**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

Факультет СПО (название факультета)

Направление подготовки (специальность) 09.02.07 Информационные системы и программирование

**О Т Ч Е Т**

**о производственной практике (по профилю специальности)**

**по ПМ.11 «Разработка, администрирование и защита баз данных»**

(наименование практики)

Тема задания: Разработка прототипа автоматизированной информационной системы для архива документов

Обучающийся \_\_\_\_\_\_Корсунов В.А.\_\_\_\_\_\_ Группа \_\_\_Y2433\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия И.О.) (номер группы)

Руководитель практики от организации: Ильяшенко Александр Сергеевич, начальник отдела НИОКР ООО «Центр реактивного программирования»

(Фамилия И.О., должность и место работы)

Руководитель практики от факультета: Говорова М.М., преподаватель  
Ефимова Т.Н., преподаватель

(Фамилия И.О., должность и место работы)

Ответственный за практику от университета: Королёв В.В., заместитель директора факультета СПО

(Фамилия И.О., должность)

Практика пройдена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подписи членов комиссии\_\_\_\_\_\_\_\_\_\_\_( )

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Говорова М.М.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Ефимова Т.Н.)

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2020

Содержание

1.1 Методологии проектирования системы 6

1.2 Архитектура системы 8

1.3 Обзор программных средств 10

1.4 Моделирование функционального поведения системы и базы данных 11

1.5 Программная реализация 12

3.1 Требования к функциональным характеристикам 28

3.2 Общие системные требования 28

3.3 Разграничения ответственности ролей при доступе 29

3.4 Описание входных и выходных данных 29

3.5 Моделирование разработки 30

3.6 Требования к технологиям разработки 31

3.7 Требования к лингвистическому обеспечению 31

3.8 Требования к защите информации от несанкционированного доступа 31

3.9 Требования к эргономике и технической эстетике 32

ПРИЛОЖЕНИЕ Б 35

введение

Целью практики является совершенствование навыков разработки информационных систем, повышение компетенции в сфере программирования.

Местом прохождения практики является ООО «Центр реактивного программирования».

Целью проекта является создание прототипа автоматизированной информационной системы (далее – АИС) для архива документов, которая будет предназначена для:

* сбора статистики добавленных архивистами данных;
* управления данными о документах, фондах и описях;
* хранения информации о внесенных документах.

Заказчик имеет потребность в создании ПО для ведения архивного делопроизводства.

Объектом рассмотрения являются документы, получаемые от клиентов архива. Когда клиенты обращаются в архив, они передают большое количество печатной/электронной документации, с целью ее организации, сортировки и дальнейшего хранения. Верхним уровнем организации является фонд, в котором хранятся описи, которые в свою очередь уже содержат документы, различаемые по типу, срокам хранения и другим параметрам.

Клиентская база архива увеличивается с большой скоростью, вследствие чего, с каждым днем становится все труднее вести учет всех данных в бумажном варианте. С целью ускорения процесса ведения архивной документации и оптимизации работы сотрудников архива будет разработана АИС для хранения всей информации об архивных единицах.

В процессе прохождения практики по профилю специальности поставлены следующие задачи:

1. Знакомство с особенностями разработки программного обеспечения.
2. Приобретение навыков изучения предметной области, поиска объектов, требований к ним, а также установление отношений между сущностями, в рамках проектируемой БД.
3. Изучение требований к оформлению и составлению документации.
4. Проектирование и реализация БД.
5. Разработка программного продукта с подключаемой базой данных.
6. Получение опыта защиты проекта.

Практическая значимость реализуемой системы заключается в том, что разрабатываемый программный продукт повысит эффективность выполнения основных операций, сотрудниками архива, а также позволит собирать необходимую статистику о проведенной работе.

1. проектирование и программная реализация
   1. Методологии проектирования системы

[Методология](https://studopedia.ru/9_174185_metodologiya.html)**проектирования** информационных систем описывает процесс создания и сопровождения систем в виде жизненного цикла (далее - ЖЦ) информационной системы (далее – ИС), представляя его как некоторую последовательность стадий и выполняемых на них процессов. Для каждого этапа определяются состав и последовательность выполняемых работ, получаемые результаты, методы и средства, необходимые для выполнения работ, роли и ответственность участников и т.д. Такое формальное описание ЖЦ ИС позволяет спланировать и организовать процесс коллективной разработки и обеспечить управление этим процессом.[12]

Цель использования методологии построения информационных систем заключается в регламентации процесса проектирования ИС и обеспечении управления этим процессом с тем, чтобы гарантировать выполнение требований, как к самой ИС, так и к характеристикам процесса разработки.

Внедрение методологии должно приводить к снижению сложности процесса создания ИС за счет полного и точного описания этого процесса, а также применения современных методов и технологий создания ИС на всем жизненном цикле ИС - от замысла до реализации.

Системы, которые осуществляют хранение и обработку информации, называют информационно-вычислительными системами. В информационную систему данные поступают от источника информации. Эти данные отправляются на хранение либо претерпевают в системе некоторую обработку и затем передаются потребителю.

Под проектированием ИС понимается процесс преобразования входной информации об объекте проектирования, о методах проектирования и об опыте проектирования объектов аналогичного назначения в соответствии с ГОСТом в проект ИС. С этой точки зрения проектирование ИС сводится к последовательной формализации проектных решений на различных стадиях жизненного цикла ИС: планирования и анализа требований, технического и рабочего проектирования, внедрения и эксплуатации ИС.

На этапе проектирования формируются модели данных. Проектировщики в качестве исходной информации получают результаты анализа требований к ИС. Построение логической и физической моделей данных является основной частью проектирования базы данных. Полученная в процессе анализа информационная модель сначала преобразуется в логическую, а затем в физическую модель данных. Параллельно с проектированием схемы базы данных выполняется проектирование процессов, чтобы получить спецификации (описания) всех модулей ИС. При проектировании модулей определяют интерфейсы программ: разметку меню, вид окон, горячие клавиши и связанные с ними вызовы. [12]

Для построения моделей информационных систем используется методология UML (Unified Modelling Language). В методологии UML функции системы представляются на диаграмме прецедентов (диаграммы вариантов использования, Use Case диаграммы), где изображены акторы и действия, которые они могут выполнять в системе.

