

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Томский государственный университет

Факультет информатики

Кафедра теоретических основ информатики

ДОПУСТИТЬ К ЗАЩИТЕ В ГАК

Зав. каф. ТОИ проф., доктор. техн. наук

\_\_\_\_\_Ю.Л. Костюк

«\_\_\_»\_\_\_\_\_2006г.

УДК 002.53:004.65

Смаль Юлия Александровна

РАЗРАБОТКА ЭЛЕКТРОННОЙ ONLINE-БИБЛИОТЕКИ

Дипломная работа

Научный руководитель:  
ст. преп. каф. ТОИ, к.т.н.

В. А. Лавров

Исполнитель:  
студ. гр. 1411

Ю. А. Смаль

Электронная версия дипломной работы помещена  
в электронную библиотеку. Файл

Администратор

Томск – 2006

## Реферат

Сведения об отчете: дипломная работа, 57 с., 16 рис., 4 табл., 13 источников, 3 приложения.

Перечень ключевых слов: ЭЛЕКТРОННАЯ БИБЛИОТЕКА, ПОЛНОТЕКСТОВЫЙ ПОИСК, ИНДЕКСИРОВАНИЕ, ПОЛНОТЕКСТОВЫЙ КАТАЛОГ, СУБД MS SQL SERVER 2000, ASP.NET, WEB-ПРИЛОЖЕНИЕ.

Объект исследования – электронные библиотеки.

Цель работы – обзор технологий создания электронных библиотек, разработка и реализация системы «Томская Электронная Библиотека».

Метод исследования – теоретический и практический, анализ информации об электронных библиотеках и его применение в создании собственной библиотеки.

Полученные результаты – разработана и реализована система «Томская Электронная Библиотека», ориентированная на пользователей Томского Интернета.

Внедрение – система «Томская Электронная Библиотека» поступила в тестовое использование.

# Содержание

<b>РЕФЕРАТ .....</b>	<b>2</b>
<b>СОДЕРЖАНИЕ.....</b>	<b>3</b>
<b>ВВЕДЕНИЕ .....</b>	<b>5</b>
<b>1. ЭЛЕКТРОННЫЕ БИБЛИОТЕКИ И ПОЛНОТЕКСТОВЫЕ БАЗЫ ДАННЫХ.....</b>	<b>6</b>
1.1. От традиционной библиотеки к электронной.....	6
1.2. Технологии обработки документов и организация информационного поиска..	6
1.3. Поисковые системы.....	8
1.3.1. Виды информационного поиска.....	9
1.3.2. Основные характеристики поисковых систем.....	9
1.3.3. Принципы работы поисковой машины Рамблер.....	10
1.3.4. Принципы работы поисковой программы Google Desktop Search ...	11
1.3.5. Сервис MS Search.....	11
1.4. Классификация электронных библиотек.....	12
1.5. Варианты построения электронного архива .....	14
1.6. Классификация и библиографическое описание книг .....	15
1.7. Анализ электронных библиотек.....	16
1.8. Выводы.....	18
<b>2. БАЗОВАЯ АРХИТЕКТУРА РАЗРАБОТАННОЙ ЭЛЕКТРОННОЙ БИБЛИОТЕКИ .....</b>	<b>19</b>
2.1. Использование ASP.NET для разработки WEB-приложений .....	19
2.1.1. Общие идеи ASP.net.....	19
2.1.2. Сравнение ASP.net и PHP.....	20
2.2. Использование СУБД MS SQL SERVER 2000.....	21
2.2.1. Особенности работы MS SQL Server 2000.....	21
2.2.2. Варианты хранения и загрузка текста в таблицы БД.....	22
2.2.3. Полнотекстовый индекс и полнотекстовый каталог .....	23
2.2.4. Этапы создания полнотекстового индекса .....	23
2.3. Структура Томской Электронной Библиотеки .....	25
2.4. Выводы.....	27
<b>3. РЕАЛИЗАЦИЯ .....</b>	<b>28</b>
3.1. Проектирование базы данных библиотеки .....	28
3.1.1. Определение множеств сущностей и атрибутов.....	28
3.1.2. Определение множеств связей.....	30
3.1.3. Определение ограничений целостности .....	32
3.1.4. Построение ER-диаграммы.....	32
3.1.5. Трансформация ER-модели в реляционную модель .....	32
3.2. Основные функциональные подсистемы ТЭБ.....	34
3.2.1. Аутентификация/регистрация .....	34
3.2.2. Добавление/удаление книг.....	34
3.2.3. Модификация поискового образа книги .....	35
3.2.4. Поиск книг .....	36
3.2.5. Скачивание книг .....	37
3.2.6. Представление данных о web-ресурсе .....	37

3.2.7. Модификация категорий из упрощенной классификации .....	37
3.3. Выводы.....	38
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>39</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....</b>	<b>40</b>
<b>ПРИЛОЖЕНИЕ А. РУКОВОДСТВО ПРОГРАММИСТА. ....</b>	<b>41</b>
<b>ПРИЛОЖЕНИЕ Б. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ .....</b>	<b>51</b>
<b>ПРИЛОЖЕНИЕ В. РУКОВОДСТВО АДМИНИСТРАТОРА .....</b>	<b>57</b>

## Введение

До недавнего времени библиотека, как социальный институт, была единственным учреждением, профессионально занимающимся сбором, систематизацией, хранением и доведением информации до пользователей (читателей). Эта исторически сложившаяся монополия на владение информацией гарантировала традиционной библиотеке прочное место в системе информационных коммуникаций: ни в каком другом месте потенциальный читатель не мог получить доступа к профессионально обработанным и тщательно сохраняемым информационным массивам.

Скачкообразное развитие Интернета в последние годы привело к информационному взрыву — явлению, хорошо знакомому не только специалистам, но и простым пользователям всемирной Сети. Формирующаяся на глазах нынешнего поколения система электронных информационных коммуникаций кардинально меняет ситуацию в сфере сбора, хранения и обработки данных. В обобщенном виде такие подходы к стандартизации информации сегодня трактуются как *создание «цифровых» или «электронных» библиотек и полнотекстовых баз данных* [1].

В общем случае под электронной библиотекой или полнотекстовой базой данных понимается распределенная информационная система, позволяющая накапливать, хранить и использовать разнообразные коллекции электронных документов. Сегодня на смену информационному обслуживанию на печатных носителях пришло обеспечение пользователей на основе доступа к электронным документам через глобальную сеть передачи данных - Интернет.

Создание электронных библиотек и полнотекстовых баз данных ведётся по всему миру, но прежде всего на Западе. Россия не сильно отстала от этого процесса: если в США и Западной Европе этими проблемами начали заниматься в 1992—1994 гг., то в России первые проекты появились уже в 1996—1999 гг. [1]. В качестве примеров существующих электронных библиотек в России можно привести: библиотеку художественной литературы Максима Мошкова ([www.lib.ru](http://www.lib.ru)), студенческую библиотеку ([www.lib.students.ru](http://www.lib.students.ru)), электронную библиотеку на сайте Эдварда Радзинского ([www.radzinski.ru](http://www.radzinski.ru)), библиотеку детской литературы «Сказки» ([www.skazki.com.ru](http://www.skazki.com.ru)) и многие другие.

На сегодняшний день, не выходя за пределы Томского Интернета, можно воспользоваться услугами следующих электронных библиотек: Томской технической библиотеки ([www.techlib.tomsk.ru](http://www.techlib.tomsk.ru)) и зеркалом библиотеки Мошкова в Томске ([www.moshkow.tomsk.ru](http://www.moshkow.tomsk.ru)). Однако эти библиотеки обладают рядом недостатков, в частности, они не предоставляют возможность полнотекстового поиска. Поэтому, целью данной дипломной работы явилось создание электронной библиотеки для пользователей Томского Интернета с полнотекстовым поиском, не ориентированной на какой-либо конкретный жанр литературы с рабочим названием «Томская Электронная Библиотека<sup>\*</sup>».

---

<sup>\*</sup> Далее ТЭБ

# 1. Электронные библиотеки и полнотекстовые базы данных

## 1.1. От традиционной библиотеки к электронной

Несомненно, что эпоха, когда основной объем знаний человечества хранился только на печатных носителях, подходит к концу. Это объясняется такими недостатками бумажных изданий как невозможность содержать в себе все виды данных, включая анимацию, звук и видео, высокая стоимость их производства и распространения, быстрая устареваемость и невозможность быстро актуализировать материал. Постепенно все эти факторы приводят к тому, что печатные издания по сравнению с электронными ресурсами на нетрадиционных носителях становятся мало пригодными для использования. Среди последних, большой интерес вызывают *полнотекстовые электронные ресурсы* – цифровые коллекции виртуальных библиотек и баз данных. Предмет такого интереса заключается в том, что для постоянных пользователей Интернета электронные библиотеки и полнотекстовые базы данных могут дать разнообразные поисковые средства, позволяющие более точно найти искомый материал, более эффективный выход на вновь созданное знание, доступ к мультимедийным ресурсам.

Для начала попытаемся найти определение библиотеки вообще.

Библиотека (от греч. *biblion* - книга и ...тека), учреждение культуры, организующее сбор, хранение, общественное пользование произведений печати [2].

В соответствии с [1] все библиотеки можно условно разделить на 3 типа:

**Аналоговая (бумажная) библиотека**, то есть классическая или традиционная библиотека с карточными каталогами.

**Автоматизированная (гибридная) библиотека**, то есть аналоговая библиотека, имеющая компьютерный электронный каталог.

**Электронная (цифровая, виртуальная) библиотека**, то есть автоматизированная библиотека, в которой значительная часть информации находится в электронном формате.

На примере Научной Библиотеки ТГУ мы можем наблюдать процесс перехода от одного типа к другому. Состоялся переход от карточного каталога к электронному, и уже начался перевод бумажных материалов в электронный вид, однако, т.к. фонд библиотеки огромен, на это понадобится более десятка лет, и пока большинство информации находится в «бумажном» виде, библиотеку следует относить ко второму типу.

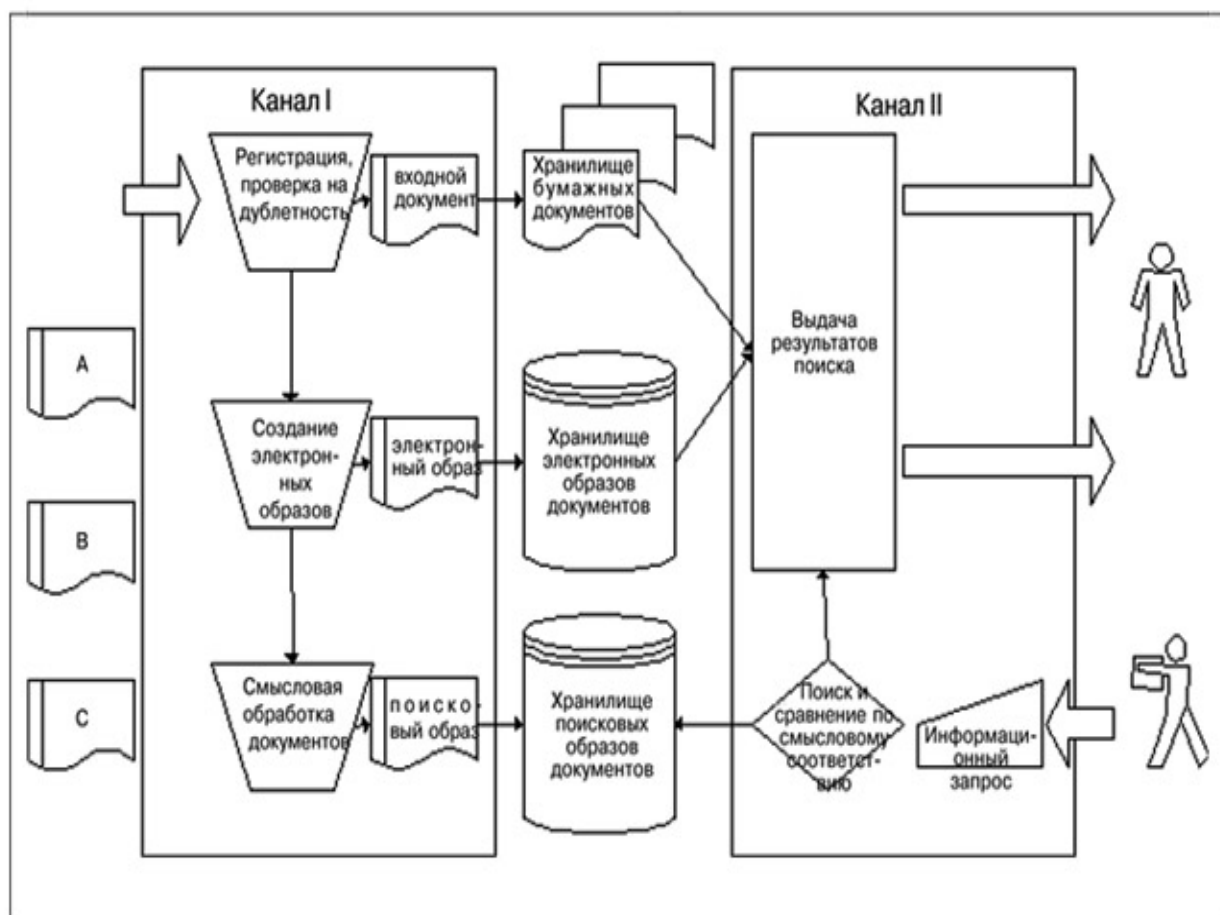
## 1.2. Технологии обработки документов и организация информационного поиска

Все библиотеки, как бы они не отличались по количеству, качеству, типу хранимой информации и носителям, обладают общими чертами. Все они созданы для того, чтобы обеспечивать некий информационный процесс, сущность которого состоит в сборе информации об объекте управления, ее упорядочивании и хранении, а также передаче информации от источника к потребителю. В электронных библиотеках перечисленные функции осуществляются с помощью средств информационной техники и собственно информационный процесс (в общем случае) можно разделить на следующие операции [3]:

- отбор документов, характеризующих объект;
- формирование информационных сообщений (образов документов) на основе отобранных документов;
- ввод сообщений в технические устройства
- хранение сообщений для последующего поиска;
- ввод запросов пользователей библиотеки;
- обработка запросов и выборка сообщений по критерию смыслового соответствия запросам;
- передача сообщений по каналам связи;
- отображение информации, содержащейся в сообщении, в доступном человеку виде.

