**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН**

Некоммерческое акционерное общество

«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ имени Гумарбека Даукеева»

Институт информационных технологии

Кафедра «IT - Инжиниринг»

**Отчёт №1**

**по научно-исследовательской работе магистранта**

На тему: Разработка интеллектуальной системы выявления сущностей в национальных стандартах для организации терминологического единства

Образовательная программа: 7M06103-Вычислительная техника и программное обеспечение

Выполнил: магистрант Сакан Ерлан М.

Группа МВТн-21-2

Приняла: доктор PhD, доцент Утегенова А.У.

Научный руководитель:

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_202\_\_\_г.

(оценка) (подпись)

Алматы 2021

Содержание

[Введение 3](#_Toc91607359)

[1 Теоретическое обоснование разработки информационной системы 4](#_Toc91607360)

[1.1 Анализ предметной области 4](#_Toc91607361)

[1.2 Существующие систем и их проблемы 4](#_Toc91607362)

[1.3 Решение проблем 11](#_Toc91607363)

[2 Описание и обоснование выбора языка программирования и СУБД 18](#_Toc91607364)

[2.1 Проектирование интелектуальной системы 18](#_Toc91607365)

[2.2 Анализ современных языков программирования 22](#_Toc91607366)

[2.3 Выбор языка программирования C# 22](#_Toc91607367)

[2.4 Особенность использования СУБД 24](#_Toc91607368)

[Список литературы 25](#_Toc91607369)

# Введение

Интелектуальные системы в данный момент активно связаны со сферой деятельности человека. По данной причине есть необходимость внедрения интелектуальные систем в сферы, где они мало развиты или слабо используются. Такое внедрение должно позволить уменьшить затраты, увеличить производство труда, а также время на обработку данных.

Интелектуальная система в дипломной работе выполнена по заказу кафедры “IT-Инжиниринг” Алматинского Университета Энергетики и Связи имени Гумарбека Даукеева.

Целью научно-исследовательской работы магистранта является разработка системы распознования сущностей в национальных стандартах. Тема выявление сущностей в национальных стандартах стала актуальна. Система позволяет позволяет полностью автоматизировать процесс.

Программа должна обеспечивать:

* выявление сущностей;
* деление сущностей по категориям;
* вывод исходного результата.

Для достижения вышеуказанной цели необходимо решить следующие задачи:

* провести анализ существующих систем;
* разработать программное обеспечение;
* провести экономическую оценку эффективности информационной системы.

В настоящее время существуют похожие интелектуальные системы, которые помогают автоматизировать процесс хранения методических трудов, а также вывода.

Большинство из них имеют довольно широкий функционал, но в них есть недостатки.

Основные недостатки:

* функционал программы не удовлетворяет всем требованиям работы;
* интелектуальные системы требуют доработку и на это могут уйти дополнительные затраты.

После изучения прочих интелектуальных систем, а также выявлений основных проблем, было принято решение о создании интелектуальной системы.

**1 Теоретическое обоснование разработки информационной системы**

* 1. **Анализ предметной области**

Кафедра IT-инжиниринга входит в состав Института систем управления и информационных технологий НАО “Алматинский Университет Энергетики и Связи имени Гумарбека Даукеева”, является общеобразовательным, учебно-научным подразделением, созданным в целях осуществления учебной, методической, научно-иссле­довательской и воспитательной работы среди обучающихся.

Кафедра ITE осуществляет свою деятельность в соответствии с действующим законодательством Республики Казахстан, приказами и распоряжениями Министерства образования и науки РК, Уставом Университета, решениями Ученого совета Университета, приказами и распоряжениями руководства, другими нормативными документами внешнего и внутреннего характера, определяющими порядок деятельности Университета, настоящим Положением и должностными инструкциями.

Основными задачами Кафедры ITE являются:

1. учебно-методическое обеспечение учебных дисциплин кафедры, подготовка учебников, учебных пособий и методических указаний по изучению дисциплин, материалов текущего, промежуточного и итогового контроля знаний по всем формам обучения;
2. рекомендация для публикаций утвержденных научных работ;
3. изучение, обобщение и распространение опыта работы лучших преподавателей;
4. проведение работ по развитию, модернизации и укреплению материально-технической базы кафедры включая постоянное совершенствование учебно-лабораторного оборудования.

Функциями Кафедры ITE являются:

1) рассмотрение и утверждение индивидуальных планов учебной, научной, учебно-методической и других работ ППС кафедры;

1. методическое обеспечение и организация самостоятельной работы студентов по дисциплинам Кафедры ITE.
   1. **Существующие систем и их проблемы**

Интелектуальные системы является системой, которая создана для хранения, поиска и обработки информации. Интелектуальные системы созданы для обеспечения информацией определенный круг людей. В основном данная система предназначена для оптимизации рабочего процесса и его автоматизации. На сегодняшний день есть достаточно много систем, которые позволяют хранить и обрабатывать информацию. Правда есть некоторые сложности и проблемы в использовании этих систем. Мы имеем такие системы, как ABBYY FineReader, Adobe Acrobat и Freemore OCR.

Главной проблемой данных сервисов является:

* безопасность хранения;
* универсальность;
* объем.