Для построения модели поведения системы используется методология DFD (Data Flow Diagram). Главная цель DFD - показать, как каждая работа преобразует свои входные данные в выходные, а также выявить отношения между этими работами. При создании DFD используются понятия:

* Процесс (англ. Process), т.е. функция или последовательность действий, которые нужно предпринять, чтобы данные были обработаны. Это может быть создание заказа, регистрация клиента и т.д. В названиях процессов принято использовать глаголы, т.е. «Создать клиента» (а не «создание клиента») или «обработать заказ» (а не «проведение заказа»). Здесь нет строгой системы требований, как, например, в IDEF0 или BPMN, где нотации имеют жестко определенный синтаксис, так как они могут быть исполняемыми. Но все же определенных правил стоит придерживаться, чтобы не вносить путаницу при чтении DFD другими людьми.
* Внешние сущности (англ. External Entity). Это любые объекты, которые не входят в саму систему, но являются для нее источником информации либо получателями какой-либо информации из системы после обработки данных. Это может быть человек, внешняя система, какие-либо носители информации и хранилища данных.
* Хранилище данных (англ. Data store). Внутреннее хранилище данных для процессов в системе. Поступившие данные перед обработкой и результат после обработки, а также промежуточные значения должны где-то храниться. Это и есть базы данных, таблицы или любой другой вариант организации и хранения данных. Здесь будут храниться данные о клиентах, заявки клиентов, расходные накладные и любые другие данные, которые поступили в систему или являются результатом обработки процессов.
* Поток данных (англ. Data flow). В нотации отображается в виде стрелок, которые показывают, какая информация входит, а какая исходит из того или иного блока на диаграмме.
* Сущность (англ. Entity). Это абстрактный объект, имеющий в конкретном контексте независимое существование, а связь – это ассоциация сущностей. Атрибут – свойство сущности или связи.

Модель «сущность – связь» (англ. Entity – Relationship, ER) позволяет описать логическую структуру базы данных. Известны несколько нотаций ER-моделей, наиболее популярной является IDEF1. [13]

* 1. Архитектура системы

Для реализации Системы была выбрана архитектура «клиент-сервер».

Клиент-серверная архитектура реализует многопользовательский режим работы и является распределенной, когда клиенты и серверы располагаются на разных узлах локальной или глобальной вычислительной сети.

Существует несколько видов этой архитектуры:

* + - архитектура "файл-сервер";
    - двухуровневая архитектура "клиент-сервер";
    - трехуровневая архитектура "клиент-сервер";
    - многоуровневая архитектура "клиент-сервер".

Файл-серверная архитектура представляет наиболее простой случай распределенной обработки данных, согласно которой на сервере располагаются только файлы данных, а на клиентской части находятся приложения пользователей вместе с СУБД. Файл-сервер представляет собой достаточно мощную по производительности и оперативной памяти ПЭВМ, являющуюся центральным узлом локальной сети. Файл-сервер в среде сетевой операционной системы организует доступ к файлам, полностью эквивалентным файлам операционной системы и расположенным во внешней памяти файл-сервера.

Двухуровневая клиент-серверная архитектура основана на использовании только сервера базы данных (DB-сервера), когда клиентская часть содержит уровень представления данных, а на сервере находится база данных вместе с СУБД и прикладными программами.

Трехуровневая клиент серверная архитектура позволяет помещать прикладные программы на отдельные серверы приложений, с которыми через API-интерфейс (Application Program Interface) устанавливается связь клиентских рабочих станций. Работа клиентской части приложения сводится к вызову необходимых функций сервера приложения, которые называются «сервисами». Прикладные программы в свою очередь обращаются к серверу базы данных с помощью SQL запросов.

Многоуровневая архитектура «Клиент-сервер» создается для территориально-распределенных предприятий. Для нее в общем случае характерны отношения «многие ко многим» между клиентскими рабочими станциями и серверами приложений, между серверами приложений и серверами баз данных. [10]

* 1. Обзор программных средств

Существует множество CASE-средств предназначенных для проектирования и документирования информационных систем и баз данных.

Для создания DFD-диаграммы используется CASE-средство, предназначенное для моделирования, анализа, документирования и оптимизации бизнес-процессов с использованием трех стандартных методологий: IDEF0 (функциональное моделирование), DFD (моделирование потоков данных) и IDEF3 (моделирование потоков работ) - AllFusion ERwin Process Modeler.

Данное программное обеспечение имеет количество обучающей документации, а также возможность работы по уровням и декомпозиции каждого процесса.

Для построения логической модели базы данных использовался MySQl WorkBench. Это инструмент для визуального проектирования [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных [MySQL](https://ru.wikipedia.org/wiki/MySQL" \o "MySQL). Продукт позволяет создавать модели в нотации Мартина, а также базы данных на основании этих моделей.

Для создания диаграммы вариантов использования использовался Drawio, который предоставляет собой инструмент, позволяющий создавать блок-схемы, сетевые диаграммы, интеллект-карты, отношения сущностей, программные блоки, UML, макеты и т. д. В списке функционала имеется возможность отслеживать и восстанавливать изменения, импортировать и экспортировать файлы различных форматов, делиться своими работами, также автоматически публиковать их. [7]

Для проектирования системы были использованы следующие программные средства:

1. AllFusion ERwin Process Modeler.
2. MySQL WorkBench.
3. Drawio.
   1. Моделирование функционального поведения системы и базы данных

Для моделирования модели процессов работы системы "Архив" была использована DFD-диаграмма главного процесса. В приложении Б на рисунке Б.1 представлена функциональная модель использования АИС.

Модель поведения системы на уровне пользователей представлена в виде Use Case диаграммы в приложении Б на рисунке Б.4. В системе представлено два типа пользователей. Оба типа должны иметь логин и пароль для входа в систему.

Архивист может добавлять фонды, описи и документы в базу. Администратор имеет доступ к полному функционалу программы, а именно может: добавлять/изменять/удалять фонды, описи и документы, формировать отчеты по необходимым фильтрам, просматривать список добавленных документов за определенный период и добавлять новых пользователей. Все типы пользователей имеют интерфейс личного кабинета в похожем стиле, откуда могут выполнять доступные им операции.