При этом основное внимание традиционно уделяется вопросам организации процесса *информационного поиска* как определяющему показателю качества функционирования системы архива в целом.

Принципиальная схема информационного поиска приведена на рис. 1.2.1.



**Рис. 1.2.1 – Общая схема информационного поиска**

Здесь можно выделить два канала передачи информации:

I — канал ввода и хранения, формирующий архив документов, (однонаправленный);

II — канал доступа пользователей, по которому осуществляется ввод запросов от абонентов и выдача информации в ответ на запрос (функционирует в обоих направлениях).

Первичные документы делятся на три типа: текстовые, графические, видео- и аудиоматериалы. Предварительная обработка документов включает регистрацию поступившего документа, проверку его на дублетность (отсутствие в хранилище), смысловую обработку (описание, аннотирование, реферирование, индексирование), а также другие операции. При этом заполняется учетная карточка документа, на которой фиксируются его поисковые признаки, иначе говоря, поисковый образ (ПОД), и информация, подлежащая выдаче при ответе на запрос (аннотация, библиографическое описание и т.п.). Сам исходный документ направляется в хранилище документов.

Создание *поискового образа документа (ПОД)* — особенно важный этап, т.к. от него напрямую зависят поисковые способности системы. [3] Существует определенная последовательность операций в процессе поиска данных по электронному архиву. От пользователя поступает информационный запрос — словесная формулировка информационной потребности абонента. По субъективным причинам она требует уточнения, которое производится как до процесса поиска информации, так и в ходе его. На основе запроса формируется поисковое предписание, в соответствии с которым в хранилище находят поисковые образы документов. В результате поиска абонент получает описания первичных документов, на основании которых принимается решение о необходимости получить оригинал из хранилища.

### 1.3. Поисковые системы

Как уже упоминалось, при проектировании электронной библиотеки большое внимание должно уделяться вопросам организации процесса *информационного поиска*.

Некоторое время назад информации даже в локальной сети предприятия было не так много, и любой поиск осуществлялся банальным перебором доступных файлов и последовательной проверкой их названий и содержимого. Такой поиск называется *прямым*, и программы (утилиты), использующие технологию прямого поиска традиционно присутствуют во всех операционных системах и инструментальных пакетах. Но даже мощности современных компьютеров не хватит для быстрого и адекватного поиска в гигантских объемах данных при прямом поиске. Перебор пары сотен документов на диске и поиск в громадной библиотеке - разные вещи. Поэтому программы прямого поиска сегодня явно уходят на второй план, если речь идет об универсальных средствах.

В основе современных технологий лежат два основополагающих процесса [4]. Это *индексация* доступной информации и *обработка запроса* с последующим выводом результатов. Что касается первого, то любая программа (будь то настольный поисковик, корпоративная информационная система или поисковый Интернет-движок) создает свою область поиска. То есть, обрабатывает документы и формирует *индекс* этих документов (организованная структура, в которой содержится информация об обработанных данных). В дальнейшем именно созданный индекс используется для работы - быстрого получения списка нужных документов согласно запросу. Далее программа обрабатывает запрос (по ключевому слову-фразе) и выводит список документов, в которых эта ключевая фраза содержится. Так как информация содержится в структурированном индексе, то обработка запроса проходит значительно (в десятки и сотни раз) быстрее, чем в случае с прямым поиском (выборка документов осуществляется не перебором файлов, а анализом текстовой информации в индексе).

Найденные документы программа выводит в результирующем списке согласно *релевантности* - соответствию документа тексту запроса. В различных технологиях, конечно, присутствуют различные методы поиска и определения релевантности



документа (количество "вхождений" слова и его частота упоминания в документе, соотношение этих параметров к общему количеству слов в документе, расстояние между словами фразы запроса в искомых файлах и так далее). На основе этих параметров определяется "вес" документа и, в зависимости от этого, тот или иной файл оказывается в списке результатов на определенной позиции.

### 1.3.1. Виды информационного поиска

Рассмотрим основные виды информационного поиска, в соответствии с [4].

*Морфологический поиск* – вид автоматизированного полнотекстового информационного поиска с учетом морфологического анализа слов запроса пользователя и текстов проиндексированных информационных ресурсов. Использование данной возможности освобождает пользователя от необходимости правильно выбирать падеж искомого слова. Использование морфологического поиска почти всегда оправдано, поэтому должно присутствовать в любом профессиональном поисковике.

*Поиск по звучанию* является нестандартной возможностью даже для профессиональных поисковиков. Суть его заключается в том, что программа будет искать слова, которые *звучат* так же, как искомое слово.

*Поиск с коррекцией ошибок* - применяется для поиска слов, содержащих синтаксические ошибки, - это могут быть как опечатки, так и ошибки в документах, полученных при помощи систем распознавания символов. Также в некоторых программах есть настройка, позволяющая определять степень возможных ошибочных символов.

*Поиск с использованием синонимов* - использует список синонимов для различных слов. Готовый список синонимов может поставляться либо нет вместе с программой (с возможностью составления собственного списка синонимов). Также, можно воспользоваться списками в Интернете (соответственно, требуется подключение, что не всегда удобно).

*Поиск с использованием фраз, состоящих из слов, соединенных логическими операциями.* Каждому слову в запросе можно устанавливать свою значимость. Как дополнение, может быть использован словарь, состоящий из не значимых слов для того, чтобы не учитывать их при поиске.

### 1.3.2. Основные характеристики поисковых систем

Рассмотрим основные характеристики поисковых систем.

*Полнота* - это одна из основных характеристик поисковой системы, которая представляет собой отношение количества найденных по запросу документов к общему числу документов в Интернете, удовлетворяющих данному запросу. Полнота поиска в большей мере зависит от работы системы сбора и обработки информации. В связи с постоянным ростом количества документов в сети, эта система в первую очередь должна быть масштабируемой.

*Точность* - еще одна основная характеристика поисковой машины, которая определяется как степень соответствия найденных документов запросу пользователя.

*Актуальность* - не менее важная характеристика поиска, которая определяется временем, проходящим с момента публикации документов в сети Интернет, до занесения их в индексную базу.

Следует упомянуть также о *скорости поиска* и *наглядности представления его результатов*, которые являются необходимыми компонентами удобного поиска.

Рассмотрим несколько технологий информационного поиска на примере поисковой машины Рамблер, поисковой программы Google Desktop Search и сервиса СУБД MS SQL Server 2000 под названием MS Search.

### 1.3.3. Принципы работы поисковой машины Рамблер

Сбором информации в поисковой системе Рамблер занимается робот-паук, который обходит страницы с заданными URL и скачивает их в базу данных, а затем архивирует и перекладывает в хранилище суточными порциями. Робот размещается на нескольких машинах, и каждая из них выполняет свое задание.

В хранилище информация в сжатом виде собирается и разбивается на куски по 50 Мб. Эти части постепенно распределяются между 70 машинами, на которых запущена программа-индексатор. В результате на первом этапе формируется много маленьких индексных баз, каждая из которых содержит информацию о некоторой части Интернета. Таким образом, вся интеллектуальная обработка данных осуществляется параллельно, поэтому ускорение процесса индексации достигается простым добавлением машин в систему.

После того, как все части информации обработаны, начинается объединение (слияние) результатов.

Повышение точности в поисковой машине Рамблер достигается за счет использования различных технологий на всех этапах обработки и поиска информации. Например, распознавание грамматических омонимов, синтаксический анализ, который позволяет с определенной вероятностью распознавать некоторые имена собственные. Еще один способ повышения точности поиска - это выделение устойчивых обозначений и поиск их как отдельных лексических единиц.

На сегодняшний день в рабочие часы к поисковой машине Рамблер приходит около 60 запросов в секунду. Каждый из этапов обработки запроса многократно продублирован и защищен системой балансировки нагрузки. Благодаря дублированию информации поисковая система Рамблер является устойчивой к сбоям на отдельных участках, авариям, отказам оборудования. Если одна из машин перестала функционировать, нагрузка перераспределяется на другие машины, и выпадения документов из поиска не происходит.

Рамблер предоставляет возможности морфологического поиска с коррекцией ошибок.

### 1.3.4. Принципы работы поисковой программы Google Desktop Search

Программа предназначена для нахождения нужной информации в почтовых архивах Outlook / Outlook Express, Netscape Mail / Thunderbird, Netscape / Firefox / Mozilla, текстовых документах, документах Word, Excel, PowerPoint, PDF, мультимедийных файлах, а также в логах чатов AOL Instant Messenger и в посещенных web-страницах, сохраненных в кэше браузера Internet Explorer.

После установки производится сканирование жесткого диска (в фоновом режиме) и индексация всего содержимого перечисленных выше документов, после чего новые документы и посещенные веб-страницы будут индексироваться в режиме реального времени. По сути, Google Desktop Search соединяет воедино веб и рабочий стол, тем самым значительно упрощая процедуру поиска и одновременно значительно расширяя его возможности.

Google Desktop Search интегрируется с браузером (выводит данные поиска в том браузере, который является браузером по умолчанию), причем полностью функционален только с Internet Explorer и другими браузерами, построенным на его движке - MyIE2 и т.п. В других браузерах (например, Opera, Firefox) не индексируется веб-история.

Что касается анализа поисковых запросов и выдачи результатов, то здесь все абсолютно идентично Google в Интернет: такая же система отображения результатов, тот же стандартный набор логических операций для поисковых запросов.

Скорости поиска и индексирования вполне приемлемые, как и размер создаваемого индекса [4]. Программа предоставляет все виды поиска, указанные в п. 2.2., кроме поиска по звучанию.

### 1.3.5. Сервис MS Search

В системе ТЭБ в качестве СУБД используется MS SQL Server 2000, в котором введена поддержка полнотекстовых запросов, что дает возможность расширенного поиска по колонкам, содержащим символьные строки. Данная возможность реализуется сервисом *MS Search*, который исполняет следующие роли [6]:

- Поддержка индексации

Реализует полнотекстовые каталоги и индексы (для БД). Обработывает определения полнотекстовых каталогов и таблиц, а также колонок, составляющие индексы в каждом каталоге. Реализует вызовы для заполнения этих индексов.

- Поддержка запросов

Обработывает запросы полнотекстового поиска. Определяет, какие вхождения в индексе соответствуют критерию выбора текста. Для каждого вхождения, которое соответствует критерию выбора, сервису MS SQL Server возвращается тождество строки и порядковый номер этой строки, а SQL Server использует эту информацию для создания окончательного результата запроса. Типы поддерживаемых запросов включают в себя поиск для:

- Слов и фраз;
- Слов, сходных с искомым;
- Формы словоизменения глаголов и существительных.

