|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Приднестровский государственный университет им. Т.Г. Шевченко | | | | | | | | | |
| Инженерно-технический институт | | | | | | | | | |
| Инженерно-технический факультет | | | | | | | | | |
| Кафедра программного обеспечения вычислительной техники  и автоматизированных систем | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | Допустить к защите | | | |
|  |  |  |  |  |  | зав. кафедрой ПОВТиАС, | | | |
|  |  |  |  |  |  | к.т.н., доцент | | | |
|  |  |  |  |  |  |  |  | С.Г. Федорченко | |
|  |  |  |  |  |  | « \_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2015 г. | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА** | | | | | | | | | |
| на соискание академической степени | | | | | | | | | |
| бакалавра | | | | | | | | | |
| по направлению 09.03.04 «Программная инженерия» | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | |
| **тема: «РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБРАБОТКИ РЕЕСТРОВ»** | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
| Расчетно-пояснительная записка | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Студент | |  |  |  |  |  | | |
|  | группы ИТ11ДР62ПИ1 | | |  |  |  | Иванович | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Научный руководитель, | | |  |  |  | Добровольская | | |
|  | ст. преподаватель | | |  |  |  | Екатерина Васильевна | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  | | |  |  |  |  | | |
|  |  | | |  |  |  |  | | |
|  |  | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| Тирасполь, 2015 | | | | | | | | | |

Приднестровский государственный университет им. Т.Г. Шевченко

Инженерно-технический институт

Инженерно-технический факультет

Кафедра программного обеспечения вычислительной техники

и автоматизированных систем

Утверждаю

зав. кафедрой ПОВТиАС,

к.т.н., доцент

С.Г. Федорченко

**«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_** 2015 г.

**ЗАДАНИЕ**

**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ БАКАЛАВРА**

**Студенту** *Ивановичу*

**Тема ВКР:**  *«Разработка программного обеспечения для обработки реестров»*

утверждена приказом по университету № 881-ОД от «09» июня 2015 г.

Срок сдачи расчетно-пояснительной записки на кафедру «25» июня 2015 г.

**Исходные данные к работе:** *база данных с формами, представлениями и примерами документов.*

**Перечень подлежащих разработке вопросов:** *доступ к базе данных посредством веб-браузера, поиск документов, обеспечение защиты от несанкционированного доступа.*

**Перечень дополнительных вопросов**: *произвести расчет затрат на выполнение данной работы в условиях института, рассмотрения вопросов охраны труда.*

Дата выдачи задания: «11» мая 2015 г

Научный руководитель, ст. преподаватель \_\_\_\_\_\_\_/ Добровольская Е.В. /

Задание принял к исполнению \_\_\_\_\_\_\_/ Харичков Н.И. /

**АННОТАЦИЯ**

В данной выпускной квалификационной работе реализовано программное обеспечение, предоставляющее доступ к документам, хранящимся в базе данных *Lotus Notes*.

Целью проектаявляется реализация веб-приложения, содержащего в себе алгоритмы выборки, поиска и открытия документов, а также защиту от несанкционированного доступа.

**ABSTRACT**

In the given graduated work a software that provides access to the documents stored in the Lotus Notes database.

The purpose of the work is to implement a web application that contains a picking algorithms, search and discovery of documents as well as protection against unauthorized access.

**СОДЕРЖАНИЕ**

|  |  |  |
| --- | --- | --- |
| ВВЕДЕНИЕ | | 5 |
| 1 ИССЛЕДОВАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ | | 6 |
|  | 1.1 Описание поставленной задачи, ее обоснование | 6 |
|  | 1.2 Обоснование актуальности исследуемой задачи | 6 |
|  | 1.3 Обзор методов решения подобных задач | 9 |
|  | 1.4 Средства реализации | 10 |
|  | 1.5 Постановка задачи, системные требования, требования к входным  данным и выходным формам | 11 |
| 2 ПРОЕКТИРОВАНИЕ СТРУКТУРЫ И АРХИТЕКТУРЫ  ПРОГРАММНОГО ПРОДУКТА | | 14 |
|  | 2.1 Выбор методов и средств для реализации, его обоснование | 14 |
|  | 2.2 Описание применяемых алгоритмов | 16 |
|  | 2.3 Структура, архитектура программного продукта | 17 |
|  | 2.4 Описание логической структуры программного продукта | 18 |
|  | 2.5 Функциональная схема, функциональное назначение  программного продукта | 19 |
| 3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА | | 27 |
|  | 3.1 Описание реализации | 27 |
|  | 3.2 Описание пользовательского интерфейса | 31 |
|  | 3.3 Методы и средства защиты программного продукта | 33 |
|  | 3.4 Тестирование и оценка надежности программного продукта | 34 |
|  | 3.5 Расчет себестоимости | 35 |
|  | 3.6 Охрана труда | 41 |
| ЗАКЛЮЧЕНИЕ | | 48 |
| ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ  И ТЕРМИНОВ | | 49 |
| СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ | | 50 |
| ПРИЛОЖЕНИЕ А – Руководство пользователя | | 51 |

**ВВЕДЕНИЕ**

Деятельность компании связана с оформлением большого количества документов: договоров, счетов-фактур и т.д. В каждой инстанции документ должен быть проверен, заверен и отправлен далее.

Автоматизация документооборота – один из способов изменить сложившийся порядок управления бизнес-процессами. Если предприятие использует *IT*-систему, вся документация находится в одной электронной базе, руководитель может в любое время увидеть, в какой стадии находится документ, и при необходимости принять меры. Документ уже не попадет к кому-либо на стол, пока не будет проверен и не пройдет соответствующих согласований в финансовой, юридической и маркетинговой службах [15].

Во многих современных предприятиях используются программные продукты, которые обеспечивают автоматизацию совместной работы и управленческой деятельности. Данные программные продукты получили повсеместное распространение в России и по всему миру. Их используют такие компании как МТС, ТНК, Балтика, а также государственные структуры. Применяемая нереляционная архитектура позволяет строить на этой платформе информационные системы, обрабатывающие сложно-структурированные данные, и автоматизировать динамически формируемые процессы.

*Lotus Notes* обладает встроенным механизмом репликации, что позволяет территориально удаленным подразделениям, оперативно обмениваться информацией и совместно работать с документами, как в интерактивном, так и в автономном режиме. Также имеется мощный механизм разделения доступа к документам, благодаря которому документ предоставляется сотруднику для просмотра или редактирования в зависимости от степени его участия в работе над документом [15].

# 1 ИССЛЕДОВАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

**1.1 Описание поставленной задачи, ее обоснование**

Задачей разрабатываемой программного продукта является получения и обработка информации из реестров. Реестры представляют собой базу данных *Lotus Notes,* в которой хранится информация о действующих юридических лицах и организациях (филиалах).

Разрабатываемый продукт должен предоставлять пользователям удобный интерфейс работы с данными, которые содержатся в базе данных. Пользователь должен быть обеспечен доступом к данным через любой веб-браузер.

Система имеет разграничение прав доступа: после входа в систему, пользователю, в зависимости от его роли, присваиваются ранее отведенные права на чтение и редактирование информации. Авторизация пользователей происходит сразу при открытии страницы в веб-браузере.

**1.2 Обоснование актуальности исследуемой задачи**

Можно выделить следующую классификацию причин, по которым существующий интерфейс не удовлетворяет потребности пользователей:

– избыточное время на обучение персонала;

– негативное влияние на производительность оператора;

– отсутствие поддержки программно-аппаратного обеспечения;

– стоит задача постепенной миграции на другое ПО.

   Время обучения очень важно при высокой частоте смены работающего персонала. Обычно это отделы продаж, различного рода обслуживание. При этом стоит задача сохранения существующей, уже работающей инфраструктуры внутри предприятия. Например, этим же программным обеспечением пользуется отдел маркетинга, где смена персонала намного реже и важны широкие возможности ПО.

Примером негативного влияния на производительность оператора, может быть организация системы продажи через 1С. Эта система является стандартом де-факто в области налогообложения, но сильно усложнит работу непосредственно продавца билетов. Хорошим решением будет написание отдельного интерфейса, который будет быстро решать задачу продажи билетов, и одновременно будет вносить изменения в исходную базу данных.

Отдельным пунктом стоит выделить моральное устаревание ПО. Не секрет, что многие компании перестают поддерживать старые версии ПО (как своего, так и того, от которого непосредственно зависит его работа), и оно перестает работать на новых версиях операционных систем. Основная сложность здесь возникает в том, что операционные системы и оборудование тоже устаревает, и очень трудно порой купить новые лицензии, или найти запасную часть к оборудованию старого образца. И хорошо, если проблему можно решить заблаговременно, и постепенно мигрировать на другое ПО. Иногда моральное устаревание происходит не так быстро, и программное обеспечение просто не может использовать новые возможности оборудования (не поддерживается новый сканер штрих-кодов, или не эффективно используется новый монитор с большей диагональю и разрешением).

Если по различным причинам было решено, что существующее ПО больше не удовлетворяет текущие или будущие потребности, и необходимо его сменить, то возникают некоторые сложности. Наиболее сложной частью является перенос существующих данных с минимальным временем простоя в работе. Но не менее сложным является вопрос переобучения рабочего персонала. Одним из решений, в таком случае и будет, как раз, написание оболочки, которая работает со старым хранилищем данных. А после того как работы по миграции будут решены - переключить источник данных на новую платформу.

Хорошим примером для того что было описано выше, является платформа *IBM Domino/Notes*. Это документ-ориентированная платформа для рабочих групп, в свое время получившая широкое распространение, обеспечивает (по замыслу разработчика) электронный документооборот на предприятиях. Начиная от почты и календаря, и заканчивая непосредственно работой с документами. Сложность интерфейса клиента данной платформы очевидна. Более того, в связи с тем, что документооборот сам по себе сложная и запутанная вещь - нельзя написать принципиально новое программное обеспечение, которое будет проще в освоении. Для сложных универсальных систем не существует простых универсальных и удобных пользовательских интерфейсов. Поэтому можно попытаться разработать индивидуальный продукт (и выполнять его поддержку, что часто очень дорого) или реализовать свой специализированный интерфейс пользователя к существующей базе данных. Этот интерфейс будет простым и удобным, по причине того, что он решает специализированную задачу для узкого круга пользователей. При этом, ваши данные будут храниться в надежном, испытанном временем и большим количеством пользователей, продукте. Объем работ, и требования к квалификации разработчиков, в таком случае, будут несравнимо меньше.

Использование фреймворка *XPages* предоставляет разработчику более широкие возможности, а значит, разрабатываемый продукт может быть более интерактивным, функциональным, в то же время упростить пользователям работу с данными. Данная технология встроена в продукт *IBM Lotus Domino*, что указывает на ее надежность, а также простое взаимодействие с другими элементами базы [5].

Для перемещения данных между элементами управления и хранилищем данных, усовершенствования функциональных возможностей или дополнительной защиты можно использовать *JavaScript* [6].

Преимуществом использования данного фреймворка являются низкий порог входа разработчика и скорость разработки. Формы и представления в базе данных могу иметь сложные структуры. Однако после их объединения процесс разработки протекает довольно легко. Например, связывание формы с элементом управления на *XPages* состоит просто в перемещении поля на элемент управления. Также если поместить *XPages* в существующее приложение *IBM Domino Designer*, то можно полностью использовать возможности существующих форм и представлений [3].

Основной идеей разработки является создание легкого программного продукта, способного предоставить стандартный функционал работы с данными, при этом не требующего установки клиентского приложения *IBM Notes*. Данный продукт должен предоставлять минимальные возможности, которые есть в клиентском приложении *IBM Domino*, с возможностью дальнейшего усовершенствования и доработки.

Актуальность создания данного продукта заключается в переходе к более оперативным и удобным способам доступа к информации. Разработка веб-интерфейса позволяет обеспечить работу с информацией в браузере пользователя, а не в клиентском приложении *Lotus*. Это также позволяет в дальнейшем расширить функциональные возможности продукта. Актуальность веб-ресурса:

– быстрый вывод в Веб существующих приложений *Notes/Domino*;

– простое создание новых веб-приложений;

– консолидация разнородных данных;

– широкие функциональные возможности;

– удобный и понятный интерфейс.

