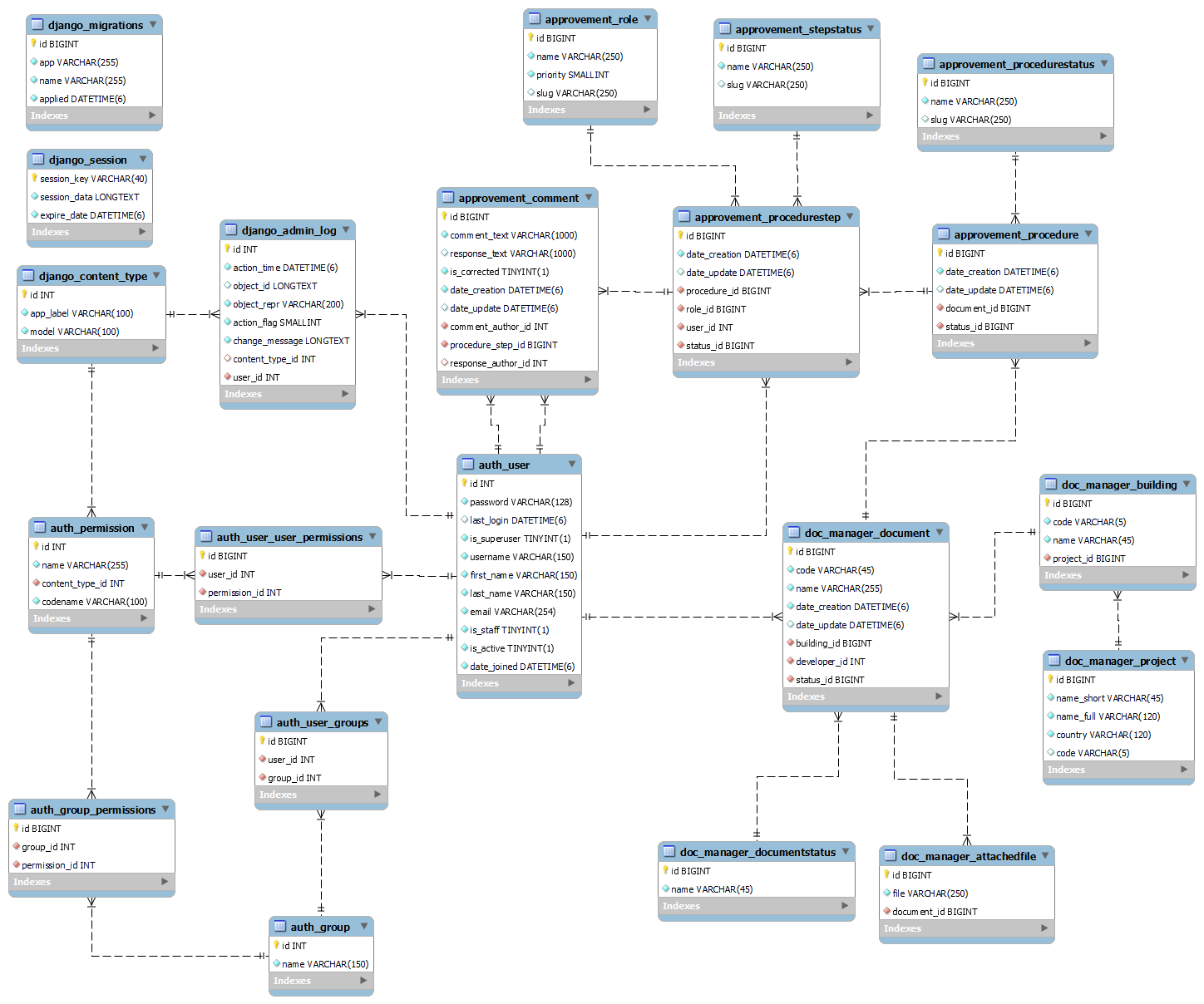


*Рисунок 24. Документ переходит в статус «Согласован» после того,   
как все замечания приобретают статус «Согласовано»*

# Структура данных

На рисунке 25 показана структура базы данных проекта. Данная структура состоит как из базовых таблиц Django с префиксами «auth» и «django», так и таблиц, определённых приложениями «Менеджер документов» (doc\_manager) и «Согласование» (approvement).