Полнотекстовые каталоги или индексы хранятся не в БД SQL Server, а в отдельных файлах, управляемых сервисом MS Search. При восстановлении SQL Server файлы полнотекстовых каталогов не восстанавливаются. Кроме того, их нельзя резервно копировать и восстанавливать T-SQL-запросами BACKUP и RESTORE. После выполнения операций RECOVERY или RESTORE полнотекстовые каталоги должны быть отдельно синхронизированы. Файлы полнотекстовых каталогов доступны только сервису MS Search или системному администратору Windows NT.

Связь между SQL Server и сервисом MS Search выполняется через полнотекстовый провайдер (full-text provider). Когда MS SQL Server получает T-SQL-запрос с полнотекстовой конструкцией, он извлекает из сервиса MS Search (с помощью полнотекстового провайдера) необходимую информацию. Полнотекстовые конструкции используют предикаты CONTAINS или FREETEXT, либо функции CONTAINSTABLE или FREETEXTTABLE. Полнотекстовые конструкции могут ссылаться на множество колонок полнотекстового индекса, если им неизвестно, в какой именно колонке могут содержаться условия поиска.

Полнотекстовые каталоги или индексы и поиск по всему тексту, предоставляемые сервисом MS Search, применимы только для таблиц в БД SQL Server.

## **1.4. Классификация электронных библиотек**

В контексте создания нового электронного архива (библиотеки), а также при переводе архива в электронную форму классификация хранилищ нужна для того, чтобы правильно выбрать необходимые технические и программные средства, а следовательно, достичь максимальных возможностей при минимальных затратах для каждого конкретного случая. Иногда необходимо обеспечить исключительную сохранность документов, например, в хранилищах редких или ветхих изданий. В других случаях важнее всего возможность оперативного поиска одновременно для множества пользователей. В третьих ситуациях важно и то, и другое в равной мере.

Как основу классификации в [3] рассматривают технологические особенности реализации основных функций архива:

- первичный ввод документов в архив;
- поиск документов;
- хранение документов.

Постараемся определить средства, нужные организаторам электронного архива в зависимости от того, какие функции преобладают.

Для *конечного пользователя* принципиально важно исполнение именно поисковых функций. Электронные архивы могут обеспечивать:

- поиск документов по картотеке;
- полнотекстовый поиск по содержанию документов.

Следует отметить, что традиционно возможность полнотекстового поиска должна предоставляться наряду с поиском по картотеке. Даже в том случае, когда в системе нет

специальной программы поддержки картотеки документов и установлены средства полнотекстового поиска, в качестве картотеки выступает файловая система ОС, описывающая характеристики файлов.

Оптимально, когда в ответ на запрос пользователю предоставляется текст документа, релевантного запросу. Этот текст может быть представлен либо отсканированным изображением документа, либо быть непосредственно текстом документа, каким-либо образом введенным в архив.

Полнотекстовый поиск предполагает значительные преимущества, но его организация требует особых усилий. Во-первых, существенно увеличивается объем работ на этапе первичного ввода документов в архив — необходим либо ручной ввод документов, либо автоматизированный, включающий распознавание текстов всех вводимых документов, устранение ошибок распознавания. Кроме того, потребуется дополнительная память на жестких дисках (до 50% от объема хранимых текстов) для хранения поисковых полнотекстовых индексов и необходимы программные средства для осуществления многопользовательского поиска по тексту документов. Возможность полнотекстового поиска существенно усложняет систему (и соответственно удорожает ее).

Для *владельца архива* на первый план выходят группы функций, связанные с созданием хранилища документов и управления им. Хранение документов в электронном архиве может быть организовано двумя способами [3]:

- с использованием средств только файлового доступа ОС (тела документов хранятся в виде файлов в файловой системе);
- на основе сервера автоматизированного документооборота (тела документов представляют собой логические разделы одного файла, доступ к которым возможен только посредством специального сервера).

Как организуется хранение электронных образов (тел) документов в файловой системе? Если их общий объем велик и превосходит объем накопителей на жестких дисках типового файлового сервера (в настоящее время около 300 ГБ), то используют библиотеки MO- или CD-дисков. Эти библиотеки представляют собой роботизированные приводы накопителей MO- или CD-дисков, выполненные в виде отдельных настольных или напольных устройств и подключаемые к серверу, как правило, по интерфейсу SCSI.

Следует отметить, что:

- в ТЭБ использование библиотек MO- или CD-дисков не предполагается;
- в состав ПО сервера, на котором может быть размещена ТЭБ, должна быть включена система управления базами данных (MS SQL Server 2000) для хранения картотеки описаний документов, непосредственно содержимого текстовых файлов, а также прочей информации;
- документы в сканированном формате хранятся в виде внешних файлов; содержимое текстовых документов хранится в базе данных в двоичном формате.

## 1.5. Варианты построения электронного архива

Рассмотрим возможные варианты построения электронного архива в соответствии с классификационными признаками, предложенными в [3] (см. таб. 1.4.1).

№ Варианта	Возможности поиска	Тип хранения
Вариант 0	Поиск по файловой системе	Документы в файловой системе
Вариант 1	Поиск по картотеке	Документы в файловой системе
Вариант 2	Поиск по картотеке	Система управления документами
Вариант 3	Полнотекстовый поиск	Документы в файловой системе
Вариант 4	Полнотекстовый поиск	Система управления документами

**Таблица 1.4.1 – Варианты построения электронного архива**

**Вариант 0** – это минимальный набор средств, обеспечивающий решение задач электронного документального архива. Все последующие варианты являются его расширениями, поэтому далее будем называть его базовым.

Образы как текстовых, так и графических документов представляют собой их сканированные изображения в формате графических файлов TIFF Group 4 (мульти-tiff или многостраничный tiff-файл). В рамках этого формата одному многостраничному документу соответствует один файл с расширением tiff, каждая страница документа может быть обработана отдельно от других. Для уменьшения объема хранимых данных используется сжатие информации.

Хранение образов документов в файловой системе позволяет реализовать простейший режим поиска документов (например, с помощью утилит командной строки). Однако это нежелательно, ведь алгоритм последовательного просмотра содержимого тома крайне медленно работает в больших объемах данных, характерных для библиотек. Поэтому базовый комплект средств не может быть рекомендован для создания электронного архива.

Самый разумный способ избежать выполнения медленных поисковых операций – ограничить доступ пользователей к файлам образов документов. Доступ к этим файлам должен быть возможен только через поисковую систему.

Варианты 1-2 предполагают поддержку поиска по картотеке, варианты 3-4, кроме поиска по картотеке, обеспечивают полнотекстовый поиск.

**Вариант 1** может быть назван вариантом архива с поиском по картотеке (картотечный архив) и хранением документов в файловой системе. Для его реализации в состав ПО ЭА должны быть введены программные средства организации картотеки документов коллективного пользования. Такие системы строятся по архитектуре «клиент-сервер» и, следовательно, состоят из двух компонент. Серверная компонента ориентирована на использование стандартной реляционной клиент-серверной СУБД типа MS SQL Server или Oracle, которую необходимо также установить на сервере. Для клиента можно использовать стандартные браузеры с протоколом HTTP. Такие системы наиболее эффективны в глобальных сетях при использовании технологии Интернет/Инtranет.

**Вариант 2** можно назвать картотечным архивом с системой управления документами. Документы хранятся в файле системы управления документами. Последняя, как правило, содержит в комплекте программ средства формирования библиотек документов и организации поиска по картотеке.

**Вариант 3** является минимальным для архива с полнотекстовым поиском (полнотекстовый архив). Для этого необходимо обеспечить ввод в систему не только изображений документов, но и их содержания в текстовом виде. Кроме того, в состав ПО сервера хранилища документов должны быть введены программные средства полнотекстового поиска, так называемая *поисковая машина*. Все вводимые документы должны быть проиндексированы по тексту поисковой машиной; результатом индексирования является *индексный файл*, который занимает дополнительный объем памяти на запоминающих устройствах сервера. Объем индексного файла ориентировочно равен объему текстов индексируемых документов.

Таким образом, вариант 3, в отличие от варианта 1, должен содержать дополнительно:

- дополнительную внешнюю память на сервере хранилища документов для хранения полнотекстовых индексов;
- в состав ПО сервера хранилища документов должна быть включена полнотекстовая поисковая машина, поддерживающая работу с несколькими языками, а в состав ПО рабочих станций операторов – клиентская часть полнотекстового поиска.

**Вариант 4** в дополнение к средствам варианта 3 предполагает использование систему управления документами (полнотекстовый архив с системой управления документами).

Планируемая реализация ТЭБ относится к варианту 3, однако следует отметить, что в системе ТЭБ на текущем этапе не предполагается использование технологии распознавания текстов документов, а полнотекстовый поиск будет осуществляться только по документам, изначально помещенным в архив в текстовом формате.

## **1.6. Классификация и библиографическое описание книг**

Для конечного пользователя библиотеки важно иметь возможность просто и с высокой степенью точности локализовывать в политематической базе данных электронных книг содержательную область, отвечающую его информационной потребности. Такую возможность дает, например, классификация книг.

Одной из общепринятых классификаций является УДК (универсальная десятичная классификация), которая существует уже более 100 лет [7]. Ее главной отличительной особенностью является иерархическое построение большинства разделов основных и вспомогательных таблиц по принципу деления от общего к частному с использованием цифрового десятичного кода.

В многочисленных разделах этой системы упорядочено множество понятий по всем отраслям знаний или деятельности. Иными словами, УДК охватывает весь универсум знаний.

УДК универсальна в применении. Благодаря обилию средств и приемов индексирования, легко сокращаемой дробности она успешно применяется для

систематизации и последующего поиска самых разнообразных источников информации в различных по объему и по назначению фондах.

Однако, в некоторых случаях, применение УДК для классификации книг не является наилучшим решением, т.к. для среднестатистического пользователя Интернета она не является интуитивно понятной. Вследствие этого, в ТЭБ была использована т.н. упрощенная классификация. Классы и подклассы были выделены на основе анализа упрощенных классификаций существующих электронных библиотек. Тем не менее, в системе ТЭБ предусмотрена конвертация упрощенных классов в классы УДК и обратно, таким образом, пользователю предоставляется возможность поиска и по упрощенным категориям, и по категориям УДК.

Применение стандартного библиографического описания документов, хранящихся в библиотеке, также имеет важное значение для процесса информационного поиска. Существует несколько стандартизованных форматов библиографического описания книг, которые с успехом применяются многими библиотеками (в т.ч. электронными). Одним из общепризнанных стандартов является UNIMARK (Universal Machine-Readable or Catalogue). Этот стандарт используется в библиотечном деле большинства развитых государств мира [8].

Впервые стандарт опубликован в 1977 году и на сегодняшний день поддерживается большинством библиотечных программных продуктов. Существует международный комитет по UNIMARK, в который входит и Россия.

Необходимо отметить, что формат UNIMARC не предназначен для поддержки коммерческих операций с товаром, он отражает специфику лишь библиотечной деятельности. Он содержит только поля библиографического описания, их более 900. В то же время, многие элементы книготорговой библиографической записи (вес издания, толщина блока, ценовые скидки, НДС и т.д.), необходимые участникам книжной торговли, не являются обязательными для библиотечной деятельности.

Относительно системы ТЭБ формат UNIMARK является наиболее подходящим форматом библиографического описания книг.

## **1.7. Анализ электронных библиотек**

Проведем сравнительный анализ некоторых существующих электронных библиотек с целью выявления их достоинств и недостатков с точки зрения конечного пользователя (см. табл. 1.7.1.). Критерии для анализа были выбраны в соответствии с [9].

В результате анализа было выяснено, что большинству критериев удовлетворяет Томская Техническая Библиотека.

Итак, с учетом анализа, при создании ТЭБ было предусмотрено следующее:

- тематический охват – без ограничений;
- поисковая система предоставляет два вида полнотекстового поиска: по описанию книги и по ее содержанию;
- используется два вида классификации книг – УДК и упрощенная;
- библиографическое описание книги соответствует стандарту UNIMARK;
- условия доступа к ресурсам – бесплатно с регистрацией;
- формат представления данных: rar, zip, djv, chm, pdf, doc, txt, html.