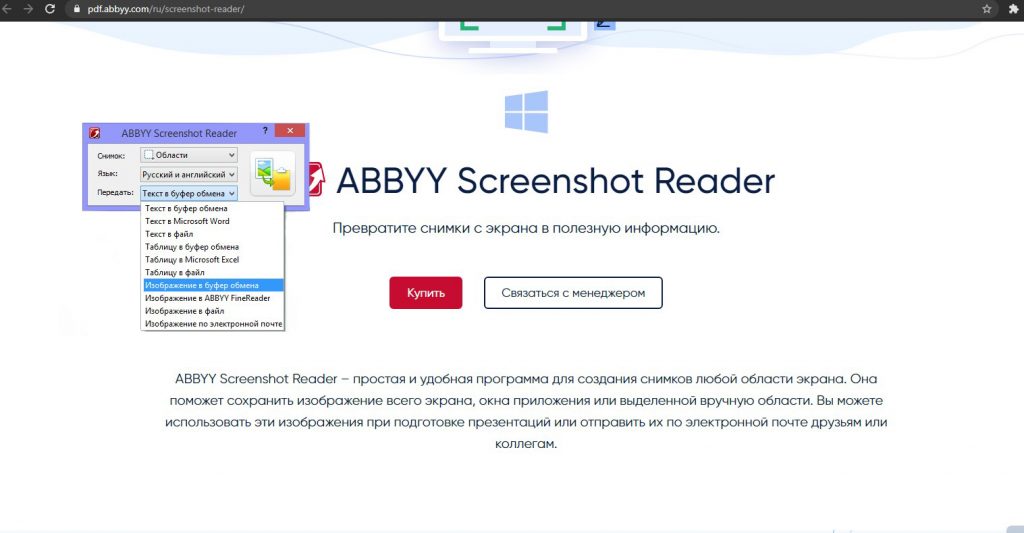
Для крупных организаций безопасность является неотъемлемо важным элементом и непозволительно допускать утечку информации. Сегодня информация имеет важный характер. Любые действия совершаются за счет, имеющийся информации. Утечка информации может привести к краху организации, так как другие организации могут собрать всю информацию и использовать ее против этой же организации. Загрузка данных в сервисы может привести к утечке, так как данные находятся во всемирной паутине. Человек, поставивший цель получения этих данных, может найти пути их получения. В данный момент поток информации, которые проходят, шифруются разными методами. Главная задача организации является надежность потока информации[1].

Для физических лиц ABBYY FineReader и Adobe Acrobat являются наиболее предпочитаемыми и удобными. Системы универсальные и позволяют работать в разных сферах деятельности человека. В основном главная задача физического лица является хранение личных файлов в данных сервисах, куда входят: личные фото, документы, но вот данное преимущество в универсальности идет в минус для государственных учреждений, организаций, частных компаний и для остальных структур. Функционал данных систем может не удовлетворять всем требованиям работы. Каждая организация имеет свою уникальную сферу деятельности, и хранения данных может быть не единственной нужной функцией для обеспечения работоспособности сферы. Возьмем ABBYY FineReader.

Довольно хорошая программа, которая сканирует и распознает текст с изображений в формате jpg, jpeg, png, gif, bmp, а также pdf документов. Полученный материал она конвертирует в файлы форматов doc, rtf, xls, html и pdf с возможностью редактирования. Не важно, каким способом получен исходник – сканированием через МФУ или сканер, съемкой через цифровой фотоаппарат или смартфон.

ABBYY FineReader работает со 179 различными языками мира, благодаря наличию технологии ABBYY OCR она распознает слова с максимальной степенью точности. В ней есть возможность сохранения оформления и стиля, также обработки и отправки полученного материала по почте или посредством публикации в интернете. Интерфейс русскоязычный, понять, как им пользоваться сможет любой, тут запутаться крайне сложно.

У данной программы есть свои определенные плюсы, которые дает ей преимущества перед выбором над остальными и является довольно распространненным продуктом, но также у программы есть свои минусы. Далее стоит рассмотреть эти плюсы и минусы.



Плюсы

* Возможность работы как с отдельными файлами и фото, так и целыми пакетами.
* Сохранение готового распознанного материала в новом редактируемом файле.
* Высокая точность распознавания текста.
* Наличие огромного количества полезных дополнений.
* Обработка фотографий, снятых на мобильный телефон.
* Регулярные обновления приложения.
* Сохранение высокого качества изображений в документе.
* Доступ к сервису ABBYY FineReader Online для зарегистрированных пользователей.

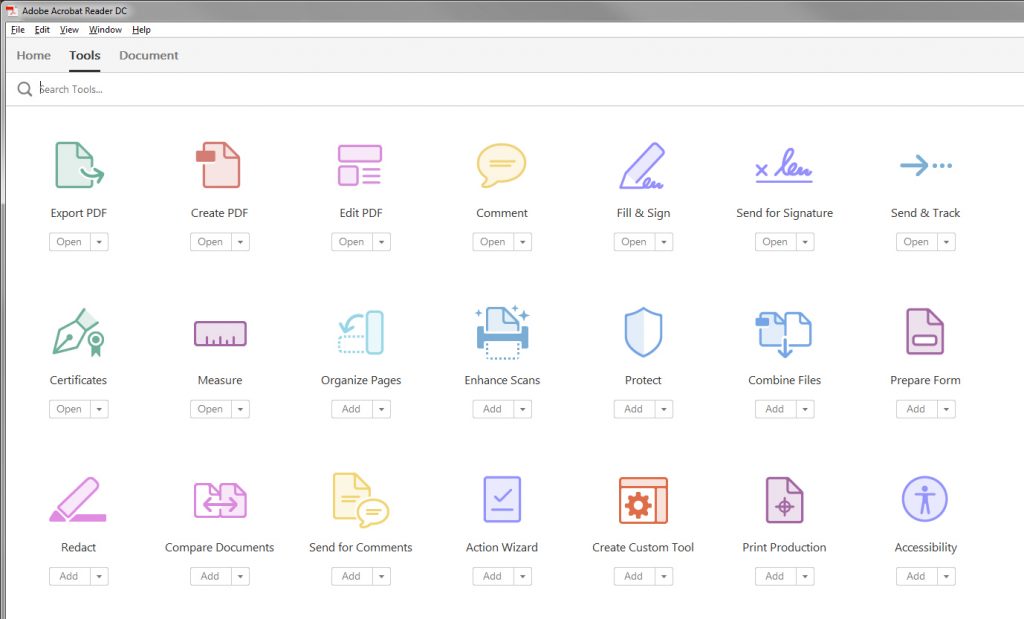
Минусы

* Платная версия. Есть пробный период, но он ограничен 15 днями, плюс в нем недостаточно обширный функционал.
* Обязательная регистрация и подписка.
* Недоступность редактирования прямо в программе – легче конвертировать материал в текст, а там уже проводить правки.
* Исходная структура документа практически не сохраняется – колонтитулы после завершения сканирования могут сильно съехать.