На рисунке Б.3 в приложении Б приведена структура базы данных, которая представляет из себя логическую модель, не привязанную к конкретной СУБД. В ней выделяют основные объекты БД и определяют связи между этими объектами. Данная модель построена методом Сущность-связь (Entity Relationship) с использование средства MySQL Workbench.

* 1. Программная реализация

На основе сформированного технического задания разработано приложение для настольных компьютеров. Проведено детальное проектирование модулей системы и структур баз данных, после чего была проведена программная реализация, на основе средств, заявленных в техническом задании. Состав программы представлен в таблице 1.

Таблица 1 – Структура приложения

|  |  |  |
| --- | --- | --- |
| Файл | Назначение | Функции |
| 1 | 2 | 3 |
| LoginForm.cs | Интерфейс авторизации в системе. | Авторизует пользователя, исходя из введенных логина и пароля, либо перенаправляет его в окно регистрации. |
| Registartion.cs | Интерфейс регистрации в системе. | Регистрирует пользователя в системе (Предоставляемый уровень доступа – «Архивист») |
| AddData.cs | Интерфейс администратора системы, откуда он может управлять всеми хранимыми данными | Оперирование (изменение, добавление, удаление) всей информацией в базе данных. Позволяет перейти на страницы просмотра таблиц. |
| Inv\_tables.cs | Интерфейс таблицы описей. | Отображает хранимые в БД описи. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Inv\_search.cs | Интерфейс окна поиска по таблице описей. | Осуществляет поиск по хранимым в БД описям |
| Unit\_table.cs | Интерфейс таблицы документов. | Отображает хранимые в БД документы. |
| Unit\_search.cs | Интерфейс окна поиска по таблице документов. | Осуществляет поиск по хранимым в БД документам. |
| Fund\_table.cs | Интерфейс таблицы фондов. | Отображает хранимые в БД фонды. |
| Fund\_search.cs | Интерфейс окна поиска по таблице фондов. | Осуществляет поиск по хранимым в БД фондам. |
| Users\_table.cs | Интерфейс таблицы пользователей. | Отображает хранимые в БД учетные записи пользователей. |
| WorkerForm.cs | Интерфейс архивиста, в котором он может добавлять данные в систему. | Добавление фондов, описей и документов в базу данных. |
| App.config | Файл конфигурации, с указанием адреса сервера БД | Передача необходимых конфигураций различным элементам программы. |
| DB.cs | Файл подключения к базе данных | Подключение базы данных к приложению |
| Program.cs | Главная точка входа для приложения. | Начало выполнения программы. |

На рисунках 1-12 продемонстрированы скриншоты окон авторизации и регистрации, а также рабочие интерфейсы сотрудников двух статусов («Администратор» и «Архивист») со всеми доступными им функциями.

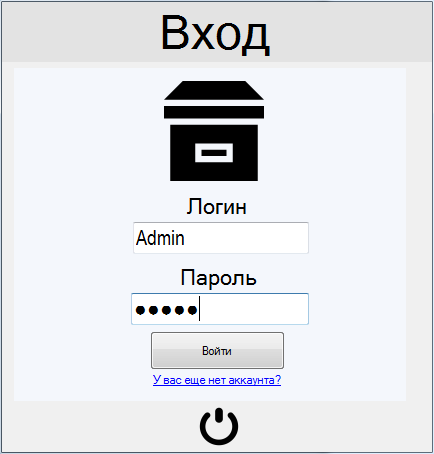


Рисунок 1 – Авторизация пользователя Admin

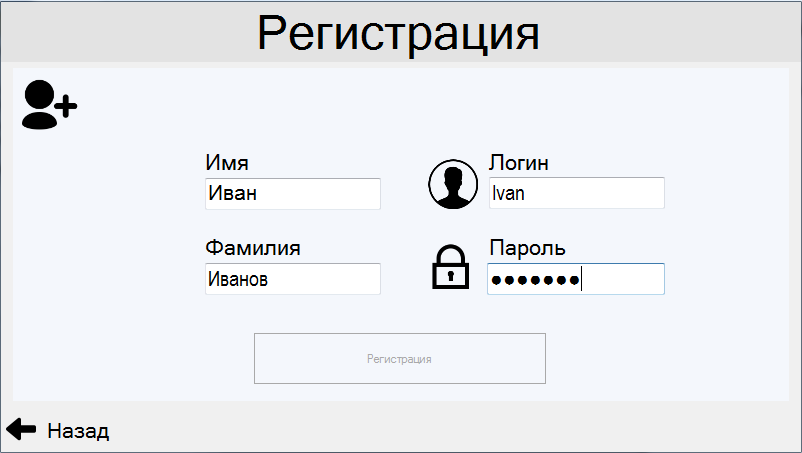


Рисунок 2 – Регистрация нового пользователя в системе

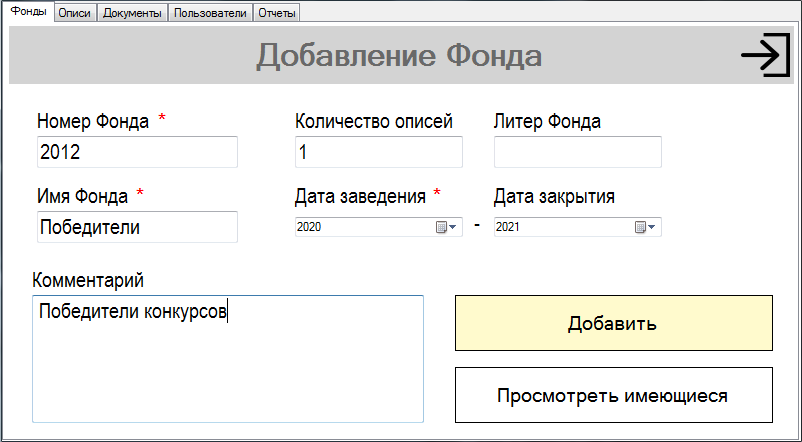


Рисунок 3 – Добавление нового фонда в базу (Администратор)