Критерий	<a href="http://www.radzinski.ru">www.radzinski.ru</a>	<a href="http://www.skazki.org.ru">www.skazki.org.ru</a>	<a href="http://www.techlib.tomsk.ru">www.techlib.tomsk.ru</a>
1. Состав ресурсов библиотеки по тематическому охвату;	Книги Эдварда Радзинского (-)	Сказки народов мира (-)	Техническая литература (+)
2. Хронологическая глубина наполнения библиотеки ресурсами;	Неизвестно (-)	С 2003 г. (+-)	С 2005 г. (+-)
3. Наличие поисковой системы и ее свойства;	Да, с помощью поисковой машины Yandex (полнотекстовый, морфологический поиск) (+)	Прямой поиск по названию и автору (слова и словосочетания) (-)	Прямой поиск по названию, автору, краткому описанию (-)
4. Использование традиционной библиотечной классификации (УДК, ГРНТИ и др.);	Нет (-)	Нет (-)	Собственная упрощенная классификация (-)
5. Представленная информация о книгах;	Издательство, объем (стр.), тираж, формат, год издания, код ISBN (+)	Название, автор, дата добавления (-)	Категория, название, автор, объем (стр.), язык, издательство, размер файла, год издания, кол-во скачиваний, формат, описание (+)
6. Количественная характеристика электронного ресурса (количество книг, количество пользователей, объем ресурса);	Около 20 книг, количество пользователей – неизвестно (-)	Книг около 10000 (+)	Книг около 8000, пользователей (зарегистрированных) около 1500 (+)
7. Новизна документов, представленных в библиотеке (период обновления);	Неизвестно (-)	Обновляется еженедельно (+)	Обновляется ежедневно (+)
8. Полнота библиотеки (количество материалов по объявленной теме);	Неизвестно (-)	Достаточно (+)	Достаточно (+)
9. Условия доступа к ресурсам;	Бесплатно (+)	Бесплатно (+)	Бесплатная регистрация и предоставление 5-ти новых книг (+)
10. Формат представления данных	Текст html (-)	Текст html (-)	Rar, zip, djv, pdf, doc, chm, txt (+)

Табл. 1.7.1. – Сравнительный анализ электронных библиотек

## **1.8. Выводы**

Становится очевидным, что электронные библиотеки имеют множество преимуществ по сравнению с традиционными. В частности, создание хранилища документов и управление им становится проще и удобнее, а информационный поиск - эффективнее.

Существует множество вариантов построения электронного архива. Выбор варианта зависит от целей создания хранилища и выделенных средств.

При разработке электронной библиотеки важное внимание должно уделяться процессу организации информационного поиска, который должен давать пользователям возможность быстро и достаточно просто локализовывать область документов, отвечающих его интересам. Книги должны быть разбиты на категории в соответствии с некоторыми признаками, а так же должны быть созданы их поисковые образы (т.е. некое библиографическое описание). Использование классификации и библиографического описания книг в соответствии с мировыми стандартами является достоинством библиотеки.

## 2. Базовая архитектура разработанной электронной библиотеки

Назначение большинства компьютерных систем – получение данных из базы и отображение их пользователю. Затем пользователь выполняет какие-либо действия и система сохраняет данные либо модифицирует их. Система ТЭБ не является исключением.

Вначале дадим не очень строгие определения:

- *web-приложение* - это набор взаимосвязанных файлов (\*.htm, \*.asp, \*.aspx, файлов изображений и т. п.), а также связанных с ними компонентов (двоичных файлов .NET или классического COM), которые размещены на web-сервере;
- *web-сервер* - это программный продукт, на котором размещаются web-приложения и который обычно обеспечивает набор связанных с web-приложениями служб, таких как интегрированные средства обеспечения безопасности, поддержка протокола FTP, поддержка средств передачи электронной почты и т. п.

Томская Электронная Библиотека реализована в архитектуре клиент-сервер, адаптированной к условиям работы в Интернет/Инtranет, и имеет конфигурацию Web-browser / Web-server / SQL-server. Это позволяет достичь оптимального сочетания, с одной стороны, универсальности и эффективности компьютерной обработки информации (за счет мощной СУБД, в качестве которой используется MS SQL Server 2000) и, с другой стороны, удобства для конечного пользователя за счет использования в качестве клиентского приложения стандартного Web-браузера. В роли Web-сервера выступает IIS (Internet Information Server) (под управлением операционной системы Windows NT/2000/2003), а непосредственно Web-приложение написано с использованием технологии ASP.net на языке C#.

### 2.1. Использование ASP.net для разработки Web-приложений

#### 2.1.1. Общие идеи ASP.net

*Active Server Pages* (ASP) - это технология Microsoft, позволяющая создавать динамические web-страницы.

«Классическая» ASP с большим успехом использовалась на web-узлах, таких, как *www.microsoft.com* и *www.deli.com*, а также на множестве других больших и маленьких сайтов [10].

ASP.net – это концептуально новая технология Microsoft, созданная в рамках идеологии .NET. Ключевыми сторонами .NET являются масштабируемость, кроссплатформность, межъязыковое взаимодействие и понятие "безопасное программирование". В ASP.net появилось много новых функций, а существовавшие в «классической» ASP значительно усовершенствованы. IIS и ASP.net предоставляют мощную и безопасную платформу для выполнения Web-приложений.

ASP.net проповедует идею полной динамики, т.е. предлагает не хранить объекты в памяти, а динамически создавать их и кэшировать. Такой подход позволяет обслужить максимальное количество пользователей. Т.е. ресурсы достаются всем поровну. Если же хранить объекты на сервере, мы получаем не количество, а качество – экземпляр приложения, созданный для конкретного пользователя и работающий для него.

## 2.1.2. Сравнение ASP.net и PHP

Проведем сравнительный анализ двух технологий создания динамических Web-страниц – ASP.net и PHP (Personal Home Page) в соответствии с [11], чтобы обусловить выбор ASP.net для разработки ТЭБ (см. табл. 2.1.2.1.).

<b>Критерий</b>	<b>PHP</b> (СУБД MySql, Web-сервер Apache, ОС Linux)	<b>ASP.net</b> (СУБД MS SQL Server, Web-сервер IIS, ОС Windows)
1. Стоимость	Открытая и бесплатная технология (однако разработка и поддержка коммерческих проектов обходится дорого)	Платная технология (придется оплатить несколько лицензий Microsoft)
2. Сложность освоения	Не нужна дорогая среда программирования, достаточно пары учебников	Необходима среда разработки Visual Studio, MSDN, доступ в Интернет
3. Основное предназначение	Мелкие и средние проекты, рассчитанные на небольшие группы программистов	Средние и большие проекты, рассчитанные на большие группы программистов под четким управлением
4. Скорость работы	Скорость работы обеспечивается тем, что все PHP-приложения работают в едином адресном пространстве. Подход более быстр, но менее надежен.	За счет сложной модели классов ASP.net многократно перепроверяет данные, удерживая каждое приложение в отдельном адресном пространстве. Подход более надежен и не менее быстр.
5. Кэширование данных из БД	Генерирует множество запросов к СУБД	Старается делать из БД как можно меньше выборки, помещая все актуальные таблицы и даже связи между ними в кэш (технология ADO.NET в случае с MS SQL)
6. Наличие отладчика	Полноценного отладчика нет, само устройство языка ведет к серьезным логическим ошибкам	Удобный отладчик Visual Studio

**Табл. 2.1.2.1. – Сравнительный анализ ASP.net и PHP**

С позиций хостинга принципиальной разницы между платформами Linux + Apache или Windows NT/2000/2003 + IIS нет. Оба варианта практически равнозначны, и выбор

того или другого зачастую основывается не на рациональных причинах, а на личных предпочтениях конкретного администратора или web-мастера. Тем не менее, по данным последних исследований аналитической компании IDC, опубликованных в конце ноября 2005г., лидером рынка серверных операционных систем является корпорация Microsoft и ее Windows.

Итак, для разработки ТЭБ как Web-приложения автором была выбрана среда ASP.NET версии 1.1, а для написания исходного ASP-текста – язык C#. Данный выбор обусловлен следующими причинами:

- В среде ASP.NET облегчено проектирование Web-приложений;
- Глубокая интеграция ASP и СУБД;
- Наличие множества удобных и гибких компонент;
- Надежность;
- ASP.net – технология для Windows-систем.

## **2.2. Использование СУБД MS SQL Server 2000**

### **2.2.1. Особенности работы MS SQL Server 2000**

Поскольку наша система является многопользовательской, и в качестве сервера баз данных мы выбрали MS SQL Server 2000, дадим краткое описание возможностей и основных принципов работы этой СУБД (в соответствии с [12]).

Microsoft SQL Server 2000 - это законченное решение для управления и анализа данных, позволяющее оперативно развертывать масштабируемые Web-приложения. SQL Server 2000 - ключевой компонент поддержки электронной коммерции, интерактивных деловых приложений и хранилищ данных, обеспечивающий масштабируемость, необходимую для поддержки растущих, динамических сред.

SQL Server 2000 - это реляционная СУБД, которая использует язык Transact-SQL для пересылки сообщений между компьютером клиента и компьютером, на котором работает SQL Server 2000. Реляционная СУБД состоит из механизма баз данных, собственно баз данных и приложений, необходимых для управления данными и компонентами реляционной СУБД. Реляционная СУБД организует данные в виде связанных строк и столбцов, составляющих базу данных. Реляционная СУБД отвечает за поддержку структуры базы данных и решает следующие задачи:

- поддерживает связи между данными в базе;
- гарантирует корректное хранение данных и выполнение правил, регламентирующих связи между ними;
- восстанавливает данные после аварии системы, переводя их в согласованное состояние, зафиксированное до сбоя.

База данных (БД) SQL Server 2000 представляет собой реляционную базу данных, совместимую с SQL (Structured Query Language) с интегрированной поддержкой XML для Интернет-приложений. SQL Server 2000 создан на основе современного расширяемого сервера SQL Server 7.0.

Как и многие серверные продукты, работающие под управлением операционной системы Windows NT/2000, MS SQL Server 2000 реализован в виде набора служб операционной системы, каждая из которых запускается самостоятельно и отвечает за

определенный круг задач. Приведем список служб SQL Server с их кратким описанием [13]:

- MSSQLServer является ядром СУБД и выполняет все основные операции, отвечает за пользователей и их права доступа, контроль свободных ресурсов, журнал транзакций, выполнение хранимых процедур, и т.д.
- SQLServerAgent предназначена для автоматизации администрирования, с ее помощью можно автоматически запускать различные задания, отправлять уведомления операторам и др. В работе этой службы применяются объекты трех видов: Jobs (задания), Operators (операторы), Alerts (события).
- Microsoft Search (MSSearch, Full-Text Search), позволяет выполнять полнотекстовый поиск по базам данных.
- MSDTC – служба занимается координацией распределенных транзакций.

Начиная с версии 7.0, компания Microsoft включила в комплект поставки Microsoft SQL Server специальную компоненту: систему полнотекстового поиска по базе данных. Эта система дает возможность пользователю находить нужные записи по разнообразным условиям, таким как поиск слов и словосочетаний, поиск слов в различных грамматических формах, а также средства поиска записей, похожих на заданный фрагмент текста. Причем для работы с этой системой пользователю не требуются дополнительные программные средства: все операторы полнотекстового поиска включены в язык Transact-SQL, используемый в этой СУБД.

Отметим разницу между механизмами полнотекстового индексирования и обычного индексирования в MS SQL Server 2000. Полнотекстовое индексирование не считается частью функции индексирования SQL Server. Полнотекстовый индекс позволяет выполнять полнотекстовые запросы для поиска в БД строки символьных данных. Этот индекс содержится в полнотекстовом каталоге. Полнотекстовый индекс и каталог обслуживается механизмом Microsoft Search, а не SQL Server.

Итак, в качестве СУБД для хранения данных ТЭБ автором была выбрана система MS SQL Server 2000 по следующим причинам:

- Архитектура «клиент-сервер» (многопользовательское приложение);
- Работает под управлением ОС Windows NT/2000/2003;
- Обеспечивает механизм полнотекстового лингвистического поиска;
- Сравнительно невысокая стоимость СУБД.

### 2.2.2. Варианты хранения и загрузка текста в таблицы БД

Текстовая информация может храниться в таблицах базы данных SQL Server 2000 в двух вариантах [5]. В первом варианте тексты представляют собой значения полей типа char, varchar, nchar, nvarchar, text, ntext. Второй вариант хранения текстов предполагает, что они являются содержимым поля типа image, в которое загружаются в формате документов MS Office или других приложений. В этом случае в таблице в дополнительном поле должно храниться расширение файла, соответствующее типу документа. Хранение расширения файла позволяет подключать соответствующие фильтры при обработке или отображении содержимого поля.

Если тексты хранятся в формате приложений, в которых они созданы, то изначально они существуют в виде файлов. Для загрузки файлов в таблицы можно использовать утилиту TextCopy. Данная утилита предназначена для обновления (Update)

поля типа text и image и заполнения его содержимым заданного файла, а также для загрузки содержимого полей данного типа в файл.

### 2.2.3. Полнотекстовый индекс и полнотекстовый каталог

Независимо от варианта хранения текста, в SQL Server можно воспользоваться функцией полнотекстового, т. е. лингвистического, поиска. Концепция полнотекстового поиска в SQL Server основана на использовании службы Microsoft Search (см. п. 1.3.5.).