Adobe Acrobat – ничем не уступает ABBYY FineReader, правда используемая не совсем для распознавания текста с фото или файлов. С ее помощью можно создавать и редактировать различные документы pdf, конвертировать их в другие форматы, редактировать отдельные элементы и так далее.

Есть две ее версии – обычная и Pro, с расширенным функционалом. Естественно, вторая распространяется не бесплатно. Standard без акции обойдется по 1777 рублей в месяц, а расширенная Pro будет стоить 1932 рубля, и это только для физических лиц. Предусмотрен 7-дневный пробный период.

Стоит учитывать, что данная программа способна работать только с файлами формата pdf. Отредактированный документ можно перевести в формат Блокнота, Word, Excel, PowerPoint, картинки jpeg и так далее.



Плюсы

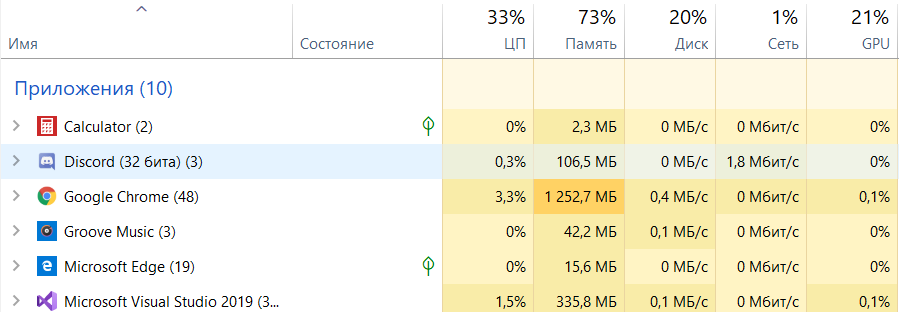
* Создание и редактирование файлов формата pdf.
* Добавление маркеров, закладок или комментариев.
* Конвертирование pdf-файла в другие форматы.
* Удобный и развитый текстовый редактор.
* Есть возможность восстановления поврежденных участков документа.
* Регулярные обновления.

Минусы

* Дорогая плата за пользование полноценной версией.
* Высокие системные требования.
* Крайне малый пробный период.

В этом и кроется проблема. Система слишком универсальна и не адаптирован для определенной среды, что не очень эффективно. Возьмем, к примеру, компьютеры средней сборки. Такими ноутбуками могут пользоваться абсолютно все желающие, так как адаптированы под повседневную жизнь пользователя.

Обычному пользователю должно хватит использование интернета, офисных приложений и предустановленного функционала: калькулятор, медиаплеер и просмотр картинок (рисунок 1.6).



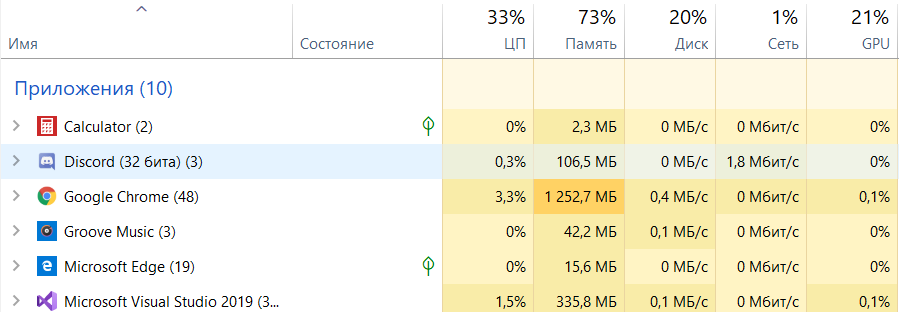


Рисунок 1.6 – Диспетчер задач (демонстрация работы процессов)

Теперь стоит упомянуть отдельную категорию людей: разработчики, монтажеры. Сейчас компьютеры играют важную роль в сфере деятельности человека. Для каждого из них нужны свои собственные комплектующие для эффективности их действий. Разработчики и монтажеры требуют более мощное железо для эффективной работы. В свою очередь монтажеры требуют более мощный графический процессор, так как приходится часто работать с аудио и видео файлами, что требует много вычислительной мощности. Особенно сейчас, когда разрешение экранов достигает 3840x2160 точек, которое в четыре раза превышает количество пикселей в формате Full HD, если еще вспомнить, что в ЭЛТ-телевизорах разрешение было 640x480, а может и меньше. Монтажеры не смогут эффективно работать с компьютерами со встроенным графическим процессором (рисунок 1.7).

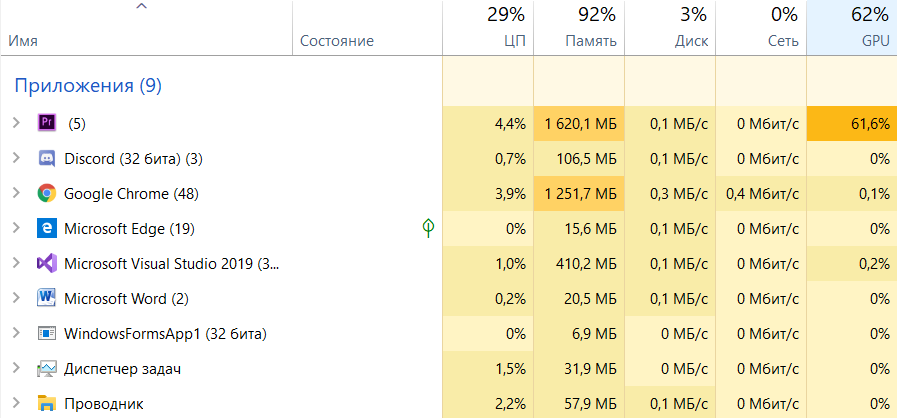
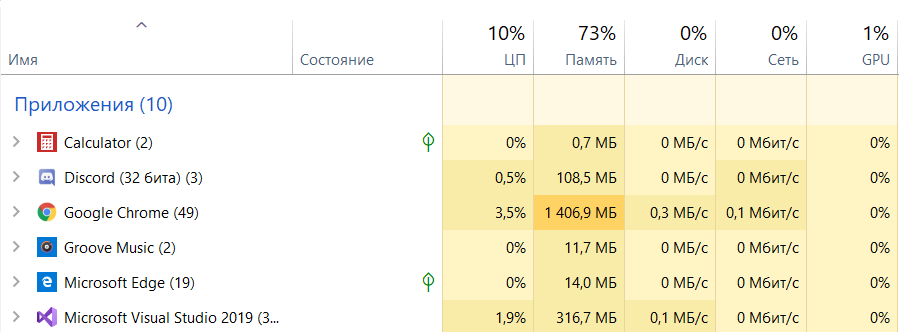


Рисунок 1.7 – Загруженность системы процессом Adobe Premiere Pro

У разработчиков проблема похожа. Для эффективной работы больше нужен центральный процессор, чем графический (рисунок 1.8).