**1.3 Обзор методов решения подобных задач**

В настоящее время у любого предприятия, вне зависимости от того, к какой области промышленности оно относится, есть своя информационная система, в которой хранятся все данные. Такие данные могут представлять информацию: бухгалтерия, о сотрудниках предприятия, ведомости, контакты сотрудничающих предприятий, расписания событий и т.п. Эти данные хранятся в электронном виде и достаточно защищены. Для удобства работы с данными, а также наглядного их представления каждое предприятие имеет свой веб-ресурс, в котором отображена вся доступная информация. Такие веб-ресурсы позволяют отображать всю необходимую информацию, а также обеспечивают обратную связь с пользователями.

Примером решения подобных задач может быть любой веб-сайт, принадлежащий различным государственным предприятиям.

**1.4 Средства реализации**

Средством реализации данного программного продукта был выбран фреймворк от *IBM* – *XPages.*

*XPages* – технология для быстрой разработки веб-приложений. Технология поддерживает данные *IBM Lotus Notes*, а также реляционные базы данных. Приложения, созданные с помощью технологии *XPages* работают в интернет-браузере и являются кроссплатформенными. Модель программирования основана на языках веб-разработки и стандартах *JavaScript*, *Ajax*, *Java*, *Dojo* *Toolkit*, *Server*-*side* *JavaScript* и *JavaServer* *Faces*. *XPages* использует *IBM Lotus Notes*, платформу *IBM* для быстрой разработки приложений, включая такой функционал, как документо-ориентированные базы данных [2].

Редактор *XPages* обладает рядом преимуществ и отличительных особенностей. Некоторые из них:

– Основная технология редактора – *JSP (JavaServicePages).*

– В *XPages* входят готовые визуальные компоненты, которые можно «перетаскивать» прямо на создаваемую страницу: *FileUpload/FileDownload* (компонент загрузки и выгрузки файлов), *DateTimePicker* (выпадающий календарь), *Section*, *TabbedPanel* и другие.

– Библиотека *JavaScript* – *Dojo*, которая решает значительную часть задач по написанию совместимого кода на стороне клиента [4].

– Совместимость с браузерами.

– Поддержка разных клиентских платформ.

Разработка *XPages*-страниц, в сущности, базируется на двух традиционных элементах дизайна: формах (*form*) для определения данных и представлениях (*view*) для вывода данных. Для перемещения данных между элементами управления и хранилищем *Lotus Domino* можно использовать *JavaScript* [1].

**1.5 Постановка задачи, системные требования, требования**

**к входным данным и выходным формам**

Целью ВКР является разработка единой точки доступа к документам, находящимся в разных базах данных. База данных должна содержать в себе все вышеперечисленные объекты и располагаться на сервере.

Интерфейс должен позволять делать выборки документов на чтение, как по фиксированным запросам по ограниченным наборам данных, так и по произвольному полнотекстовому поиску.

Интерфейс пользователя должен представлять собой веб-интерфейс, работающий в основных браузерах Интернета актуальной версии. Базы данных представляют собой четыре реестра, состоящие из объектно- ориентированных баз данных *Lotus Notes* с произвольной структурой. Название базы данных: «Архив юридических лиц».

Доступ к базам данных осуществляется по внутренней компьютерной сети органа с реестрами на его центральных серверах.

*Информационные объекты хранения для базы данных системы.* Общего назначения: орган государственной власти, регистрационный номер, наименование организации, филиал (представительство), наименование, дата регистрации, документ, уникальный идентификатор. Служебные: запись о пользовательской транзакции, уровень доступа в систему, учетная запись полномочий пользователя (для организации защиты от несанкционированного доступа).

*Функциональные характеристики системы:*

– просмотр документов в БД;

– изменение содержания БД:

– изменение названий документов;

– удаление документов;

– осуществление поиска документов по запросу пользователя;

– удаленный доступ к серверу по протоколу *TCP/IP*;

– обеспечение защиты и безопасности данных, в частности:

– разграничение прав доступа к ресурсам сервера.

*Цели и задачи системы.* Система будет обеспечивать доступ различных групп пользователей к документам, а именно:

– Предоставлять информацию о действующих юридических лицах с сортировкой: по названиям, по дате регистрации, по местонахождению, по организационно-правовой форме, по пакетам.

– Предоставлять информацию о действующих государственных органах с сортировкой: по названиям, по пакетам.

– Предоставлять информацию о действующих представительствах с сортировкой: по названиям, по местонахождению, по пакетам.

– Предоставлять информацию о предприятиях малого бизнеса с сортировкой по наименованию.

– Обеспечивать выборку документов по фиксированным запросам по ограниченным наборам данных и по произвольному полнотекстовому поиску.

*Категории пользователей.* База данных содержит список управления доступом, предназначенный для определения уровней доступа пользователей и серверов в *IBM Lotus Notes*. Уровни доступа, назначенные пользователям, определяют круг задач, выполняемых этими пользователями в базе данных. Уровни доступа, назначенные серверам, определяют круг сведений в базе данных, реплицируемых этими серверами. Изменять список управления доступом может только пользователь с доступом на уровне управляющего. Разработчик и управляющий базой данных могут создать одну или несколько ролей, позволяющих уточнить круг доступа к отдельным представлениям, формам, разделам и полям базы данных. В таблице 1.1 представлены уровни доступа, их действия и назначение:

Таблица 1.1 – Уровни доступа в БД *Lotus Notes*

|  |  |  |
| --- | --- | --- |
| **Уровень доступа** | **Действия** | **Назначение** |
| Управляющий | Удаление базы данных  Шифрование базы данных  Изменение параметров репликации  Изменение СУД базы данных | Пользователи, ответственные за эту базу данных. Если один пользователь отсутствует, базой данных может управлять другой. |
| Разработчик | Создание полнотекстового индекса  Изменение всех элементов структуры базы данных  Выполнение всех задач, разрешенных для более низких уровней доступа | Разработчик базы данных и (или) пользователь, ответственный за будущие обновления структуры. |
| Редактор | Создание документов  Изменение всех документов, включая созданные другими пользователями  Чтение всех документов, если в форме нет поля **Читатели** | Любой пользователь, которому разрешено создание и изменение документов в базе данных. |
| Автор | Создание документов  Изменение документов с полем **Авторы**, в котором указан данный пользователь  Чтение всех документов, если в форме нет поля **Читатели** | Пользователи, которым требуется добавлять документы в базу данных. |
| Читатель | Чтение документов | Пользователи, которым требуется чтение документов в базе данных. Создание и изменение документов запрещено. |
| Нет доступа | Только чтение общих документов и запись общих документов | Пользователи с ограниченным доступом, которым не требуется доступ к базе данных, либо пользователи, не имеющие особого доступа. |

Минимальные системные требования:

– операционная система: *Windows XP*;

­– оперативная память: 1024 Мб;

– тактовая частота процессора: 1,8 Ггц;

– пространство на жестком диске: 2 Гб;

Рекомендуемые системные требования:

– операционная система: *Windows XP / Windows 7*;

– оперативная память: 2048 Мб;

– тактовая частота процессора: 2,4 Ггц;

– пространство на жестком диске: 10 Гб;

– манипулятор «мышь», клавиатура.

**2 ПРОЕКТИРОВАНИЕ СТРУКТУРЫ И АРХИТЕКТУРЫ**

**ПРОГРАММНОГО ПРОДУКТА**

**2.1 Выбор методов и средств для реализации, его обоснование**

В ВКР разработан программный продукт, представляющий собой клиент-серверное приложение для работы с информационным реестром, представленным в виде базы данных *Lotus Notes*. Реализованы следующие функции: выборка, сортировка и редактирование заголовков данных во всех таблицах, поиск данных в таблицах, защита от несанкционированного доступа, защита совместно используемой базы данных.

База данных располагается на сервере *Lotus Notes*. Сервер обеспечивает взаимодействие с базой данных, т.е. принимает запросы от клиентских приложений, обрабатывает их, и возвращает значение. Сервер может одновременно принять в любой момент запрос от клиентских приложений и производить работу с базой данных.

На рисунке 2.1 представлена диаграмма вариантов использования программного продукта:

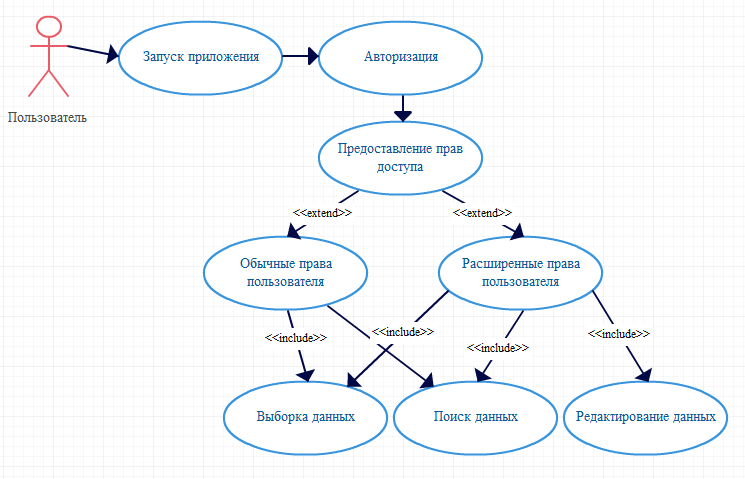


Рисунок 2.1 – Диаграмма вариантов использования клиентского приложения

В таблице 2.1 приведено описание и назначение актеров и прецедентов, участвующих в программе.

Таблица 2.1 – Описание актеров и прецедентов диаграммы ВИ

|  |  |  |
| --- | --- | --- |
| **Актер** | | **Краткое описание** |
| Пользователь | | Пользуется программой, взаимодействует с сервером. |
| **Прецедент** | **Краткое описание** | |
| Запуск приложения | В интернет-браузере вводится адрес сервера. | |
| Авторизация | Ввод пользователем логина и пароля, определение его группы пользователей. | |
| Предоставление прав доступа | Включение и отключение некоторых функций в программе в зависимости от прав доступа. | |
| Обычные права пользователя | Группа обычных пользователей с некоторыми ограничениями. | |
| Расширенные права пользователей | Группа пользователей, наделенная всеми правами. | |
| Выборка данных | Просмотр всех данных реестра с указанной сортировкой. | |
| Поиск данных | Выборка определенных данных по запросу пользователя. | |
| Редактирование данных | Редактирование заголовков данных. | |

Полное описание по Алистеру Коберну:

*Название* – считать данные из информационного реестра.

*Контекст использования* – разработка приложения, осуществляющего доступ к документам, находящимся в разных базах данных.

*Область действия* – система хранения документов.

*Бизнес-требования:* система должна сократить время доступа к данным.

*Требования пользователей:* программа должна быть представлена в виде веб-приложения.

*Функциональные требования:* продукт должен позволять делать выборки документов на чтение, как по фиксированным запросам по ограниченным наборам данных, так и по произвольному полнотекстовому поиску.

*Основное действующее лицо* – сотрудник службы.

*Предусловие* – копия базы данных.

*Минимальные гарантии* – ограничение доступа и защита данных.

*Гарантии успеха* – отсортированные данные, удовлетворяющие результатам запроса.

*Триггер* – отсутствует, управление программой осуществляется вручную.

*Основной сценарий*:

шаг 1: запуск программы;

шаг 2: авторизация;

шаг 3: ввод параметров поиска;

шаг 4: группировка документов;

шаг 5: считывание результатов.

*Расширения:*

шаг 5, расширение 1: печать результатов.

**2.2 Описание применяемых алгоритмов**

*Разграничение прав доступа*. Для реализации разделения прав доступа в программном продукте используется алгоритм, в основе которого лежит список контроля доступа, называемый *Access Control List (ACL).* Этот список определяет, кто или что может получать доступ к конкретному объекту, и какие именно операции разрешено или запрещено этому субъекту проводить над объектом. Список доступа представляет собой структуру данных, содержащую записи, определяющие права индивидуального пользователя или группы на специальные системные объекты [7]. В разрабатываемом программном продукте авторизация пользователей производится сразу при входе на сайт. Во время авторизации определяется, к какой группе принадлежит этот пользователь, после чего эти данные сохраняются на протяжении всей сессии. Каждый объект в реестре, т.е. таблица, содержит в себе список групп пользователей и права, которыми они обладают. При попытке пользователя перейти на страницу просмотра какой-либо таблицы, производится соответствие группы пользователя и ее полномочий, после чего пользователь наделяется определенными правами на действия с таблицей.