Каждая таблица, относящаяся к приложениям проекта, соответствует отдельной сущности, прописанной в виде Python-класса в файле models.py соответствующего проекта. Взаимосвязи между таблицами также задавались при определении классов посредством указания Foreign Key при помощи Django-поля ForeignKey.



*Рисунок 25. Структура Базы Данных MySQL*

# Заключение

Было разработано веб-приложение «DocMan», представляющее собой систему электронного документооборота проектной документации с возможностью согласования документов.

Приложение было спроектировано в соответствии с требованиями сдачи проектной документации объектов ядерной энергетики, оно отвечает требованиям безопасности в рамках текущей концепции импортозамещения программного обеспечения. Узкая специфика проектирования и высокие требования Заказчика делают крайне затруднительным поиск готового решения, реализующего все необходимые возможности. Разработанное приложение сочетает в себе весь требуемый функционал, для получения которого ранее приходилось использовать два отдельных приложения, что значительно уменьшает количество человеко-часов на выполнение всех действий по согласованию документации и организации её последующего хранения.

Для этого были достигнуты следующие цели:

1. Были проанализированы требования к проектируемому ПО:
   1. учтены требования Заказчика;
   2. учтены пользовательские истории.
2. Разработана База Данных MySQL для хранения информации:
   1. о пользователях;
   2. о проектах;
   3. о документации, соответствующей проектам;
   4. о процедурах согласования и роли для экспертов.

3. На языке Python с использованием фреймворка Django разработано интуитивно понятное web-приложение, позволяющее пользователям работать с информацией из Базы Данных, в котором учтены все заявленные функциональные требования.

Трудности, с которыми пришлось столкнутся при создании дипломного проекта были связаны в основном с визуальной частью приложении, в частности с разметкой HTML и стилями CSS, для решений этих задач, понадобилось ознакомиться с различными материалами, взятыми из интернета [8, 9, 10].

На базе сделанной работы можно реализовать следующие возможности:

1. Отправка уведомлений по электронной почте сотрудникам, когда они становятся участниками согласования документации.
2. Добавление функционала для работы технического архива с возможностью присвоения архивных номеров, выпущенной документации.

# Список литературы

1. ГОСТ 2.501-88 – Единая система конструкторской документации. Правила учета и хранения
2. ГОСТ 2.001-2013 – Единая система конструкторской документации. Общие положения
3. Меле А. Django 4 в примерах / пер. с англ. А. В. Логунова. – М.: ДМК Пресс,   
   2023. — 800 с.
4. Лутц М. Изучаем Python, том 1, 5-е изд.: Пер. с англ. — СПб.: ООО “Диалектика”, 2019. — 832 с.
5. Python 3.11.4 documentation: [Электронный ресурс] URL: <https://docs.python.org/3/> (Дата обращения 17.07.2023)
6. Django documentation: [Электронный ресурс] URL: <https://docs.djangoproject.com/en/4.2/> (Дата обращения 24.07.2023)
7. Руководство по MySQL: [Электронный ресурс] URL: <https://metanit.com/sql/mysql/> (Дата обращения 15.07.2023)
8. Онлайн учебники и справочные материалы по HTML, CSS и JavaScript: [Электронный ресурс] URL: <https://ru.w3docs.com> (Дата обращения: 20.07.2023)
9. HTML Color Codes: [Электронный ресурс] URL: <https://htmlcolorcodes.com/color-picker/> (Дата обращения: 18.07.2023)
10. Dofactory: [Электронный ресурс] URL: <https://www.dofactory.com/html/size> (Дата обращения: 23.07.2023)