

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**

(Н И У « Б е л Г У »)

**ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ИНФОРМАЦИОННЫХ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ**

**МОДУЛЬ АВТОМАТИЗАЦИИ ДОКУМЕНТООБОРОТА ДЛЯ
СИСТЕМЫ «ПЕГАС»**

Выпускная квалификационная работа студента
обучающегося по направлению подготовки 09.03.02 Информационные системы
и технологии
очной формы обучения, группы 07001409
Чеботарева Алексея Андреевича

Научный руководитель
старший преподаватель
Нестерова Е.В.

БЕЛГОРОД 2018

РЕФЕРАТ

Модуль автоматизации документооборота для системы «Пегас». – Чеботарев Алексей Андреевич, выпускная квалификационная работа бакалавра. Белгород, ФГАОУВО Белгородский государственный национальный исследовательский университет (НИУ «БелГУ»), количество страниц 47, включая приложения 47, количество рисунков 37, количество таблиц 1, количество использованных источников 25.

КЛЮЧЕВЫЕ СЛОВА: документооборот, гетерогенная система, верификация данных, цифровая подпись, PGP, метод N-грамм, метод гибкой маски, нечеткий поиск, метод расширенной выборки, метод карты указателей, SpringFramework, виртуальный тоннель.

ОБЪЕКТ ИССЛЕДОВАНИЯ: процессы обработки информации, направленный на верификацию документов и их безопасную передачу по сети интернет.

ПРЕДМЕТ ИССЛЕДОВАНИЯ: методы самокоррекции и извлечения данных из текстовой информации, а также методы аутентификации информации.

ЦЕЛЬ РАБОТЫ: автоматизация процесса внесения документальной информации в гетерогенную распределенную информационную систему.

ЗАДАЧИ ИССЛЕДОВАНИЯ: анализ существующих подходов реализации аутентификации, верификации, фильтрации, и нечеткого поиска в тексте. Разработка модуля автоматизации документооборота для системы «Пегас».

МЕТОДЫ ИССЛЕДОВАНИЯ: метод гибкой маски, метод N-грамм, метод расширенной выборки, метод карты указателей, метод аутентификации цифровой подписью.

ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ: в результате работы был разработан проект модуля автоматизации аутентификации и верификации текстовой информации, извлечение значащих данных с последующей их публикацией в системе «Пегас» и распространением в смежных системах ИнфоБелГУ.

СОДЕРЖАНИЕ

Введение.....	4
1 Исследование предметной области.....	6
1.1 Анализ процесса обеспечения документооборота.....	6
1.2 Обоснование необходимости модернизации	8
1.3 Постановка задачи.....	11
2 Проектирование и моделирование модуля	14
2.1 Технологическое обеспечение проекта.....	14
2.2 Информационное обеспечение проекта.....	25
2.3 Проектирование модуля	26
2.4 Выбор программных средств.....	35
3 Программная реализация и тестирование модуля	38
3.1 Программная реализация.....	38
3.2 Тестирование	42
Заключение	44
Список используемых источников.....	45

ВВЕДЕНИЕ

В текущее время темпы развития информационных технологий и систем претерпевает пик своего развития: доступность недорогих компьютеров, богатый рынок программных продуктов для решения различных задач. Как следствие этого, в бизнесе и производстве значительно повысился уровень автоматизации различных процессов.

К сожалению, из-за особенностей развития информационных технологий в Российской Федерации во многих организациях используют устаревший подход к внедрению информационных технологий.

Это в свою очередь вызывает появление большого количества ошибок, возникающих по вине человека (спешка, невнимательность, переутомление) и увеличенные затраты времени.

На данный момент, наиболее распространённый подхода для избегания этих ошибок – это внедрение централизованной системы управления. Но такой подход не является совершенным, так как современные централизованные информационные системы не способны решать весь спектр задачи, возникающих на предприятии. В связи с этим на больших предприятиях присутствуют несколько изолированных от общей экосистемы узкоспециализированных программных продуктов[10].

В этом сценарии четко прослеживается проблема автоматического обмена данными в экосистеме с низким уровнем взаимной интеграции информационных систем и, как следствие – неоптимальное использование человеческих ресурсов, значительно более, высокое количество ошибок и ограничение максимальной производительности в целом[17].

В данной выпускной квалификационной работе рассматривается частный случай решения проблемы обмена данными в гетерогенной информационной системе. От организации НИУ «БелГУ» был получен заказ на увеличение производительности отдела, выполняющего задачу обеспечение документооборота во внутренней подсистеме «Пегас».

После детального изучения рынка решений, позволяющих полностью или частично решать поставленную задачу, найдено не было. Причиной этого послужило то, что использующиеся в НИУ «БелГУ» информационные системы созданы по индивидуальному заказу и созданы для решения внутренних задач. На основе этого было решено создать модуль расширения функциональности для гетерогенной информационной системы «ИнфоБелГУ», частью которой и является подсистема «Пегас» (далее «модуль»).

Объектом исследования данной работы являются процессы обработки информации, направленный на верификацию документов и их безопасную передачу по сети интернет.

Предметом исследования данной работы являются методы самокоррекции и извлечения данных из текстовой информации, а также методы аутентификации информации.

Цель данной работы – автоматизация процесса внесения документальной информации в гетерогенную распределенную информационную систему, позволяющая с минимальным участием человека произвести проверку документа на подлинность с последующим сохранением его значащего содержимого.

Задачи данной работы являются:

- анализ актуальных методов верификации текстовой информации;
- анализ актуальных методов нечеткого поиска;
- анализ актуальных методов защиты информации от структурных ошибок;
- анализ актуальных методов аутентификации данных, передаваемых по сети интернет;
- создание функциональной модели модуля;
- создание полной модели потока данных модуля;
- реализация модуля.

1 Исследование предметной области

1.1 Анализ процесса обеспечения документооборота в подсистеме «Пегас»

Процесс обеспечения документооборота в подсистеме «Пегас» в рамках деятельности организации-заказчика является линейный маршрут с обратной связью, в котором каждый из этапов выполняется исключительно в ручном режиме с малой степенью интеграции информационных решений, направленных на автоматизацию процесса [15]. Участвующие роли и обязанности, возложенные на них отображены на рисунке 1.1.

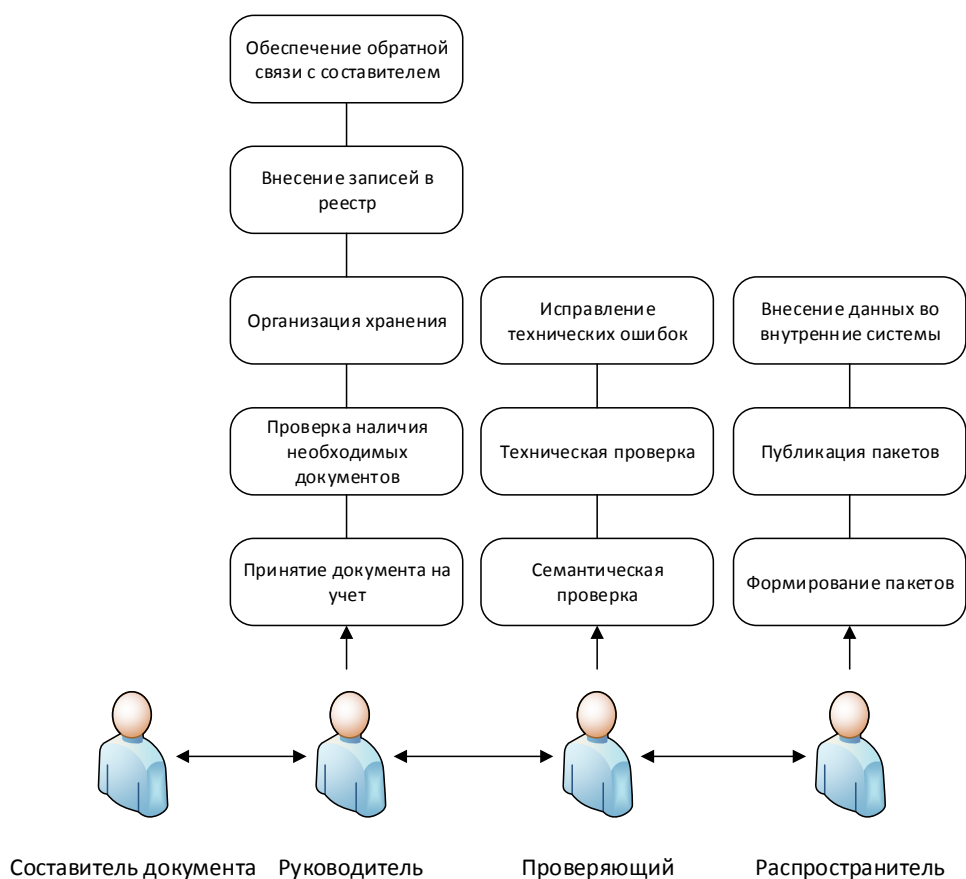


Рисунок 1.1 – Схема ролей исполнителей процесса

Сам процесс состоит из девяти основных этапов:

- формирование набора документов согласно перечню, описанному в регламенте организации;
- проверка наличия всех необходимых документов и, при необходимости, оцифровка документов, предоставляемых в печатном виде;
- проверка и подтверждение учетной информации из документов;
- внесение учетной информации в внутренний реестр для ведения контроля и отчетности;
- смысловая (семантическая) проверка документов;
- техническая проверка документов;
- формирование единого пакета документов системы «Пегас»;
- публикация единого пакета документов системы «Пегас»;
- внесение значащей информации из единого пакета документов в подсистемы ИнфоБелГУ «Пегас» и ИнфоБелГУ «Деканат».

Выполнение всех вышеописанных этапов распределено между четырьмя сотрудниками: составитель документа, руководитель, проверяющий и распространитель. Должности перечислены в порядке структурной иерархии процесса. В случае возникновения ошибок на любом из вышеописанных этапов сотрудник, отвечающий за этот этап должен сообщить о проблеме вышестоящему в процессе сотруднику и, в зависимости от принятого решения, исправить возникшую ошибку самостоятельно либо приостановить обработку документа до получения его исправленной версии.

Все этапы, перечисленные выше выполняются в строгой последовательности только при окончательном завершении предыдущего этапа.

Видно, что исполнители ролей «Проверяющий» и «Руководитель» обязаны выполнять избыточные операции, выходящие за рамки линейного маршрута.

Сотрудник на позиции распространителя обязан обладать хорошими навыками работы с персональным компьютером. Это связано с тем, что в его

обязанности входит работа со специализированным программным обеспечением и средствами удаленного администрирования веб-ресурса.

Сотрудники на позициях руководителя и проверяющего обязаны обладать хорошими навыками работы с пакетом программ MicrosoftOffice 2013. Это связано с требованиями специального программного обеспечения, формирующего единый пакет документов системы «Пегас». Кроме этого весь внутренний реестр подразделения основан на программах MicrosoftWord 2013 и MicrosoftExcel 2013.

1.2 Обоснование необходимости модернизации

При анализе процесса были сделаны следующие выводы.

Составитель документа является сотрудником другого подразделения, а также по причине наличия человеческого фактора время затрачиваемое на обратную связь является очень непостоянным.

В подразделении на руководителя возлагаются дополнительные обязанности обеспечения прямой и обратной связи с составителем документа. Данное решение создает прецедент неравномерного распределения нагрузки в линии и в теории может стать причиной задержек.

В случае отсутствия сотрудника, выполняющего роль проверяющего, руководителю придётся взять на себя все обязанности проверяющего. Это, на фоне уже имеющейся излишней загруженности руководителя, может значительно увеличить время выполнения линии.

Ведение реестра в подразделении не ведется при помощи специализированных инструментов, что в теории может стать причиной возникновения неотслеживаемых ошибок в реестре и отчетах. Кроме этого использование неспециализированных инструментов для подобных задач является причиной задержек.

В результате изучения статистики работы подразделения за последние три года была собрана информация о наиболее часто встречающихся

осложнениях и их влиянии на итоговое время прохождения. Кроме этого, в качестве встречных мер был проведен анонимный опрос среди всех участников линейного маршрута. Для наглядности собранная информация была классифицирована и собрана воедино в виде таблицы для более наглядного представления.

В таблице 1 приведено сравнение влияния ошибок, возникающие в процессе прохождения документа.