Алгоритм работы программы представлен на рисунке 2.2:

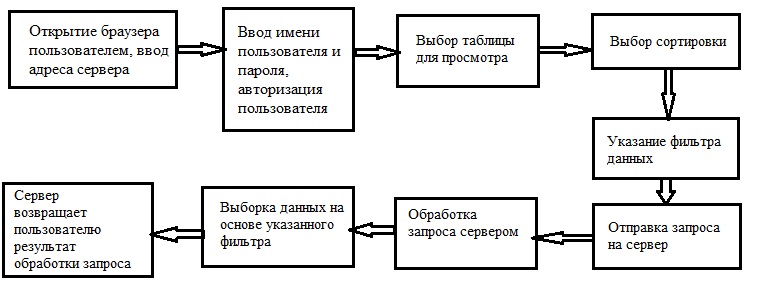


Рисунок 2.2 – Алгоритм взаимодействия клиента и сервера разрабатываемого программного продукта

**2.3 Структура, архитектура программного продукта**

Серверное приложение представляет собой сервер *Domino*, где будет развернуто разрабатываемое *XPages*-приложение при помощи *IBM Lotus Domino Administrator,* а также при помощи механизмов репликации и синхронизации, встроенных в *IBM Lotus Domino*. Серверы *Domino* и рабочие станции *Notes* (клиенты) формируют интегрированную архитектуру клиент/сервер для совместной работы и обмена информацией [8]. На рисунке 2.3 отображена архитектура системы *Notes/Domino:*

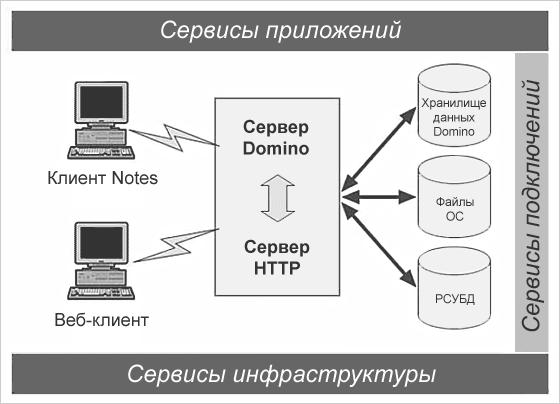


Рисунок 2.3 – Архитектура *Notes/Domino*

Система *Notes/Domino* представляет собой комбинацию следующих компонентов архитектуры клиент/сервер:

Таблица 2.2 – Компоненты системы *Notes/Domino*

|  |  |
| --- | --- |
| **Компонент** | **Функции** |
| Сервер *Domino* (веб-сервер) | * Сервер обеспечивает хранение информации, безопасность, передачу и синхронизацию данных с другими серверами *Domino*. * В зависимости от запроса клиента, сервер может предоставлять информацию из различных источников |
| Клиент *Notes* и веб-клиент | * Программное обеспечение клиента *Notes* позволяет получать доступ к данным *Domino*, которые могут располагаться как на сервере, так и локально. * Веб-клиент имеет возможность получать доступ к данным *Domino* через веб-браузер. |

Одним из базовых компонентов архитектуры *Notes* является хранилище объектов (*Object Store*), известное как база данных *nsf* (*Notes Storage Facility*).

В таблице 2.3 приведено описание элементов базы данных *Domino*:

Таблица 2.3 – Элементы базы данных

|  |  |
| --- | --- |
| **Элементы базы данных** | **Описание** |
| Документы (*Documents*) | Содержат данные различных типов – текст, графику, файлы, созданные в других приложениях. |
| Формы (*Forms*) | Используются для создания и отображения документов через веб-браузер. |
| Представления (*Views*) | Используются для отображения документов в виде динамического списка. |
| Агенты (*Agents*) | Программы, которые могут быть запланированы на выполнение по расписанию; например, программа для обработки группы документов, выбранных по определенному критерию. |

**2.4 Описание логической структуры программного продукта**

Такие задачи, как управление, обновление, поддержка и распространение данных выполняют службы сервера *Domino.* Различают следующие категории служб: приложений, подключений и инфраструктуры. В таблице 2.4 приведены категории служб *Domino* сервера:

Таблица 2.4 – Категории служб сервера *Domino*

|  |  |
| --- | --- |
| **Службы** | **Описание** |
| Приложений (*Application Service*) | Обеспечивают средства для создания приложений:   * Клиент *Domino Designer*, позволяющий разрабатывать веб-приложения для *Notes*. * Шаблоны *Notes*. * Поддержка языков програмирования *LotusScript*, *Java*, Javascript. |
| Подключений (*Connection Service*) | Позволяет использовать *Domino* с существующими реляционными базами данных. |
| Инфрастуктуры (*Infrastructure Service*) | Обеспечивают фундамент *Domino*:   * Базовые службы, такие как справочник, безопасность, управление агентами и репликации. * Протоколы, определяющие правила подключений к серверу. |

Обмен данными осуществляется посредством объектов *OLE*:

– связывание и внедрение объектов;

– создание связей с другими приложениями или данными;

– внедрение данных в документ *Notes;*

– добавление пользовательского объекта *OLE* в документ *Notes;*

– разрешение вопросов, связанных с внедрением объектов *OLE* и элементов управления *ActiveX;*

– разрешение вопросов, связанных с обменом данными.

**2.5 Функциональная схема, функциональное назначение**

**программного продукта**

База данных имеет три раздела, которые логически разделяют таблицы по смысловому принципу: *Действующие*, *Иерархия изменений*, *Прекращение деятельности* (содержит подраздел *Исключенные из гос. реестра*). Каждый из этих разделов содержит в себе представления, названия которых могу совпадать. Далее будут приведено содержимое разделов и описана структура каждого представления.

Раздел *Действующие* содержит следующие представления:

Структура представления *Юридические лица* с сортировкой по названиям представлена в таблице 2.5:

Таблица 2.5 – Структура Юридические лица (По названиям)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Рег. номер | *Anumreg* | Числовой |
| Наименование организации | *FullName* | Текстовый |

Структура представления *Юридические лица* с сортировкой по дате регистрации изменений представлена в таблице 2.6:

Таблица 2.6 – Структура Юридические лица (По дате рег. изменений)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Дата рег. изменений | *Formula* | Дата |
| Рег. номер | *Anumreg* | Числовой |
| Наименование организации | *FullName* | Текстовый |

Структура представления *Юридические лица* с сортировкой по дате регистрации представлена в таблице 2.7:

Таблица 2.7 – Структура Юридические лица (По дате регистрации)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Дата регистрации | *Formula* | Дата |
| Рег. номер | *Anumreg* | Числовой |
| Наименование организации | *FullName* | Текстовый |

Структура представления *Юридические лица* с сортировкой по местонахождению представлена в таблице 2.8:

Таблица 2.8 – Структура Юридические лица (По местонахождению)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Местонахождение | *Atown* | Текстовый |
| Рег. номер | *Anumreg* | Числовой |
| Наименование организации | *FullName* | Текстовый |

Структура представления *Юридические лица* с сортировкой по организационно-правовой форме представлена в таблице 2.9:

Таблица 2.9 – Структура Юридические лица (По орг. прав. форме)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Тип формы | *Formula* | Текстовый |
| Рег. номер | *Anumreg* | Числовой |
| Наименование организации | *FullName* | Текстовый |

Структура представления *Юридические лица* с сортировкой по пакетам представлена в таблице 2.10:

Таблица 2.10 – Структура Юридические лица (По пакетам)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Пакет | *Formula* | Текстовый |
| Рег. номер | *Anumreg* | Числовой |
| Наименование организации | *FullName* | Текстовый |

Структура представления *Государственные органы* с сортировкой по названиям представлена в таблице 2.11:

Таблица 2.11 – Структура Государственные органы (По названиям)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Орган гос. власти | *FullName* | Текстовый |

Структура представления *Государственные органы* с сортировкой по пакетам представлена в таблице 2.12:

Таблица 2.12 – Структура Государственные органы (По пакетам)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Пакет | *Formula* | Текстовый |
| Орган гос. власти | *FullName* | Текстовый |

Структура представления *Филиалы, представительства* с сортировкой по названиям представлена в таблице 2.13:

Таблица 2.13 – Структура Филиалы, представительства (По названиям)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Учредитель | *Formula* | Текстовый |
| Рег. номер | *Anumreg* | Числовой |
| Филиал (представительство) | *FullName* | Текстовый |

Структура представления *Филиалы, представительства* с сортировкой по дате регистрации изменений представлена в таблице 2.14:

Таблица 2.14 – Структура Филиалы, представительства (По дате рег. изменений)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Дата рег. изменений | *Formula* | Дата |
| Рег. номер | *Anumreg* | Числовой |
| Филиал (представительство) | *FullName* | Текстовый |

Структура представления *Филиалы, представительства* с сортировкой по местонахождению представлена в таблице 2.15:

Таблица 2.15 – Структура Филиалы, представительства (По местонахождению)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Местонахождение | *Atown* | Дата |
| Филиал (представительство) | *FullName* | Текстовый |

Структура представления *Филиалы, представительства* с сортировкой по пакетам представлена в таблице 2.16:

Таблица 2.16 – Структура Филиалы, представительства (По пакетам)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Пакет | *Formula* | Текстовый |
| Рег. номер | *Anumreg* | Числовой |
| Филиал (представительство) | *FullName* | Текстовый |

Структура представления *Предприятия малого бизнеса* представлена в таблице 2.17:

Таблица 2.17 – Структура Предприятия малого бизнеса

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Филиал | *Formula* | Текстовый |
| Наименование | *ShortName* | Текстовый |
| Дата регистрации | *Adatereg* | Дата |
| Рег. номер | *Anumreg* | Числовой |

Раздел *Иерархия изменений* содержит следующие представления:

Структура представления *Организации* представлена в таблице 2.18:

Таблица 2.18 – Структура Организации

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Иерархия изменений в организациях | *Formula* | Текстовый |
| Дата рег. изменений | *Formula* | Дата |

Структура представления *Государственные органы* представлена в таблице 2.19:

Таблица 2.19 – Структура Государственные органы

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Иерархия изменений в органах гос. власти | *Formula* | Текстовый |
| Дата рег. изменений | *Formula* | Дата |

Структура представления *Филиалы, представительства* представлена в таблице 2.20:

Таблица 2.20 – Структура Филиалы, представительства

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Иерархия изменений в филиалах | *Formula* | Текстовый |
| Дата рег. изменений | *Formula* | Дата |

Раздел *Прекращение деятельности* содержит следующие представления:

Структура представления *Уведомление* представлена в таблице 2.21:

Таблица 2.21 – Структура Уведомление

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Дата уведомления | *BeginLiquid* | Дата |
| Ликвидируемые организации | *FullName* | Текстовый |

Структура представления *В стадии ликвидации* представлена в таблице 2.22:

Таблица 2.22 – Структура В стадии ликвидации

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Дата уведомления | *BeginLiquid* | Дата |
| Ликвидируемые организации | *FullName* | Текстовый |
| В стадии ликвидации | *SartedLiquid* | Дата |

Структура представления *Реорганизованные* представлена в таблице 2.23:

Таблица 2.23 – Структура Реорганизованные

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Названия реорганизованных оргнанизаций | *FullName* | Текстовый |
| Форма | *Formula* | Текстовый |

Подраздел *Исключенные из государственного реестра* содержит следующие представления:

Структура представления *Юридические лица* с сортировкой по названиям представлена в таблице 2.24:

Таблица 2.24 – Структура Юридические лица (По названиям)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Рег. номер | *Anumreg* | Числовой |
| Наименование организации | *FullName* | Текстовый |

Структура представления *Юридические лица* с сортировкой по дате ликвидации представлена в таблице 2.25:

Таблица 2.25 – Структура Юридические лица (По дате ликвидации)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Дата ликвидации | *DataLiquid* | Дата |
| Рег. номер | *Anumreg* | Числовой |
| Наименование организации | *FullName* | Текстовый |
| Дата регистрации | *Adatereg* | Дата |
| Дата нахождения в стадии ликвидации | *SartedLiquid* | Дата |

Структура представления *Юридические лица* с сортировкой по местонахождению представлена в таблице 2.26:

Таблица 2.26 – Структура Юридические лица (По местонахождению)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Местонахождение | *Atown* | Текстовый |
| Рег. номер | *Anumreg* | Числовой |
| Наименование организации | *FullName* | Текстовый |

Структура представления *Юридические лица* с сортировкой по организационно-правовой форме представлена в таблице 2.27:

Таблица 2.27 – Структура Юридические лица (По орг. прав. форме)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Тип формы | *Formula* | Текстовый |
| Рег. номер | *Anumreg* | Числовой |
| Наименование организации | *FullName* | Текстовый |

Структура представления *Юридические лица* с сортировкой по пакетам представлена в таблице 2.28:

Таблица 2.28 – Структура Юридические лица (По пакетам)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Пакет | *Formula* | Текстовый |
| Рег. номер | *Anumreg* | Числовой |
| Наименование организации | *FullName* | Текстовый |

Структура представления *Государственные органы* с сортировкой по названиям представлена в таблице 2.29:

Таблица 2.29 – Структура Государственные органы (По названиям)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Орган гос. власти | *FullName* | Текстовый |

Структура представления *Государственные органы* с сортировкой по пакетам представлена в таблице 2.30:

Таблица 2.30 – Структура Государственные органы (По пакетам)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Пакет | *Formula* | Текстовый |
| Орган гос. власти | *FullName* | Текстовый |

Структура представления *Филиалы, представительства* с сортировкой по названиям представлена в таблице 2.31:

Таблица 2.31 – Структура Филиалы, представительства (По названиям)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Учредитель | *Formula* | Текстовый |
| Рег. номер | *Anumreg* | Числовой |
| Филиал (представительство) | *FullName* | Текстовый |

Структура представления *Филиалы, представительства* с сортировкой по дате ликвидации изменений представлена в таблице 2.32:

Таблица 2.32 – Структура Филиалы, представительства (По дате ликвидации)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Дата ликвидации | *DataLiquid* | Дата |
| Рег. номер | *Anumreg* | Числовой |
| Филиал (представительство) | *FullName* | Текстовый |

Структура представления *Филиалы, представительства* с сортировкой по местонахождению представлена в таблице 2.33:

Таблица 2.33 – Структура Филиалы, представительства (По местонахождению)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Местонахождение | *Atown* | Дата |
| Филиал (представительство) | *FullName* | Текстовый |

Структура представления *Филиалы, представительства* с сортировкой по пакетам представлена в таблице 2.34:

Таблица 2.34 – Структура Филиалы, представительства (По пакетам)

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Пакет | *Formula* | Текстовый |
| Рег. номер | *Anumreg* | Числовой |
| Филиал (представительство) | *FullName* | Текстовый |

Структура представления *Уведомление о прекращении деятельности* представлена в таблице 2.35:

Таблица 2.35 – Структура Уведомление о прекращении деятельности

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Имя поля | Домен |
| Наименование | *FullName* | Текстовый |
| Дата уведомления | *Activity* | Дата |

**3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА**

**3.1 Описание реализации**

Разработка программного продукта производится в среде *Lotus Domino Designer,* язык программирования выбран *Javascript.* Программный продукт представляет собой веб-приложение, обеспечивающее безотказную работу в таких современных веб-браузерах, как: *Internet Explorer*, *Mozilla Firefox*, *Google Chrome* и т.д.

Основной задачей разработки является выборка данных. Данные в *Lotus* находятся в хранилищах, при этом они не описаны с помощью таблиц, жестко задающих формат данных. Единица данных называется документом. Каждый отдельный документ является результатом заполнения формы. Открытый документ представляется в виде заполненной формы, в том виде, в котором она была непосредственно перед сохранением. Форма содержит в себе поля, обладающие уникальным идентификатором, которые могут обладать различными форматами ввода данных: текстовый, числовой, дата/время и т.д. На рисунке 3.1 приведен конструктор форм:

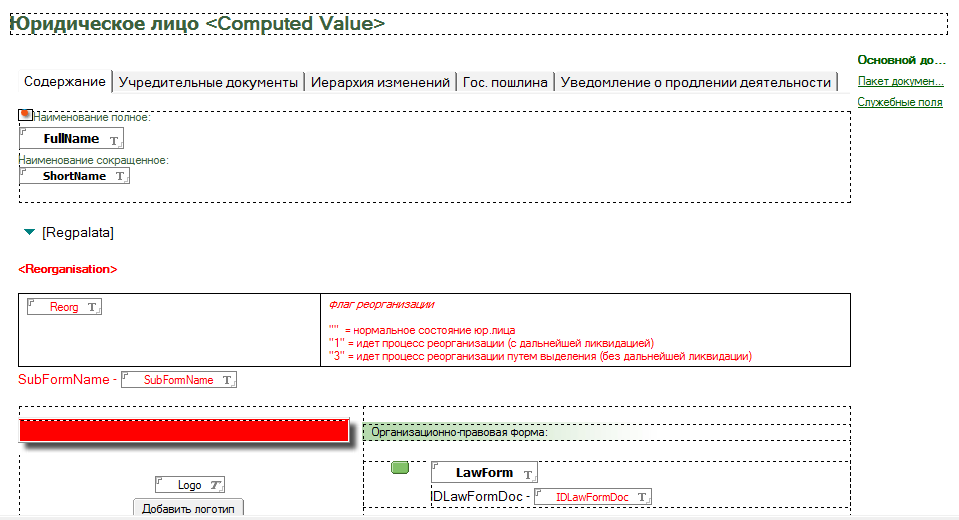


Рисунок 3.1 – Конструктор формы

Для отображения списка документов в *Lotus* используются представления. Представление конструируется на основе привязки к данным хранилища с помощью идентификатора, а также путем написания формул на языке формул в *Lotus* [13]*.* На рисунке 3.2 приведен конструктор представлений:

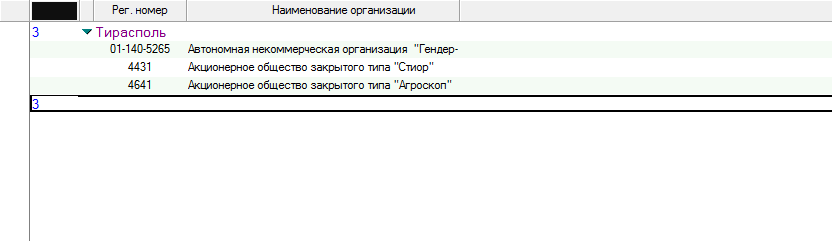


Рисунок 3.2 – Конструктор представления

Представление выбирает документы, данные которых удовлетворяют всем требованиям. Требованием может быть наличие записи в конкретном поле или выполнение условия формулы. Далее приведен код, написанный на языке формул, задающий условие для отображения данных:

|  |
| --- |
| SELECT (Form = "Register" & !@IsResponseDoc) & (@IsMember("4";CurStatusCreatReorgLiquidIzmen) & @IsNotMember("5";CurStatusCreatReorgLiquidIzmen))&!@Contains(@LowerCase(Liquid);"суд") & IsArhivDoc=1  @If(DataLiquid != "";110;1) |

Листинг 3.1 – Запрос на выборку определенных данных

Представление является строгой структурой представления данных, с которой будет работать разрабатываемый программный продукт. Соответственно, для работы необходимо создать набор представлений, на основе которых в дальнейшем и будет работать веб-приложение.

*XPage* – основной элемент разрабатываемого программного продукта, который представлен в виде отдельной страницы. Данный элемент предоставляет возможность размещения на нем как встроенных, так и нестандартных элементов управления, при этом он широко взаимодействует с другими элементами *Lotus.* Основой работы приложения является связка *XPage-View,* т.е. подразумевается, что для каждого представления будет создана отдельная страница в веб-приложении. Каждая страница привязана к представлению, которое предоставит необходимые данные. На рисунке 3.3 представлены параметры страницы *XPage,* одним из которых является привязка к представлению:

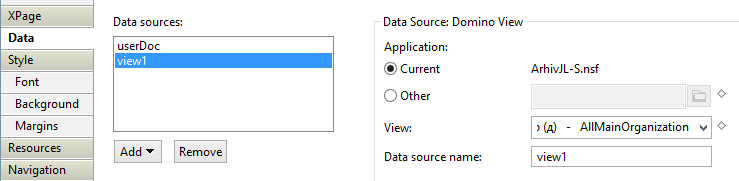


Рисунок 3.3 – Свойства страницы *XPages*

В роли «контейнера», который будет заполняться данными, выбран встроенный элемент управления *Repeater.* Данный элемент управления является аналогом списка, который может быть привязан к какой-либо структуре данных, с тем отличием, что формат этих данных задается заранее [14]. В нашем случае, содержимым *Repeater* является элемент управления *LinkLabel.* Их количество напрямую зависит от количества столбцов в представлении. На рисунке 3.4 представлен конструктор страницы *XPage*, на которой размещены элементы управления *Repeater* и *LinkLabel*:

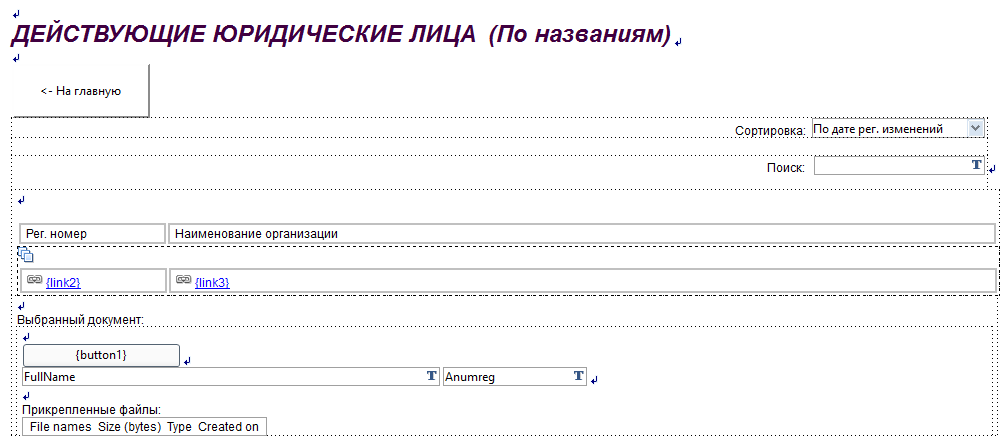


Рисунок 3.4 – Конструктор страницы *XPages*

Каждый элемент *LinkLabel* привязывается к определенному столбцу в представлении, поэтому в свойстве *Text* указывается индекс столбца в заполненной ранее структуре данных. Далее приведен код, который выполняет вышеописанное действие:

|  |
| --- |
| rowData.getColumnValues()[3] |

Листинг 3.2 – Привязка элемента *LinkLabel*

Применение элемента управления *LinkLabel* объясняется тем, что есть возможность нажатия кнопкой мыши на какую либо запись. В свою очередь, в дальнейшем это дает возможность работы с документом, которому принадлежит эта запись. Для того, чтобы была возможность получить идентификатор документа, был описан обработчик события *Click* элемента *LinkLabel* [12]*.* Далее приведен код, выполняющий вышеописанные действия:

|  |
| --- |
| viewScope.put("UNID", rowData.getUniversalID()) |

Листинг 3.3 – Получение уникального идентификатора

Поиск в программе работает по следующему принципу: во время передачи данных из хранилища в связанный элемент, программно производится проверка каждого документа. Если в названии документа содержится текст, введенный в строке поиска, то этот документ будет отображен. Проверка производится каждый раз во время выборки данных. Далее приведен код, выполняющий вышеописанные действия:

|  |
| --- |
| var inputText3:com.ibm.xsp.component.xp.XspInputText = getComponent("inputText4");  var str = inputText3.getValue();  if(str == null)  { return true; }  else  { return (rowData.getColumnValues()[0].indexOf(str)>-1 || rowData.getColumnValues()[1].indexOf(str)>-1 || rowData.getColumnValues()[2].indexOf(str)>-1); } |

Листинг 3.4 – Алгоритм поиска документов

После выбора какого-либо документа нажатием по нему кнопкой мыши, в нижней части страницы отображается элемент управления *FileDownload* позволяющий просмотреть и сохранить на диск прикрепленные к этому документу файлы. На рисунке 3.5 представлена настройка привязки элемента управления, осуществляющая выборку прикрепленных файлов:

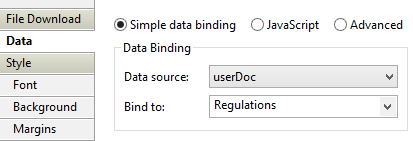


Рисунок 3.5 – Свойства элемента управления *FileDownload*

Также, после выбора документа, его можно открыть нажатием кнопки на веб-форме. Для этого, был описан обработчик события *Click* элемента *Button*. Далее приведен код, выполняющий вышеописанные действия:

|  |
| --- |
| viewScope.put("UNID", rowData.getUniversalID());  importPackage(pack1);  var v = new linkSrv();  v.setId(rowData.getDocument().getUniversalID());  var lin = v.mainCode();  var doc = v.getId();  var url = lin+"/0/"+doc+"?OpenDocument";  view.postScript("window.open('"+ url +"')") |