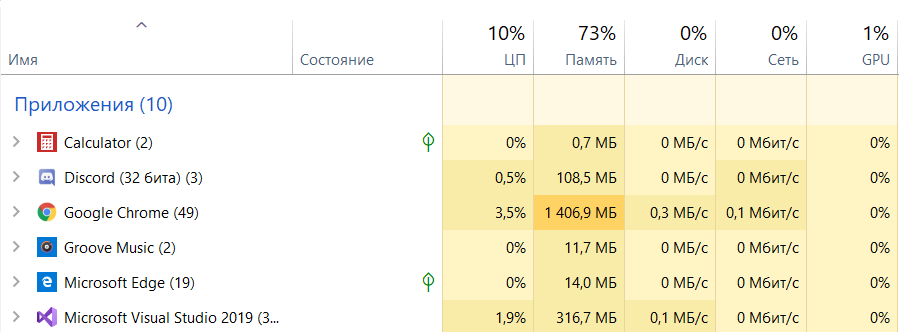


Рисунок 1.8 – Загруженность системы процессом Visual Studio Code

Расход ресурсов приложения в среде разработки Visual Studio (рисунок 1.9).

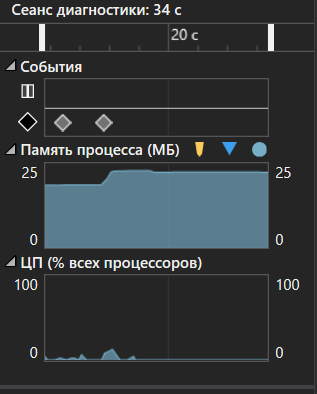


Рисунок 1.9 – Расход ресурсов в Visual Studio

Здесь приведены примеры для сравнения комплектующих разных сфер в целях понятия эффективности (рисунок 1.10).

|  |  |
| --- | --- |
| https://upload.wikimedia.org/wikipedia/commons/thumb/f/f2/Adobe_Premiere_Pro_Logo.svg/1200px-Adobe_Premiere_Pro_Logo.svg.png | https://pngimage.net/wp-content/uploads/2018/06/gpu-png-5.png |
| https://redtech.lk/wp-content/uploads/2019/05/i7.png |
| https://upload.wikimedia.org/wikipedia/commons/thumb/c/cd/Visual_Studio_2017_Logo.svg/1200px-Visual_Studio_2017_Logo.svg.png | https://redtech.lk/wp-content/uploads/2019/05/i7.png |

Рисунок 1.10 – Требование к определенным приложениям

Разработчикам эффективно не тратить средства на графический процессор и могут вложить больше средств больше средств на центральный процессор. Здесь ситуация имеет некоторую схожесть. В этой ситуации дается абсолютно пустой бланк, который нужно вручную заполнить, так как он не адаптирован под сферу деятельности предприятия. Необходимо тратить много времени на исполнение работы, которую можно было потратить более эффективно.

* 1. **Решение проблем**

Будут получены результаты исследования моделей формирования единого репозитория метаданных, проведен анализ, исследование и проработка научно-методических разработок Национального органа по стандартизации. Впервые будет осуществлено комплексное исследование модели базы специалистов в области разработки стандартов для выявления проблемных участков. Осуществлено комплексное исследование методов создания цифрового следа при разработке национального стандарта, разработаны концептуальные подходы к анализу полученных данных, на основе которых возможно формирование диагностики причин и сбоев в работе групп разработчиков и контролируемых органов, проведен анализ операционного цикла разработки стандарта.

**Задачи для решения проблемы:**

1. **Проведен анализ, исследование и проработка научно-методических разработок и операционного цикла Национального органа по стандартизации.**

- Разработана модель требований для управления государственными данными в органах и организациях государственного сектора;

- Выявлены и проработаны требования по организации управления данными, требования по описанию данных и оценки эффективности управления метаданными.

- В рамках представленной подзадачи исследована модель архитектур данных на примере Группы национальных стандартов «Информационные технологии».

- Одним их ценных исследований в области управления данными в представленном отчете занимает управление качеством данных. Определены возможности выявлять качественные данные из общего объема данных, это одна из главных задач выдвигаемая в качестве результата работы участниками проекта.

- При управлении государственными данными в органах и организациях государственного сектора необходимо учитывать методы хранения и возможные операции над данными, согласно этим вопросам, приводится попытка создания базовой архитектуры системы хранения данных, с представлением вариантов развития архитектуры с учетом критериев совместного использования данными и возможностями восстановления метаданных.

- Исследован и изучен операционный цикл разработки национального стандарта на примере Национального органа по стандартизации - РГП "Казахстанский институт стандартизации и сертификации"

- Изучены и приняты во внимание существующие разработки в представленной организации, список программных решений представлен в открытом доступе на официальном ресурсе организации https://ksm.kz/. Выполнены переговоры по заключению мемарандума о сотрудничестве, по задачам консультационного характера в рамках проекта.

- Изучены перспективные направления организации, план стандартизации, ознакомлены с действующими ТК на территории РК, выявлены новые направления для изучения в рамках Научно-исследовательских работ, выполненных Казахстанским институтом стандартизации и метрологии в сфере технического регулирования.

1. **Проведено комплексное исследование модели оценки специалистов в области разработки стандартов и моделей формирования единого репозитория метаданных.**

- Разработана и апробирована модель оценки специалистов в области национальных стандартов.

- Проведен анализ, исследование и проработка научно-методических разработок Национального органа по стандартизации.