Полнотекстовый каталог занимает довольно много места. Размер его может превышать размер текстовой информации, для работы с которой он создавался, если информации мало. Служебные файлы и папки в полнотекстовом каталоге имеют объем от 4 Мбайт. Полнотекстовые каталоги на сервере нумеруются от 1 до 256, и информация о них хранится в системной таблице sysfulltextcatalogs. Дополнительно в таблице sysindexes в поле ftcatid для таблицы хранится номер ее полнотекстового каталога.

Каждая ли таблица должна иметь свой полнотекстовый каталог? Официальный ответ в документации звучит так: если таблица содержит или будет содержать миллион записей, тогда ей необходим отдельный полнотекстовый каталог. В остальных случаях количество полнотекстовых каталогов и их распределение между таблицами определяется характером работы с таблицами и режимом обновления каталога. Обновляемые таблицы часто имеют общий полнотекстовый каталог для упрощения сопровождения, но использование общего каталога может снизить скорость выполнения запросов. Использование отдельных каталогов для таблиц ограничено числом 256 для сервера в целом.

При резервном копировании базы полнотекстовый индекс не помещается в резервную копию, но при копировании базы на другой сервер он может быть создан заново.

### 2.2.4. Этапы создания полнотекстового индекса

Полнотекстовый индекс строится в несколько этапов.

Если тексты хранятся в поле типа image, то они сначала фильтруются по типу документа. Фильтр представляет собой динамическую библиотеку, которая извлекает из документа его содержимое в виде потока текста. Нужный фильтр определяется по расширению документа, хранящемуся в отдельном поле. Механизм полнотекстового поиска загружает соответствующий фильтр документа, и тот выбирает текстовую информацию для индексирования. В SQL Server 2000 реализованы фильтры для следующих типов документов: doc, xls, ppt, txt, htm. На сайте MSDN можно найти средства написания и подключения новых фильтров для файлов других типов. В случае хранения текстов не в полях типа image фильтрация упрощается, так как фильтр известен заранее.

На следующем этапе поток текста должен быть разбит на слова с помощью специального делителя (Word Breaker). Разбиение на слова зависит от локализации, т. е. используемого языка. В Microsoft Search применяется несколько встроенных библиотек локализации. Встроенные библиотеки существуют для следующих языков: китайского, английского, французского, немецкого, итальянского, японского, корейского, испанского, шведского. К сожалению, русский язык не входит в этот список, поэтому

пользователь не сможет искать информацию с учетом *всех* грамматических особенностей русского языка без использования дополнительных морфологических модулей. В качестве примера такого модуля можно привести коммерческую поисковую систему «Следопыт для MS SQL Server 7.0/2000» от компании «МедиаЛингва».

Встроенные библиотеки локализации обеспечивают разбор соответствующих языку грамматических конструкций. Для всех прочих языков применяется нейтральный (Neutral) делитель, который в качестве разделителя слов использует пробелы. Язык для хранения данных в полнотекстовом индексе определяется параметром Unicode collation locale, выбираемым во время установки SQL Server 2000. Третий этап заключается в отбрасывании незначащих слов (Noise words), или стоп-слов, к которым относятся междометия, союзы, предлоги

Списки незначащих слов хранятся в \Program Files\Microsoft SQL Server\MSSQL\FTDATA\Config под именами Noise. Расширение файла соответствует названию языка. Списки можно редактировать как обычные текстовые файлы. Имя папки файловой системы, содержащей полнотекстовый каталог, соответствует уникальному идентификатору полнотекстового каталога. Путь к каталогу можно найти в таблице sysfulltextcatalogs.

Следующий этап: сопровождение полнотекстового индекса. Здесь выделяются три основные задачи: резервное копирование полнотекстового каталога, обновление полнотекстового индекса и настройка сервера для оптимального выполнения полнотекстовых запросов.

Заполнение полнотекстового индекса и его обновление может принимать различные формы:

- Полное заполнение.

Вхождения индексов строятся для всех строк всех таблиц, которые охватываются полнотекстовым каталогом. Обычно полное заполнение происходит при первом заполнении каталога.

- Инкрементальное заполнение.

Настраивает индексы только для добавленных, измененных или удаленных (после последнего заполнения) строк. Для этого индексируемая таблица должна содержать колонку с типом данных *timestamp*. В противном случае может выполняться только полное заполнение (даже после запроса инкрементального заполнения).

Если для таблицы, которая не была перед этим ассоциирована с каталогом, определен новый полнотекстовый индекс, запрос на инкрементальное заполнение приведет к созданию всех вхождений для таблицы.

Запрос на инкрементальное заполнение реализуется как полное заполнение и в случае изменения метаданных для данной таблицы после последнего заполнения, например, изменения определения какой-либо колонки, индекса или полнотекстового индекса.

Обычно полнотекстовый каталог бывает определен, и для него уже выполнено полное заполнение. После этого для передачи новых данных, добавленных к индексируемым колонкам SQL Server, выполняется инкрементальное заполнение. Эти периодические операции по инкрементальному заполнению обычно определяются как задания (jobs) и автоматически обрабатываются сервисом SQL ServerAgent.



## 2.3. Структура Томской Электронной Библиотеки

Схематически общая архитектура электронной библиотеки ТЭБ представлена на рис. 2.3.1.

Пользователь может обращаться к ТЭБ посредством любого HTTP-браузера (например, Internet Explorer, Netscape Navigator, Opera). После ввода URL-адреса ТЭБ, в окне браузера открывается html-страница с некоторыми общими сведениями о библиотеке, статистическими данными (такими, как количество книг в библиотеке, количество зарегистрированных пользователей и др.), информацией о регистрации. Непосредственная работа с библиотекой допускается только после аутентификации, поэтому пользователю будет предложено ввести свои логин и пароль, либо зарегистрироваться. При аутентификации и регистрации происходит обращение к базе данных Library к таблице «Пользователь».

В случае успешной аутентификации пользователю становятся доступны возможности поиска, скачивания и добавления книг, а также возможность редактирования полей книг и собственных регистрационных данных (таких, как пароль и e-mail).

При добавлении книги в БД возможны два варианта:

1. Если книга в текстовом формате (doc, txt, html) – ее содержимое копируется в поле типа image специальной таблицы БД и индексируется механизмом MS Search. При попытке скачивания такой книги, ее содержимое выгружается во временный файл, который и пересылается запросившему пользователю.
2. Другие книги добавляются непосредственно на сервер и хранятся в виде внешних файлов. Скачивание таких книг происходит непосредственно с сервера, а не из БД, что дает более высокую скорость передачи данных по сети.

Механизм полнотекстового индексирования запускается по расписанию дважды в сутки.

При добавлении книги на сервер (вне зависимости от варианта добавления), в БД формируется ее поисковый образ (ПОД): информация об авторах, издательстве и т.д. ПОД формируется вручную, но предполагается усовершенствовать систему и реализовать возможность автоматического заполнения ПОД (для книг в текстовом формате).

В ТЭБ каждый пользователь имеет свой статус. На данный момент выделяется два статуса: «читатель» и «администратор». Пользователь со статусом «администратор» обладает всеми правами «читателя», а также имеет право удалять книги и пользователей, редактировать ПОД книг, добавлять, редактировать и удалять категории (в дереве категорий упрощенной классификации). «Читатель» также имеет право удалять редактировать поля книг, который были им добавлены, но только в течение определенного времени, с момента добавления.

Пользователю предлагается несколько видов поиска:

- По полям автор, название, издательство (т.н. быстрый поиск, требует введения поискового выражения);

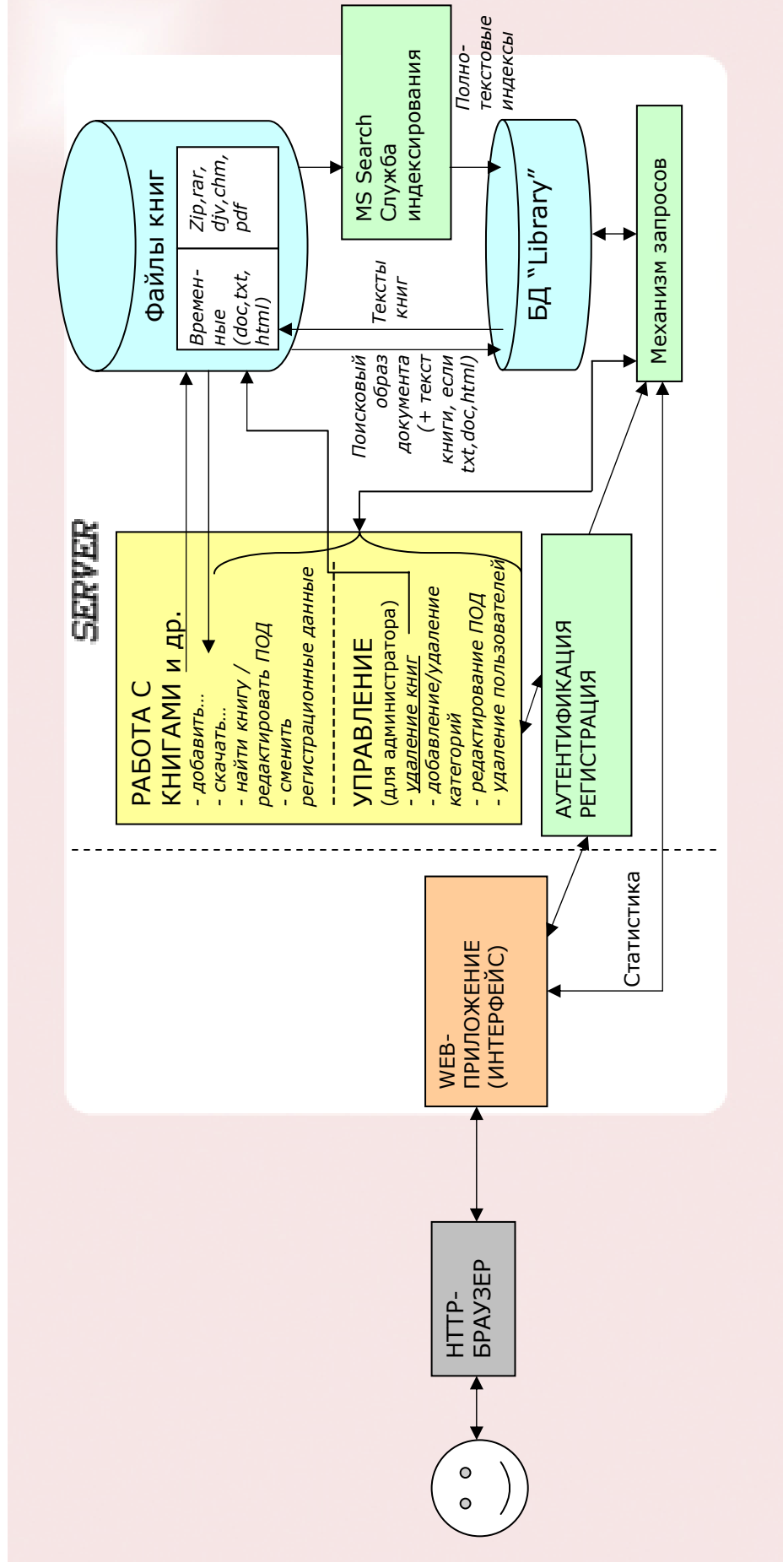


Рис 2.3.1 – Общая архитектура ТЭБ

- По содержимому (полнотекстовый поиск осуществляется с помощью механизма запросов Microsoft Search посредством функции ContainsTable());
- По категориям упрощенной классификации и по классам УДК (необходимо выбрать интересующую категорию или класс УДК из предложенного списка);
- По дате добавления книги в библиотеку.

На текущий момент автором разработана ER-модель БД, осуществлен переход к реляционной модели, разработана т.н. упрощенная классификация книг и механизм конвертации ее в УДК и обратно, реализованы процедуры аутентификации/регистрации, добавления книг, поиска и скачивания книг, управления сайтом. ТЭБ поступила в тестовое использование и уже наполняется книгами. Планируется усовершенствовать систему безопасности, реализовать автоматическое заполнение ПОД для книг в текстовом формате, повысить эффективность хранения и поиска информации за счет механизма выделения редко используемых книг и помещения их в отдельный архив, а также за счет технологии кластеризации.

Система может быть размещена на любом Web-сервере с установленной операционной системой Windows NT/2000/2003 Server, СУБД MS SQL Server 2000, .net FrameWork версии не ниже 1.1, Internet Information Server (IIS) версии не ниже 5.1. Web-сервер должен поддерживать технологию ASP.net.