Листинг 3.5 – Открытие документа

**3.2 Описание пользовательского интерфейса**

При запуске веб-приложения отображается окно авторизации. В нем находятся два текстовых поля для ввода имени пользователя и пароля. На рисунке 3.6 представлено окно авторизации:

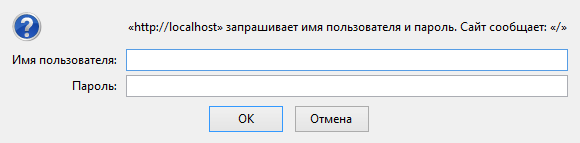


Рисунок 3.6 – Окно авторизации

Если авторизация прошла успешно, пользователь может продолжить работу с приложением, ему открывается главная страница. Главная страница содержит в себе элементы управления, позволяющие перейти к просмотру документов из различных разделов. На рисунке 3.7 представлена главная страница веб-приложения:

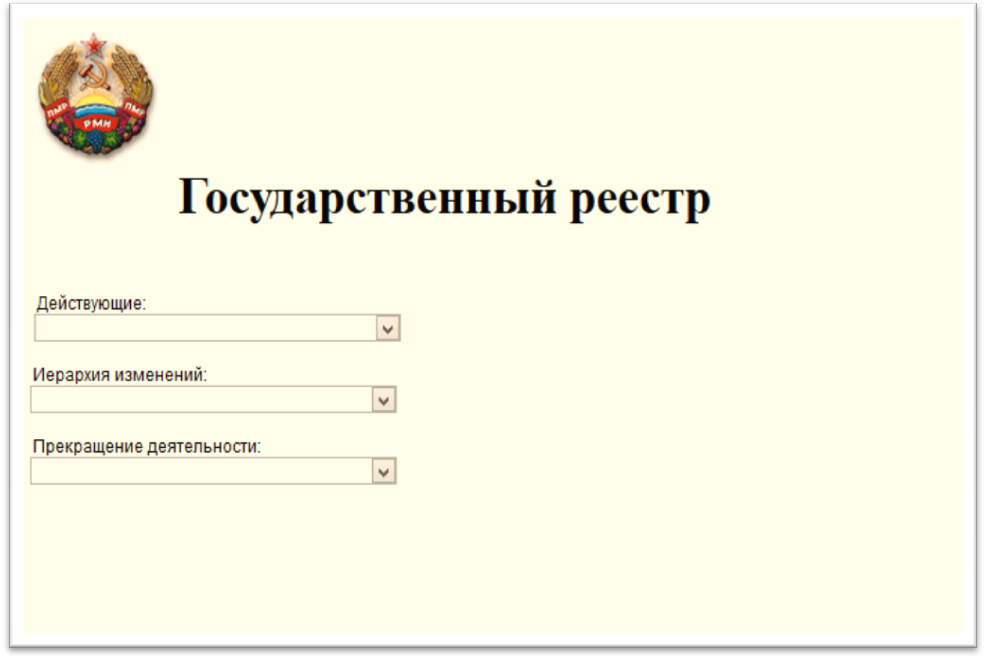


Рисунок 3.7 – Главная страница

Страница просмотра документов содержит в себе заголовок, кнопки управления, текстовое поле для осуществления поиска, а также элемент управления *Repeater* для отображения выбранных документов. Элемент *Repeater* содержит в себе несколько элементов *LinkLabel*, заключенных в рамку. Над каждым из этих элементов расположен элемент *Label*, отображающий название текущего столбца в представлении. В итоге, выбранные документы представлены в виде таблицы. На рисунке 3.8 представлен пример вышеописанной веб-страницы:

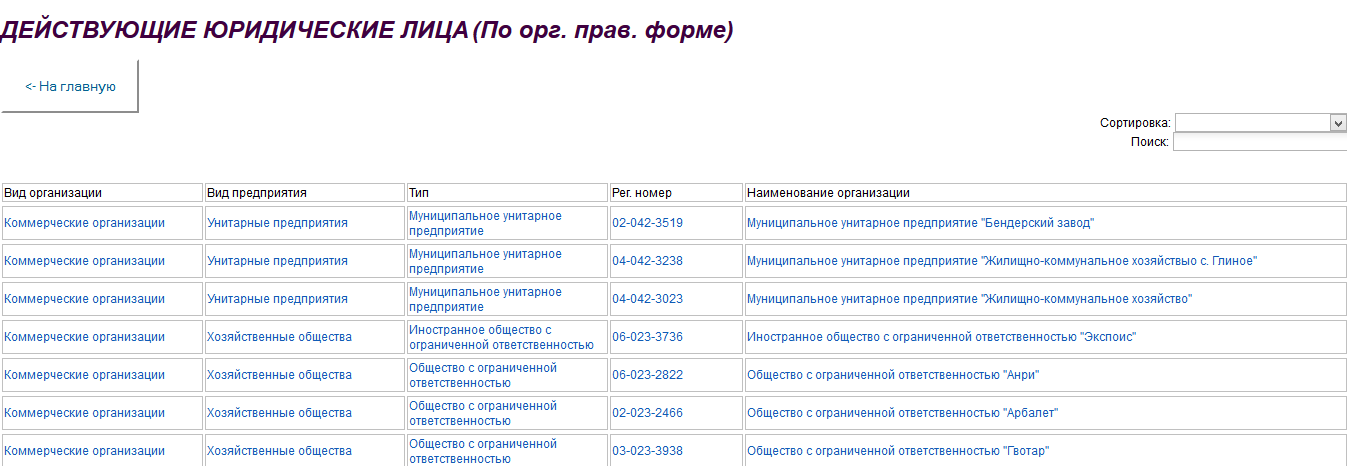


Рисунок 3.8 – Страница просмотра документов

**3.3 Методы и средства защиты программного продукта**

В программном продукте используются следующие средства защиты:

– аутентификация пользователей;

– авторизация пользователей;

– разграничение доступа.

*Аутентификация пользователей.* При входе в систему необходимо ввести имя пользователя и пароль. Эта информация отправляется на сервер, там производится поиск заданного имени пользователя в базе данных. В случае успеха проверяется совпадение введенного пароля, с паролем, указанным при регистрации пользователя. Если пароли совпали, то на клиентское приложение возвращается результат, который производит переход к авторизации пользователя.

*Авторизация пользователей.* Каждый пользователь принадлежит определенной группе пользователей: администраторы, клиенты с различными уровнями доступа, гости. После аутентификации клиентская программа получает параметр, в котором указана принадлежность данного пользователя к одной из вышеперечисленных групп. После чего программа сделает доступным только строго определенный набор возможностей.

*Разграничение доступа.* Информация о пользователе сохраняется на протяжении всего времени работы с приложением. При попытке пользователя произвести переход к какой-либо части программы, будет автоматически произведена проверка. Проверка заключается в поиске данного пользователя или группы, к которой принадлежит данный пользователь, в служебном списке доступа. На рисунке 3.9 представлен пример списка доступа к элементу веб-приложения:

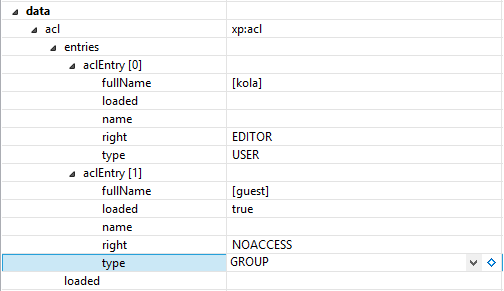


Рисунок 3.9 – *ACL* страницы *XPages*

**3.4 Тестирование и оценка надежности программного продукта**

В результате многократного тестирования в процессе разработки было выявлено, что ошибки, возникающие в ходе работы программы, можно разделить на группы. Частота появления этих ошибок относительно друг друга сильно варьируется, так как ошибки, возникающие вследствие сбоя аппаратной части или касаемые исходных данных, достаточно высоки, а практически полное отсутствие арифметических расчетов в программе снижает коэффициент проявления сбоев по этой причине. В таблице 3.1 приведены примерные коэффициенты возможности проявления ошибок различной степени, полученные в результате проведения 400 тестов:

Таблица 3.1 – Частота появления ошибок в ПО (% из 100)

|  |  |
| --- | --- |
| Тип ошибки | Частота появления,  % |
| Неполная или ошибочная спецификация | 3 |
| Отклонение от спецификации | 3 |
| Пренебрежение правилами программирования | 10 |
| Ошибочная выборка данных | 1 |
| Ошибочная последовательность операций | 10 |
| Ошибочные арифметические операции | 1 |
| Нехватка времени для решения | 13 |
| Ошибка при работе оборудования | 38 |
| Ошибка в исходных данных | 20 |
| Неточная запись | 1 |

При финальном тестировании программного продукта, было установлено, что надежность работы «удовлетворительная». Но для более точных результатов необходимо произвести нагрузочное тестирование и проверить работу продукта в реальных условиях при продолжительной работе.

3.5 Расчет себестоимости

Стоимость разработки программного продукта определяется по фактическим затратам, произведенным за счет собственных финансовых средств. Следовательно, необходимо подсчитать все затраты по различным критериям и получить конечную стоимость разработки.

В структуре капитальных вложений, связанных с автоматизацией управления, выделяют капитальные вложения на разработку проекта автоматизации (предпроизводственные затраты) и капитальные вложения на реализацию проекта (затраты на внедрение):

**  (3.1)

где *Кп* – капитальные вложения на проектирование;

*Кр* – капитальные вложения на реализацию проекта.

*Расчет капитальных вложений на проектирование*. Капитальные вложения на проектирование ПС определяются путем составления смет расходов и определяются по формуле:

 (3.2)

где *Км –* стоимость материалов;

*Кпр –* заработная плата основная и дополнительная с отчислениями в соцстрах инженерно-технического персонала, непосредственно занятого разработкой проекта;

*Кмаш –* затраты, связанные с использованием машинного времени на отладку программы;

*Кс –* оплата услуг сторонним организациям, если проектирование производится с привлечением сторонних организаций;

*Кн –* накладные расходы отдела проектирования.

Все расчеты будут производиться в условных единицах (у.е.), что соответствует стоимости одного доллара США в Приднестровском Республиканском Банке на момент разработки ПС.

*Затраты на материалы.* Определим смету затрат и рассчитаем стоимость материалов *Км*, пошедших на разработку ПС. Перечень материалов обусловлен темой дипломной работы. В их состав входит следующее: носители информации (бумага, магнитные диски) и быстроизнашивающиеся предметы труда (ручка, карандаш, резинка). Смета затрат на материалы представлена в таблице 3.2.

Таблица 3.2 – Смета затрат на материалы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Материал | Единица  измерения | | Цена за  единицу (у.е.) | Количество | Сумма (у.е.) |
| *CD-RW* диск | Шт. | | 0,5 | 1 | 0,5 |
| Бумага | Пач. | | 4,1 | 1 | 4,5 |
| Тонер картридж | Шт. | | 15,5 | 1 | 15,5 |
| Папка | Шт. | 0,25 | | 1 | 0,25 |
| ИТОГО |  | | | | 20,75 |

*Затраты на оплату труда*. Затраты на основную заработную плату проектировщика (*Кпр*) рассчитывается на основе данных о квалификационном составе разработчиков, их должностных окладах и общей занятости по теме. Дополнительная заработная плата начисляется в размере 10% от суммы основной заработной платы, а отчисления на социальные страхования – в размере 39% от фонда заработной платы. Затраты на оплату труда представлены в таблице 3.3.

Таблица 3.3 – Смета затрат на оплату труда

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Должность работника | Должностной оклад (у.е.) | Дневная ставка | Занятость  по теме | Сумма основной з/п (у.е.) |
| Программист | 221 | 10,03 | 115 | 1153,45 |
| Руководитель программного продукта | 255 | 11,05 | 75,5 | 834,28 |
| ИТОГО |  | | | 1987,7 |

Итого *Кпр =* 1987,7 у.е

*Затраты на отладку программы*. Затраты, связанные с использованием машинного времени на отладку программы (*Кмаш*) учитываются для следующих этапов проектирования: разработка рабочего проекта; внедрение - проведение опытной эксплуатации задач и сдача их в промышленную эксплуатацию.

Затраты на отладку программы определяются по формуле:

 (3.3)

где *Смч* – стоимость одного часа машинного времени;

*tотл* – время отладки программы (ч);

*Snp*– количество программистов.

