

УДК: 004.4'22

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ КАТАЛОГИЗАЦИИ ДОКУМЕНТОВ ДЛЯ АРХИВИРОВАНИЯ

Боронина Е.А., студент гр. 430-1, 4 курс

Научный руководитель: Захарова А.А., д.т.н., проф. каф. АСУ

Томский университет систем управления и радиоэлектроники
г. Томск

В современном мире сфера информационных технологий непрерывно развивается, а задачи по цифровизации данных приобретают все большие масштабы. Одной из таких задач является перевод проектно-сметной документации в электронный формат, так как на данный момент большая часть соответствующей документации хранится в бумажном формате, что приводит к значительным размерам архивов и существенно замедляет процесс поиска необходимых данных.

При таких объемах данных процесс цифровизации документов вручную является времязатратным и может быть связан с ошибками в каталогизации, вызванными человеческим фактором.

Оптимальным решением данной проблемы является разработка информационной системы каталогизации проектно-сметной документации для архивирования.

Проект реализуется в рамках прохождения преддипломной практики на базе единого цифрового оператора Группы Газпром - Газпром Цифровые проекты и сервисы.

Практической значимостью проекта является автоматизация подготовки технической документации к отправке на хранение в архив, что поможет разгрузить команду специалистов от рутинных операций по обработке документации и упорядочить информационные потоки, а также обеспечить максимальную прозрачность и контроль процесса получения и согласования проектно-сметной документации.

Целью проекта является разработка веб-приложения, упрощающего процесс подготовки технической документации к архивированию.

Первостепенными задачами являются:

1. Проведение анализа аналогов системы.
2. Определение этапов обработки документации.
3. Выделение пользователей системы.
4. Определение функциональных требований к базовой версии системы.
5. Определение средств разработки веб-приложения.

При поиске и анализе аналогов удалось выявить, что, как правило, подобные программные решения разрабатываются под нужды конкретного заказчика, в связи с чем имеют определенные ограничения для внедрения в другие

организации. Примерами такого программного обеспечения являются:

1. «Логика: ТЕХДОК», обеспечивающий построение и ведение централизованной системы сбора, регистрации, учета и хранения технической документации, порождаемой на всех стадиях жизненного цикла проектных разработок [1].

2. «Технический архив» от «Digital Design», позволяющий компании сделать систему управления потоками документации понятной и быстрой и предоставить сотрудникам возможность оперативно походить необходимые материалы. Также он призван вести учет места хранения и факта выдачи бумажных оригиналов [2].

В нашем случае проект также реализуется под нужды конкретного заказчика с возможностью динамического изменения некоторых структур, что определяет актуальность разработки нового программного продукта

В настоящий момент можно выделить следующие этапы ручной подготовки документов к архивированию:

1. Младший сотрудник получает доступ к папке со сканами, каждый из которых ему необходимо просмотреть, выделить значимые атрибуты документа и в соответствии с их значениями перенести файл в новую директорию согласно заданной структуре.

2. Младший сотрудник заносит в Excel таблицу данные о выделенные атрибуты документа и конечный путь к соответствующей структуре директории хранения файла.

3. Старший сотрудник проверяет корректность заполнения атрибутов и размещения структурированных файлов.

4. Старший сотрудник по итогам проверки принимает решение отправить файл с неверно заполненными атрибутами на доработку или принять верно обработанный файл для дальнейшей отправки на хранение в архив.

5. Руководитель отдела оценивает объемы выполненных работ.

Прежде чем перейти к формулированию функциональных требований к системе определим пользователей системы и их основной функционал. Для разграничения прав доступа пользователей к системе используем концепцию ролевой модели, где чем выше роль, тем больше вариантов взаимодействия с системой доступно пользователю. При подробном рассмотрении этапов процесса перевода физического архива в электронный достаточно очевидно выделение следующих акторов, каждому из которых соответствует свой функционал:

1. «*Оператор*» осуществляет загрузку файлов, редактирует их атрибуты в соответствии с доступными справочными данными и отправляет файлы на проверку модератору.