Таблица 1 – отчето осложнениях при процессе

Описание	Среднее время исправления, ч.	Относительная доля, проц.	Причина возникновения
отсутствие необходимых документов	24	36	составитель документа не ознакомился с правилами подачи документов; оповещение было получено спустя более чем 24 часа после принятия документов на учет; ошибка по невнимательности составителя; ошибка при принятии документов на учет
нарушение правил составления фонда тестовых заданий	3	20	составитель документа не ознакомился с правилами подачи документов; оповещение было получено спустя более чем 24 часа после принятия документов на учет; ошибка по невнимательности составителя
нарушение правил составления иерархии оглавлениях	2	14	составитель документа не ознакомился с правилами подачи документов; неочевидность изложения правил, касающихся иерархии оглавления; оповещение о нарушении правил; ошибка по невнимательности составителя

Продолжение таблицы 1

Описание	Среднее время исправления, час.	Относительная доля, %	Причина возникновения
нарушение правил включения схем	2	10	отсутствие изложения правил, описывающих стандарт включения схем в цифровой документ в правилах подачи документов; оповещение было получено спустя более чем 24 часа после принятия документов на учет; ошибка по невнимательности составителя
нарушение правил включения изображений	2	10	отсутствие изложения правил, описывающих стандарт включения изображений в цифровой документ в правилах подачи документов; ошибка по невнимательности составителя
нарушение правил заполнения документов с учетной информацией	1	5	составитель документа не ознакомился с правилами подачи документов; оповещение было получено спустя более чем 24 часа после принятия документов на учет; ошибка по невнимательности составителя
несоответствие данных в информационных документах учетной информации	24	3	составитель документа не ознакомился с правилами подачи документов; оповещение было получено спустя более чем 24 часа после принятия документов на учет; ошибка при принятии документов на учет
технические неисправности	4	2	несовершенство используемых в ИнфоБелГУ технологий; ошибка по невнимательности разработчик; неподходящие для выполнения задач программные решения

Из представленного отчета видно, что основными причинами возникновения осложнений являются ошибки человеческого характера (переутомление, спешка, стресс, игнорирование инструкций). Очевидно, что

минимизация участия человека в процессе снизит процент возникающих ошибок более чем на 50%, среднее время задержки уменьшится с нескольких часов до нескольких минут. Кроме этого оповещение составителя документов о найденных ошибках на этапе принятия документов на учет сократит долю ошибок, возникающих по причине запоздалого оповещения о обнаруженной ошибке.

После детального рассмотрения отчета о осложнениях можно сделать следующие выводы:

- 54% ошибок – это ошибки, возникающие на этапе технической проверки;
- 49% ошибок – это ошибки структуры содержимого документа;
- 39% ошибок возникают по причине несовершенства ведения внутреннего учета;
- 81% ошибок можно обнаружить без участия человека;
- 49% ошибок могут решаться без участия составителя;
- 24% ошибок всегда решаются без участия составителя;
- среднее время задержки из-за осложнений – 7 часов;
- нормальное время задержки из-за осложнений – 2 часа.

1.3 Постановка задачи

В результате анализа процесса «как-есть» стало очевидным, что большинство этапов, а именно проверка наличия всех необходимых документов, внесение информации реестр, техническая проверка документа, формирование единого пакета документов и его публикация в линейном маршруте являются монотонными однотипными повторяющимися действиями со строгими четко расписанными инструкциями. По Л. И. Селевцову данные действия являются полностью автоматизируемыми[21].

На основе результатов исследования предметной области можно сделать вывод, что полная автоматизация этапов, следующих за семантической

проверкой уменьшит частоту возникновения ошибок минимум на 45%, ускорит прохождение документа минимум на 2 часа и уменьшит нагрузку на персонал минимум в два раза, а создание специализированного инструмента для ведения реестра снизит частоту возникновения ошибок на 10%, а время прохождения документа сократится на 1 час.

На рисунке 1.2 можно наблюдать предполагаемые изменения в существующей системе ролей, а также степень автоматизации обязанностей.

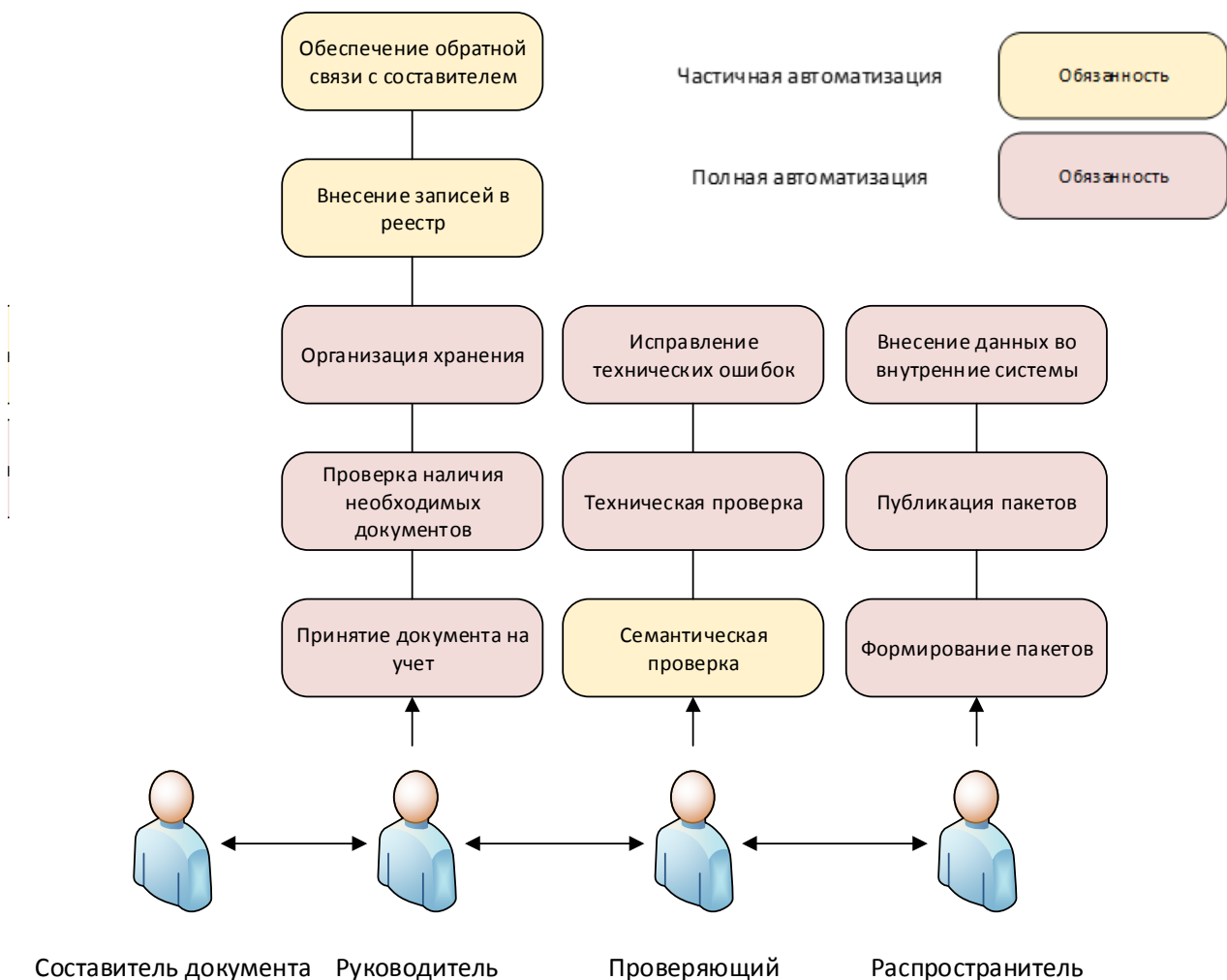


Рисунок 1.2 – Предлагаемая схема ролей процесса

Стоит обратить внимание, что все задачи, выполняемые распространителем полностью автоматизируемые. Из этого можно сделать вывод, что после модернизации у предприятия освободятся человеческие ресурсы, которые можно будет использовать для решения более требовательных задач.

Как следствие из сформулированного предложение следует цель – автоматизация процесса внесения документальной информации в гетерогенную распределенную информационную систему, позволяющую с минимальным участием человека произвести проверку документа на правильность заполнения с последующей обработкой его значащего содержимого.

И для достижения поставленной цели, было выявлено задачи:

- разработка постановки задачи;
- анализ существующих методов верификации текстовой информации;
- анализ существующих методов нечеткого поиска ключевой информации в тексте;
- анализ актуальных методов защиты информации от структурных ошибок;
- анализ актуальных методов аутентификации данных, передаваемых по сети интернет;
- создание функциональной модели модуля;
- создание полной модели потока данных модуля;
- реализация модуля.

Выводы по первому разделу:

В данном разделе было произведено исследование процесс документооборота в системе «Пегас», который включал анализ процесса «как-есть» для выведения перечня недостатков имеющегося процесса; обоснование необходимости исправления обнаруженных недостатков на основе анализа воздействия оных на имеющийся процесс; формулировку способов исправления обнаруженных недостатков.

2 Проектирование и моделирование модуля

2.1 Технологическое обеспечение проекта

В данном проекте выделяются четыре основные функциональные задачи:

- определение вида анализируемого документа;
- поиск значащей информации в документе;
- исправление структурных ошибок в документе;
- безопасная передача обработанных данных по сети интернет.

Определение вида анализируемого документа – это верификация, необходимая для первичной проверки правильности заполнения документов, а также способ определения типов, значащих данных, которые хранятся в обрабатываемом документе[22]. Так как согласно регламенту предприятия, в данном процессе участвует исчисляемое количество документов со строгой формой, то данная функциональная задача сводится к определению класса информации по шаблону.

Поиск значащей информации в документе в данном проекте осложняется особенностями языка и человеческим фактором. В результате искомый фрагмент может незначительно отличаться от эталона, не теряя при этом своего значащего статуса. В связи с этим задача поиска значащей информации сводится к нечеткому поиску с коррекцией.

Исправление структурных ошибок – это процесс поиска и коррекции ошибок, возникших по несоблюдению правильности структуры заполнения документа[18]. Для решения данной проблемы принято использовать самовосстанавливающиеся структуры данных– это обуславливается относительно высокой производительностью современных систем и максимальной надежностью подхода по сравнению с существующими альтернативами [5]. В связи с этим задача исправления структурных ошибок сводится ко внедрению самовосстанавливающихся структур данных.

Безопасность передачи данных по сети на сегодняшний день является одной из самых острых и актуальных проблем в сфере информационных технологий. Обеспечение секретности, сохранности и однозначности данных совершенствовалось на протяжении всего периода существования удаленных соединений. На сегодняшний день выделяют два основных подхода к обеспечению безопасности передачи данных:

- симметричное шифрование;
- асимметричное шифрование.

Важно отметить, что два этих подхода, хоть и являются взаимозаменяемыми в некоторых случаях, решают разные задачи. Поэтому для аргументации дальнейшего выбора важно уточнить задачу, представленную выше [7].

Безопасность передачи данных в контексте данного проекта – это защита системы от ошибочного внесения информации в систему злоумышленником. Данное понятие включает в себя как создание новых экземпляров данных, так и изменение существующих, в том числе тех, над которыми еще не завершилась обработка.

Из представленного выше определения можно сделать вывод, что задача защиты информации от хищения не входит в задачи данного проекта, так как вся обрабатываемая значащая информация является публичной и не подпадает под действие закона №152-ФЗ «О персональных данных» [20].

Вывод из проведенного выше анализа: задача безопасная передача обработанных данных по сети интернет сводится к обеспечению надежной цифровой подписи для передаваемых данных.

2.1.1 Метод гибкой маски

Метод гибкой маски – это один из методов проверки соответствия информации определенному шаблону. Данный метод широко используется в современных информационных системах и реализован в различных интерпретаторах регулярных выражений и средствах сжатия информации (архиваторы, экономные форматы изображений и звука), а также в машинном обучении [3].

Пример работы алгоритма поиска на основе метода гибкой маски изображен на рисунке 2.1.

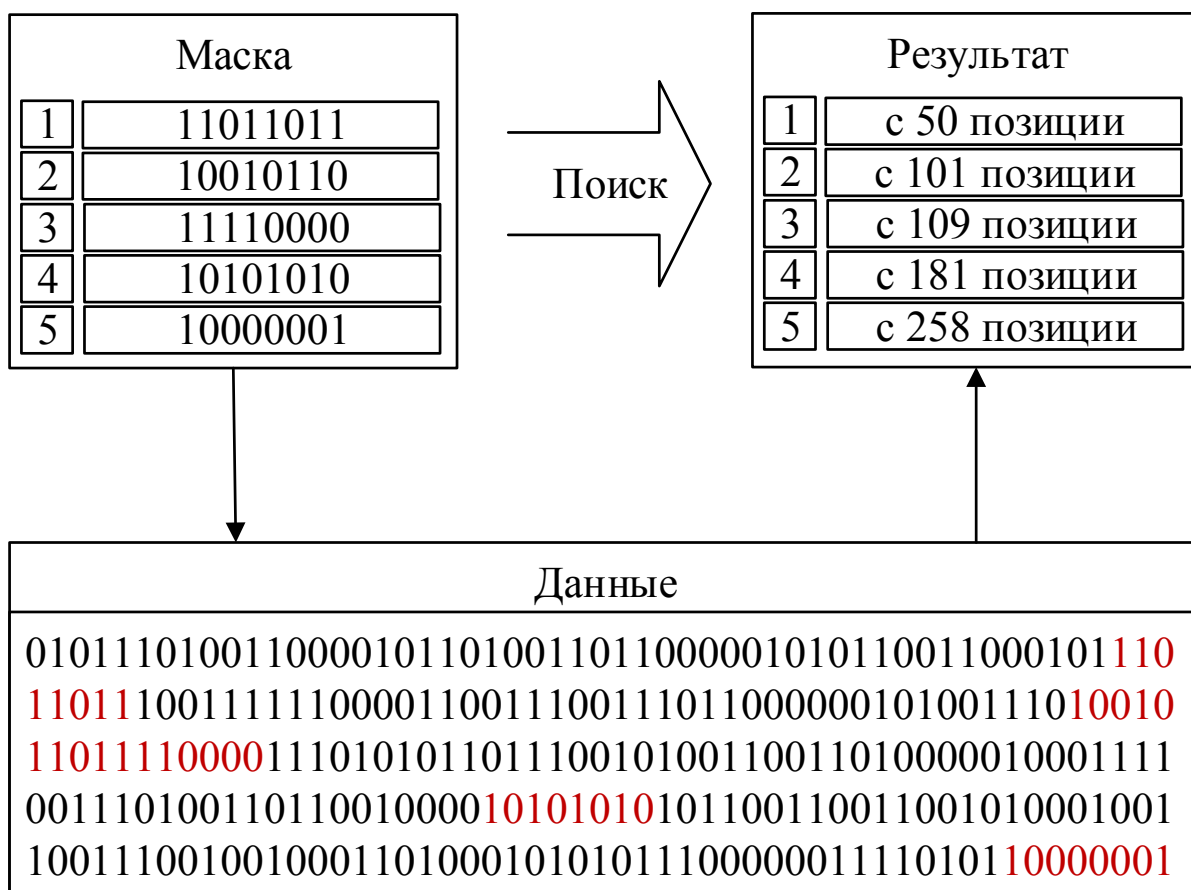


Рисунок 2.1 –Схема работы

Метод гибкой маски заключается в поиске некоего упорядоченного набора различных групп символов в потоке информации с последующим анализом результатов поиска. В зависимости от того, какие именно группы были найдены в исходном потоке, на каком расстоянии они находятся друг от друга

и какими отличительными характеристиками обладает информация, находящаяся между ними и формируется окончательное решение о принадлежности анализируемого потока информации тому или иному шаблону[4].