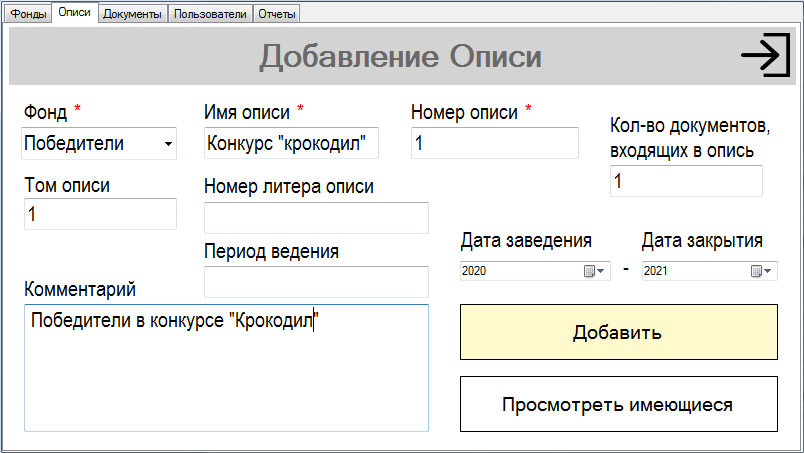


Рисунок 4 – Добавление новой описи в базу (Администратор)

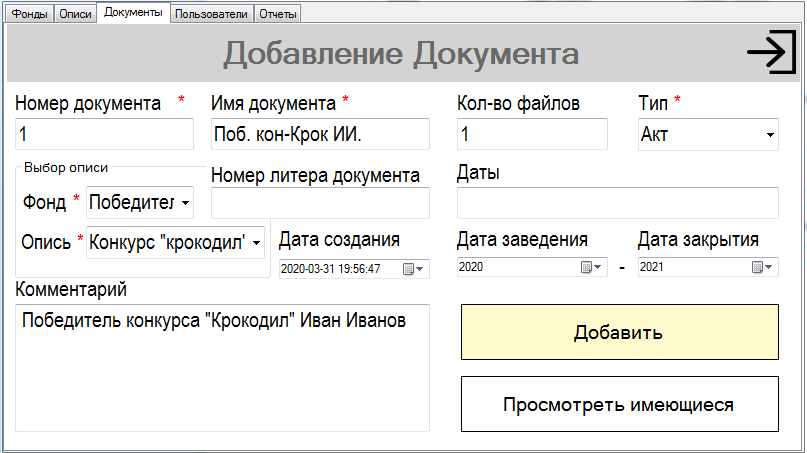


Рисунок 5 – Добавление нового документа в базу (Администратор)

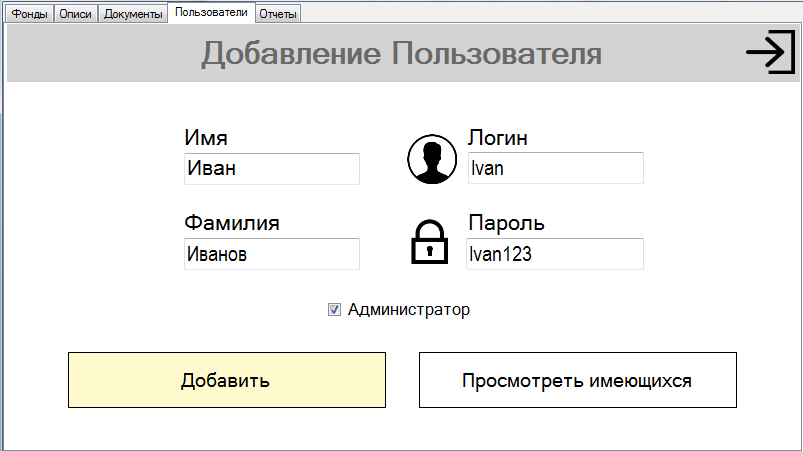
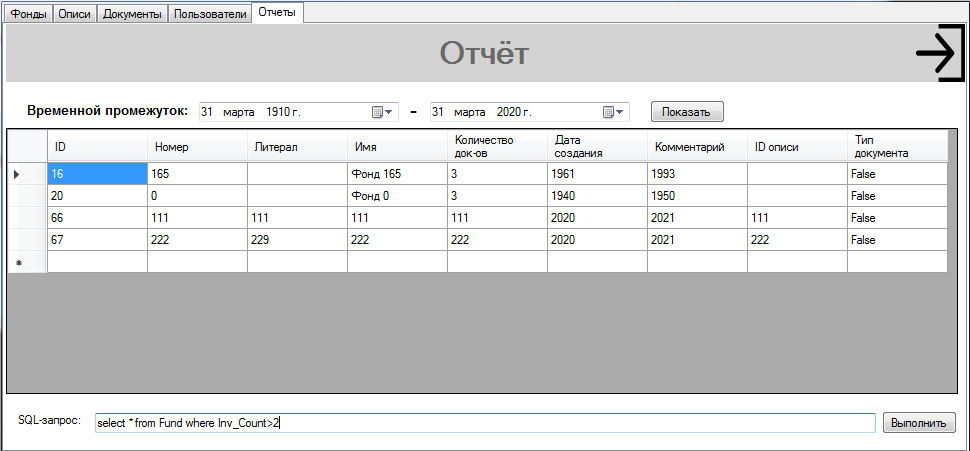


Рисунок 6 – Добавление нового пользователя в базу (Администратор)

Рисунок 7– Окно отчетности Администратор)