- Получены рекомендации по организации структуры представления ПРНС (Программа разработки национальных стандартов), для разработки алгоритма модели оценки специалистов, задействованных во всех этапах разработки национальных стандартов.

- Разработана целевая архитектура системы хранения метаданных, рассмотрены модели Инфраструктуры как услуга (Iaas) и платформа как услуга (PaaS).

- Проведена оценка текущих возможностей систем хранения, загруженности, ожидаемым требованиям, использованным ресурсам.

- Проведены переговоры с конечными потребителями результатов выполнения данного проекта, с участием Национального органа по стандартизации, РГП "Казахстанский институт стандартизации и сертификации", Комитета технического регулирования и метрологии Министерства торговли и интеграции Республики Казахстан.

- Проведены переговоры с практикующими экспертами и разработчиками национальных стандартов, получены рекомендации для детализации алгоритма по формированию метрик оценки специалистов в области разработки стандартов.

По результатам выполнения представленной подзадачи были проработаны и исследованы основные нормативные документы и национальные стандарты:

- Закона РК «О стандартизации» от 5 октября 2018 года № 183-VІ ЗРК (<http://adilet.zan.kz/rus/docs/Z1800000183/info>)

- Казахстанский институт стандартизации и сертификации (<https://kazinst.kz/institute/index.php>)

- Научно технический журнал Метрология, ISSN 2522-1744 <https://ksm.kz/scientific-activities/journal/>

- Приказы о применении национальных стандартов государств-членов ЕАЭС взаимосвязанных с ТР ТС/ЕАЭС <https://ksm.kz/activities/sp/orders/>

- СТ РК 1.2 Национальная система стандартизации Республики Казахстан. Порядок разработки документов по стандартизации

- Об утверждении Правил разработки, согласования, экспертизы, утверждения, регистрации, учета, изменения, пересмотра, отмены и введения в действие национальных стандартов (за исключением военных национальных стандартов), национальных классификаторов технико-экономической информации и рекомендаций по стандартизации

1. **Осуществлено комплексное исследование методов создания цифрового следа проекта национального стандарта, без изменений существующих требований к гармонизируемым стандартам.**

В настоящий момент вся процедура согласования стандартов проходит по электронной почте, нередко только из-за этого происходят задержки по принятию решений. Разработчики высылают экспертам проект стандарта на трех языках, казахский, русский и английский, а также сопроводительное письмо. Колоссальная нагрузка на экспертов вызвана несоблюдением требований и неосведомленностью руководства об общей загруженности того или иного эксперта. Неравномерное распределение трудовых нагрузок ведет к нерациональному функционированию всей системы. Говоря об автоматизации процессов необходимо начинать с нижних слоев всех этапов, и только после этого получать реальные данные и принимать действенные, а в некоторых случаях кардинальные решения по изменению методов работы существующей системы. Необходимо учитывать персональную ответственность при формировании позиции экспертов и создании цифрового следа документа для исключения коррупционных сделок заинтересованных сторон. В связи с вышеизложенным, выполнены следующие работы:

- Осуществлено комплексное исследование методов создания цифрового следа при разработке национального стандарта, разработаны концептуальные подходы к анализу полученных данных, на основе которых возможно формирование диагностики причин и сбоев в работе групп разработчиков и контролируемых органов

- Разработана модель пользователей для обеспечения прозрачности доказательной базы по квалификации разработчиков, организаций участников, ТК, заказчиков, для формирования цифрового следа и дальнейшей обработки полученных данных, в той части, которая не рассматривается как государственная тайна.

1. **Разработаны концептуальные подходы к анализу полученных данных, на основе которых возможно формирование диагностики причин и сбоев в работе групп разработчиков и контролируемых органов.**

- Сформирована модель диагностики причин сбоев в работе групп разработчиков и контролируемых органов

- Проведено комплексное исследование модели базы специалистов в области разработки стандартов для выявления проблемных участков.

1. **Проведено исследование и получены промежуточные результаты по текущим задачам анализа применимости методологии онтологического инжиниринга.**

- Рассмотрена основная проблема для разработчиков стандарта на государственном (казахском) языке в части небольшого отклонения от семантическо-терминологического смысла и логики, смежных или взаимосвязанных стандартов. Приводится решение задач проблем обработки неструктурированной информации.

- При разработке алгоритма учитывался правильный и однозначный перевод ключевых понятий на русском языке изложения. При этом четко определены понятия присущие различным отраслям разрабатываемых нормативных документов.

При исследованиях и проработке представленной подзадачи были рассмотрены труды: Ермакова А.Е., Маликова А.В. и Целиковского А.С. по описанию методов автоматического извлечения сущностей. Обзор предшествующих научных исследований, проведенных в мире и Республике Казахстан, показал активную заинтересованность исследователей всего мира. В частности, на территории Казахстан помимо представленной рабочей группы интерес к изучению онтологического инжиниринга продемонстрировали ученые университета ВКГТУ им. Серикбаева. Работы трудов Рахметуллина С.Ж. посвящены мониторингу на основе онтологических моделей ситуаций, непосредственной разработкой самих баз знаний занимались такие ученные как Балова Т.Г., Рохас Криулько Н.П.

Теоретические основы области онтологии описывают основные типы онтологий, основные компоненты моделирования онтологий, основанные на фреймах или логике описания, критерии проектирования для построения онтологий, а также отношения с другими методами моделирования, которые используются в разработке программного обеспечения и базы данных.

Согласно общепринятой категоризации онтологий первоначально выделили следующие четыре вида: 1. Онтологии содержания для повторного использования знаний. Эти онтологии включают другие подкатегории: онтологии задач, онтологии предметной области и общие онтологии. 2. Онтологии коммуникации («скажи и спроси») для обмена знаниями. 3. Индексирование онтологий для поиска случаев. 4. Мета-онтологии эквивалентны тому, что другие авторы называют онтологией представления знаний.

Обобщая историю категоризации необходимо выделить, что полнота использования того или иного вида существенно зависит от первоначальных требований использования онтологий и области применения. В представленном исследовании авторы отчета пришли к выводу разработки формального языка представлений знаний для полноценного охвата задач, направленных на создание репозитория знаний по ключевым сущностям, данные задачи планируется начать второй год выполнения проекта.