Как пример, ниже на рисунке 2.2 представлена маска, позволяющая определить является ли обрабатываемый документ заявлением от студента.

Маска	
1	Директору
2	студент
3	группы
4	Заявление
5	Прошу
6	Дата:
7	Подпись:

Рисунок 2.2 – Пример маски документа

Метод гибкой маски лучше всего подходит для определения типа информации неизвестной длины. Особо хорошо данный способ показывает себя при анализе текстовой и звуковой информации. Кроме этого, использование данного метода совместно со средствами нечеткого поиска и исправления ошибок позволяют эффективно использовать данный метод для верификации [12].

2.1.2 Методы нечеткого поиска в тексте

Из-за специфики поставленной задачи в данном проекте возникает проблема поиска текстовых данных, в которых могут присутствовать различные структурные (опечатки, пропуски, замены) и семантические (сокращения, падежи, рода, числа) отклонения. В связи с этим в проекте используется два различных метода нечеткого поиска: метод N-грамм и метод

расширенной выборки. Это обусловлено их взаимодополнением и различной спецификой подходов.

Метод N-грамм – это широко используемый метод нечеткого поиска, основывающийся на идее того, что даже при возникновении структурной ошибки в обрабатываемой информации в ней все еще остаются неизменные упорядоченные фрагменты из исходных данных [24].

Схему работы алгоритма на основе метода N-грамм рассмотрен на рисунке 2.3.

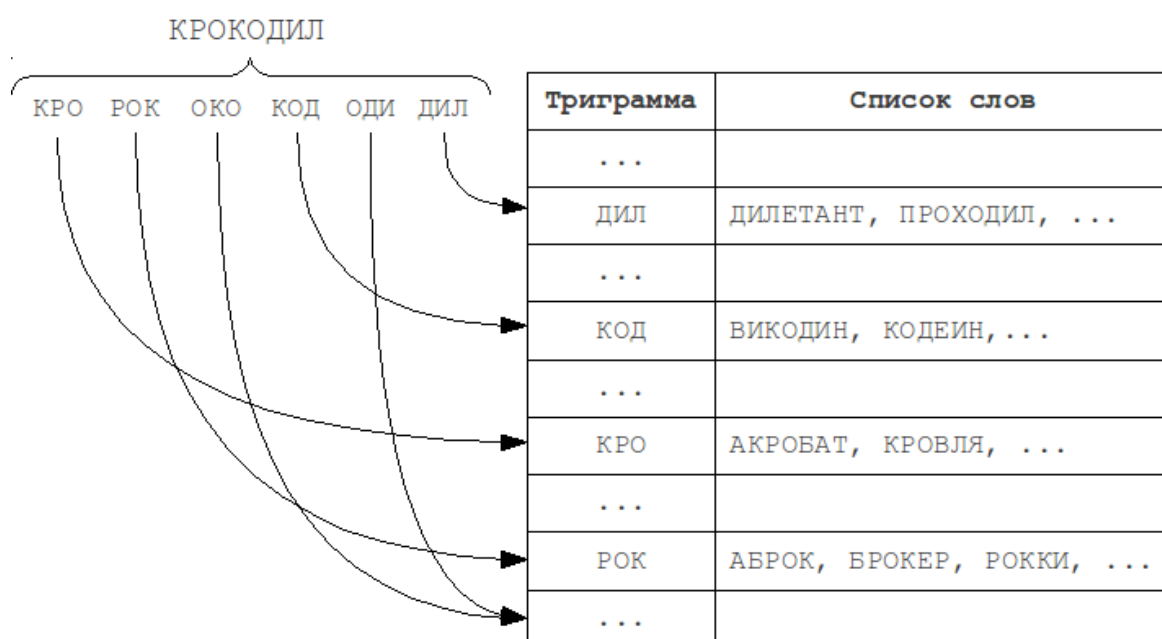


Рисунок 2.3 – Схема работы

Метод заключается в поиске подобных фрагментов в исходной информации с последующим совмещением множеств результатов поиска по различным фрагментам в целях обнаружения элемента, входящего в максимальное количество пересечений сформированных множеств [13].

Метод N-грамм уже успел хорошо себя зарекомендовать в среде информационных технологий. На данный момент это самый простой и самый часто используемый метод нечеткого поиска, и область его применения начинается генетики и заканчивается исследованиями космических снимков и индексации результатов поиска. Эффективность работы данного метода при работе с текстовой информацией наглядно продемонстрировала компания

Google в сентябре 2006 года – после долгосрочного исследования компанией был разработан и опубликован инструмент для статического машинного перевода, который согласно результатам исследования, превосходил актуальный на тот момент технологию статического перевода более чем на 80% [11].

У поиска методом N-грамм выделяют два основных недостатка [14]:

- относительно невысокая точность, связанная с ложноположительными результатами поиска;
- необходимость создания предварительного индексирования перед поиском.

Так как время, необходимое на индексирование документа средних размеров несоизмеримо меньше со временем проведения документа, было решено пренебречь последним описанным недостатком. Для решения проблемы избыточности в итоговой выборке было решено использовать вторичный поиск методом расширенной выборки.

Метод расширенной выборки – это метод нечеткого поиска в тексте, основной идеей которого является превращение одной задачи нечеткого поиска в множество задач четкого поиска [14]. Это обеспечивается за счет формирования словаря мутаций – расширяющей выборки на основе всевозможных изменений искомого слова, а именно:

- замена;
- перестановка;
- вставка;
- удаление.

Особенности работы метода позволяют легко расширить словарь мутаций за счет расширения области мутации с двух символов до трех и более, а так же создания мутаций на основе нескольких изменений.

На рисунке 2.4 изображен наглядный пример формирования словаря мутаций с базовым размером в 1 символ.

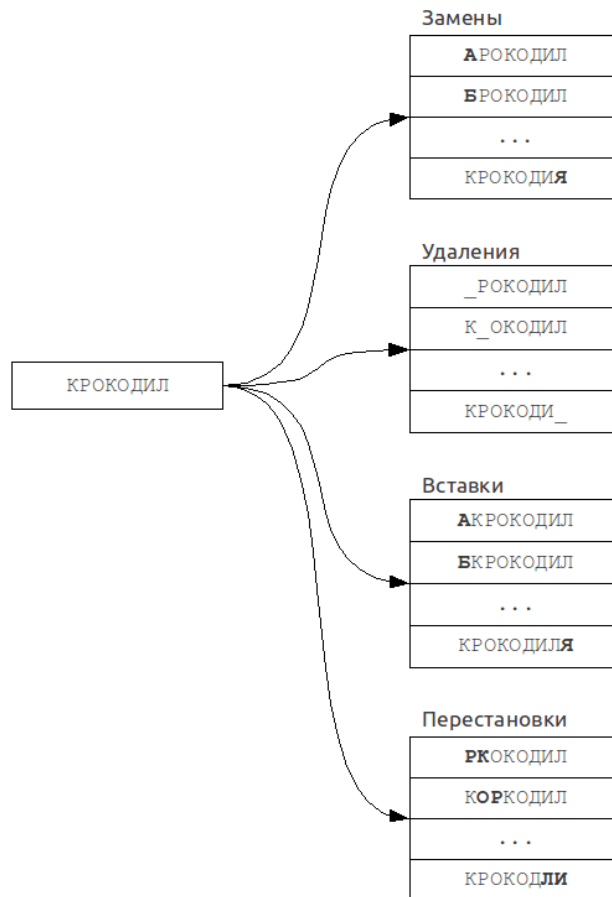


Рисунок 2.4 – Пример словаря мутаций

Одной из самых важных особенностей данного метода то, что данный метод является эффективным только при поиске семантически нейтральных слов (число, род и падеж искомого не отличаются от такового у анализируемого). Именно эта особенность не позволяет ограничиться данным методом как единственно верным для решения задачи нечеткого поиска в целом [16].

На практике метод расширенной выборки показывает точность бесконечно близкую к 100%. Эта особенность сделала данный метод наиболее популярным средством для алгоритмов исправления орфографических ошибок и алгоритмов распознавания речи[6].

2.1.3 Метод карты указателей

Метод карты указателей – это метод организации самовосстанавливающейся структуры данных, основывающейся на упорядоченных индексах[19].

Рисунок 2.4 иллюстрирует упрощенный принцип работы алгоритмов, основанных на методе карты указателей.

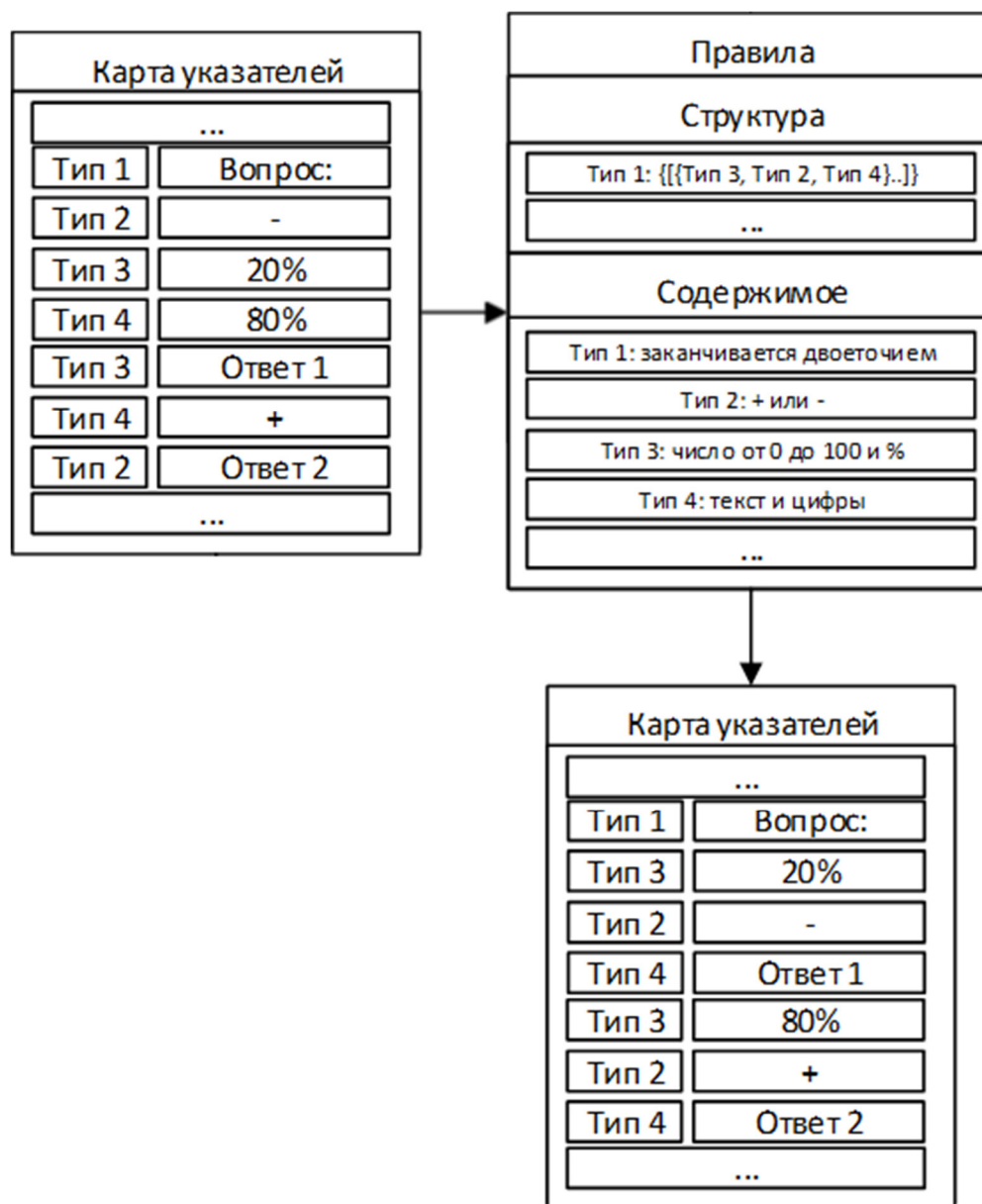


Рисунок 2.5 – Схема работы карты указателей

Данный метод заключается в формировании двух сопроводительных структур вокруг существующих данных:

- карта указателей, которая состоит из столбцов: кванта информации и его класса;
- правила карты, в которых описываются варианты шаблонные структуры классов.