Подставляя фактические данные, получаем величину затрат на отладку программы:

*Смч =* 0,5 у.е*., tотл =* 49 часов,

*Sпр =* 1 программист *Кмаш* = 0,5·49·1 =24,5 у.е.

Поскольку сторонние организации не привлекались к работе, то *Кс* = 0.

Накладные расходы на разработку дипломной работы берутся в размере 45% от основной заработной платы разработчиков для покрытия административно-хозяйственных и других непредусмотренных расходов:

 (3.4)

Так как затраты на основную заработную плату проектировщика (*Кпр*) равны 1987,7 у.е., то накладные расходы составят: *Кн =* 1987,7\*0,45 = 894,5 у.е.

Так как при реализации данной задачи не производилось специальных закупок техники и переустройства рабочих мест, капитальные вложения на реализацию задачи *Кр* равны нулю и общая величина капитальных вложений определяется затратами на предпроизводственные затраты. Общая величина капитальных вложений приведена в таблице 3.4.

Таблица 3.4 – Общая смета затрат на проектирование

|  |  |  |
| --- | --- | --- |
| Статьи | Затраты | |
| Сумма (у.е.) | Удельный вес статьи в общей стоимости (%) |
| Материалы и покупные полуфабрикаты | 20,75 | 0,56 |
| Основная заработная плата | 1987,7 | 54,21 |
| Дополнительная заработная плата | 198,5 | 5,41 |
| Отчисления на единый социальный налог | 540,50 | 14,70 |
| Затраты на отладку программы | 24,5 | 0,66 |
| Накладные расходы | 894,5 | 24,39 |
| ИТОГО: | 3666,45 | 100,00 |

Итого общая величина капитальных вложений на реализацию проекта составляет 3666,45 у.е.

К затратам текущего характера относятся затраты, связанные с обеспечением нормального функционирования разработанного программного средства.

Это могут быть затраты на ведение информационной базы, эксплуатацию технических средств, реализацию технологического процесса обработки информации по задачам, эксплуатацию системы в целом.

Затраты, связанные с эксплуатированием задачи вычисляются по формуле:

 (3.5)

где *Смч* – стоимость одного часа работы технических средств;

*Тэ* –время эксплуатации задачи в течение года.

Подставляя реальные значения, полученные в ходе опытной эксплуатации задачи, получаем величину годовых эксплуатационных расходов с учетом оплаты за расход электроэнергии компьютера в год:

*Сэз* = 0,17·2175 = 369,75 у.е.

*Определение экономической эффективности**от внедрения программы.* Экономический эффект, как реальная экономия, обусловлена следующими факторами: сокращением времени обработки информации; сокращением потерь рабочего времени.

Рассчитаем абсолютную годовую экономию на основе сокращения потерь рабочего времени, образующуюся в виде экономии на заработной плате за счет: снижение затрат на оплату простоев служащих; сокращение численности служащих; увеличение эффективности фонда времени одного служащего; сокращение сверхурочных работ.

Сокращения затрат при использовании программных средств для решения поставленной задачи обусловлено снижением трудоемкости работ по обработке информации и снижением затрат на оплату простоев сотрудников.

*Расчет экономии за счет снижения трудоемкости решения задачи.* Экономия за счет снижения трудоемкости решения определенного класса задач, рассчитывается по формуле:

 (3.6)

где *А* – коэффициент, учитывающий дополнительную заработную плату;

*В* – коэффициент, учитывающий отчисления на соцстрах;

*Тр* – трудоемкость решения задачи вручную (ч);

*Зчас* – среднечасовая тарифная ставка работника (у.е.);

*Кр* – коэффициент использования технических средств;

*Тоб* – трудоемкость при автоматизированной обработке (ч);

*Смч* – стоимость одного машинного часа работы (у.е.);

*Ue* – периодичность решения задачи (раз/год).

Подставляя реальные данные, полученные в результате исследований при ручном (полуавтоматизированном) и автоматизированном способах планирования деятельности предприятия, получаем величину экономии за счет снижения трудоемкости решения задачи при условии, что

*А* = 1,1; *В* = 1,27; *Тр* = 4 ч; *Зчас* = 1,45 у.е. (при основной заработной плате 150 у.е., 8-мичасовом рабочем дне, 22 рабочих дня в месяц);

*Кр* = 1,17; *Тоб* = 0,13 ч; *Смч* = 0,8 у.е.; *Ue* = 650 раз в год.

*Этр* = 4371,45 у.е.

*Определение годового экономического эффекта*. Основной экономический показатель, определяющий экономическую целесообразность затрат на создание программного продукта – это годовой экономический эффект, который определяется по формуле:

 (3.7)

где *Этр* – годовая экономия от применения внедренной задачи;

*Ен* – нормативный коэффициент экономической эффективности капитальных вложений (*Ен*= 0,15);

*Кп* – единовременные затраты, связанные с внедрением задачи;

Подставляя в формулу (3.7) реальные данные, определяем величину годового экономического эффекта при *Кп =* 80 у.е:

*Эс* = 3989,7 у.е.

*Расчет экономической эффективности.* Экономическая эффективность капитальных вложений, связанных с разработкой и внедрением программного продукта определяется по формуле:

(3.8)

Подставляя в формулу фактические данные, определяем величину экономической эффективности: *Ерс* = 49,87.

Так как *Ерс* > *Ен*, то внедрение экономически эффективно. Определяем срок окупаемости внедренной задачи:

*Те = Кп/Эс* = 0,02 года.

Расчеты показали, что использование данного программного продукта является экономически оправданным и ведет к сокращению потерь рабочего времени за счет уменьшения времени решения «вручную», что в свою очередь приводит к значительной экономии человеческих ресурсов и финансовых средств.

3.6 Охрана труда

При разработке программного продукта человек на протяжении всего рабочего дня вынужден взаимодействовать с различными техническими средствами, поэтому встает вопрос об обеспечении безопасности труда для здоровья рабочего.

Внедрение компьютерных технологий обработки информации способствовало совершенствованию организации и эффективности управленческого труда. Вместе с тем, являясь источником целого ряда неблагоприятных физических факторов воздействия на функциональное состояние и здоровье пользователей, компьютерная техника при неправильной эксплуатации и расстановке, особенно в неприспособленных для этого помещениях [9].

Наиболее важными возможными последствиями неблагоприятного воздействия на здоровье работников являются:

– заболевания глаз и зрительный дискомфорт;

– изменения костно-мышечной системы;

– нарушения, связанные со стрессом;

– кожные заболевания и др.

К числу факторов, ухудшающих состояние здоровья пользователей компьютерной техники, следует отнести:

– электромагнитные и электростатические поля;

– акустический шум;

– пониженная ионизация воздуха.

Немаловажную роль играют и состояние освещенности на рабочем месте, параметры мебели и характеристики помещения, где расположена компьютерная техника.

Соответствующие рабочие места запрещено обустраивать в подвальных или цокольных помещениях домов. В оборудовании помещений запрещается использование полимерных материалов, выделяющих вредные химические вещества. Также следует уделить внимание обеспечению достаточным для осуществления работы уровнем освещения (естественного и искусственного - в темное время суток) и звукоизоляции. Для регуляции уровня освещение естественным светом желательно применять жалюзи. Кроме того, в помещениях, где осуществляется работа с компьютерами, ежедневно должна производиться влажная уборка с целью недопущения запыленности пола и мебели [9].

Заземленные конструкции, находящиеся в помещениях, где размещены рабочие места операторов (батареи отопления, водопроводные трубы, кабели с заземленным открытым экраном), должны быть надежно защищены диэлектрическими щитками или сетками с целью недопущения попадания человека под напряжение [10].

Площадь, отведенная на одно рабочее место должна составлять не менее 6 кв. м., а объем - не менее 20 куб. м. Конструкция рабочего места должна обеспечивать поддержание оптимальной рабочей позы (то есть такой, которая позволяет работнику выполнять работу с минимальным напряжением тела, и которая позволяет избежать переутомления в ходе и после окончания рабочего процесса). Рациональная рабочая поза имеет важное значение для сохранения здоровья работника, поскольку длительное пребывание его в неудобной и напряженной позе может привести к таким заболеваниям, как сколиоз (искривление позвоночника), варикозное расширение вен, плоскостопие и т.д. Установлено, что работа в согнутом положении увеличивает затраты энергии на 20%, а при значительном наклонении - на 45% по сравнению с прямым положением корпуса [10].

При необходимости особой концентрации внимания во время выполнения работ смежные рабочие места операторов необходимо отделять друг от друга перегородками высотой 1,5 - 2 м.

Рабочие места следует располагать в отношении источника естественного света (окон) таким образом, чтобы свет падал сбоку, преимущественно слева. Также рабочее место должно отвечать современным требованиям эргономики:

– стол должен иметь высоту поверхности 680 - 800 мм, ширину 600 - 1400 мм и глубину 800 - 1000 мм;

– рабочий стул рабочий стул должен быть подъемно-поворотным, с возможностью регулировки высоты, желательно со стационарными или переменными подлокотниками и полумягкой нескользкой поверхностью сидения, которое легко чистится и не электризуется.

– экран компьютера должен находиться на оптимальном расстоянии от пользователя, которое составляет 600 - 700 мм, но не меньше 600 мм.

*Электробезопасность.* Лаборатория относится к помещениям повышенной опасности. Основной мерой защиты от поражения электрическим током являются: применение для облицовки современных электроизоляционных материалов; выполнение электропроводки закрытого типа с возможностью быстрого отключения на легкодоступном щите; обязательное заземление [10].

*Расчет выносного заземления*. Рассчитаем выносное заземляющее устройство. Преимуществом такого типа заземляющего устройства является возможность выбора места размещения электродов с наименьшим сопротивлением грунта (сырое, глинистое).

Сопротивление группового заземлителя рассчитывается, если:

– мощность установки менее 2 кВА;

– вертикальный заземлитель – прут диаметром 20 мм и длиной 2 м;

– горизонтальный заземлитель – полоса длиной 20 мм, толщиной 4 м;

– удельное сопротивление грунта (глина) 70 Ом\*м.

Сопротивление одиночного вертикального заземлителя рассчитывается по формуле:

, (3.9)



где  − удельное сопротивление грунта (Ом\*м);

*l* − длина вертикального заземлителя (м);

*d* − диаметр вертикального заземлителя (м);

*t* − глубина заложения.

 (3.10)

.

Расстояние между заземлителями (м):

 (3.11)

.

Ориентировочное количество вертикальных заземлителей (шт):

 (3.12)

где *Rзаз* – нормируемая величина сопротивления заземления (*Rзаз=4Ом*);



Количество вертикальных заземлителей определяется по формуле:

 (3.13)

где  – коэффициент использования вертикальных заземлителей (так как ориентировочное *n=*8 и *la=*6, поэтому *ηв*=0,85).



Длина горизонтального заземлителя (м):

 (3.14)

.

Сопротивление горизонтального заземлителя рассчитывается по формуле:

 (3.15)

где *b1* – ширина полосы (м)

.

Сопротивление группового заземлителя:

 (3.16)

где  – коэффициент использования горизонтальных заземлителей ()

, 

Рассчитанное заземление подходит для помещения, в котором проводилась реализация программного продукта, и обеспечит защиту персонала от поражения электрическим током в случае неисправности оборудования (при пробое на корпус).

*Пожарная безопасность.* Степень огнестойкости зданий принимается в зависимости от их назначения, категории по взрывопожарной и пожарной опасности, этажности, площади этажа в пределах пожарного отсека.

Здание, в котором находится помещение, по пожарной опасности строительных конструкций относится к категории *K1* (малопожароопасное), поскольку здесь присутствуют горючие вещества (книги, мебель, оргтехника и т.д.), которые при взаимодействии с огнем могут гореть без взрыва [11].

По конструктивным характеристикам здание можно отнести к зданиям с несущими и ограждающими конструкциями из естественных или искусственных каменных материалов, бетона или железобетона.

Следовательно, степень огнестойкости здания можно определить как третью (III).

Помещение по функциональной пожарной опасности относится к классу Ф1.3 − многоквартирные жилые дома.

Здание оборудовано пожарным водопроводом высокого давления с пожарными кранами.

Требования, предъявляемые к пожарной безопасности:

– выполнение скрытой электропроводки в стенах;

– устранение неисправных выключателей и розеток;

– запрет на использование оголенных проводов для соединения;

– необходимо иметь в доступном видимом месте углекислотные огнетушители (по характеру помещения – минимум 1 шт.).