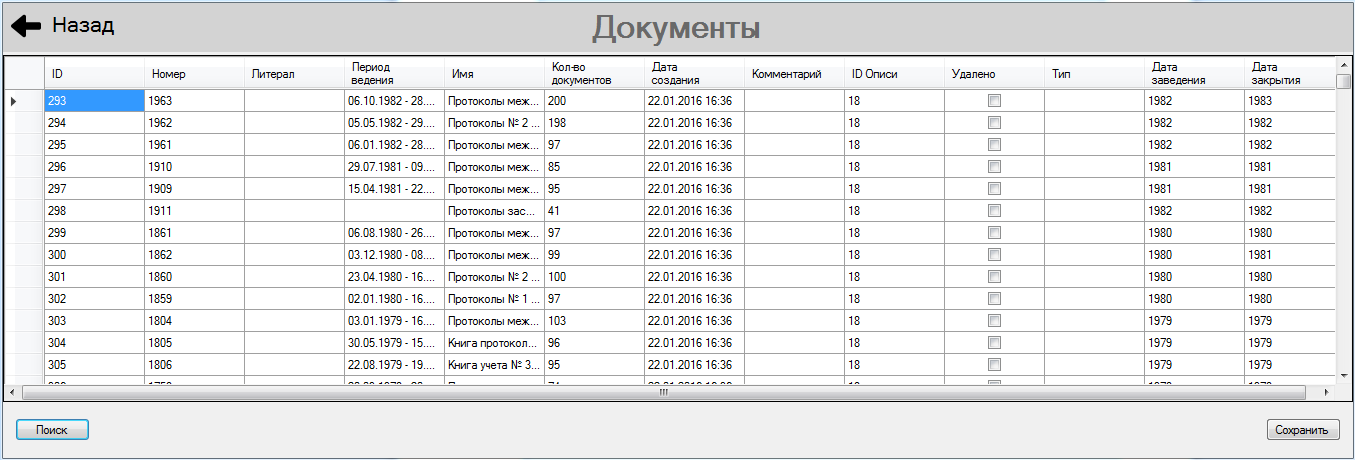


Рисунок 8 – Вывод данных, содержащихся в БД (Администратор)

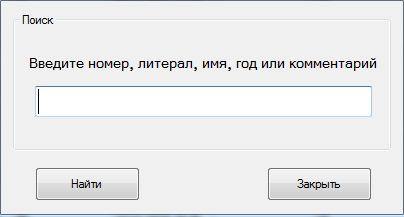


Рисунок 9 – Окно поиска по таблице базы (Администратор)

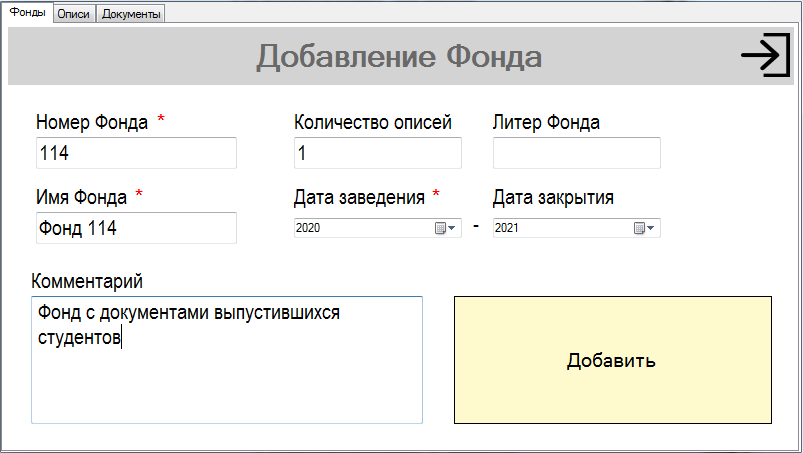


Рисунок 10 – Добавление нового фонда в базу (Архивист)

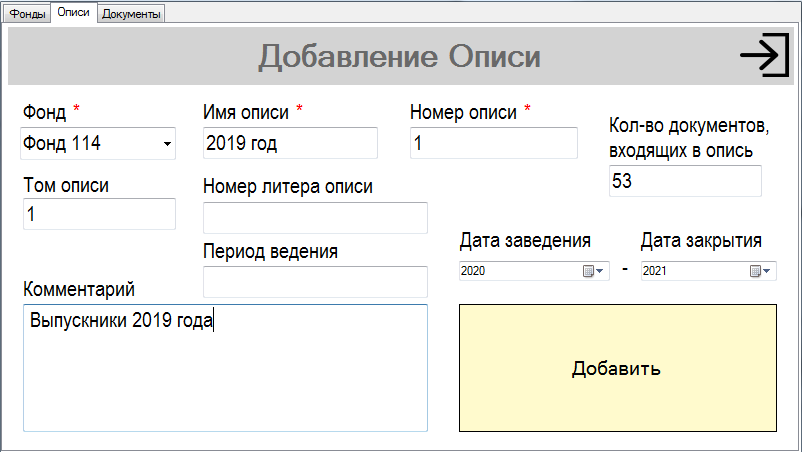


Рисунок 11 – Добавление новой описи в базу (Архивист)

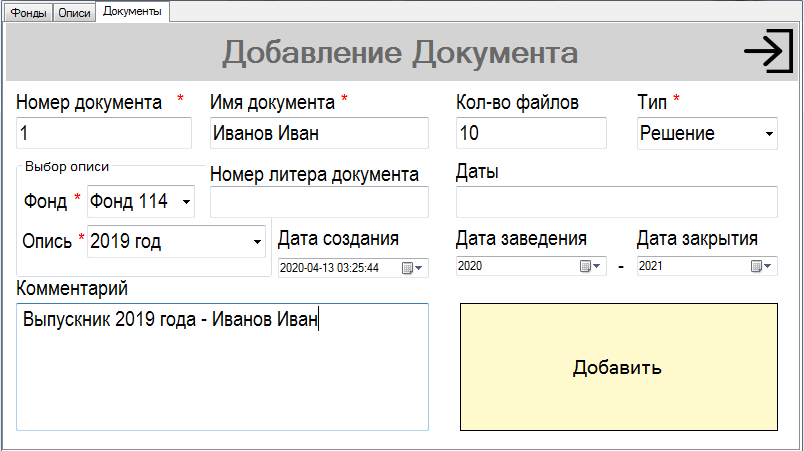


Рисунок 12 – Добавление нового документа в базу (Архивист)

заключение

В ходе выполнения производственной практики была спроектирована информационная система «Архив». На основе анализа предметной области разработан основной алгоритм работы разрабатываемого приложения. После чего была разработана база данных, а также настольное приложение для работы с ней. Приложение имеет простой и понятный графический интерфейс, позволяющий пользователю быстро освоиться в системе и начать работу в ней. С точки зрения обучения, разработка данного приложения помогла приобрести и укрепить навыки работы с базами данных в среде «СУБД MS SQL Server», а также навыки создания и использования «.NET» приложений.