За счет подобного способа хранения информации даже значительные нарушения в порядке классов или ошибочное указание класса для любого кванта информации может быть исправлено за счет исчерпывающих правил карты[19].

Ниже на рисунке 2.6 изображен пример правил для одного из классов используемой в проекте карты.

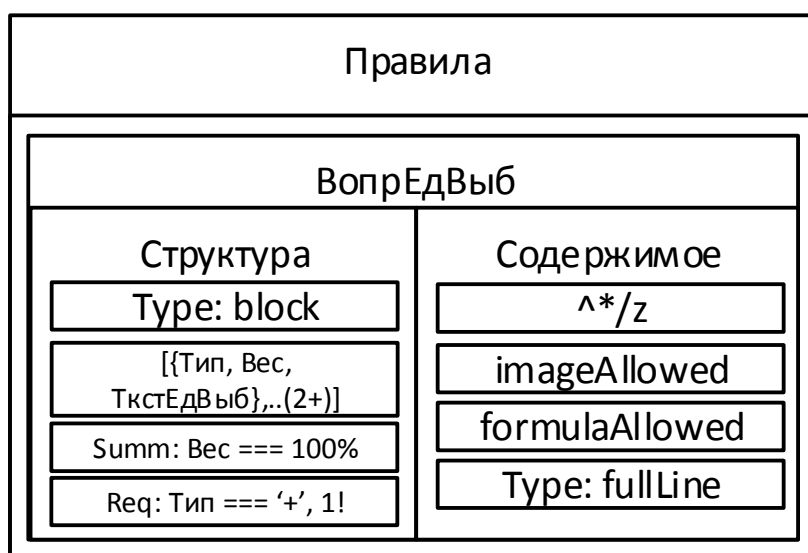


Рисунок 2.6 – Пример правил карты

Основным недостатком карты указателей является требование к частичному соблюдению правил в изначальной карте указателей. Без этого требования во время анализа карты указателей может возникнуть ложноотрицательный результат при проверке сегмента на отсутствие ошибок. Кроме этого важным является изначально продуманная организация правил карты, которые позволят однозначно определить тип рассматриваемого

сегмента, иначе может возникнуть коллизия при выборе правил, которым должен удовлетворять обрабатываемый сегмент [12].

Главным преимуществом метода карты указателей его концептуальная простота и наглядность – при правильной реализации данного метода даже пользователь без высокого уровня информационной образованности может составить документ, который в дальнейшем после процесса самокоррекции будет полностью верно структурирован и пригоден к дальнейшей обработке. Это преимущество и нетребовательность к способу реализации делает данный метод идеальным для решения поставленной функциональной задачи [20].

2.1.4 Цифровая подпись OpenPGP

Аутентификация при помощи цифровой подписи – это способ аутентификации сообщения, заключающийся на формировании специальной реквизита (цифровой подписи), позволяющая однозначно подтвердить отправителя сообщения и проверить оное на наличие непреднамышленных изменений в последнем[2].

Общую концепцию работы цифровой подписи данных можно рассмотреть на рисунке 2.7.

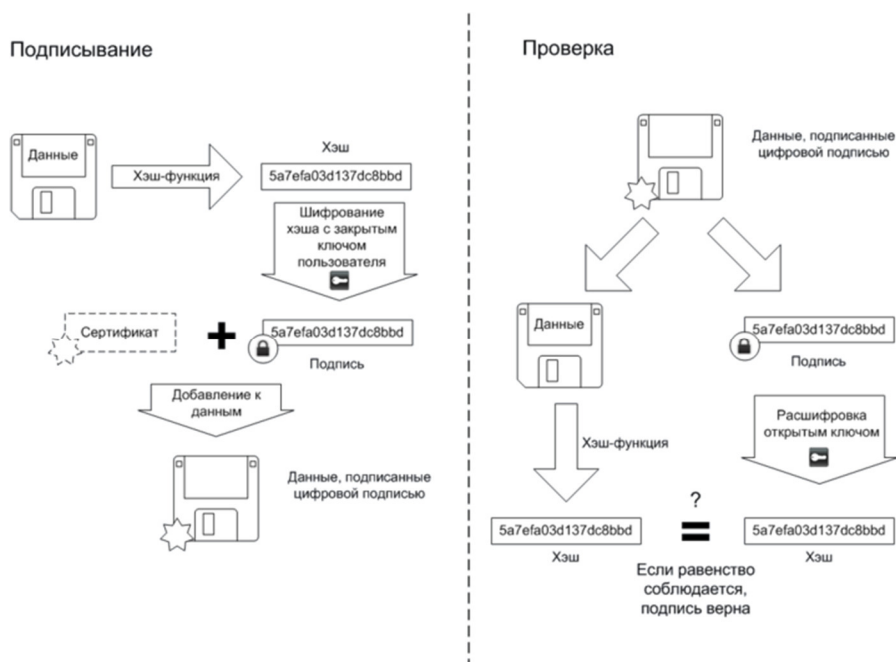


Рисунок 2.7 – Схема работы цифровой подписи

В процессе формирования цифровой подписи участвует закрытый ключ, а при проверке – открытый. Данная пара ключей заблаговременно генерируется отправителем локально. Закрытый ключ всегда остается секретным и никак не распространяется. Открытый же ключ является общедоступным и передается отдельно от сообщения перед подписанием

Безопасность цифровой подписи полностью опирается на секретность закрытого ключа. Злоумышленник, обладающий копией закрытого ключа сможет самостоятельно подписывать сообщения и представлять их от лица владельца закрытого ключа.

В коммерческих системах цифровой подписи, таких как TSL или MicrosoftSecureOffice, за генерацию закрытых ключей и подтверждения владельцев открытых отвечает специальный представитель сети (сертифицирующий центр). Это позволяет предотвратить компрометацию публичного ключа при помощи атаки «человек-по-середине» [25].

К сожалению алгоритмы OpenPGP являются открытыми. Это значит, что любой злоумышленник может сгенерировать ключевую пару от имени доверенного узла. Для того, чтобы предотвратить это в системе цифровой подписи OpenPGP используется принцип сети доверия (рисунок 2.8).

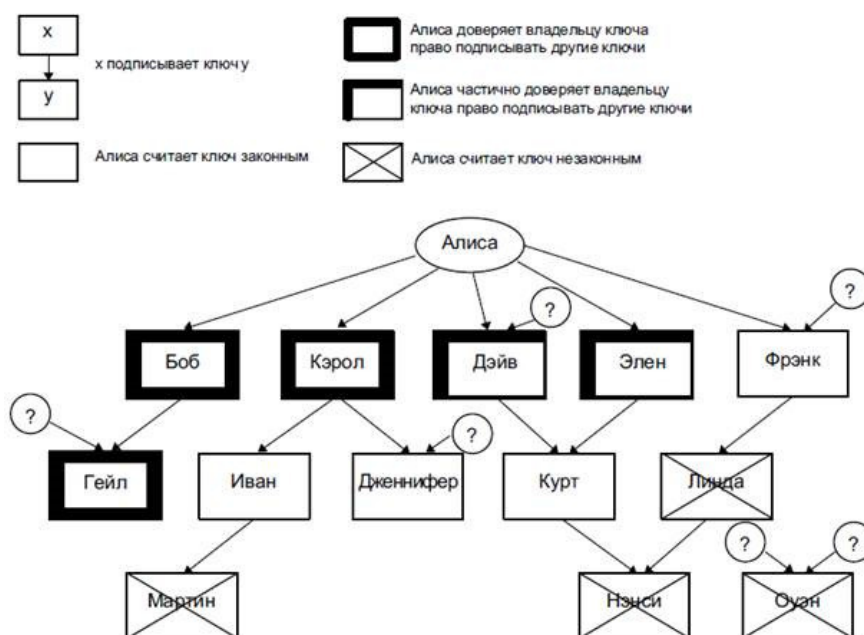


Рисунок 2.8 – Схема работы Сети доверия

Сеть доверия – это сообщество клиентов в единой сети, каждый из которых обладает полномочиями центра сертификации. При этом степень доверия к каждому клиенту является субъективной [8].

2.2 Информационное обеспечение проекта

В качестве информационного обеспечения данного проекта используются три источника информации:

- файлы с тексто-графической информацией в форматах MicrosoftOfficeWord (.doc, .docx) и PlainTextCompatibleFormats (.txt, .rtf, .cvs);
- файлы с не анализируемой информацией в различных форматах;
- реляционная база данных.

Для стандартизации передачи информации между системами, с которыми взаимодействует модуль был разработан формат единого пакета документов. Он представляет из себя ZIP-архив, который включает в себя:

- директорию «Программная часть», содержащую в себе оцифрованные копии регистрационного листа, рабочего плана дисциплины, выписки кафедры, внутренней и внешней рецензии;

- директорию «Методическая часть», содержащую в себе оцифрованные копии конспекта лекций, заданий для семинарских, лабораторных и практических работ, глоссария и руководства к изучению дисциплины;

- директорию «Фонд тестовых заданий», содержащую в себе вопросы для оценивания знаний с картой указателей вариантов ответов, а также оцифрованную копию паспорта фонда тестовых заданий;

- ZIP-архив «Дидактические материалы».

Важно отметить, что конспект лекций, темы для семинаров и задания для лабораторных и практических работ не являются обязательными документами. Согласно регламенту предприятия заказчика, в едином пакете документов должен присутствовать хотя бы один из выше перечисленных документов – в ином случае комплект документов считается неполным и формирование единого пакета документов происходить не должно.

2.3 Проектирование модуля

2.3.1 Модель бизнес процесса

Проектируемый модуль полностью автоматизирует следующие процессы:

- сопоставление представленных документов с учебным курсом;
- техническая проверка предоставленных документов;
- публикация курса в системе «Пегас» путем создания курса;
- внесение записей в систему ИнфоБелГУ «Рейтинг».

Так же проектируемый модуль частично автоматизирует следующие процессы или предоставляет адаптивный инструментарий для оптимизации выполнения данных процессов человеком:

- управление реестром подразделения;
- семантическая проверка предоставленных документов;
- проверка сопроводительных документов;
- проверка фонда тестовых заданий;
- обратная связь с составителем.

На основе этого функционала была составлена модель итогового бизнес-процесса в нотации IDEF0 с двухуровневой декомпозицией сложных процессов.

Функциональный блок «Модуль автоматизации документооборота для системы «Пегас»», изображенный на рисунке 2.9 является представлением всех процессов, происходящих в проектируемом модуле

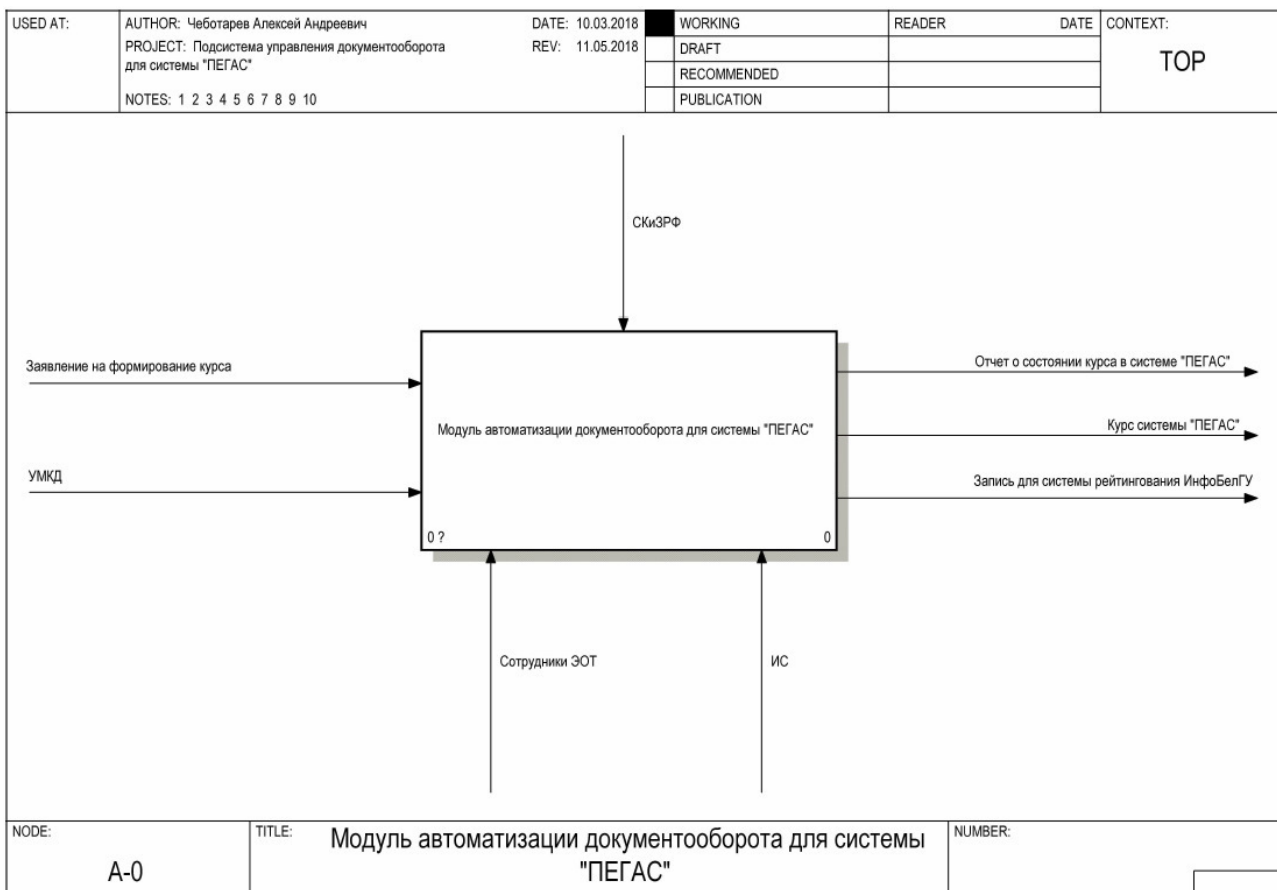


Рисунок 2.9 – IDEF0 модель конечного бизнес-процесса

Входными параметрами модуля являются Заявление на формирование курса и УМКД. Заявление на формирование курса формируется без участия составителя на основании заказов кафедры и результатов заседания факультетов. УМКД в данной модели используется как алиаст к набору документов, подвываемых составителем для проведения одного единого пакета документов.