## 2.4. Выводы

На сегодняшний день наиболее популярными технологиями разработки Web-приложений являются ASP.net (сервер IIS под управлением ОС Windows, СУБД MS SQL Server) и PHP (сервер Apache под управлением ОС Linux, СУБД MySQL).

Проведенный сравнительный анализ данных технологий показывает, что каждая из них обладает рядом достоинств и недостатков и выбор той или иной технологии зависит от целей создания web-приложения, его особенностей и выделенных средств.

Для разработки ТЭБ была выбрана технология ASP.net в связке IIS+Windows NT/2000/2003+MS SQL Server 2000.

Рассматривая особенности работы MS SQL Server 2000, мы отметили, что в комплект поставки данной СУБД входит система полнотекстового поиска по базе данных. Для работы с этой системой не требуются дополнительные программные средства, т.к. все операторы полнотекстового поиска включены в язык Transact-SQL.

Также мы рассмотрели схему общей архитектуры ТЭБ, где были выделены основные функциональные разделы и объяснено их назначение.

### 3. Реализация

#### 3.1. Проектирование базы данных библиотеки

##### 3.1.1. Определение множеств сущностей и атрибутов

В этом параграфе подробно опишем основные типы сущностей с их атрибутами.

###### 1) Множество сущностей «Книга»<sup>1</sup>

Атрибут	Описание атрибута	Тип данных
Книга_id	Уникальный идентификатор книги	Int
Название	Название книги	Nvarchar
Дата добавления	Дата добавления книги на сервер	Datetime
Скачена раз	Значение счетчика показывает, сколько раз данная книга была скачена с сервера	Int
Год издания	Год издания книги	Datetime
Размер файла	Размер файла документа (книги) в байтах	int
Краткое описание	Краткое описание содержания книги	Text
Ссылка	Местоположение файла книги на физическом диске (относительное)	Nvarchar
Количество страниц	Сведения о количестве страниц в книге	Int
Код ISBN	Уникальный код ISBN книги	Nvarchar
Номер тома/выпуска	Номер тома книги (если книга в нескольких томах) или номер выпуска периодического издания	Nvarchar
Место издания	Название населенного пункта (города или др.), где книга была издана	Nvarchar

###### 2) Множество сущностей «Пользователь»

Атрибут	Описание атрибута	Тип данных
Пользователь_id	Уникальный идентификатор пользователя	Int
Логин	Логин пользователя	Nvarchar
Пароль	Пароль пользователя	Nvarchar
Дата регистрации	Дата, когда пользователь прошел процедуру регистрации	Datetime
E-mail	Адрес электронной почты пользователя	Nvarchar

<sup>1</sup> Здесь и далее под «книгой» понимается электронный документ

Количество закачанных книг	Счетчик, значение которого показывает, сколько книг пользователь поместил в библиотеку (по умолчанию равен 0)	Int
Количество скачанных книг	Счетчик, значение которого показывает, сколько книг пользователь скачал из библиотеки (по умолчанию равен 0)	Int
Дата последнего посещения	Дата, когда пользователь последний раз прошел процедуру аутентификации	Datetime

### 3) Множество сущностей «Автор»

Атрибут	Описание атрибута	Тип данных
Автор_id	Уникальный идентификатор автора	Int
Имя	Имя автора	Nvarchar
Отчество	Отчество автора	Nvarchar
Фамилия	Фамилия автора	Nvarchar

### 4) Множество сущностей «Статус»

Атрибут	Описание атрибута	Тип данных
Статус_id	Уникальный идентификатор статуса	Int
Тип статуса	На данный момент выделяются два вида статусов – «администратор» и «читатель».	Nvarchar

### 5) Множество сущностей «Категория»

Атрибут	Описание атрибута	Тип данных
Категория_id	Уникальный идентификатор категории	Int
Имя категории	Название категории (относится к упрощенной классификации)	Nvarchar

### 6) Множество сущностей «Вид издания»

Атрибут	Описание атрибута	Тип данных
Вид издания_id	Уникальный идентификатор вида издания	Int
Вид издания	На данном этапе выделяются такие виды издания как «книга», «брошюра», «периодика», «специальное» и «другое»	Nvarchar

### 7) Множество сущностей «Язык издания»

Атрибут	Описание атрибута	Тип данных
Язык издания_id	Уникальный идентификатор языка издания	Int

Язык	Наименование естественного языка, на котором написана книга	Nvarchar
------	---	----------

#### 8) Множество сущностей «Обложка»

Атрибут	Описание атрибута	Тип данных
Обложка	Картинка, являющаяся обложкой книги	Image

#### 9) Множество сущностей «Текст книги»

Атрибут	Описание атрибута	Тип данных
Текст	Непосредственно текстовое содержание книги	Image
Расширение	Расширение файла книги, чье содержание загружается в поле «Текст». Необходимо для автоматического определения фильтра документа, в соответствии с которым будет проводиться полнотекстовое индексирование	Char

#### 10) Множество сущностей «Издательство»

Атрибут	Описание атрибута	Тип данных
Издательство_id	Уникальный идентификатор издательства	Int
Название издательства	Название издательства (издательского дома)	Nvarchar

#### 11) Множество сущностей «УДК»

Атрибут	Описание атрибута	Тип данных
УДК_id	Уникальный идентификатор класса УДК	Int
Номер УДК	Универсальный десятичный номер класса УДК	Nchar
Название УДК	Название класса УДК	Nvarchar

### 3.1.2. Определение множеств связей

В данном параграфе определим множества связей над множествами сущностей, описанных в п. 3.1.1. (см. табл. 3.1.2.1.).

Тип связи	Семантика связи	Тип отношения
Книги, добавленные пользователем	Каждая книга может быть добавлена в библиотеку одним и только одним	1:M

	пользователем. В свою очередь пользователь может добавлять в библиотеку неограниченное количество книг.	
Книга - автор(ы)	Одна книга может быть написана несколькими людьми, в то же время один и тот же автор может написать несколько книг. Учесть, что автор книги может быть неизвестен.	M:N
Статус пользователя	Каждый пользователь имеет определенный статус, причем несколько пользователей могут иметь один и тот же статус.	1:M
Категория книги	Каждая книга относится к определенной категории, причем несколько книг могут относиться к одной и той же категории.	1:M
Потомок категории	У каждой категории есть т.н. категория-родитель, причем только одна. У каждой же категории-родителя может быть сколько угодно много потомков. Корнем дерева категорий является категория «Неотсортированное».	1:M
Вид издания книги	Каждая книга относится к какому-либо виду издания, причем к одному и тому же виду издания могут относиться несколько книг.	1:M
Язык издания книги	Каждая книга написана на определенном естественном языке (мы полагаем, что если книга написана сразу на двух и более языках – один из них считать основным), причем несколько книг могут быть написаны на одном и том же языке.	1:M
Обложка книги	Некоторые книги могут иметь обложки, причем каждая книга имеет свою собственную неповторимую обложку.  Множество сущностей «Обложка» является слабым, по отношению к множеству сущностей «Книга».	1:1
Текст книги	Содержание книги в текстовом формате помещается в таблицу «Текст книги». Каждая такая книга имеет свое собственное содержание, отличное от других.  Множество сущностей «Текст книги» является слабым, по отношению к множеству сущностей «Книга».	1:1
Книга-издательство	Полагаем, что каждая конкретная книга издана одним издателем. Причем каждый издатель может издавать сколько угодно много книг.	1:M
Категория-УДК	Полагаем, что установлено соответствие: каждой категории из упрощенной классификации сопоставлен один/несколько	M:N

	классов УДК. И наоборот: одному классу УДК могут быть поставлены в соответствие одна/несколько категорий из упрощенной классификации.	
--	---	--

**Таблица 3.1.2.1. – Множества связей**

### 3.1.3. Определение ограничений целостности

Рассмотрим ограничения целостности по существованию (зависимость существования). В полученной ER – диаграмме множества сущностей «Обложка» и «Текст книги» зависимы по существованию от множества сущностей «Книга». Все зависимости существования и их описание приведены при определении множеств связей.

Рассмотрим ограничения целостности на значения некоторых атрибутов:

- Множество сущностей «Пользователь»: не может существовать двух пользователей с одинаковым значением атрибута «Логин»;
- Множество сущностей «Автор»: не может существовать двух авторов с одинаковыми значениями атрибутов «Фамилия», «Имя» и «Отчество»;
- Множество сущностей «Издательство»: не может существовать двух издателей с одинаковыми значениями атрибута «Название издательства».

Рассмотрим значения по умолчанию, заданные для некоторых атрибутов:

- Количество закачанных книг («Пользователь») = 0;
- Количество скачанных книг («Пользователь») = 0;
- Дата последнего посещения («Пользователь») = Дате регистрации;
- Скачана раз («Книга») = 0;

### 3.1.4. Построение ER-диаграммы

В процессе проектирования базы данных библиотеки была построена ER-диаграмма (см. рис. 3.1.4.1).

### 3.1.5. Трансформация ER-модели в реляционную модель

С учетом существующих правил преобразования ER-модели в реляционную модель, получаем следующую совокупность таблиц. На рис. 3.1.5.1 приведена реляционная схема базы данных «Библиотека».



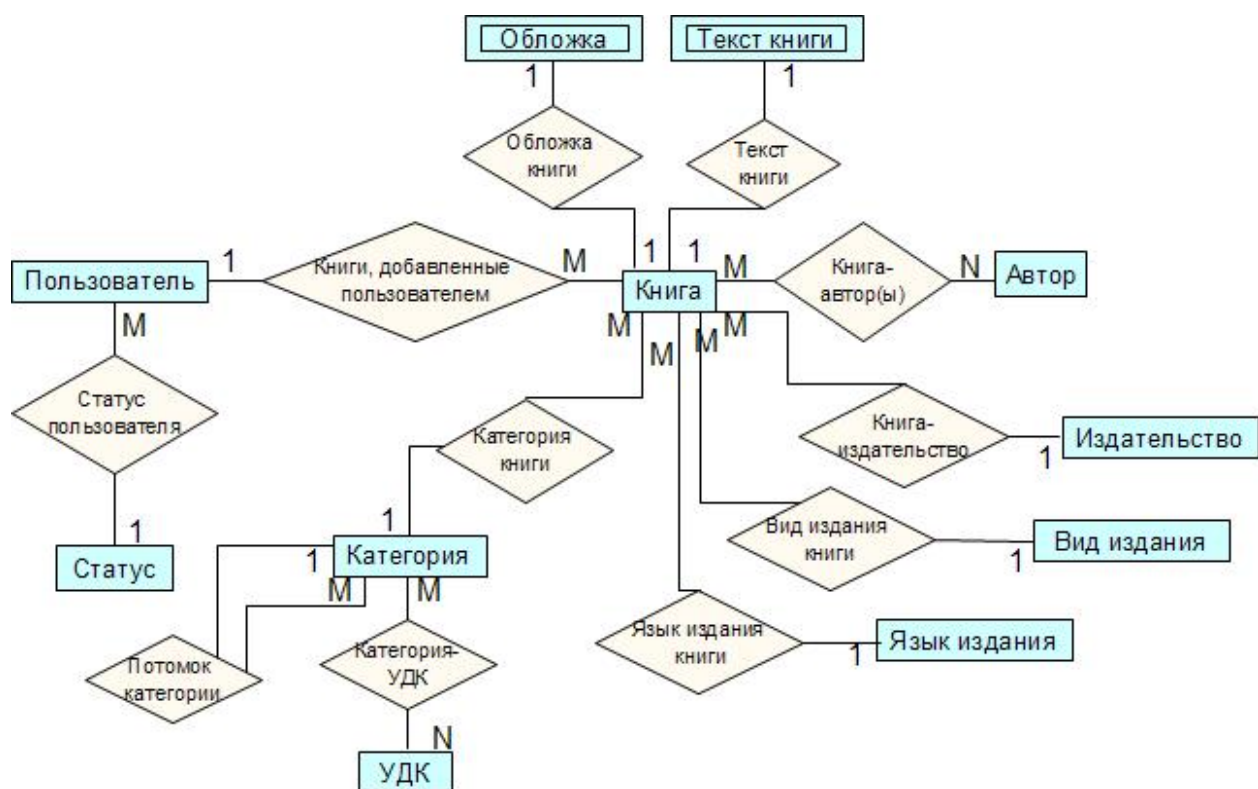


Рис. 3.1.4.1. – ER-диаграмма БД «Библиотека»