Все задачи, поставленные на производственную практику, были выполнены.

Список источников

1. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. (Переиздание (Ноябрь 1987 г.) с Изменением № 1, утвержденным в июле 1981 г (ИУС 7-81)) [Электронный ресурс] // RUGOST1.0 - разработка документации по ГОСТ 34, 19, РД-50: [сайт]. URL: http://www.rugost.com/index.php?option=com\_content&view=article&id=54:19201-78&catid=19 (дата обращения: 25.03.2020).
2. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы [Электронный ресурс] // RUGOST1.0 - разработка документации по ГОСТ 34, 19, РД-50: [сайт]. URL. http://www.rugost.com/index.php?option=com\_content&view=article&id=96:gost-34602-89&catid=22&Itemid=53 (дата обращения: 25.03.2020)
3. Ульман Д.Д., Уидом Д. Реляционные базы данных. Лори, 2014. 384c.
4. Рудаков А.В., Федорова Г.Н. Технология разработки программных продуктов. Практикум. А.В.: Издательский центр «Академия», 2010.
5. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. Вильямс, 2017.
6. Основы работы с базами данных [Электронный ресурс] // Национальный открытый университет ИНТУИТ [сайт], 2003-2020. URL. <https://www.intuit.ru/studies/courses/93/93/info> (дата обращения: 28.03.2020).
7. Описание Draw.io // Startpack URL: https://startpack.ru/application/draw-io (дата обращения: 30.03.2020).
8. Работа с базой данных в C# // IT-BLACK Компьютер - это просто! URL: https://it-black.ru/rabota-s-bazoj-dannyh-v-ci\_sharp/ (дата обращения: 21.03.2020).
9. Выполнение команд и SqlCommand // METANIT.COM URL: https://metanit.com/sharp/adonet/2.5.php (дата обращения: 22.03.2020).
10. Проектирование ИС // ФПИ URL: http://fpi-kubagro.ru/proektirovanie-is/11/ (дата обращения: 22.03.2020).
11. О.Г. Инюшкина Проектирование информационных систем (на примере методов структурного системного анализа). Екатеринбург: Форт-Диалог Исеть, 2014.
12. ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ // Studme.org URL: https://studme.org/205620/informatika/tehnologii\_proektirovaniya\_informatsionnyh\_sistem#112 (дата обращения: 24.03.2020).
13. Что такое DFD (диаграммы потоков данных) // Trinion URL: https://www.trinion.org/blog/chto-takoe-dfd-diagrammy-potokov-dannyh (дата обращения: 21.03.2020).

ПРИЛОЖЕНИЕ А

Техническое задание

1. Назначение разработки

Назначением разработки является автоматизация документооборота и рабочих процессов (поиск данных, извлечение необходимых документов для передачи заказчику, статистический сбор информации) архива документов, при помощи внедрения автоматизированной информационной системы.

Целевой аудиторией являются люди, непосредственно взаимодействующие с архивом документов (архивисты), а также начальство, ведущее отчетную работу.

1. Цели и задачи

Цель – разработка информационной системы для автоматизации услуг документооборота и архивирования.

Задачи, решаемые в процессе достижения цели:

1. Уточнение функциональных требований (при необходимости).

2. Уточнение функциональной модели (при необходимости).

3. Реализация базы данных.

4. Детальное проектирование прикладных модулей.

5. Разработка и программная реализация алгоритмов функционирования модулей.

6. Тестирование модулей.

1. Требования к разработке
   1. Требования к функциональным характеристикам

Программа должна обеспечить:

* ввод, хранение, изменение, и обработку передаваемой клиентом информации;
* обеспечение разграничения прав доступа определённой категории пользователей, для поддержки соответствующего функционала.

В системе должны быть предусмотрены две категория пользователей:

* администратор;
* архивист.

При работе с системой администратору архива должны быть доступны следующие функции:

* добавление, редактирование и удаление данных из архива;
* формирование отчётов. Отчетами в данном процессе считаются:
* список внесенных в архив документов;
* вывод информации исходя из запроса, введенного администратором БД;
* Вывод информации по фильтрам;
* Поиск информации по словам или словосочетаниям.

При работе с системой оператору-архивисту должны быть доступны следующие функции:

* добавление данных в архив.
  1. Общие системные требования

Система должна быть реализована как настольное приложение, представляющее собой - АИС с программным интерфейсом и многопользовательским режимом работы.

* 1. Разграничения ответственности ролей при доступе

В системе должны быть определены роли пользователей менеджер, оператор. Распределение прав пользователей согласно ролям, в системе представлено в таблице 2.

Таблица 2. Распределение прав пользователей согласно ролям

|  |  |  |
| --- | --- | --- |
| Роль  Функции | Администратор | Архивист |
| 1 | 2 | 3 |
| Авторизация | + | + |
| Вывод хранимой в базе информации | + | - |
| Формирование отчетов | + | - |
| Редактирование/удаление информации в базе | + | - |
| Внесение информации о новых фондах, описях и документах | + | + |
| Добавление новых пользователей | + | - |

* 1. Описание входных и выходных данных

Входными данными является информация о документах, предназначенных для хранения в архиве, и фильтры для формирования выборки данных и отчётов.

Выходными данными является информация, хранимая в БД, и полученная на основании запросов пользователей или сформированная в виде отчетов.

Структура входных и выходных данных представлена в таблице 3.

|  |  |  |
| --- | --- | --- |
| Таблица 3. Входные и выходные данные программных модулей АИС для формирования отчетов | | |
| Функция | Входные данные | Выходные данные |
| Авторизация | Логин/пароль | Пользовательские полномочия для проведения последующих действий (редактирование/добавление/удаление данных) |
| Регистрация документов | Сведения о документе (номер, тип, размер, дата добавления, время хранения и т.д.) | Данные о документах в базе |
| Вывод информации по фильтру | Фильтр для сортировки хранимой информации | Отсортированная информация |
| Формирование отчётов (по срокам хранения, по времени добавления, по количеству документов) | Фильтр запроса для составления отчета/акта | Отчет по срокам хранения, отчет по времени добавления документов, отчет по количеству документов |

* 1. Моделирование разработки
     1. Функциональная модель

Система должна удовлетворять функциональной модели, представленной в приложении Б, на рисунках Б.1 – Б.2.