Так как предприятие-заказчик является государственным учреждением, то весь процесс регламентируется Сводом Кодексов и Законом Российской Федерации.

Все операции в моделируемом процессе производятся сотрудниками отдела электронных образовательных технологий при помощи инструментов управления информационных систем.

Конечным результатом всех операций являются:

- опубликованный в виде учебного курса в системе «Пегас», сформированного на основе единого пакета документов;
- служебные записи во всех смежных системах ИнфоБелГУ;
- отчет о процессе проведения документа и информация о состоянии сформированного в системе «Пегас» курса.

Ниже на рисунке 2.10 представлены все основные процессы, происходящие в модуле.

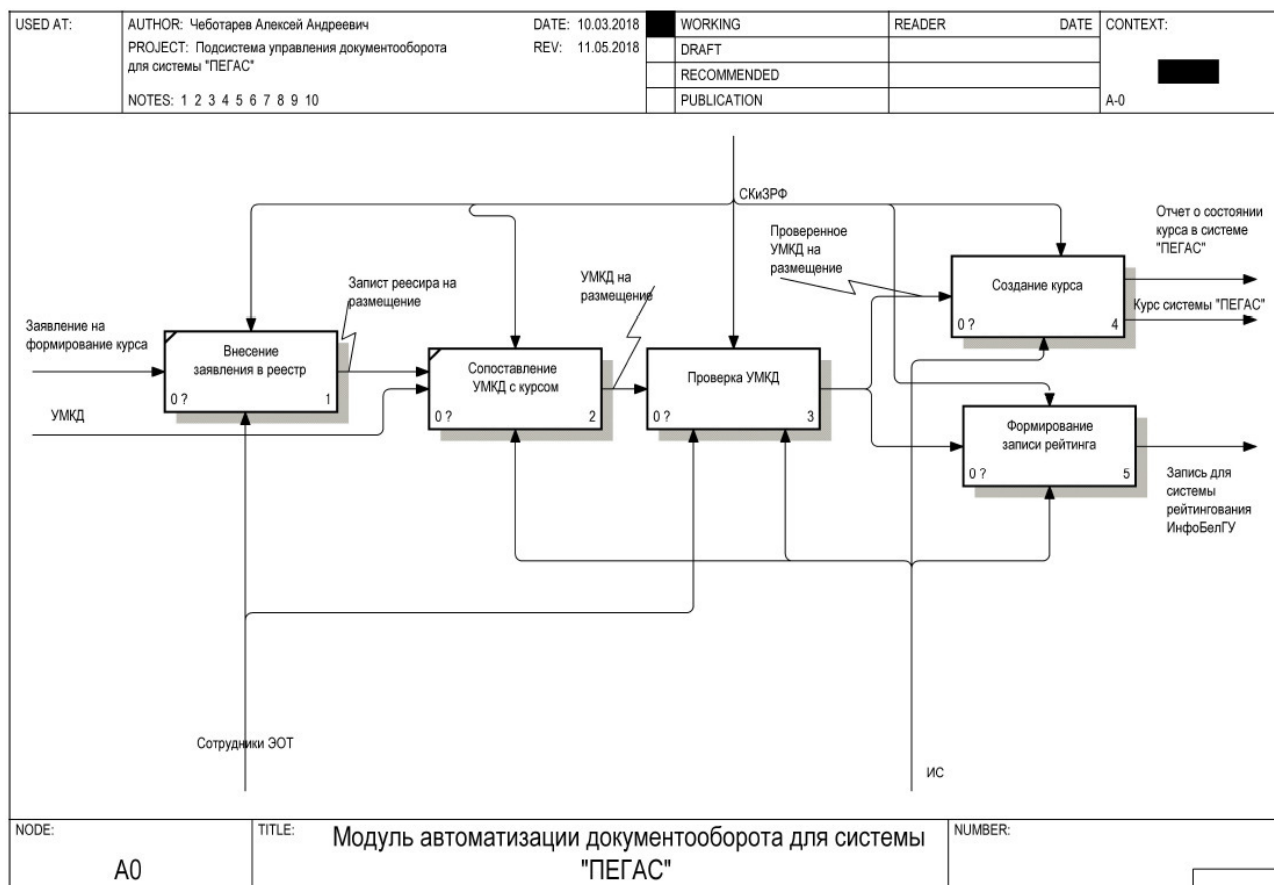


Рисунок 2.10 – Декомпозиция процессов, происходящих в модуле автоматизации

На рисунке 2.10 можно наблюдать все основные процессы, происходящие в модуле. Так же на нем можно четко наблюдать этапы распространения информации из проверенного пакета документов смежным системам. На представленной схеме видно, что сотрудник принимает участие только в двух этапах формирования пакета документов, а именно «Внесение заявления в реестр» и «Проверка УМКД». Данный факт вызван тем, что:

- на этапе внесения заявления в реестр формат и формы предоставляемых данных не стандартизирован, в связи с чем автоматизация данного этапа невозможна;
- на этапе проверки УМКД возможны сценарии возникновения ошибки, которую невозможно исправить без участия оператора и/или составителя документов.

На рисунке 2.11 представлено детальное представление процесса проверки документов, сопряженных с составляемым единым пакетом.

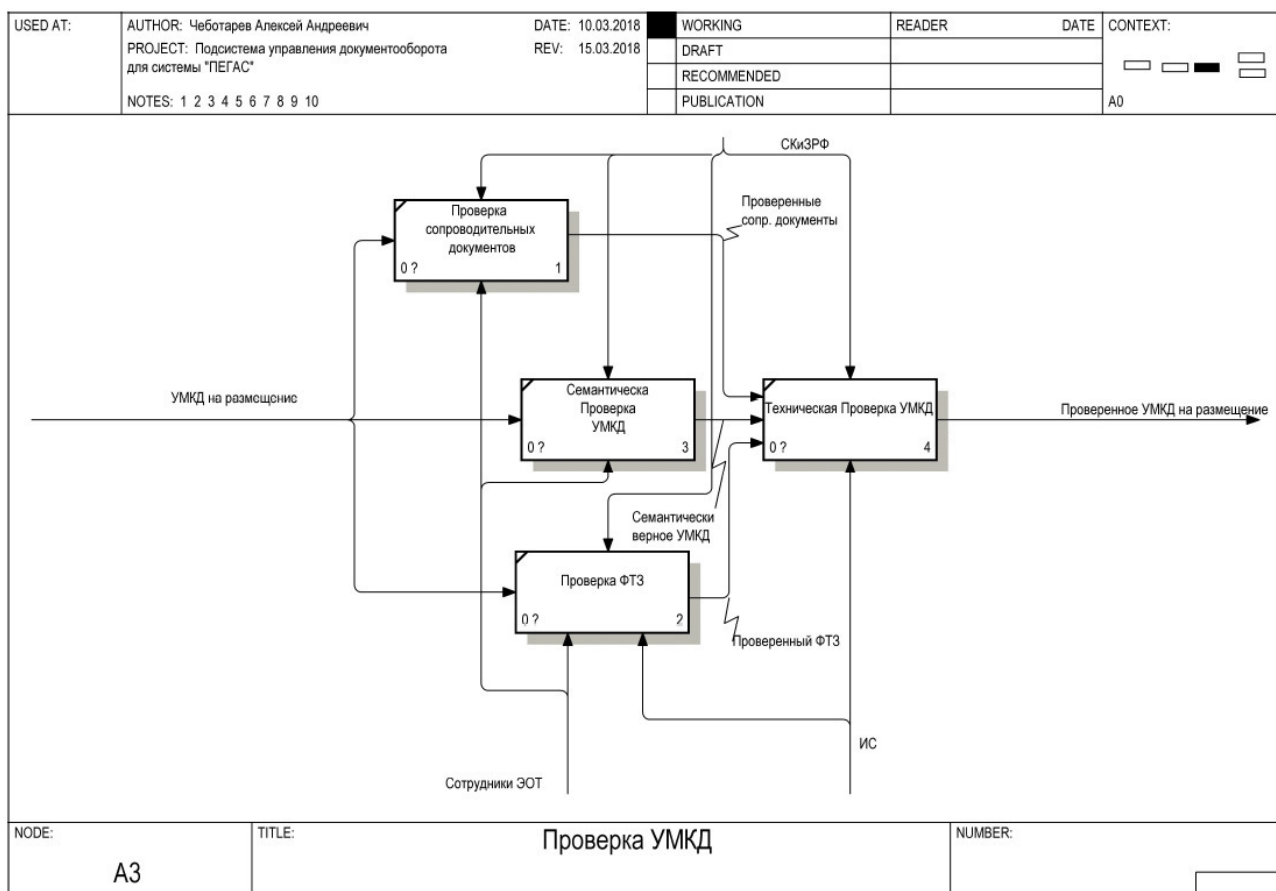


Рисунок 2.11 – Декомпозиция процесса «Проверка УМКД»

На этапе «Проверка УМКД» оператор участвует в процессах «Проверка сопроводительных документов», «семантическая проверка УМКД» и «Проверка ФТЗ». Данное решение обосновывается следующим:

- проверка сопроводительных документов включает в себя верификацию документов, предоставляемых в устаревших форматах;

- при семантической проверке производится проверка соответствия смыслового содержания с регистрационной информацией, что требует большого количества вычислительных ресурсов при автоматизации;
- при проверке ФТЗ могут возникнуть ошибки, неразрешимые без смыслового анализа содержания документа.

На рисунке 2.12 представлена все этапы публикации единого пакета в системе «Пегас».

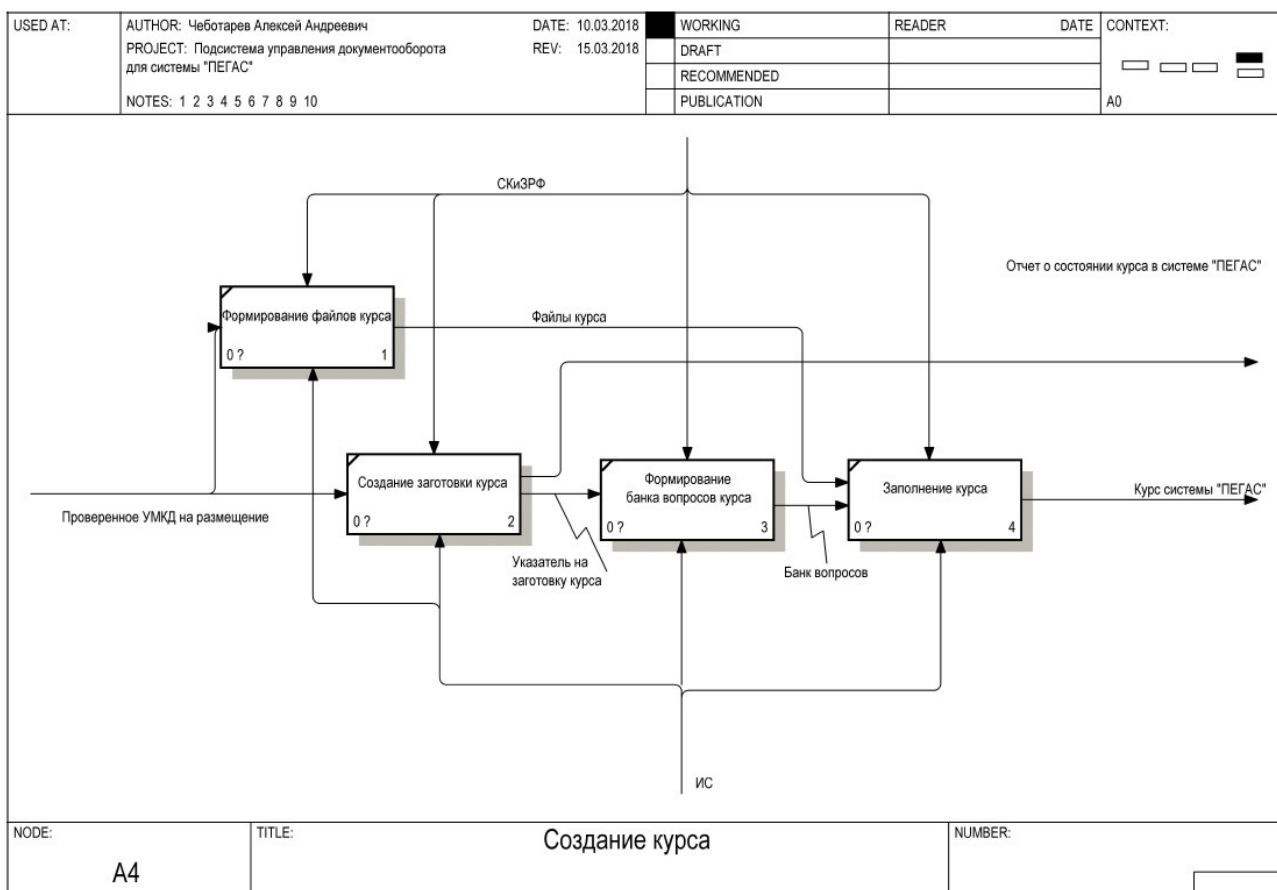


Рисунок 2.12 – Декомпозиция процесса «Создание курса»

Так как шанс возникновения ошибки при формировании пакета документов согласно вышеописанному процессу крайне мал, то возможно безусловно гарантировать отсутствие ошибок при формировании файлов курса и формировании банка вопросов курса.