*Причины возникновения пожара*. Пожар в помещении может привести к очень неблагоприятным последствиям (потеря ценной информации, порча имущества, гибель людей и т.д.), поэтому необходимо: выявить и устранить все причины возникновения пожара; разработать план мер по ликвидации пожара в помещении; план эвакуации людей из помещения [11].

Причинами возникновения пожара могут быть:

– неисправности электропроводки, розеток и выключателей которые могут привести к короткому замыканию или пробою изоляции;

– использование поврежденных (неисправных) электроприборов;

– использование в помещении электронагревательных приборов с открытыми нагревательными элементами;

– возникновение пожара вследствие попадания молнии в здание;

Профилактика пожара. Пожарная профилактика представляет собой комплекс организационных и технических мероприятий, направленных на обеспечение безопасности людей, на предотвращении пожара, ограничение его распространения, а также создание условий для успешного тушения пожара. Для профилактики пожара чрезвычайно важна правильная оценка пожароопасности здания, определение опасных факторов и обоснование способов и средств пожаропредупреждения и защиты [11].

В случае возникновения пожара необходимо отключить электропитание, вызвать по телефону пожарную команду, эвакуировать людей из помещения и приступить к ликвидации пожара. При наличии небольшого очага пламени можно воспользоваться подручными средствами с целью прекращения доступа воздуха к объекту возгорания.

**ЗАКЛЮЧЕНИЕ**

В настоящее время наблюдается высокая степень информатизации и компьютеризации предприятий и общества в целом. Для оперативного, гибкого и эффективного документооборота выделяются автоматизированные системы, ядром которых является база данных. При большом объеме информации и сложности, производимых с ней операций проблема эффективности средств организации хранения и передачи данных приобретает особое значение.

Для достижения поставленной цели и решения возникших задач была проделана следующая работа:

– изучена предметная область документооборота с применением методологии системного анализа;

– изучена и проанализирована учебная литература по теме ВКР;

– обоснован выбор проектных решений;

– реализовано веб-приложение, предназначенное для решения поставленных задач;

– составлена документация к программному продукту.

В результате выполнения выпускной квалификационной работы была разработана система получения данных из государственных реестров, находящихся в ведении огранизации на основе традиционных методов и подходов проектирования информационных систем, позволяющая снизить время поиска и чтения данных из баз данных *Lotus Notes*, и соответствующая предъявленным требованиям.

**ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ,**

**ЕДИНИЦ И ТЕРМИНОВ**

БД – база данных.

ВКР – выпускная квалификационная работа.

ГСС –служба.

ИС – информационная система.

ПО – программное обеспечение.

ПС – программная система.

СМИ – средство массовой информации.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Евсеев И. *Lotus Notes* для пользователя – М. : ООО «И.Д. Вильямс», 2002. – 854 с.

2. Ионцев Н. *Lotus Notes R.4*: разработка приложений. – М.: Издательство: «ИнтерТраст», 1996. – 567 с.

3. Монсон Ф. *Lotus Domino 7* разработка приложений. – СПб.: БХВ-Петербург, 2007. – 150 с.

4. Мэтью А. Расселл. ***Dojo*. Подробное руководство** – М.: Издательство: «Символ-Плюс», 2009. – 171 с.

5. Поляков Е. Краткий справочник по разработке *Web*-приложений на базе *Lotus Domino R5*. – М.: Издательство: «ИнтерТраст», 2000. – 146 с.

6. Прохоренок Н. *HTML, JavaScript, PHP* и *MySQL***. –** СПб.: БХВ-Петербург, 2010. – 900 с.

7. Райан П. Вопросы безопасности в *Lotus Notes* и *Domino 7*. – СПб.: БХВ-Петербург, 2007. – 256 с.

8. Рудаков А. В. Технология разработки программных продуктов. Учебное пособие. – М.: Издательство: «Академия», 2006. – 208 с.

9. СанПин 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы, 2007.

10. СНиП ПМР 31-20-02. Электротехнические устройства, 2002.

11. СНиП 21-01-03. Пожарная безопасность зданий и сооружений, 2003.

**12.** Стефанов С**. *JavaScript.* Шаблоны. –** М.: Издательство: «Символ-Плюс», 2011. – 272 с.

13. *Domino Designer 9.0.1 product documentation* [Электронный ресурс]. – 2015. – Режим доступа:

*www-10.lotus.com/ldd/ddwiki.nsf*

14. *Introducing XPages* [Электронный ресурс]. – 2012. – Режим доступа:

*http://www.xpages.info*

*15. NotesIn9 Episode Guide* [Электронный ресурс]. – 2014. – Режим доступа:

*http://www.notesin9.com/xpages-jumpstart/*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Приднестровский государственный университет им. Т.Г. Шевченко | | | | | | | | | |
| Инженерно-технический институт | | | | | | | | | |
| Инженерно-технический факультет | | | | | | | | | |
| Кафедра программного обеспечения вычислительной техники  и автоматизированных систем | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | | | |
|  |  |  |  |  |  |  | | | |
|  |  |  |  |  |  |  | | | |
|  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА** | | | | | | | | | |
| на соискание академической степени | | | | | | | | | |
| бакалавра | | | | | | | | | |
| по направлению 09.03.04 «Программная инженерия» | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | |
| **тема: «РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБРАБОТКИ РЕЕСТРОВ»** | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
| Руководство пользователя | | | | | | | | | |
|  | | | | | | | | | |
| Эксплуатационный документ | | | | | | | | | |
| в текстовом виде | | | | | | | | | |
| на \_7\_ листах | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Студент | |  |  |  |  |  | | |
|  | группы ИТ11ДР62ПИ1 | | |  |  |  | Иванович | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  | | |  |  |  |  | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| Тирасполь, 2015 | | | | | | | | | |

**СОДЕРЖАНИЕ**

|  |  |  |
| --- | --- | --- |
| ВВЕДЕНИЕ | | 53 |
| 1 РУКОВОДСТВО ПО УСТАНОВКЕ | | 54 |
|  | 1.1 Системные требования | 54 |
|  | 1.2 Установка программного продукта | 54 |
| 2 РУКОВОДСТВО ПО ПОЛЬЗОВАНИЮ | | 56 |
|  | 2.1 Запуск и работа с программным продуктом | 56 |

**ВВЕДЕНИЕ**

Данный программный продукт представлен в виде утилиты для базы данных *Lotus Notes,* предоставляющей доступ к документам посредством *web*-браузера. Программа предоставляет следующие функциональные возможности: выбор сортировки документов, поиск документов и отображение их в табличном виде, открытие документа для просмотра.

В данном руководстве будет описан порядок установки программного продукта, а также продемонстрирована его работа.

**1 РУКОВОДСТВО ПО УСТАНОВКЕ**

**1.1 Системные требования**

Минимальные системные требования:

– операционная система: *Windows XP*;

­– оперативная память: 1024 Мб;

– тактовая частота процессора: 1,8 Ггц;

– пространство на жестком диске: 2 Гб;

Рекомендуемые системные требования:

– операционная система: *Windows XP / Windows 7*;

– оперативная память: 2048 Мб;

– тактовая частота процессора: 2,4 Ггц;

– пространство на жестком диске: 10 Гб;

– манипулятор «мышь», клавиатура.

**1.2 Установка программного продукта**

Для начала работы с программным продуктом необходимо произвести его установку. Установка производится следующим образом: из тестовой базы данных в «реальную» импортируются новые объекты (представления, формы, страницы, классы и т.д.). Так как структура данных в обоих базах одинаковая, следовательно, проблем и побочных эффектов при работе программного продукта быть не должно (в противном случае были допущены ошибки при установке). Импорт объектов включает в себя следующие действия:

1) В раздел *Forms* необходимо перенести следующие формы: *EmptyForm*

2) В раздел *Views* необходимо перенести следующие представления: *Empty, По местонахождению (д2), По местонахождению (л2), По местонахождению (ф д2), По местонахождению (ф л2), По пакетам (г д2), По пакетам (г л2), По пакетам (д2), По пакетам (л2), По пакетам (ф д2), По пакетам (ф л2), По типу форм собственности (л2), Реорганизованные (2)*

3) В раздел *Xpages* необходимо перенести следующие страницы: *Главная, Д\_ГО\_Назв, Д\_ГО\_Пакет, Д\_ПМБ\_Назв, Д\_ФП\_Датаизм, Д-ФП\_место, Д\_ФП\_Назв, Д\_ФП\_Пакет, Д\_ЮЛ\_Датаизм, Д\_ЮЛ\_Датарег, Д\_ЮЛ\_Назв, Д\_ЮЛ\_Место, Д\_ЮЛ\_Оргформ, Д\_ЮЛ\_Пакет, ИИ\_ГО, ИИ\_О, ИИ\_ФП, ПД\_ГО\_Назв, ПД\_ГО\_Пакет, ПД\_Реорганиз, ПД\_СтадЛикв, ПД\_Уведом, ПД\_ФП\_Дата, ПД\_ФП\_Место, ПД\_ФП\_Назв, ПД\_ФП\_Пакет, ПД\_ЮЛ\_Дата, ПД\_ЮЛ\_Место, ПД\_ЮЛ\_Назв, ПД\_ЮЛ\_Оргформ, ПД\_ЮЛ\_Пакет.*

4) В раздел *Code/Java* необходимо перенести следующие классы: *pack1/linkSrv.java*

5) В классе *linkSrv* изменить содержимое строки *lnk:* указать *IP*-адрес сервера и название базы данных.

На этом установка программного продукта закончена.

**2 РУКОВОДСТВО ПО ПОЛЬЗОВАНИЮ**

**2.1 Запуск и работа с программным продуктом**

Для начала работы с программным продуктом пользователь должен открыть любой веб-браузер и ввести следующую адресную строку: [*http://www.IP\_адрес\_сервера/название\_БД/Главная.xsp*](http://www.IP_адрес_сервера/название_БД/Главная.xsp)

Если сервер работает, то на экране появится окно, в котором пользователь должен ввести имя пользователя и пароль (рисунок А.1). Если авторизация прошла успешно, то пользователь будет перенаправлен на главную страницу веб-приложения.

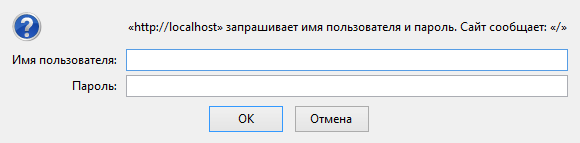


Рисунок А.1 – Окно авторизации

На главной странице есть элементы управления, с помощью которых пользователь сможет перейти в нужный раздел базы данных (рисунок А.2).

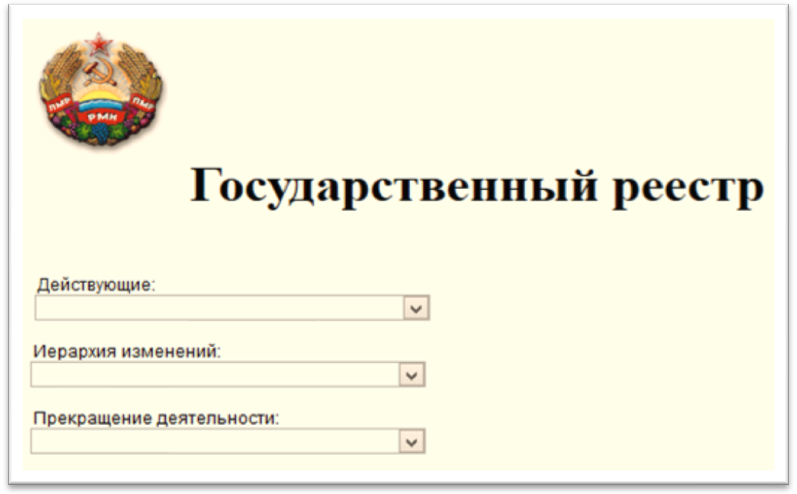


Рисунок А.2 – Главная страница

Далее пользователь перенаправляется на страницу просмотра документов. Для выборки документов необходимо ввести ключ в одно из полей поиска (быстрого поиска или поиска по полям документа) и нажать на кнопку «Поиск». В итоге, если по запрошенному ключу документы будут найдены, то они отобразятся в табличном виде (рисунок А.3).