2. «*Модератор*» проверяет правильность заполнения атрибутов файлов после их заполнения оператором и имеет возможность вернуть файлы на доработку или отправить проверенные файлы на каталогизацию. Он также может загружать файлы в систему и самостоятельно редактировать их атрибуты.

3. «*Администратор*» имеет доступ к возможностям менять роли

пользователей, редактировать справочники, удалять файлы с сервера, просматривать информацию о статистике взаимодействия других пользователей с системой. Также ему доступны все возможности оператора и модератора.

Исходя из проведенного анализа этапов каталогизации документов и выделенного функционала каждого из предполагаемых участников процесса, определим требования к составу выполняемых функций, которые должна обеспечить базовая версия разрабатываемой системы:

1. Управление доступом и авторизация пользователей. Должен быть обеспечен доступ пользователей к системе по логину и паролю, при этом каждому пользователю должна быть присвоена роль в соответствии с концепцией, описанной выше. Должно быть реализовано соответствующее ролям разграничение к функционалу системы. Для администратора должна быть реализована возможность присвоения различных ролей другим пользователям.

2. Загрузка и настройка атрибутов проектно-сметной документации. Должна быть реализована возможность загрузки одного или нескольких сканированных файлов. Необходима реализация функции массового присвоения атрибутов группе файлов при одновременной их загрузке. Также должна быть возможность присвоения или редактирования атрибутов одного или нескольких документов уже после их загрузки на сервер.

3. Удаление загруженных документов с сервера. Необходима возможность полного удаления документа с сервера для случаев загрузки посторонних файлов.

4. Просмотр и управление статусами документов. Пользователи должны иметь возможность просмотра загруженных файлов в системе. Для удобства дальнейшей модерации файлам должны быть присвоены различные статусы с возможностью просмотра файлов с одинаковыми статусами в отдельных вкладках. Всего система предусматривает 5 статусов документа:

- 1) Документ еще не обработан только загружен на сервер.
- 2) Документ обработан оператором, все значимые атрибуты заполнены, но не произведена отправка на проверку модератором.
- 3) Документ отправлен на проверку модератору.
- 4) Документ проверен модератором, все атрибуты заполнены верно.
- 5) Документ автоматически каталогизирован в соответствии с заданной структурой по значениям атрибутов.

5. Массовое изменение статусов документов. Аналогично с массовым присвоением и изменением атрибутов документов должно быть реализовано изменение статусов группы документов.

6. Модерация документов. В системе предусмотрена модерация документов, включая возможность их проверки или возврата на доработку.

7. Каталогизация и настройка структуры документов. Система позволяет автоматически группировать документы по атрибутам в древовидную структуру каталога, а также настраивать параметры этой структуры, т.е. указывать какие атрибуты будут участвовать в каталогизации и в какой приоритетности

они будут расположены.

8. Автоматическое изменение имени файла. Так как у нескольких проектов могут быть файлы с одинаковым названием, то при загрузке на сервер все документы с одинаковым названием должны автоматически переименовываться на допустимое имя файла, в целях избегания утечки данных, аналогично они должны переименовываться на оригинальное название при каталогизации, когда у файла больше нет дубликатов и нет необходимости отличать его от других файлов с тем же именем.

9. Настройка справочников для заполнения атрибутов документов. Чтобы упростить обработку документов и минимизировать число ошибок необходимо создать справочные таблицы для используемых в каталогизации атрибутов, которые сможет редактировать администратор.

10. Просмотр статистики. Администратор имеет доступ к консолидированной информации о количестве пользователей системы и количестве файлов с различными статусами, отображаемой на дашбордах.

На рисунках 1-2 представлена IDEF0 диаграмма, построенная на основе описанных выше функциональных требований к системе.