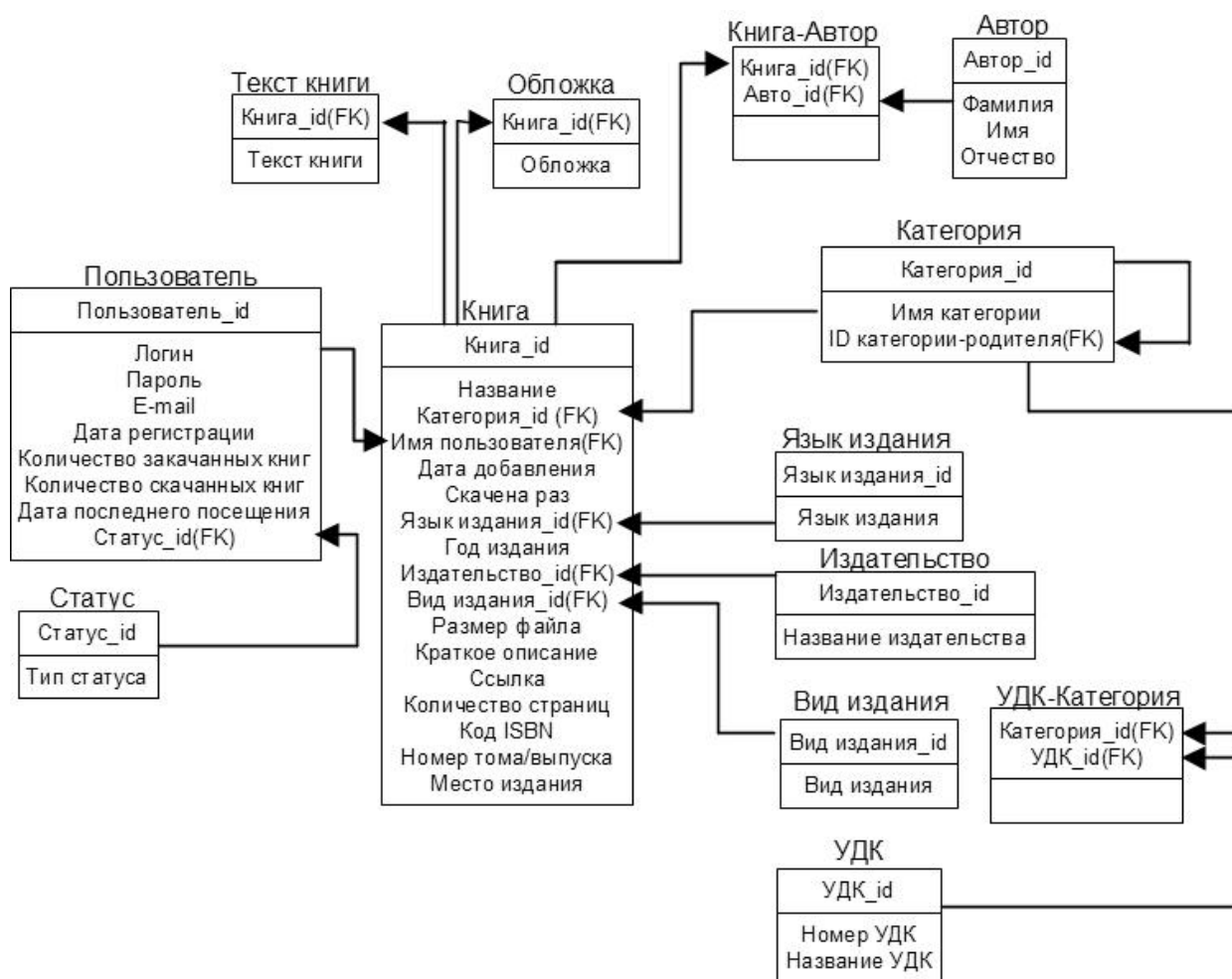


Рис. 3.1.5.1. – Реляционная схема БД «Библиотека»

## 3.2. Основные функциональные подсистемы ТЭБ

В данном параграфе рассмотрим основные функциональные подсистемы разработанной библиотеки.

### 3.2.1. Аутентификация/регистрация

В ASP.net предоставлены три способа аутентификации:

- Windows
- Формой
- Паспортом (требуется регистрация сайта в службе Passport)

Для нашего WEB-приложения была выбрана аутентификация формой по следующим причинам:

- Имена пользователей и пароли не хранятся в учетных записях Windows (хотя аутентификация на основе форм не запрещает применение учетных записей Windows);
- Приложение развертывается через Интернет;
- Требуется поддержка всех браузеров и клиентских операционных систем;
- В качестве страницы входа – собственная UI-форма;
- Есть возможность программно проверить имя пользователя и пароль.

В нашем WEB-приложении ранее зарегистрированный клиент идентифицируется с помощью временного Cookie-файла – небольшого блока данных, изначально передаваемого сервером клиенту. Затем в каждом HTTP-запросе cookie-файл передается обратно на сервер, и наличие такого блока может указывать, что клиент уже аутентифицирован. ASP.net предоставляет в модуле CookieAuthenticationProvider механизм, позволяющий использовать cookie при аутентификации на основе форм. Cookie поддерживаются большинством Web-браузеров, в том числе Internet Explorer и Netscape Navigator [10].

Информация о зарегистрированных пользователях хранится в базе данных MS SQL Server, причем пароли передаются на сервер и хранятся там не в открытом виде, а хешируются с помощью алгоритма “SHA1” (Secure Hash Algorithm версии 1).

### 3.2.2. Добавление/удаление книг

В системе ТЭБ любой зарегистрированный пользователь имеет право добавлять книги в библиотеку. Для этого он должен заполнить т.н. «карточку» книги, где обязательными полями являются:

- Основное заглавие;
- Категория;
- Язык;
- Вид издания;
- Сведения о количестве страниц;

- Файл книги.

Затем содержание данной формы передается на сервер, где заполненные поля (обязательные и необязательные) проверяются на корректность (в частности, с помощью серверных элементов управления - валидаторов). В случае ошибок заполнения полей книга добавлена не будет, а сервер вернет сообщение об ошибках заполнения, которые необходимо исправить. В случае корректного заполнения полей запускается механизм добавления, который выполняется в следующем порядке (если поле является необязательным и данные не были введены – шаг опускается):

1. Выделяет название издательства из поля «Издатель» и добавляет в БД в таблицу «Издательство», если сведений о таковом там еще нет;
2. Формирует параметры и добавляет сведения о книге в БД в таблицу «Книга»;
3. Выделяет авторов из поля «Авторы», и добавляет их в БД в таблицу «Автор», если сведений о таковых авторах там еще нет;
4. Добавляет записи в БД в таблицу «Книга-Автор»;
5. Добавляет содержимое файла с обложкой в БД в таблицу «Обложка»;
6. Если файл в текстовом формате – содержимое файла добавляется в БД в таблицу «Текст книги», иначе – файл загружается на сервер в специальный каталог /Data/.

Право удалять книги имеет пользователь со статусом «Администратор». Также, пользователь со статусом «Читатель» может удалить книгу, которая была им добавлена в течение часа с момента добавления. Механизм удаления книги работает следующим образом:

1. Удаляется запись из таблицы «Текст книги», если книга была в текстовом формате, либо удаляется файл из каталога /Data/, по ссылке, указанной в поле «Ссылка» в таблице «Книга»;
2. Удаляется запись из таблицы «Обложка», если у книги была обложка;
3. Удаляются записи из таблицы «Книга-Автор», если были указаны авторы книги;
4. Удаляется запись из таблицы «Книга».

### 3.2.3. Модификация поискового образа книги

Модифицировать поисковый образ (или «учетную карточку») любой книги имеет право пользователь со статусом «Администратор». «Читатель» же может редактировать поля книги, добавленной им самим в течение часа с момента добавления.

Для того, чтобы изменить значения полей ПОД, нужно перейти на web-страницу модификации. При этом в качестве параметра ей необходимо передать идентификатор модифицируемой книги. Сервер передает на форму те значения полей, которые были установлены ранее. Для изменения значения какого-либо поля необходимо ввести туда или выбрать из предложенного списка новое значение. Проверка на корректность вводимых данных производится аналогично как при добавлении книги (см. п. 3.2.2.).

### 3.2.4. Поиск книг

Пользователю предоставляется несколько способов локализации содержательной области книг, удовлетворяющих его информационной потребности:

- Система категоризации;
- Отбор по дате добавления;
- «Быстрый» поиск по «карточке» книги;
- Полнотекстовый поиск по содержанию.

Система категоризации работает следующим образом: после выбора категории из упрощенной классификации или классификации УДК формируется SQL-запрос к базе данных «Библиотека» и отбираются и отображаются только те книги, которые удовлетворяют запросу, а именно: книги выбранной категории.

Для удобства реализации возможности отбора книг в соответствии с *датой их добавления* на web-страницы помещен элемент управления “Calendar”. В соответствии с выбранной датой в этом элементе формируется SQL-запрос к БД.

Т.н. «*быстрый*» поиск – это полнотекстовый поиск по некоторым полям поискового образа книги (автор, название, краткое описание). Для реализации такого вида поиска, поля «Название» и «Краткое описание» из таблицы «Книга» и поле «Фамилия» из таблицы «Автор» были зарегистрированы для полнотекстового поиска. Данные этих полей индексируются механизмом MS Search в соответствии с настройками SQL-сервера. На текущий момент полнотекстовый индекс обновляется в соответствии с расписанием инкрементально (см. п. 2.2.4.) дважды в сутки. Для получения результатов используется основная функция полнотекстового поиска CONTAINS. Данная функция используется для того, чтобы определить, содержат или нет данные, хранящиеся в зарегистрированном для полнотекстового использования столбце, определенные слова и выражения. В случае «быстрого» поиска накладывается условие близкого расположения слов из поискового выражения за счет использования оператора NEAR().

Для осуществления *полнотекстового поиска по содержанию* для столбца «Текст» типа *image* таблицы «Текст книги» был также создан полнотекстовый индекс. Для получения результатов используется функция CONTAINSTABLE(), которая, кроме прочего, возвращает оценку соответствия: насколько хорошо строка удовлетворила критерию выборки. Создана возможность использования спецсимволов для генерации полнотекстовых запросов:

Синтаксис	Что означает оператор
&	Логическое И
	Логическое ИЛИ
+	Обязательное наличие слова
-	Обязательное отсутствие слова (И НЕ)
“ ”	Поиск фразы
()	Группирование слов
~	Слово должно идти после предыдущего

### 3.2.5. Скачивание книг

Скачивание интересующей книги происходит следующим образом:

- для книг в текстовом формате формируется временный файл, в который копируется содержимое поля «Текст» таблицы «Текст книги» и осуществляется переход по относительной ссылке на этот файл;
- для остальных книг осуществляется переход по относительной ссылке на файл (поле «Ссылка» в таблице «Книга»).

Сервер возвращает браузеру клиента документ, и механизм скачивания осуществляется непосредственно браузером.

### 3.2.6. Представление данных о web-ресурсе

Важно заинтересовать потенциального пользователя в посещении библиотеки, поэтому вне зависимости – зарегистрирован он или нет – ему предоставляется некоторая информация об объеме web-ресурса, такая, как

- количество книг;
- количество зарегистрированных пользователей;
- количество пользователей online;
- рейтинг «самые скачиваемые книги».

Количество книг и количество зарегистрированных пользователей отображаются как результаты запроса к базе данных к таблицам «Книга» и «Пользователь» соответственно: “Select count (\*) from Книга”, “Select count (\*) from Пользователь”

Рейтинг «самые скачиваемые книги» также определяется с помощью запроса к БД к таблице книга: “Select top 10 Название,[Скачена раз] from Книга order by [Скачена раз] DESC ”.

Количество пользователей online определяется как количество открытых сессий (Session) средствами ASP.net (в файле Global.asax).

### 3.2.7. Модификация категорий из упрощенной классификации

Добавлять, изменять и удалять категории из упрощенной классификации имеет право только пользователь со статусом «администратор».

При добавлении новой категории (или изменении существующей) необходимо указать категорию-родителя. Если новая (изменяемая) категория является корнем нового поддерева, то в качестве родительской следует указать категорию «Не отсортированные». Затем необходимо указать класс УДК, соответствующий добавленной (изменяемой) категории и подтвердить добавление (изменение).

При удалении категории необходимо учесть, что все книги, относящиеся к удаляемой категории будут автоматически перенесены в «Не отсортированные».

### **3.3. Выводы**

В данной главе было рассмотрено проектирование базы данных «Библиотека»: определение множеств сущностей, множеств связей, построение ER-модели и трансформация ее в реляционную модель. Также мы выделили основные функциональные подсистемы ТЭБ и рассмотрели их назначение и принципы реализации.