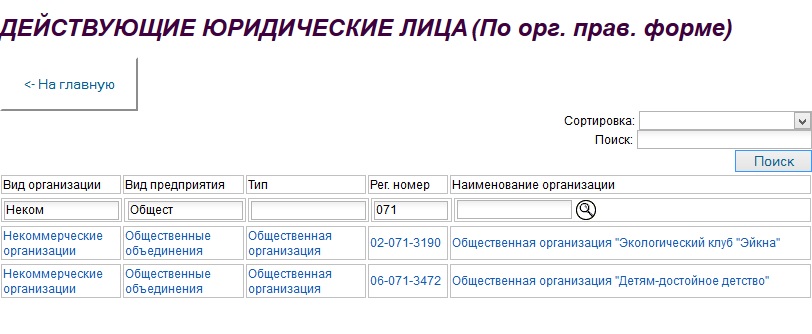


Рисунок А.3 – Поиск документов

Также если нажать кнопкой мыши на любую из записей, то можно будет открыть документ для просмотра (рисунок А.4).

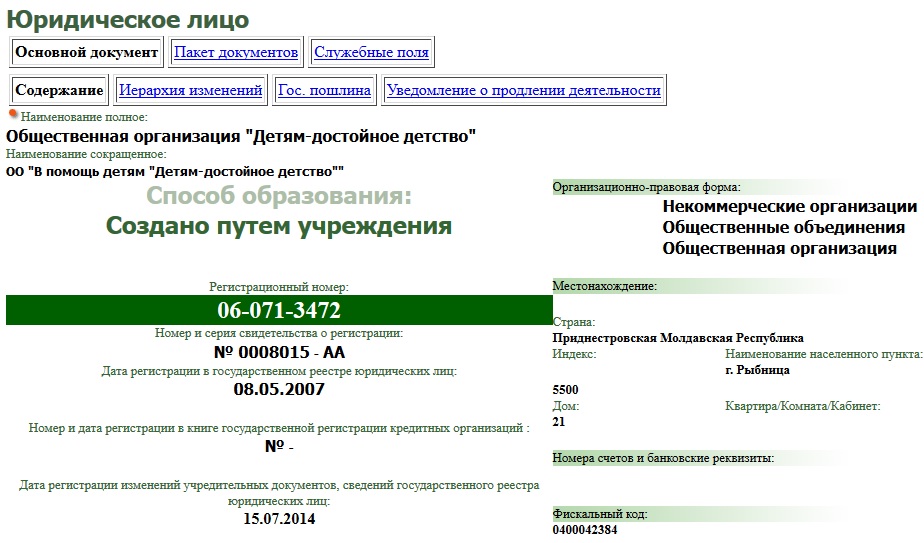


Рисунок А.4 – Просмотр документа

**ОТЗЫВ НАУЧНОГО РУКОВОДИТЕЛЯ**

**ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

**Студента** *Ивановича*

**Институт** *Инженерно-технический институт*

**Кафедра** *программного обеспечения вычислительной техники и автоматизированных систем*

**Группа** *ИТ11ДР62ПИ1*

**Направление**  *09.03.04 Программная инженерия*

**Квалификация (степень) выпускника** *бакалавр*

**Наименование темы:** *«Разработка программного обеспечения для обработки реестров»*

**Научный руководитель** *ст. преподаватель Добровольская Е.В. кафедра ПОВТиАС ИТИ*

**ПОКАЗАТЕЛИ ОЦЕНКИ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **№** | **Показатели оценки** | **Оценка** | | | |
| **5** | **4** | **3** | **0\*)** |
| Профессиональная | 1 | Оригинальность и новизна полученных результатов, научных,  конструкторских и технологических решений |  |  |  |  |
| 2 | Степень самостоятельного и творческого участия студента в работе |  |  |  |  |
| 3 | Уровень и корректность использования в работе методов  исследований, математического моделирования, инженерных расчетов |  |  |  |  |
| Справочно-  Информационная | 4 | Степень комплексности работы. Применение в ней знаний  естественнонаучных, социально-экономических,  обще-профессиональных и специальных дисциплин |  |  |  |  |
| 5 | Использование информационных ресурсов *Internet* |  |  |  |  |
| 6 | Использование современных пакетов компьютерных программ  и технологий |  |  |  |  |
| 7 | Наличие публикаций, участие в н.-т. конференциях,  награды за участие в конкурсах |  |  |  |  |
| Оформительская | 8 | Степень полноты обзора состояния вопроса |  |  |  |  |
| 9 | Ясность, четкость, последовательность и обоснованность изложения |  |  |  |  |
| 10 | Качество оформления пояснительной записки (общий уровень  грамотности, стиль изложения, качество иллюстраций, соответствие требованиям стандарта к этим документам) |  |  |  |  |
| 11 | Объем и качество выполнения графического материала,  его соответствие тексту записки и стандартам |  |  |  |  |
| **ИТОГОВАЯ ОЦЕНКА** | | |  | | | |

\*) не оценивается (трудно оценить)

**Отмеченные достоинства:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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

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

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

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

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

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

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

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

**Отмеченные недостатки:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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

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

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

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

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

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

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

**Заключение**:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_ г.

Научный руководитель / Добровольская Е.В. / \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Приднестровский государственный университет им. Т.Г. Шевченко

**ОТЗЫВ РЕЦЕНЗЕНТА**

**О ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ**

**Студента** *Ивановича*

**Институт** *Инженерно-технический институт*

**Кафедра** *программного обеспечения вычислительной техники и автоматизированных систем*

**Группа** *ИТ11ДР62ПИ1*

**Направление**  *09.03.04 программная инженерия*

**Квалификация (степень) выпускника** *бакалавр техники и технологии*

**Наименование темы:** *«Разработка программного обеспечения для обработки реестров»*

**Рецензент\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( Фамилия, И., О., место работы, должность, ученое звание, степень)

**ОЦЕНКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **№** | **Показатели оценки** | **Оценка** | | | | |
|  | **5** | **4** | **3** | **2** | **0\*** |
| Справочно-информационная | 1 | Соответствие представленного материала техническому заданию |  |  |  |  |  |
| 2 | Раскрытие актуальности тематики работы |  |  |  |  |  |
| 3 | Степень полноты обзора состояния вопроса |  |  |  |  |  |
| 4 | Уровень и корректность использования в работе методов исследований, математического моделирования, инженерных расчетов |  |  |  |  |  |
| 5 | Степень комплексности работы, применение в ней знаний  естественнонаучных, социально-экономических,  обще профес­сиональных и специальных дисциплин |  |  |  |  |  |
| 6 | Использование информационных ресурсов *Internet* |  |  |  |  |  |
| 7 | Использование современных пакетов компьютерных программ и  Технологий |  |  |  |  |  |
| Твор-ческая | 8 | Оригинальность и новизна полученных результатов, научных,  конструкторских и технологических решений |  |  |  |  |  |
| 9 | Ясность, четкость, последовательность и обоснованность изложения |  |  |  |  |  |
| Оформительская | 10 | Уровень оформления пояснительной записки: |  |  |  |  |  |
| * общий уровень грамотности |  |  |  |  |  |
| * стиль изложения |  |  |  |  |  |
| * качество иллюстраций |  |  |  |  |  |
| 11 | Объем и качество выполнения графического материала, его соответствие тексту записки |  |  |  |  |  |
| 12 | Соответствие требованиям стандарта оформления пояснительной  записки и графического материала |  |  |  |  |  |
| **ИТОГОВАЯ ОЦЕНКА** | | |  | | | | |

\*) не оценивается (трудно оценить)

**Отмеченные достоинства:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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

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

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

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

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

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

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

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

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

**Отмеченные недостатки:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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

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

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

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

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

**Заключение**:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Приднестровский государственный университет им. Т.Г. Шевченко

**ЛИСТ НОРМОКОНТРОЛЯ ВКР**

**Институт** *Инженерно-технический институт*

**Кафедра** *программного обеспечения вычислительной техники и автоматизированных систем*

**Группа** *ИТ11ДР62ПИ1*

**Направление**  *09.03.04 Программная инженерия*

**Квалификация (степень) выпускника** *бакалавр*

**Наименование темы:** *«Разработка программного обеспечения для обработки реестров»*

**Научный руководитель** *ст. преподаватель Добровольская Е.В. кафедра ПОВТиАС ИТИ*

Нормоконтроль проводится в два этапа:

1) после представления ВКР на предзащите (пояснительная записка к ВКР не переплетена);

2) после сдачи ВКР в установленные сроки на кафедру (пояснительная записка к ВКР переплетена).

Данный лист заполняется после второго этапа нормоконтроля и прикладывается к пояснительной записке к ВКР.

**Анализ ВКР на соответствие требованиям методических указаний**

|  |  |  |
| --- | --- | --- |
| **№**  **п/п** | **Объект** | **соответствует: +**  **не соответствует: -** |
| **1** | Соответствие темы работы приказу |  |
| **2** | Тип шрифта, размер шрифта |  |
| **3** | Междустрочный интервал |  |
| **4** | Абзац, выравнивание текста |  |
| **5** | Поля (мм) |  |
| **6** | Нумерация страниц |  |
| **7** | Общий объем без приложений |  |
| **8** | Объем введения |  |
| **9** | Объем основной части |  |
| **10** | Объем заключения |  |
| **11** | Соответствие оглавления содержанию |  |
| **12** | Последовательность структурных частей работы |  |
| **13** | Оформление структурных частей работы |  |
| **14** | Наличие перечня условных обозначений, символов, единиц  и терминов |  |
| **15** | Оформление списка использованной литературы |  |
| **16** | Наличие ссылок на использованную литературу в тексте |  |
| **17** | Оформление приложений |  |
| **18** | Оформление содержания |  |
| **19** | Качество оформления графического материала и таблиц |  |
| **20** | Наличие реферата к ВКР |  |

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

фамилия, имя, отчество подпись дата

С результатами нормоконтроля ознакомлен:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ФИО подпись

Приднестровский государственный университет им. Т.Г. Шевченко

**РЕФЕРАТ**

**К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ**

**Институт** *Инженерно-технический институт*

**Кафедра** *программного обеспечения вычислительной техники и автоматизированных систем*

**Группа** *ИТ11ДР62ПИ1*

**Направление**  *09.03.04 Программная инженерия*

**Квалификация (степень) выпускника** *бакалавр*

**Наименование темы:** *«Разработка программного обеспечения для обработки реестров»*

**Научный руководитель** *ст. преподаватель Добровольская Е.В. кафедра ПОВТиАС ИТИ*

Выпускная квалификационная работа содержит 57 с., 16 рис., 40 табл.,   
15 литературных источников, 1 приложение.

Ключевые слова: выборка, поиск, ключ, элемент управления, привязка, сортировка, база данных, представление, форма, страница, класс, веб-браузер.

Цель работы – разработка программного продукта для получения и обработки информации из реестров.

В процессе выполнения ВКР проводились работы по выбору и изучению средств реализации программного продукта; изучалась предметная область; разрабатывались алгоритмы основных функций.

В результате выполнения ВКР был получен отлаженный программный продукт с упрощенным пользовательским интерфейсом, который выполняет все задачи, предъявленные к нему на начальном этапе его жизненного цикла.

Область применения: предприятия малого и среднего бизнеса

Планируемое развитие работы: внедрение программного продукта

Студент \_­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ФИО подпись

Научный руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ФИО подпись

Приднестровский государственный университет им. Т.Г. Шевченко

**АННОТАЦИЯ**

**ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

**Студента** *Харичкова Николая Ивановича*

**Институт** *Инженерно-технический институт*

**Кафедра** *программного обеспечения вычислительной техники и автоматизированных систем*

**Группа** *ИТ11ДР62ПИ1*

**Направление**  *09.03.04* *Программная инженерия*

**Квалификация (степень) выпускника** *бакалавр*

**Наименование темы:** *«Разработка программного обеспечения для обработки реестров»*

**Научный руководитель** *ст. преподаватель Добровольская Е.В. кафедра ПОВТиАС ИТИ*

**ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

1. Вид выпускной квалификационной работы *бакалаврская работа*

2. Цель исследования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

3. Число литературных источников, использованных при составлении обзора\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Полное число литературных источников, использованных в работе\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. В том числе литературных источников:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Отечественных** | | | **Иностранных** | | |
| Последние 5 лет | От  5 до 10 лет | Более  10 лет | Последние  5 лет | От  5 до 10 лет | Более  10 лет |
|  |  |  |  |  |  |

6. Использование информационных ресурсов *Internet* (да, нет, число ссылок в списке литературы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

7. Использование современных пакетов компьютерных программ и технологий (указать какие именно, и в каком разделе работы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

8. Наличие публикаций и выступлений на конференциях по теме выпускной работы\_\_\_\_\_\_\_\_

(Да, нет)

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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

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

Выпускник / Харичков Н.И. / \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Научный руководитель / Добровольская Е.В. / \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

“\_\_\_\_\_\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.