Особое внимание уделяется методам обучения онтологии, которые сокращают усилия в процессе приобретения знаний; объединение онтологий, которое генерирует уникальную онтологию из нескольких онтологий; согласование онтологий, которое устанавливает различные типы отображения между онтологиями и оценка содержания онтологии.

В планах реализации программного решения планируется применение онтологического инжиниринга в области автоматизации процессов для гармонизации стандартов согласно Национальному плану стандартизации на основе результатов сравнительного анализа моделей и методов по применению метаданных, цифрового следа, методов по автоматизации разметки текстов, методов интеллектуального управления в процессах операционного цикла разработки стандарта.

Разработана проектная документация для механизмов мониторинга и контроля процессов при разработке национальных и международных стандартов, а также операционного цикла разработки стандарта на основе модели с использованием системного подхода по методологии онтологического инжиниринга. Проведено исследование и получены промежуточные результаты по текущим задачам анализа применимости методологии онтологического инжиниринга. Получены согласованные с экспертами в области стандартизации в лице курирующих органов, проектные решения по реализации программного обеспечения.

1. **Описание и обоснование выбора языка программирования и СУБД**
2. **Проектирование интелектуальной системы**

На рисунок 2.1 представлена диаграмма возможностей для пользователя.

Интелектуальные системы дают возможность использовать следующие функции пользователю[8]:

- импорт национальных стандартов в базу данных (путем заполнения нужных ячеек в самой информационной системе);

- экспорт из БД (возможность экспорта загруженных стандартов из базы данных);

- редактирование (редактирование в самой системе).

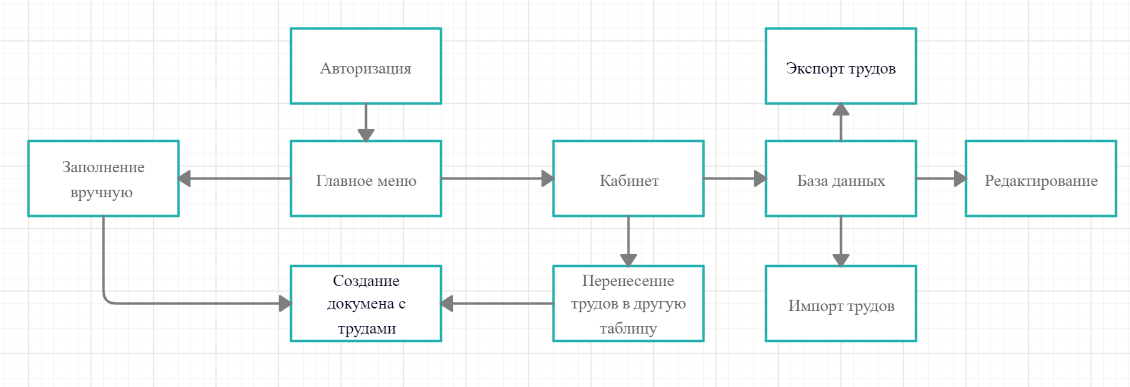


Рисунок 2.1 – Диаграмма возможностей для пользователя

На рисунке 2.2, 2.3, 2.4 представлена физическая модель, схема и диаграмма взаимодействия с базой данных. Рисунок содержит детали физической модели базы данных.