На рисунке 2.13 изображена декомпозиция процесса формирования и отправки записи о УМКД в подсистему рейтинговая ИнфоБелГУ.

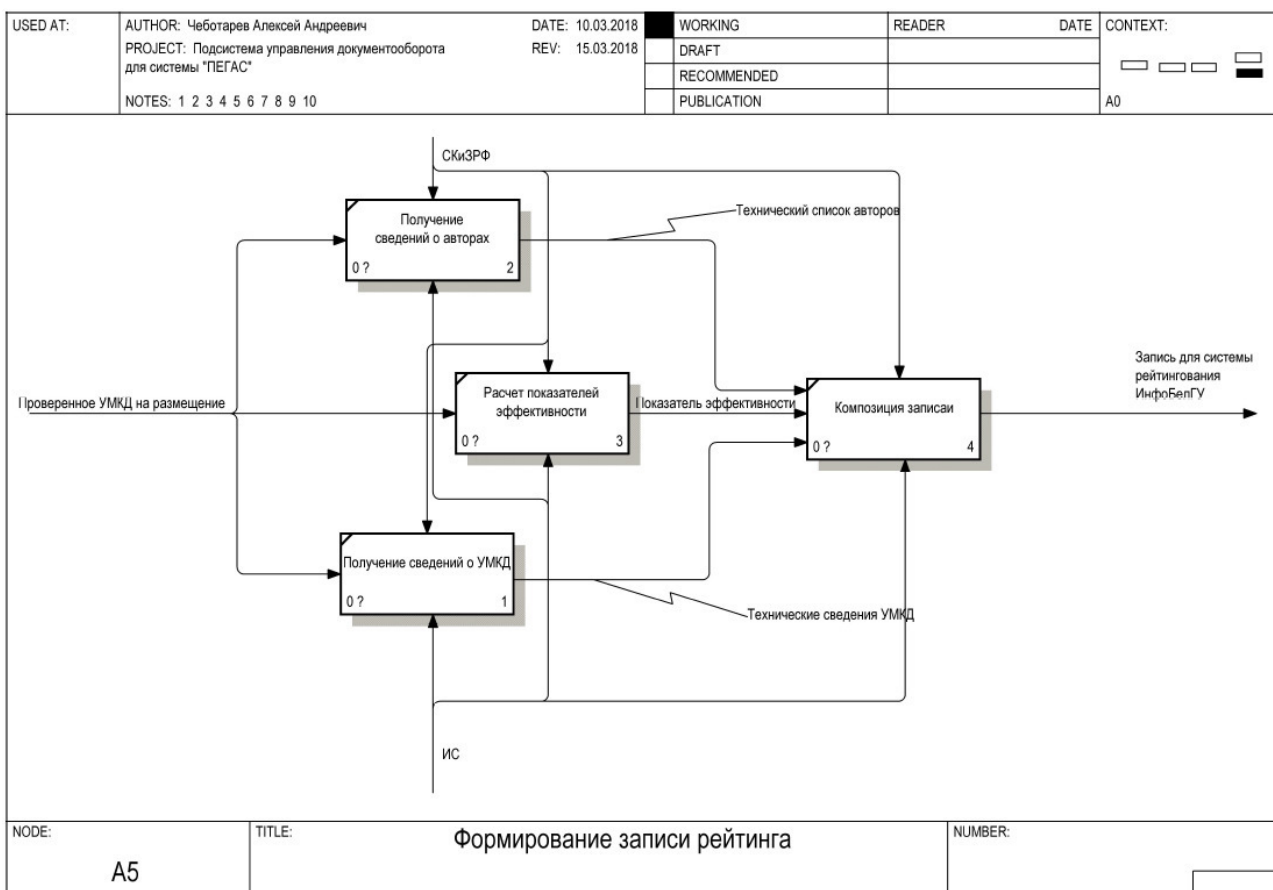


Рисунок 2.13 – Декомпозиция процесса «Формирование записи рейтинга»

Данный подход позволяет оптимизировать прохождение документа за счет параллельного выполнения двух задач:

- подготовка курса для публикации информации из пакета документов в системе «Пегас»;
- подготовка файлов для публикации информации из пакета документов в учебном курсе системы «Пегас».

Формирование записи рейтинга – это процесс, который использует данные из трех различных систем. Как следствие этого на рисунке 2.13 можно наблюдать высокий уровень квантования данного этапа на специализирующиеся подзадачи. Подобный подход позволяет достигнуть следующих целей:

- оптимизировать процесс за счет параллельной обработки данных из разных источников;

– уменьшить шанс возникновения ошибок за счет рассмотрения каждой подзадачи как самостоятельного проекта с соответствующим подходом к разработке;

– изолировать области вызова данных из различных источников, и, как следствие, упростить процесс конфигурации и реализации.

2.3.2 Модель потока данных

В качестве решения была выбрана схема виртуального туннелирования на основе приватных цифровых подписей (рисунок 2.16). Данное решение обосновано простотой реализации, высоким уровнем надежности и существованием свободных для использования технологий [23].

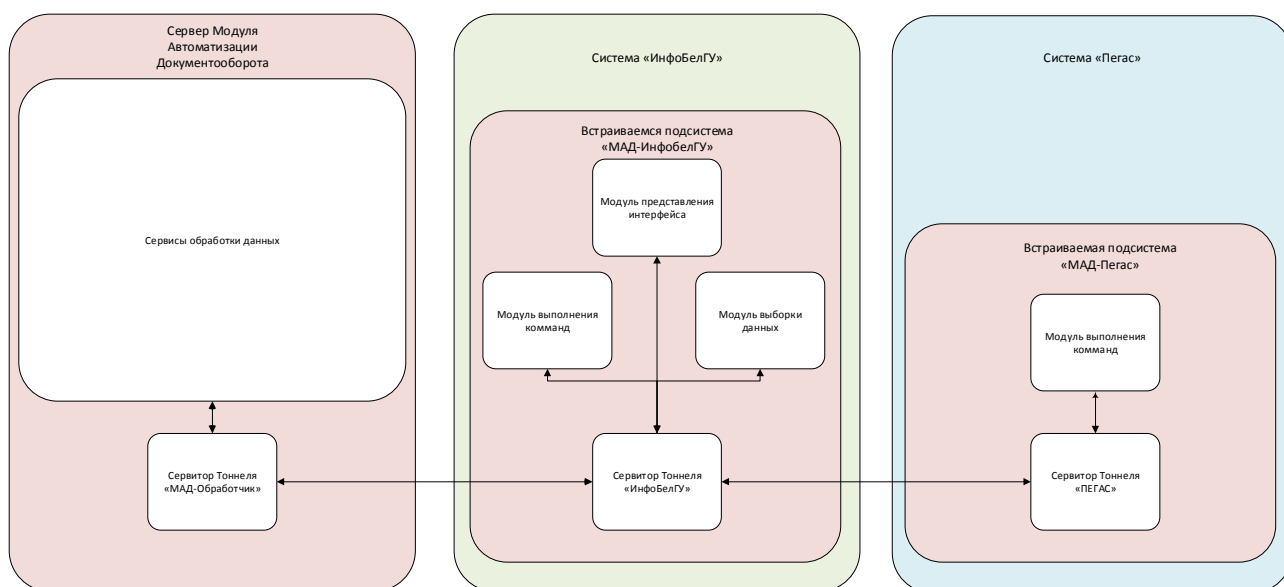


Рисунок 2.16 – Схема туннелирования данных

Можно увидеть, что для обеспечения унификации в каждую из рассматриваемых в проекте подсистем решено внедрить специализированный модуль, отвечающий за модуляцию информации между туннелем и локальной системой.

Работа моделируемой системы завязана на оптимизации и упрощении обмена данным между несколькими объектами без взаимной интеграции. Следствием из этого является необходимость структурирования порядка

обращений, их содержимого и его информационного наполнения, а также обеспечение целостности передаваемой информации. Из-за того, что в существующих подсистемах ИнфоБелГУ отсутствует единый стандарт децентрализованной передачи информации, то было решено разработать онный для гарантии безопасности при дальнейшей эксплуатации.

Для наглядности информации на рисунке 2.15 изображена схема отражающая перемещение данных, обрабатываемых модулем.

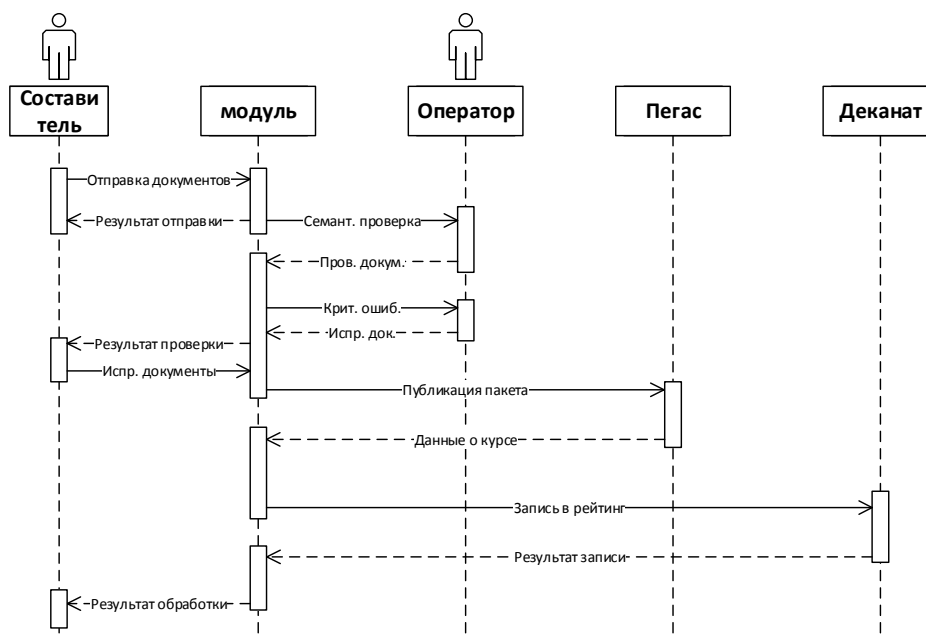


Рисунок 2.15 – Схема последовательностей данных

Видно, что из всех взаимодействий между различными объектами межсистемными являются две. Из этого можно сделать вывод о том, что данные взаимодействия включают в себя передачу большого объема данных по сети интернет.

Использование туннеля продиктовано тем, что ни одна система ИнфоБелГУ не использует HTTPS для внутренних и внешних обращений. Из-за небезопасности данного подхода и прогрессирующим отказом большинства разработчиков от HTTP для сохранения совместимости с используемыми в

реализации проекта библиотеками и фреймворками решено использовать WebSocket как основной протокол обмена данными.

2.3.3 Модель базы данных

Одной из основных задач проектируемой задачи является объединение разрозненных хранилищ данных в единую инфраструктуру. Для того, чтобы этого было достижимо необходимо чтобы у модуля присутствовал функционал обеспечения целостности данных. В этих целях у самого модуля есть внутренняя база данных.