Для представления поведения системы и всех возможных вариантов действий была использована диаграмма вариантов использования (UML Use Case). В диаграмме вариантов использования выделены следующие актёры:

* Администратор;
* Архивист.

Роли пользователей в системе должны удовлетворять модели, представленной на рисунке Б.4 в приложении Б.

* + 1. Модель данных

Модель данных должна удовлетворять реляционной модели, представленной в приложении Б и на рисунке Б.3.

* 1. Требования к технологиям разработки

Разрабатываемое ПО должно быть многопользовательской АИС с графическим интерфейсом.

Для разработки ПО и графического интерфейса должна быть использована среда Visual Studio. Для разработки информационного обеспечения должна использоваться технология баз данных с интерфейсом управления MySQL Workbench.

Для разработки должны использоваться следующие программные средства:

* СУБД Microsoft SQL Server (не ниже версии 11.0.3128.0);
* программная платформа Microsoft .NET Framework.
  1. Требования к лингвистическому обеспечению

При реализации системы должны применяться следующие языки высокого уровня: SQL, С#. Также для реализации графического интерфейса должна быть использована платформа .NET Framework.

Ресурс должен быть реализован на русском языке.

* 1. Требования к защите информации от несанкционированного доступа

Система должна поддерживать разграничение прав доступа с возможностью формирования групп пользователей и присвоение группе и каждому пользователю определенных полномочий на доступ к данным системы.

Обеспечение информационной безопасности подсистемы должно удовлетворять следующим требованиям:

* защита от несанкционированного доступа (хранение информации в базе данных);
* обеспечение разделения прав доступа между пользователями.
  1. Требования к эргономике и технической эстетике

Взаимодействие пользователей с Системой должно осуществляться посредством приложения с графическим интерфейсом. Интерфейс должен обеспечивать удобный доступ к основным функциям и операциям. Разрабатываемая АИС должна функционировать на русском языке.

Страницы пользовательского интерфейса должны проектироваться с учетом требований унификации:

* страницы должны быть выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;
* в разделах интерфейса для обозначения сходных операций должны использоваться сходные графические значки, кнопки и т.п. управляющие (навигационные) элементы;
* внешнее поведение сходных элементов интерфейса (реакция на наведение указателя «мыши», переключение фокуса, нажатие кнопки и т.п.) должны реализовываться одинаково для однотипных элементов.

Страница администратора включает:

* интерфейс добавления, просмотра и редактирования описей;
* интерфейс добавления, просмотра и редактирования фондов;
* интерфейс добавления, просмотра и редактирования документов;
* интерфейс добавления, просмотра и редактирования пользователей;
* интерфейс просмотра данных по указанным фильтрам;
* интерфейс поиска информации в базе.

Страница пользователя категории Архивист включает интерфейс добавления фондов, описей и документов.

1. Методы тестирования

Тестирование должно осуществляться на модульном и интеграционном уровнях.

На этапе разработки программного обеспечения должна применяться технология тестирования "белого ящика" с использованием следующих видов тестирования:

* переходов между состояниями.