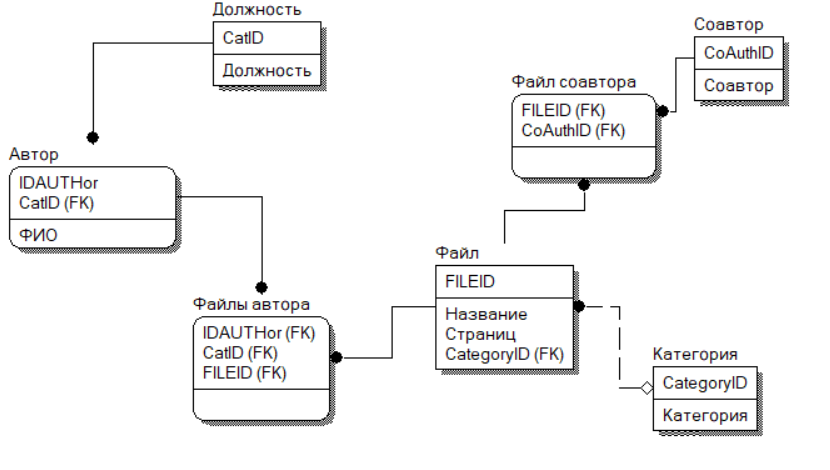


Рисунок 2.2 – Физическая модель базы данных

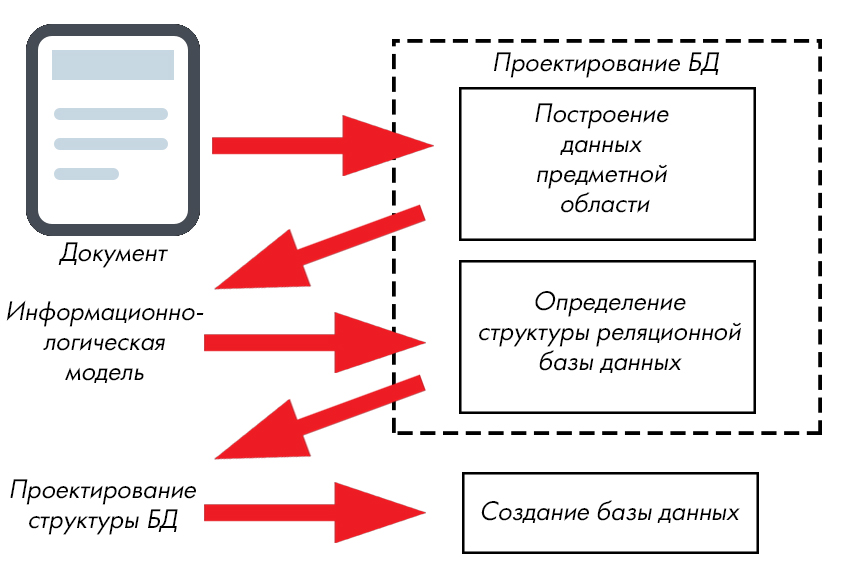


Рисунок 2.3 – Схема базы данных

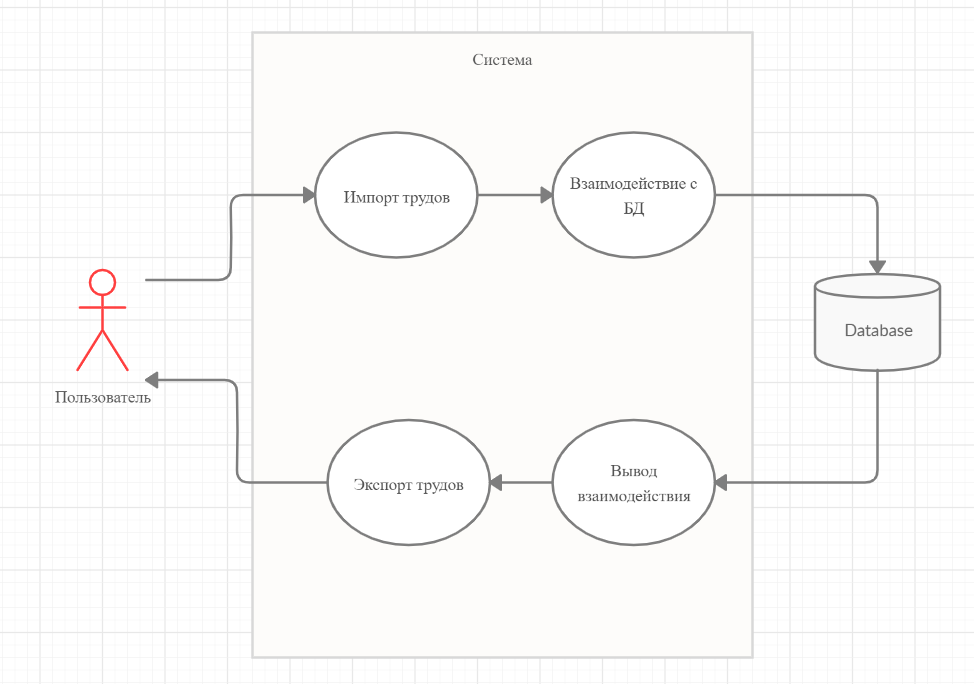
****

Рисунок 2.4 – Диаграмма взаимодействия с базой данных

На рисунке 2.5 представлена диаграмма последовательности работы информационной системы.

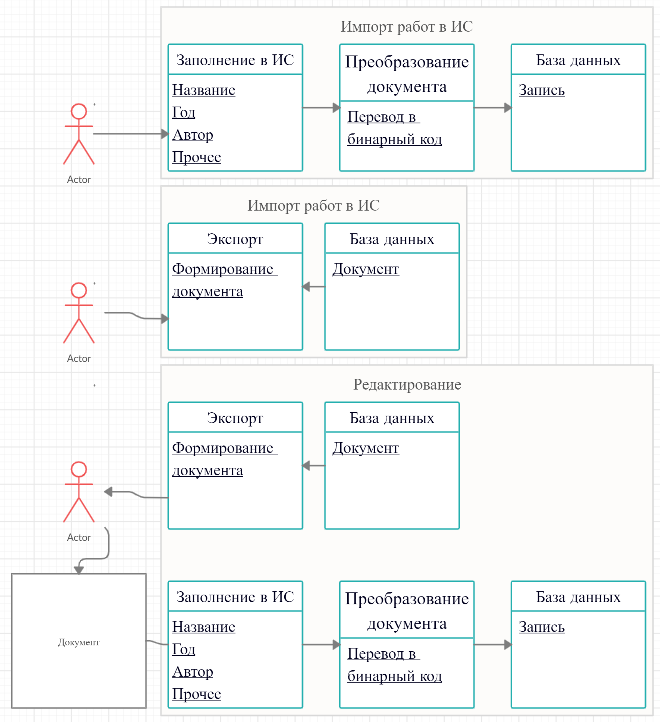
Диаграмма последовательности 

Рисунок 2.5 – Диаграмма последовательности

Ниже приведена блок-схема работы данного элемента сайта:

Вход в программу

Добавление работ

Ввод данных работ

Проверка введённых данных

Импорт работ

КОНЕЦ

Открытие работ

1. **Анализ современных языков программирования**

Среди разработчиков существует множество сред языков программирования[10].

Среда разработки программного обеспечения используется программистами для разработки программного обеспечения. Обычно в среде разработки входит компоненты, компилятор и отладчик. В такой среде можно производить весь цикл разработки[11]. Также существуют среды для разработки на нескольких языков – Microsoft Visual Studio и прочие.

Современные языки программирования:

- Delphi;

- C/C++;

- C#;

- Visual Basic;

- Java;

- JavaScript;

- PHP.

Рассмотрим плюсы и минусы разных языков программирования.

C++, C плюсы:

- возможность полного контроля;

- полный контроль над памятью;

- вы лучше понимаете все ошибки во время компиляции

C++, C минусы:

- требуется заранее оптимизировать;

- бедная библиотека.

JAVA, C# плюсы:

- можно не управлять памятью;

- богатая библиотека;

- компилируется в байт-код.

Одним из минусов языков JAVA и C# является наибольшая сложность выявления ошибки среди сравниваемых языков.

1. **Выбор языка программирования C#**

Язык программирования C# был выбран по причине легкости и доступности. C# является современным объектно-ориентированным и безопасным язык программирования. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript.

C# является объектно-ориентированным языком, который поддерживается также и компонентно-ориентированным программированием[2]. Разработка приложений сильнее тяготеет к производству программных компонентов в виде автономных пакетов, исполняющих отдельные функциональные возможности[12]. Важная особенность этих компонентов в том, что выражают из себя модель программирования со свойствами, методами и событиями. У них есть атрибуты, предоставляющие декларативные сведения о компоненте[13]. Эти компоненты включают в себя собственную документацию. C# дает возможность использовать языковые конструкции, которые поддерживают такую концепцию работы. Благодаря этому C# подходит для создания и применения программных компонентов.