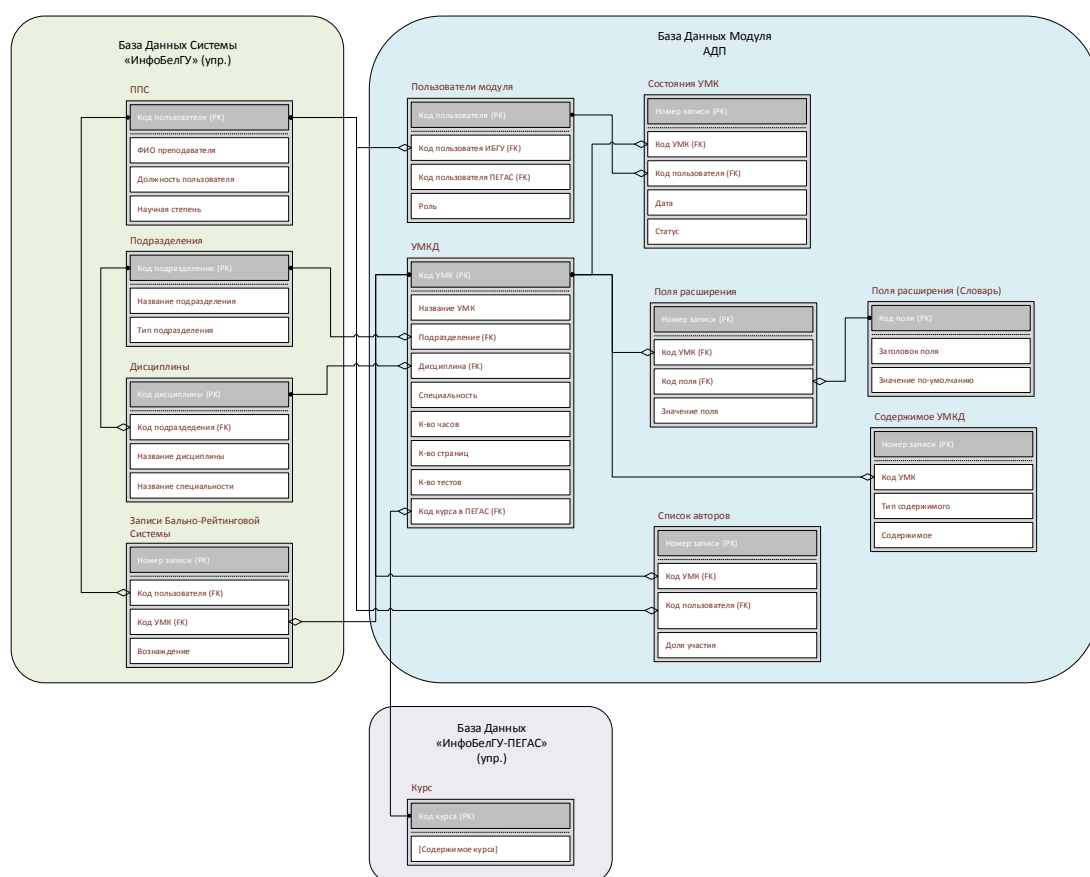


Рисунок 2.14 – Упрощенная схема базы данных

В моделируемом модуле присутствует большое количество связей между разными базами данных. Это обеспечивается за счет промежуточного синтаксического анализатора запросов в каждой системе – данное расширение используя расширенный синтаксис запросов и динамические временные

таблицы, создаваемые обработчиком самостоятельно получает необходимые данные из смеженной базы данных, передает их по защищенному протоколу и включает результат выборки в анализируемый запрос.

Важно заметить, что представленная на рисунке 2.14 схема базы данных является сильно упрощенной. Это связано с большим количеством таблиц широкой выборки в системах ИнфоБелГУ, которые используются в моделируемом процессе.

Так как все затрагиваемые в данном проекте используют реляционные базы данных, стало возможным обеспечение интерфейса бесшовного взаимодействия.

2.4 Выбор программных средств

Из проекта модуля, описанного выше, вытекают следующие требования к программным средствам:

- обладание инструментарием для работы с байтовым содержимым файлов;
- возможность созданного предложения выполнять роль активного сервера;
- наличие функционала работы с файлами MicrosoftOffice;
- наличие функционала работы с цифровой подписью PGP;
- совместимость с СУБД MySQL;
- поддержка протокола HTTP 1.1.

Исходя из данных требований было решено использовать в качестве основного языка разработки язык Java с использованием SpringFramework.

SpringFramework – это бесплатный модульный универсальный фреймворк, созданный как альтернатива EnterpriseJavaBeans от Oracle. На рисунке 2.15 изображены все основные модули фреймворка.

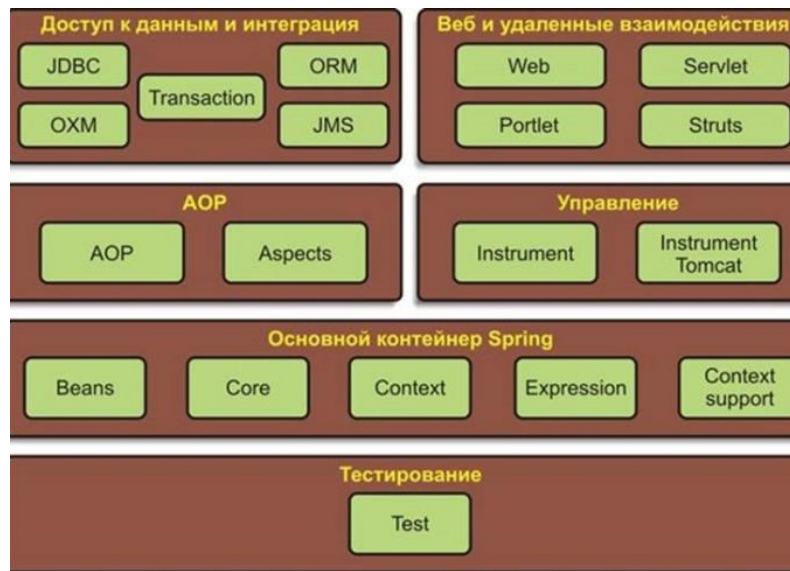


Рисунок 2.15 – Архитектура SpringFramework

Платформа Spring состоит из 6 основных модулей в составе SpringFramework, а так же модули расширения функционала:

- SpringBoot, облегчающий задачу конфигурации приложения;
- SpringCloud, предоставляющий удобный инструментарий для разработки облачных систем;
- SpringIntegration, обеспечивающий поддержку множества промышленных протоколов и шаблонов коммуникации;
- SpringBatch, облегчающий работу с загруженными потоками данных;
- SpringSecurity, позволяющий организовать системы авторизации и контроля доступа в приложении;
- SpringHATEOS, реализующий REST-архитектуру по принципам HATEOS;
- SpringWebFlow, предназначенный для создания веб-приложений;
- SpringWebService, упрощающий разработку SOAP-сервисов;
- SpringSession, предоставляющий инструментарий для работы с пользовательскими сессиями;
- SpringFlo, реализующий создания веб графики на стороне сервера при помощи JavaScript и HTML5;

- SpringStatemachine, упрощающий разработку конечных автоматов;
- SpringIOP, предназначенный для реализации продвинутой промышленной системы на основе взаимных зависимостей;
- SpringMVC, облегчающий реализацию шаблона «Модель-Наблюдатель-Контроллер».

Подобная универсальность набора модулей и их взаимная независимость делает данную платформу одним из самых привлекательных решений для решения любых производственных задач, а наличие детальной документации и большого заинтересованного сообщества вокруг Spring и проектов созданных на его основе облегчает задачу освоения фреймворка и технологий, предлагаемых им для реализации.

Выводы по второму разделу

В данном разделе было произведено проектирование и моделирование модуля. Это включает в себя анализ технологического обеспечения проекта, в котором были описаны и обоснованы решения выбора методов решения функциональных задач модуля; анализ информационного обеспечения проекта с целью организации знаний о моделируемом процессе; проектирование модуля, несущее в себе задачу описания всех архитектурных и функциональных решений, а также отображение их взаимосвязей в действительности.

3 Программная реализация и тестирование модуля

3.1 Программная реализация

3.1.1 Проектная реализация центра обработки данных

В качестве примера проектной реализации центра обработки данных будет приведен сервис обратной связи с активным сервером.

Для работы с базовыми понятиями разработанной схемы баз данных проекта в фреймворке Spring необходимо провести связывание таблиц с сущностями Java проекта. Пример такой реализации показан на рисунке 3.1.

```
@Data
@Entity
@EntityListeners(AuditingEntityListener.class)
@ToString
public class Notice {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE,
        generator = "notice_seq")
    @SequenceGenerator(sequenceName = "notice seq",
        allocationSize = 1, name = "notice_seq")
    private Long id;
    private String senderName;
    @Enumerated(EnumType.STRING)
    private NotifierTypeEnum notifierType;
    @Enumerated(EnumType.STRING)
    private NoticeStatusEnum status;
    private String subject;
    private String message;
    @CreatedDate
    @Temporal(TemporalType.TIMESTAMP)
    private Date createTime;
    @Temporal(TemporalType.TIMESTAMP)
    private Date sentTime;
```

Рисунок 3.1 – Связывание сущности с таблицей базы данных

Механизм скрепления сущности подразумевает полное описание таблицы базы данных с применением конфигурационных аннотаций объектного модуля фреймворка для установления ограничений в перечисляемых полях либо указания специфики их поведения [9].

Последующая работа с объектами проводится при помощи репозитория, образец реализации которого показан на рисунке 3.2.

```

@CrossOrigin
@RepositoryRestResource
public interface NoticeRepository extends JpaRepository<Notice, Long> {
    List<Notice> findByIdIn(@Param("-ids") List<Long> ids);
}

```

Рисунок 3.2 – Реализация репозитория

Встроенные реализации типовых моделей репозитория в Spring позволяют быстро реализовать доступ к основным функциям манипулирования физическими сущностями базы данных, а также обеспечивает интерфейс формирования, структурирования и расширения способов фильтрации выдачи запросов.

Для реализации доступа к механизмам манипулирования данными средствами интернет необходимо реализовать сервисы – промежуточный программный слой, реализующий основную бизнес логику приложения и интерфейс взаимодействия с репозиториями. Имплементация такого компонента показана на рисунке 3.3.

```

@Service
public class NoticeService {
    @Autowired
    private NoticeRepository noticeRepository;
    @Autowired
    private RecipientRepository recipientRepository;
    public List<Notice> getAll () { return noticeRepository.findAll(); }
    public Notice getById (Long id) { return noticeRepository.findOne(id); }
    public Notice add (Notice notice) {
        Iterable<Recipient> recipients = recipientRepository
            .save(notice.getRecipients());
        notice.setRecipients((List<Recipient>) recipients);
        return noticeRepository.save(notice);
    }
}

```

Рисунок 3.3 – Программная реализация сервиса

Сетевое взаимодействие между компонентами системы производится посредством обмена HTTP сообщениями с использованием архитектурного стиля передачи состояния представления взаимодействия компонентов распределенного приложения.

Основным приемом реализации сетевого программного интерфейса приложения в Spring является создание контроллеров, связывающих входящие сетевые запросы и соответствующие им классы реализации. Демонстрация имплементации такого подхода показан на рисунке 3.4.

```

@CrossOrigin
@RestController
public class NoticeRestController {
    @Autowired private NoticeService noticeService;
    @GetMapping("/notice")
    public List<Notice> getAll () { return noticeService.getAll(); }
    @GetMapping("/notice/one/{id}") public Notice getById
        (@PathVariable("id") Long id) { return noticeService.getById(id); }
    @PostMapping("/notice")
    public Notice add (@RequestBody Notice notice) return noticeService.add(notice); }
    @PutMapping("/notice")
    public Notice update (@RequestBody Notice notice) {
        return noticeService.update(notice);
    }
    @DeleteMapping("/{notice/fid}")
    public ResponseEntity delete (@PathVariable("id") Long id) {
        noticeService.delete(id);
        return ResponseEntity.ok().build();
    }
}

```

Рисунок 3.4 – Программная реализация контроллера

Результатом проведенной работы является подготовленный к работе обработчик, который можно связывать с коммутируемыми сервисами центра и подчиненными узлами.

3.1.2 Проектная реализация подчиненных узлов

Подчиненные узлы – это смежные системы, обладающие функционалом удаленного выполнения команд, полученных от центра обработки информации. Все подчиненные узлы были разработаны на языке программирования PHP для совместимости с MoodleCMS. Рассмотрение реализации будет производиться на примере автоматического создания итогового теста в системе «Пегас».

На рисунке 3.5 показан фрагмент аутентификации команды от центра обработки.

```

function verify($data)
{
    GLOBAL $CFG;
    putenv('GNUPGHOME=' . $CFG->dirroot . '.gnupg');
    error_reporting(E_ALL);
    $gpg = new gnupg();
    $PublicData = file_get_contents('public.key');
    $gpg->import($PublicData);
    return ($gpg->verify($signed_text, false, $plaintext) ?
        (json_decode($plaintext)) : false;
}

```

Рисунок 3.5 – Функция аутентификации

На данном этапе проводится проверка цифровой подписи полученного сообщения при помощи библиотекиgnupg.

В связи с необходимостью промежуточной обработки результатов выполнения функций ядра вызов последних происходит в специальных методах-контейнерах, отражающих собой определенные конечные команды.

При помощи упорядоченного вызова подобных функций и обеспечивается выполнение полученных команд. Пример этого можно наблюдать на рисунке 3.6.

```
function add_question($quizid, $sumgrades){
    GLOBAL $DB;
    list($thispageurl, $contexts, $cmid, $cm, $quiz, $pagevars) =
        question_edit_setup('editq', '/mod/quiz/addrandom.php', true,
            false, true, $quizid);
    $course = $DB->get_record('course', array('id' => $quiz->course));
    $context_id = explode(',', $pagevars['cat'])[1];
    if ($cat_ids = $DB->get_records_sql("SELECT id FROM moodle.mdl_question_categories
        where contextid = '$context_id.' and parent = 0")) {
        if (count($cat_ids) > 1) {
            $par_cat = $DB->insert_record('question_categories',
                array('name' => 'Все вопросы курса',
                    'contextid' => $context_id, 'info' => '',
                    'stamp' => make_unique_id_code(),
                    'parent' => 0, 'sortorder' => 999));
            foreach ($cat_ids as $cat_id)
                $DB->update_record('question_categories',
                    array('id' => $cat_id->id, 'parent' => $par_cat));
        } else $par_cat = array_shift($cat_ids)->id; }
    echo 'Cat ' . $par_cat . '<br/>';
    quiz_add_random_questions($quiz, 0, $par_cat, $sumgrades, 1, true);
    quiz_delete_previews($quiz); quiz_update_sumgrades($quiz); }
```

Рисунок 3.6 – Функция генерации теста

Для выполнения команд в автоматическом режиме библиотеки ядра Moodle были модифицированы для поддержки вызова операций без проверки авторизации с правами специального системного пользователя. Пример такой реализации можно наблюдать на рисунке 3.7.

```
if(!$is bot)
    require_capability( capability: 'moodle/question:useall', $catcontext);
```

Рисунок 3.7 – Модификация функции создания теста

Благодаря закрытости систем ИнфоБелГУ можно гарантировать, что параметр \$is_bot устанавливается только после обработки команды полученной от центра обработки информации.