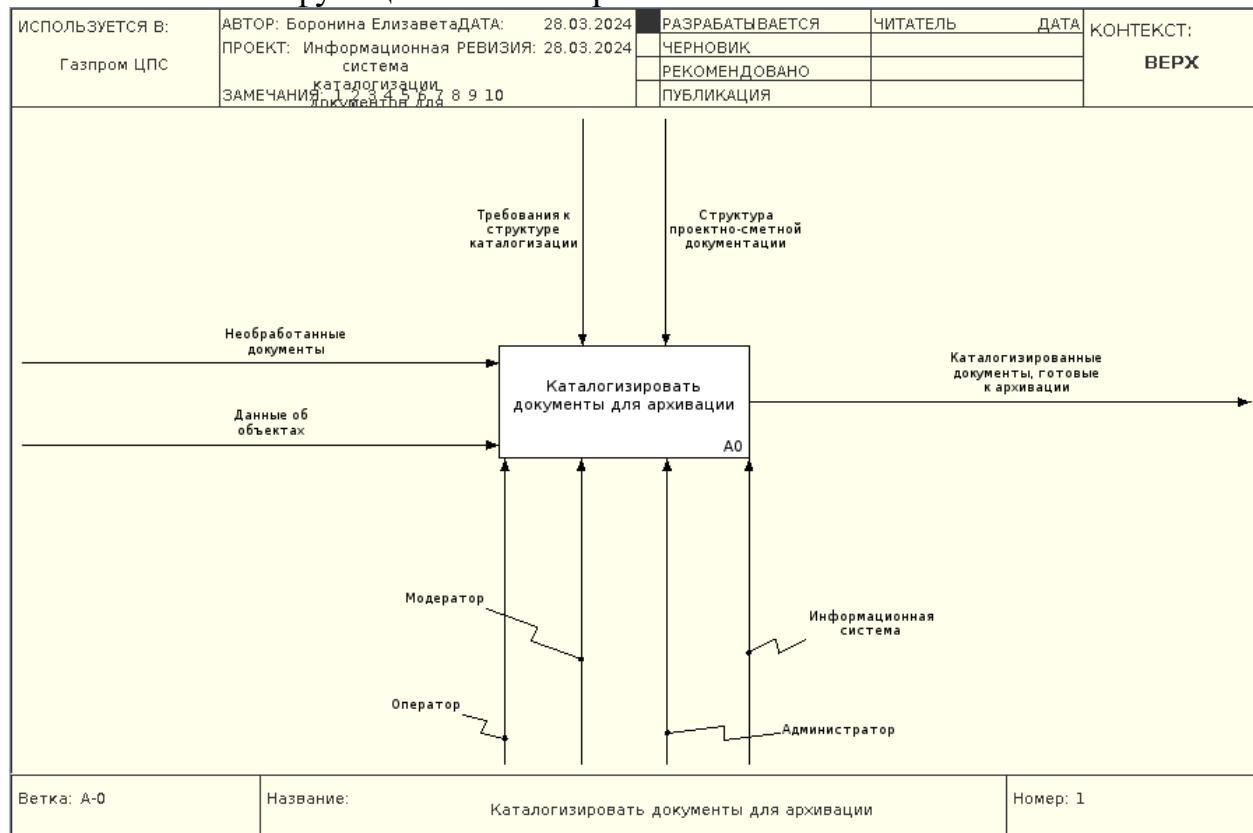


Рисунок 1 – Контекстная диаграмма А-0 функциональной модели IDEF0

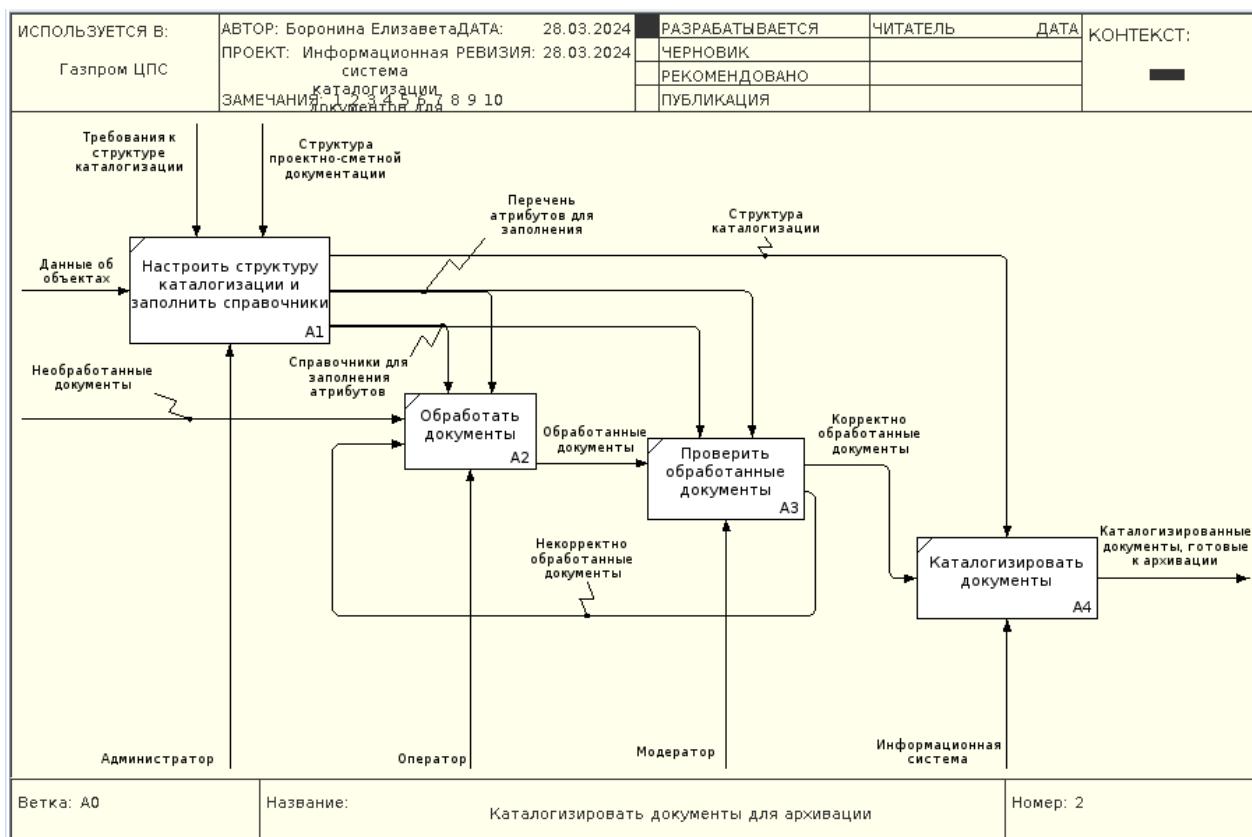


Рисунок 2 – Декомпозиция контекстной диаграммы первого уровня

Для реализации проекта был выбран следующий стек технологий:

1. Python – это язык программирования, часто используемый в веб-приложениях, разработке программного обеспечения и машинном обучении. Python широко используют, за счет его эффективности, простоты в изучении и хорошей совместимости с различными платформами [3].
2. PostgreSQL – мощная объектно-реляционная система баз данных с открытым исходным кодом. Имеет прочную репутацию за счет своей надежности, функциональной устойчивости и высокой производительности [4].
3. React – JavaScript-библиотека позволяющая создавать интерактивные пользовательские интерфейсы [5].
4. Nginx — это HTTP-сервер и TCP/UDP прокси-сервер общего назначения. Он позволяет работать с настройками CORS, чтобы позволить агенту пользователя получать разрешения на доступ к ресурсам сервера на источнике (домене), отличном от того, что используется клиентской частью [6].
5. Docker – программное обеспечение позволяющее автоматизировать развертывание и управление приложениями в средах с поддержкой контейнеризации [7].

Список литературы:

1. ЛОГИКА: ТЕХДОК [Электронный ресурс]. – URL: <https://blogic.ru/products/prikladnye-sistemy/storage/>
2. Автоматизация технического архива [Электронный ресурс]. – URL:

<https://digdes.ru/products/tehnicheskij-arhiv>

3. Что такое Python? [Электронный ресурс]. – URL:

<https://aws.amazon.com/ru/what-is/python/>

4. PostgreSQL: The World's Most Advanced Open Source Relational Database [Электронный ресурс]. – URL: <https://www.postgresql.org/>

5. React [Электронный ресурс]. – URL: <https://ru.legacy.reactjs.org/>

6. nginx [Электронный ресурс]. – URL: <https://nginx.org/ru/>

7. Docker Builds: Now Lightning Fast [Электронный ресурс]. – URL:
<https://www.docker.com/>