Функции языка C#, обеспечивающих надежность и устойчивость приложений:

- сборка мусора автоматически освобождает память, занятую недостижимыми неиспользуемыми объектами;

- обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок, а также к восстановлению;

- безопасная структура языка делает невозможным чтение из неинициализированных переменных, индексацию массивов за пределами их границ или выполнение непроверенных приведений типов.

В C# есть единая система типов. Все типы C#, включая типы-примитивы, такие как int и double, наследуют от одного корневого типа object[14]. Это говорит о том, что все типы используют общий набор операций, и значения любого типа можно хранить, передавать и обрабатывать схожим образом. Таким образом, C# поддерживает пользовательские ссылочные типы и типы значений, позволяя как динамически выделять память для объектов, так и хранить упрощенные структуры в стеке.

При разработке C# много внимания было уделено управлению версиями, чтобы обеспечить совместимость программ и библиотек C# при дальнейшем развитии[15]. Большинство языков программирования стараются не обращать внимания на этот вопрос[3]. Из-за чего эти программы на этих языках ломаются чаще, чем хотелось бы, при выходе новых версий зависимых библиотек. Вопросы управления версиями существенно повлияли на такие аспекты разработки C#, как раздельные модификаторы virtual и override, правила разрешения перегрузки методов и поддержка явного объявления членов интерфейса.

Элементы языка C#[16]:

- структура C# основывается на программах, пространствах имен, типах, членах и сборках;

- можно получить сведения, такие как типы значения, ссылочные типы и переменные в языке C#;

- выражения производится из операндов и операторов (выражения также могут возвращать значения);

- можете использовать инструкции, чтобы описать действия, которые исполняются программой;

**- классы – это самый важный тип в языке C# (объекты представляют собой экземпляры классов, классы производятся путем описания их членов, которые описаны в этой статье)**;

**- массив является структурой данных, которая содержит несколько переменных, доступ к которым можно осуществить по вычисляемым индексам**;

**- интерфейс реализован классами и структурами (интерфейс может содержать методы, свойства, события и индексаторы)**;

**- тип delegate выражает из себя ссылки на методы и типом возвращаемого значения (делегаты позволяют использовать методы как сущности, сохраняя их в переменные и передавая в качестве параметров)**;

**- атрибуты дают возможность программам указывать дополнительные описательные данные о типах, членах и других сущностях**.

1. **Особенность использования СУБД**

Информационная система требует использования базы данных. Для работы требуется хранения методических трудов непосредственно в базе данных. Visual Studio имеет в себе локальную базу данных, которая очень удобна для использования[17]. Все синтаксисы запросов является общей среди всех языков базы данных и можно без проблем адаптироваться[4].

MySQL – свободная субд, где гибкость СУБД MySQL выражается поддержкой большого количества типов таблиц:

- пользователи имеют возможность выбрать как таблицы типа MyISAM;

- поддерживает текстовый поиск, так и таблицы InnoDB;

- поддерживает транзакции на уровне отдельных записей.

СУБД MySQL ставится с особенным типом таблиц EXAMPLE, показывающие принципы производства новых типов таблиц. В СУБД MySQL перманентно появляются новые типы таблиц.

MySQL имеет API и коннекторы для многих языков программирования[18]:

* Delphi;
* C/C++/C#;
* JAVA;
* PHP;
* Python.

При этом – это ещё не полный список языков.

# Список литературы

1 Перлова, О.Н. Проектирование и разработка информационных систем: Учебник / О.Н. Перлова, О.П. Ляпина, А.В. Гусева. - М.: Academia, 2017. - 416 c;

2 Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C# / Джеффри Рихтер. - М.: Питер, 2013. - 928 c;

3 Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C# / Джеффри Рихтер. - М.: Питер, 2013. - 928 c;

4 Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. - М.: Вильямс, 2015. - 266 c;

5 Бишоп, Дж. C# в кратком изложении / Дж. Бишоп, Н. Хорспул. - М.: Бином. Лаборатория знаний, 2013. - 472 c;

6 Вагнер, Билл С# Эффективное программирование / Билл Вагнер. - М.: ЛОРИ, 2013. - 320 c;

7 Зиборов, В.В. Visual C# 2012 на примерах / В.В. Зиборов. - М.: БХВ-Петербург, 2013. - 480 c;

8 Зиборов, Виктор Visual C# 2010 на примерах / Виктор Зиборов. - М.: "БХВ-Петербург", 2011. - 432 c;

9 Ишкова, Э. А. Самоучитель С#. Начала программирования / Э.А. Ишкова. - М.: Наука и техника, 2013. - 496 c;

10 Касаткин, А. И. Профессиональное программирование на языке си. Управление ресурсами/А.И. Касаткин. - М.: Высшая школа, 2012. - 432 c;

11 Лотка, Рокфорд C# и CSLA .NET Framework. Разработка бизнес-объектов / Рокфорд Лотка. - М.: Вильямс, 2010. - 816 c;

12 Мак-Дональд, Мэтью Silverlight 5 с примерами на C# для профессионалов / Мэтью Мак-Дональд. - М.: Вильямс, 2013. - 848 c;

13 Марченко, А. Л. Основы программирования на С# 2.0 / А.Л. Марченко. - М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2011. - 552 c;

14 Подбельский, В. В. Язык С#. Базовый курс / В.В. Подбельский. - М.: Финансы и статистика, Инфра-М, 2011. - 384 c;

15 Прайс, Джейсон Visual C# 2.0. Полное руководство / Джейсон Прайс, Майк Гандэрлой. - М.: Век +, Корона-Век, Энтроп, 2010. - 736 c;

16 Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C# / Джеффри Рихтер. - М.: Питер, 2013. - 928 c;

17 Смоленцев, Н. К. MATLAB. Программирование на Visual С#, Borland JBuilder, VBA (+ CD-ROM) / Н.К. Смоленцев. - М.: ДМК Пресс, 2011. - 456 c;

18 Троелсен, Эндрю Язык программирования C# 5.0 и платформа .NET 4.5 / Эндрю Троелсен. - М.: Вильямс, 2015. - 486 c.