3.2 Тестирование

С точки зрения составителя пакета документов процесс эксплуатации системы начинается с загрузки архива, содержащего все представляемые на обработку документы. Интерфейс предоставлен на картинке 3.8.

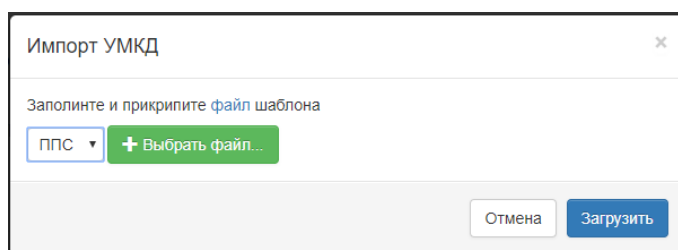


Рисунок 3.8 – Окно импорта

На этапе импорта система отправляет в центр обработки переданный архив и выводит пользователю-составителю информацию о процессе загрузки. При возникновении ошибок на данном этапе документы отвергаются, а на странице импорта выводится детальная информация о причине ошибки и ссылка на рекомендации к исправлению.

После успешного импорта документов система формирует запись на странице статистики пользователя (рисунок 3.9)

Вы зашли под именем Чебогарев Алексей Андреевич (Выход)

Belarusian National Research University
 Белорусский государственный
 национальный исследовательский
 университет

Кафедральные, институтские показатели

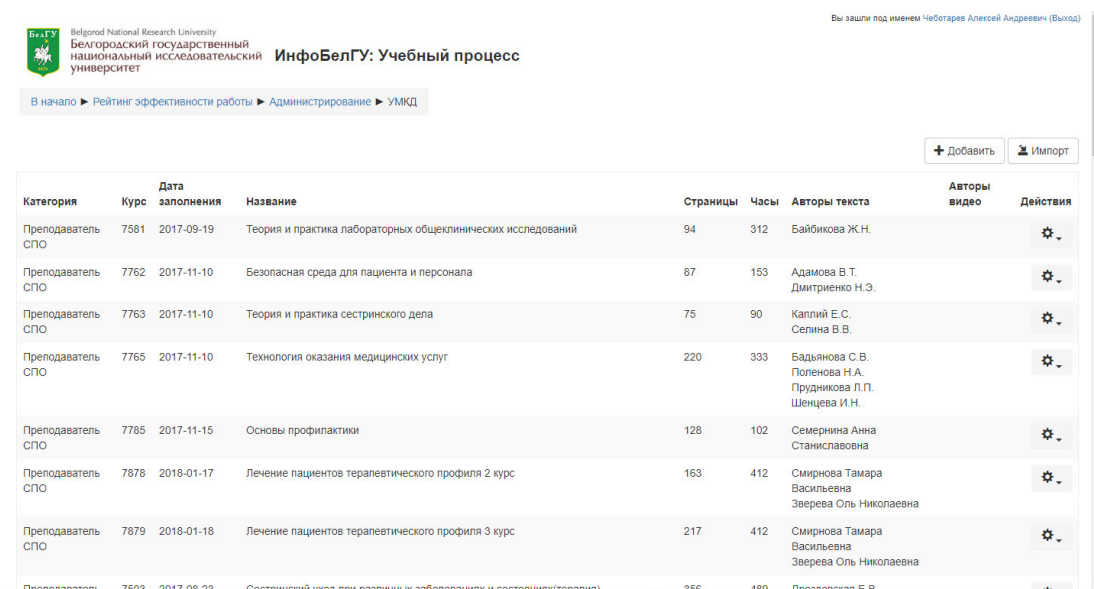
[В начало](#) > [Управление УМКД](#) > [Реестр УМКД](#)

Номер	Название	Подразделение	Специальность	Авторы	Дата приема	Дата проверки	Дата загрузки	Статус
1	Тестовый УМКД 1	110205. Факультет иностранных языков педагогического института	44.03.05 Педагогическое образование	Аматов Александр Михайлович	01.03.2018	05.04.2018	-	Ошибка в ФТЗ
2	Тестовый УМКД 2	11070001. Кафедра прикладной информатики и информационных технологий	44.03.05 Педагогическое образование	Гальцев Олег Владимирович Штирляков Андрей Иванович	10.02.2018	15.02.2018	25.02.2018	Успешно
3	Тестовый УМКД 3	110205. Факультет иностранных языков педагогического института	44.03.05 Педагогическое образование	Аматов Александр Михайлович	01.03.2018	-	-	Нет выписки с кафедры

Рисунок 3.9 –Страница статистики

На странице статистики пользователя выводится информация о статусе обрабатываемых документов, учетная информация и даты прохождения ключевых этапов обработки.

Информация для сотрудников отдела, отвечающего за документооборот представлена на отдельной странице, изображенной на рисунке 3.10.



Категория	Курс	Дата заполнения	Название	Страницы	Часы	Авторы текста	Авторы видео	Действия
Преподаватель СПО	7581	2017-09-19	Теория и практика лабораторных общеклинических исследований	94	312	Байбикова Ж.Н.		⚙️ -
Преподаватель СПО	7762	2017-11-10	Безопасная среда для пациента и персонала	87	153	Адамова В.Т. Дмитриенко Н.Э.		⚙️ -
Преподаватель СПО	7763	2017-11-10	Теория и практика сестринского дела	75	90	Каплий Е.С. Селина В.В.		⚙️ -
Преподаватель СПО	7765	2017-11-10	Технология оказания медицинских услуг	220	333	Бадьянова С.В. Поленова Н.А. Прудникова Л.П. Шенцева И.Н.		⚙️ -
Преподаватель СПО	7785	2017-11-15	Основы профилактики	128	102	Семернина Анна Станиславовна		⚙️ -
Преподаватель СПО	7878	2018-01-17	Лечение пациентов терапевтического профиля 2 курс	163	412	Смирнова Тамара Васильевна Зверева Ольга Николаевна		⚙️ -
Преподаватель СПО	7879	2018-01-18	Лечение пациентов терапевтического профиля 3 курс	217	412	Смирнова Тамара Васильевна Зверева Ольга Николаевна		⚙️ -
Преподаватель СПО	7503	2017-08-23	Сестринский уход при различных заболеваниях и состояниях (теория)	356	480	Прокофьев Е.Б.		⚙️ -

Рисунок 3.10 – Страница отчета

В отличие от страницы статистики, ответственным сотрудникам доступна детальная информация по всем заказанным, обрабатываемым и опубликованным пакетам документов, прошедших обработку, а также ссылку на страницу с полной информацией о пакете.

Выводы по третьему разделу

В данном разделе описаны особенности процесса программной реализации. В этом разделе были рассмотрены примеры реализации технологий, необходимых для решения поставленных задач; тестирование, где обозначены проблемы, возникшие при испытательном введении разработанного решения в эксплуатацию, описаны их причины и решения с частичной демонстрацией реализации.

ЗАКЛЮЧЕНИЕ

В ходе реализации проекта в рамках выпускной квалификационной работы была проведена работа по автоматизации документооборота организации, использующей гетерогенную систему. В результате внедрения проекта были достигнуты следующие цели:

- за счет автоматизации подавляющего числа этапов влияние человеческого фактора на результат процесса снизилось, что как следствие привело к падению количества ошибок при проверке на 76%;
- за счет превентивного направления на исправление документов, не прошедших верификацию, время проведения документов сократилось на 300%;
- за счет формализации и верификации процесса ведения реестра ошибки отчетности сократились на 100%;
- за счет создания инструментария для формирования отчетов удалось улучшить качество мониторинга и статистики работы подразделения и предприятия;
- за счет оптимизации использования человеческих ресурсов подразделение стало способным решать новые поставленные перед ним задачи.

Разработанный программный продукт отвечает всем современным требованиям к системам подобного класса. Внедрение такого решения позволило улучшить эффективность и скорость прохождения документов. Также внедрение данного решения является примером положительного эффекта внедрения подобных решений, что в перспективе может повлечь за собой распространение данной практики.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1 Аверченков, В. Автоматизация проектирования технологических процессов: учебное пособие /В. Аверченков, Ю. Казаков.–Litres, 2015.–229с.
- 2 Основы криптографии: учебное пособие //А.П. Алферов и др.—М.: Гелиос АРВ, 2005.–480 с.
- 3 Барсегян, А. Методы и модели анализа данных: OLAP и DataMining / А. Барсегян.– БХВ–Петербург, 2004.– 340с.
- 4 Барсегян, А. Технологии анализа данных: DataMining, TextMining, VisualMining, OLAP / А. Барсегян 2 изд.– БХВ–Петербург, 2008.– 384с.
- 5 Брейман, А.Д. Сети ЭВМ и телекоммуникации: учебное пособие / А.Д. Брейман // Часть 1. Общие принципы построения сетей. Локальные сети.– АД–М.: МГАПИ, 2001.–75с.
- 6 Гниловская, Л. П. Автоматическая коррекция орфографических ошибок / Л.П. Гниловская, Н. Гниловская.– 2004.–203с.
- 7 Петров, А. Компьютерная безопасность. Криптографические методы защиты / А. Петров.–Litres, 2017.– 448с.
- 8 Garfinkel,S. PGP: pretty good privacy / S. Garfinkel.– O'Reilly Media, Inc, 1995.– 364с.
- 9 Spring Data: modern data access for enterprise Java / М. Pollack et al.– O'ReillyMedia, Inc, 2015.– 314с.
- 10 Развитие интернета в регионах России [Электронный ресурс] //Яндекс.– Москва: 2016.– Режим доступа: https://yandex.ru/company/researches/2016/ya_internet_regions_2016, свободный.
- 11 Бабин, Д.Н. О перспективах создания системы автоматического распознавания слитной устной русской речи / Д.Н. Бабин, И.Л. Мазуренко, А.Б. Холоденко //Интеллектуальные системы.– 2004.– Т.8.– №. 1–4.– С. 45–70.
- 12 Балакин, А.В. Использование стеганографических методов для защиты текстовой информации /А.С. Елисеев, А.Ю. Гуфан //Т–Сотт–Телекоммуникации и Транспорт.–2009.– Т. S–DSPА.– 72с.

- 13 Бойцов, Л. М. Классификация и экспериментальное исследование современных алгоритмов нечеткого словарного поиска /Л.М. Бойцов //Труды 6 Всеросс. науч. конф. – 2004. – Т. 6. – 250с.
- 14 Восстановление формата данных /А.И. Гетьман и др. //Труды Института системного программирования РАН.– 2010.– Т. 19. – 312с.
- 15 Данилов, А.Д. GTD, коллаборация и автоматизация документооборота на крупных предприятиях с распределенной структурой / А.Д. Данилов, А.Н. Боровцов//Вестник Воронежского государственного технического университета.– 2011.– Т. 7.– 103с.
- 16 Довгаль, В.А. Методы повышения безопасности в сфере «облачных» технологий /В.А. Довгаль//Вестник Адыгейского государственного университета. Серия 4: Естественно–математические и технические науки.– 2014.– 186с.
- 17 Егорова, М.А. Финансовый аспект теории эффективного управления /М.А. Егорова, Л.Г. Селютина //Общество. Среда. Развитие (TerraHumana). – 2009. – №. 3.– 169с.
- 18 Комашинский Д. В., Котенко И. В. Исследование структурных особенностей вредоносных документов методами DataMining //Информационные технологии и вычислительные системы. – 2012. – №. 2. – С. 76–92.
- 19 Метод динамического создания связей между информационными объектами базы знаний/Обухова О. Л. и др. //Труды Института вычислительных технологий СО РАН. – 2009. – С. 39–45.
- 20 Петрыкина, Н.И. Правовое регулирование оборота персональных данных/Петрыкина, Н.И.–М.: Статут, 2011. – 134 с.
- 21 Селевцов, Л.И. Автоматизация технологических процессов: учебник/Л.И. Селевцов.–2–е изд., М.: Академия, 2012.–352 с.
- 22 Сеницын, С.В. Верификация программного обеспечения/ С.В. Сеницын, Н.Ю. Налютин.– М.: БИНОМ. – 2008. – 156 с.

23 Ходашинский И. А. Идентификация нечетких систем: методы и алгоритмы //Проблемы управления. – 2009. – №. 4.

24 О построении статистических языковых моделей для систем распознавания русской речи /Холоденко А.Б. и др. //Интеллектуальные системы. – 2002. – Т. 6. – С. 381–394.

25 Хореев П. Б. Методы и средства защиты информации в компьютерных системах: учеб. пособие для студ. высш. учеб. заведений/ПБ Хорев.—3-е изд., стер.—М.: Издательский центр «Академия», 2007.—256 с.

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

« » _____ Г.