ПРИЛОЖЕНИЕ Б

**Модель разработки**

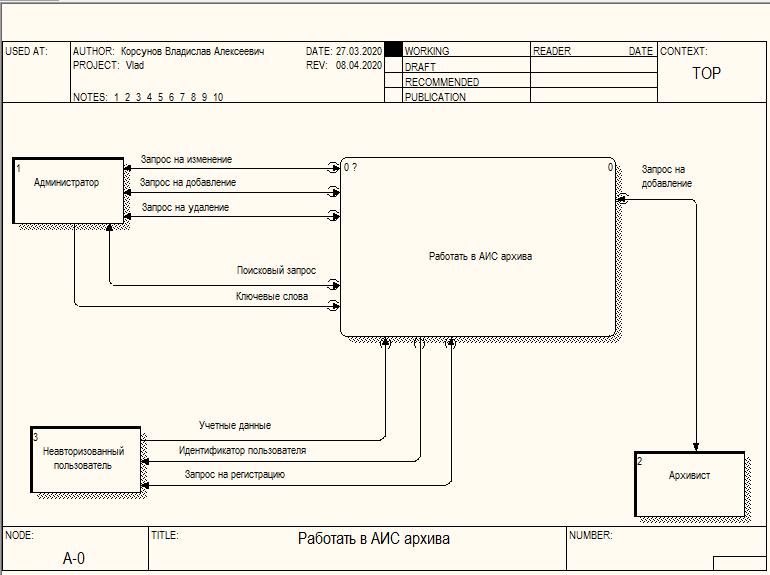


Рисунок Б.1 – Функциональная модель DFD: Работать в АИС, уровень А-0

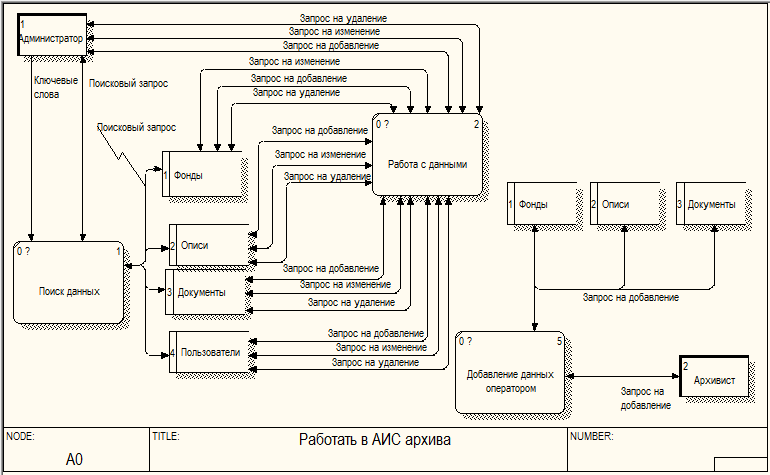


Рисунок Б.2 Функциональная модель DFD: детализация главного процесса, уровень А0

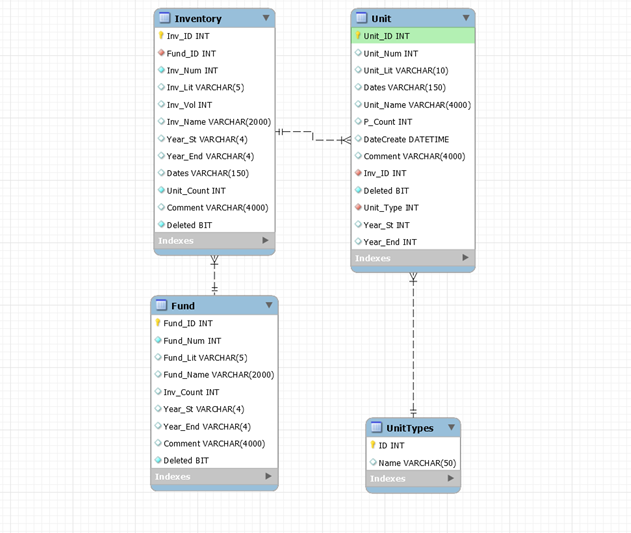


Рисунок Б.3 - Структура базы данных в нотации Мартина

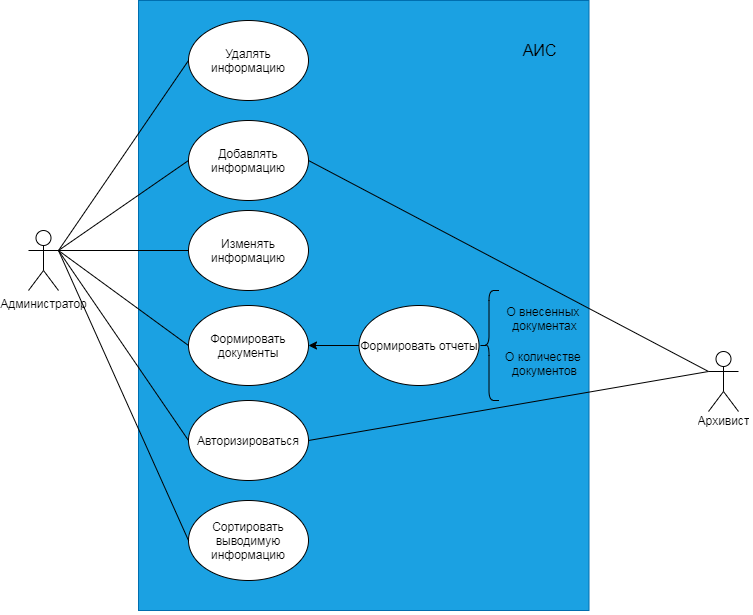


Рисунок Б.4 - Use case диаграмма, Акторы: Администратор, Архивист

Таблица Б1 **–** Фонды (Fund)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснение |
| Fund\_ID | INT | + | + | - | UNIQUE INDEX | ID фонда |
| Fund\_Num | INT | + | - | - |  | Номер фонда |
| Fund\_Lit | VARCHAR(5) | - | - | - |  | Литер фонда |
| Fund\_Name | VARCHAR(2000) | + | - | - | UNIQUE INDEX | Имя фонда |
| Inv\_Count | INT | - | - | - |  | Количество описей, которые в него входят |
| Year\_St | VARCHAR(4) | + | - | - |  | Дата заведения |
| Year\_End | VARCHAR(4) | - | - | - |  | Дата закрытия |
| Comment | VARCHAR(4000) | - | - | - |  | Комментарий |
| Deleted | BIT | - | - | - |  | Флаг удаления |

Таблица Б2. Описи (Inventory)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснение |
| Inv\_ID | INT | + | + | - | UNIQUE INDEX | ID описи |
| Fund\_ID | INT | + | - | + | UNIQUE INDEX | ID фонда |
| Inv\_Num | INT | + | - | - |  | Номер описи |
| Inv\_Lit | VARCHAR(5) | - | - | - |  | Номер литера описи |
| Inv\_Vol | INT | - | - | - |  | Том описи |
| Inv\_Name | VARCHAR(2000) | + | - | - |  | Имя описи |
| Year\_St | VARCHAR(4) | - | - | - |  | Дата заведения |
| Year\_End | VARCHAR(4) | - | - | - |  | Дата закрытия |
| Dates | VARCHAR(150) | - | - | - |  | Период ведения |
| Unit\_Count | INT | - | - | - |  | Количество документов, входящих в опись |
| Comment | VARCHAR(4000) | - | - | - |  | Комментарий |
| Deleted | BIT | - | - | - |  | Флаг удаления |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснение |
| Unit\_ID | INT | + | + | - | UNIQUE INDEX | ID Документа |
| Unit\_Num | INT | - | - | - |  | Номер документа |
| Unit\_Lit | VARCHAR(10) | - | - | - |  | Литер документа |
| Dates | VARCHAR(150) | - | - | - |  | Даты ведения |
| Unit\_Name | VARCHAR(4000) | - | - | - |  | Имя документа |
| P\_Count | INT | - | - | - |  | Количество файлов, входящих в документ |
| DateCreate | DATETIME | - | - | - |  | Дата создания |
| Comment | VARCHAR(MAX) | - | - | - |  | Комментарий |
| Inv\_ID | INT | + | - | + | UNIQUE INDEX | ID описи, в которой состоит документ |
| Deleted | BIT | - | - | - |  | Флаг удаления |
| Unit\_Type | INT | + | - | + | UNIQUE INDEX | Тип документа |
| Year\_St | VARCHAR(4) | - | - | - |  | Дата создания |
| Year\_End | VARCHAR(4) | - | - | - |  | Дата закрытия |

Таблица Б3. Документы (Units)

Таблица Б4. Типы документов (UnitTypes)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснение |
| ID | INT | + | + | - | UNIQUE INDEX | ID типа |
| Name | VARCHAR(50) | + | - | - | UNIQUE INDEX | Имя типа |