## Заключение

Электронные библиотеки и полнотекстовые базы данных становятся неотъемлемой частью деятельности почти в любой области: науки, культуры, образования, институтов власти и т.д. Для постоянных пользователей Интернета электронные архивы могут предложить разнообразные поисковые средства, позволяющие более точно найти искомый материал, доступ к мультимедийным ресурсам, чего никогда не смогут предоставить хранилища печатных материалов.

Результатом данной работы явилось создание электронной online-библиотеки для пользователей Томского Интернета со следующей спецификой:

- тематический охват неограничен;
- два принципа категоризации книг: в соответствии с УДК и с разработанной упрощенной классификацией;
- особенности работы для книг в текстовом формате: предоставление возможностей полнотекстового поиска по их содержанию.

В ходе работы были рассмотрены возможные варианты создания электронных библиотек, принципы организации информационного поиска по документам в хранилище, разработана и реализована архитектура для системы «Томская Электронная Библиотека»:

- спроектирована база данных;
- разработана т.н. упрощенная классификация книг и механизм конвертации ее в УДК и обратно;
- реализованы процедуры:
  - аутентификации и регистрации пользователей-«читателей»;
  - добавления, удаления книг;
  - модификации поискового образа книги;
  - поиска и скачивания книг;
  - представления данных об объеме web-ресурса и некоторых рейтинговых данных;
  - добавления, изменения и удаления категорий из упрощенной классификации;
- разработан базовый интерфейс для библиотеки.

Планируется усовершенствовать систему безопасности, реализовать автоматическое заполнение ПОД для книг в текстовом формате, повысить эффективность хранения и поиска информации за счет механизма выделения редко используемых книг и помещения их в отдельный архив, а также за счет технологии кластеризации, установить дополнительную поддержку русского языка для повышения качества полнотекстового поиска.

Система «Томская Электронная Библиотека» поступила в тестовое использование.

## Список использованных источников

1. [www.library.tver.ru](http://www.library.tver.ru) - Электронные библиотеки и полнотекстовые базы данных. // Губанова С.А. - Электронный читальный зал. Лекторий, 2003.
2. Большой энциклопедический словарь.-2-е изд., перераб. и доп.-М.: Большая Российская Энциклопедия, 1998.- 1456с.: ил.
3. [www.ci.ru](http://www.ci.ru) – «Эврика»: варианты построения электронного архива. // Пашков Юрий. - Электронный журнал «Компьютер-Информ», №16 (63) 1999.
4. [www.ci.ru](http://www.ci.ru) - Системы поиска.// Маглас М. – Электронный журнал «Компьютер-Информ», №23(208) 2005.
5. [www.infocity.kiev.ua](http://www.infocity.kiev.ua) – Хранение и поиск текстов в базах данных SQL Server 2000. // Крамарская Татьяна.- Виртуальный город компьютерной документации, раздел «Базы данных», 20.03.2005г.
6. Архитектура современного sql-сервера.  
<http://www.optim.su/cs/1999/2/sql7architecture/sql7archetecture.asp>
7. [www.science.viniti.ru](http://www.science.viniti.ru) - Универсальная десятичная классификация.// Всероссийский Институт Научной и Технической информации.
8. Ершова Г.Н. Информационные технологии в книжном деле. - Учебное пособие. - 2002 – 83с. <http://www.bookresearch.ru/itbook.htm>
9. Критерии оценки для научных электронных ресурсов. //Козлова Е.  
[http://www.viniti.msk.su/icsti\\_papers/russian/Kozlova.pdf](http://www.viniti.msk.su/icsti_papers/russian/Kozlova.pdf)
10. Рейли Д. Создание приложений Microsoft ASP.net. – Пер. с англ. – М.: Русская редакция, 2002. – 455 с.
11. PHP4 или ASP.net – что лучше. //Филатов Алексей.- Компьютерра+Новосибирск, №35 (559) 24.09.2004г.
12. Microsoft Corporation. Проектирование и реализация баз данных Microsoft SQL Server 2000. – Пер. с англ. – М.: Русская редакция, 2003. – 474с.
13. Мамаев Е. MS SQL Server 2000. - СПб: «БХВ-Петербург». – 2001. - 1280с.



## Приложение А. Руководство программиста.

Электронная библиотека ТЭБ состоит из базы данных Library под управлением СУБД MS SQL Server 2000 и web-интерфейса, разработанного в среде MS Visual Studio 2003 на базе платформы .net Framework 1.1 по технологии ASP.net 1.1.

База данных состоит из двух файлов: собственно файл с данными Library\_data.mdf и журнал транзакций Library\_log.ldf. По умолчанию эти файлы создаются и хранятся в папке /Microsoft SQL Server/mssql/data/.

Запуск сервера производится при помощи графической утилиты SQL Server Service Manager. Можно назначить автоматический запуск сервера при старте операционной системы. Другие параметры и свойства базы данных можно посмотреть или установить, используя консоль SQL Server Enterprise Manager. Там же предусмотрена возможность создания пользователей системы, назначение им прав доступа, создание заданий и др.

Создание базы данных, а также таблиц осуществлялось средствами графического интерфейса Enterprise Manager. Спроектированная БД состоит из 13 таблиц (см. п. 3.1.).

Для полнотекстового поиска были зарегистрированы следующие столбцы: Название («Книга»), Краткое\_описание («Книга»), Фамилия («Автор»), Текст («Текст\_книги»). В таблицы «Книга», «Автор» и «Текст\_книги» было добавлено специальное поле *timestamp*, дающее возможность настроить обновление полнотекстового каталога как инкрементальное. Полнотекстовый каталог *BookSearch* для всех таблиц один. Расписание обновления: дважды в сутки, инкрементальное.

### Хранимые процедуры БД:

- AddAuthor – добавляет новую запись в таблицу «Автор» и возвращает уникальный идентификатор добавленной записи;
- AddBook – добавляет новую запись в таблицу «Книга» и возвращает уникальный идентификатор добавленной записи;
- AddBookAuthor – добавляет новую запись в таблицу «Книга\_Автор»;
- AddCover – добавляет новую запись в таблицу «Обложка»;
- AddPub – добавляет новую запись в таблицу «Издательство» и возвращает уникальный идентификатор добавленной записи;
- AddText – добавляет новую запись в таблицу «Текст\_книги»;
- AddUser – добавляет новую запись в таблицу «Пользователь»;
- BookDownloaded – увеличивает счетчик «Количество\_скаченных\_книг» у пользователя, скачавшего книгу (таблица «Пользователь»); увеличивает счетчик «Скачено\_раз» у скачанной книги (таблица «Книга»);
- CounterInc – увеличивает счетчик «Количество\_закаченных\_книг» у пользователя, закачавшего книгу на сервер (таблица «Пользователь»);
- FindAuthor – возвращает идентификатор автора с заданными фамилией, именем и отчеством (причем, значения параметров «имя» и «отчество» могут быть <NULL>);
- FindLogin – возвращает количество пользователей с заданным логином;
- FindPub – возвращает идентификатор издательства с заданным названием;

- FindUser – возвращает количество пользователей с заданным логином и паролем;
- UpdateLastVisitDate – устанавливает значение поля «Дата\_последнего\_посещения» в таблице «Пользователь».

WEB-приложение Library представляет собой набор файлов, из которых выделим следующие:

- Файлы проекта Library.sln, Library.csproj, Library.csproj.webinfo;
- Группы файлов с расширениями aspx, aspx.cs, resx (web-формы):
  - Index (стартовая страница);
  - Default (основная страница);
  - Addbook (страница добавления книги);
  - BookAbout (страница просмотра подробной информации о книге);
  - BookModify (страница для изменения полей поискового образа книги);
  - Manage (страница для администратора);
  - Credentials (страница для изменения своих регистрационных данных);
- Группы файлов с расширениями ascx, ascx.cs, resx (пользовательские элементы управления):
  - LoginControl (элемент для ввода логина, пароля и последующей авторизации);
  - RegControl (элемент для заполнения полей регистрации);
  - MenuControl (элемент для перехода между страницами);
  - BookRepeater (элемент для отображения краткой информации о книгах библиотеки);
  - Calendar (элемент для отображения календаря);
  - Categories (элемент для отображения категорий из упрощенной классификации);
  - MostPopular (элемент для отображения списка наиболее скачиваемых книг);
  - Udk (элемент для отображения категорий УДК);
  - Volume (элемент для отображения данных об объеме web-ресурса);
  - LoginInfo (элемент для отображения логина авторизованного пользователя или слова «Гость», а также кнопки «Выход» для уничтожения временного cookie-файла)
- Классы данных c# (файлы с расширением cs):
  - BookInfo (класс, содержащий свойства и методы для удобной работы с атрибутами таблицы «Книга»);
  - Functions (класс, содержащий общедоступные методы);
- Файл таблицы стилей MyStyle.css;
- Файл конфигурации web.config;
- Группа файлов global.asax, global.asax.cs, global.asax.res (содержит методы, выполняющиеся в ответ на определенные события web-приложения).

## Functions

Рассмотрим группу общедоступных свойств и методов класса Functions.

### Свойства:

- ConnectionString – задает строку подключения к БД «Библиотека»;

- NumBooksOnPage – задает количество блоков с краткой информацией о книгах, которые будут отображены на web-странице default;

#### Методы:

- *string DeleteOverSymbols(string StringBefore, string sym, bool addsymend)* – удаляет из строки StringBefore повторяющийся символ sym, если addsymend=true, то добавляет в конец строки символ ‘;’;
- *ArrayList rprBookInfoFill(string CommandString, int PageNum)* – выполняет запрос к БД (текст запроса в CommandString) к таблице «Книга», выбирает те книги, которые будут отображены на странице default (количество книг определяется глобальной переменной NumBooksOnPage, а переменная PageNum по сути определяет, какие именно книги из всех, удовлетворяющих критерию выборки, будут отображены), формирует и возвращает ArrayList, элементами которого являются экземпляры класса BookInfo;
- *BookInfo BookInfoFill(SqlDataReader drBook, bool extra, bool fullText)* – заполняет и возвращает экземпляр класса BookInfo (в соответствии со значением переменных extra и fullText учитываются либо нет некоторые дополнительные свойства класса BookInfo);
- *ArrayList ddlCategoryFill(ArrayList values, DataTable dt, int parent\_id, string st\_level, string st\_catname)* – формирует и возвращает ArrayList, чьими элементами являются названия категорий упрощенной классификации, сформированные в удобочитаемой форме в соответствии с деревом категорий;
- *string FullCategoryName(int catID, DataTable DT, string curStr)* – по заданному идентификатору категории возвращает ее «полное» имя (т.е. с учетом отношения родитель-потомок);
- *bool AdminOrNot(string login)* – возвращает значение, обладает ли пользователь с заданным логином статусом «администратор»;

Рассмотрим группу классов, соответствующих web-страницам.

### Index

Назначение web-страницы: стартовая страница, предоставляет некую вводную информацию о содержании сайта, предлагает зарегистрироваться и /или авторизоваться;

Размещенные пользовательские элементы управления: Volume, Menu, MostPopular, Categories, Calendar, LoginControl, RegControl, LoginInfo;

Другие элементы управления (кратко): приветствие;

Методы: нет;

Передаваемые параметры и их назначение: нет.

### Default

Назначение web-страницы: основная страница, служит для предоставления блоков с краткой информацией о книгах, удовлетворяющих критерию выборки;

Размещенные пользовательские элементы управления: Volume, Menu, MostPopular, Categories, Calendar, LoginInfo, UDK, BookRepeater;

Другие элементы управления (кратко): две под-формы для осуществления быстрого поиска и поиска по содержанию;

Методы:

*PageLoad()* – если пользователь не авторизован – перенаправляет его на страницу index, формирует строку отображения критерия отбора книг и показывает ее;

*btnFastSearch\_click()* – по нажатию на кнопку «быстрый поиск» формирует соответствующие параметры и страница обновляется;

*btnFullSearch\_click()* – аналогично, по нажатию на кнопку «поиск по содержанию»;

Передаваемые параметры и их назначение: Параметры задают критерий выборки книг:

- [“catid”] – идентификатор категории из упрощенной классификации;
- [“udkid”] – идентификатор категории УДК;
- [“date”] – дата добавления книги;
- [“ss”] – поисковое выражение для «быстрого» поиска;
- [“a”] – поиск по автору;
- [“t”] – поиск по названию;
- [“d”] – поиск по описанию;
- [“Fs”] – поисковое выражение для поиска по содержанию.

Обработкой значений переданных параметров занимается пользовательский элемент управления BookRepeater (размещен на странице default).

## AddBook

Назначение web-страницы: предназначена для добавления книг на сервер (заполнения соответствующих полей для описания книги и указания ее физического файла);

Размещенные пользовательские элементы управления: Volume, Menu, MostPopular, Categories, Calendar, LoginInfo;

Другие элементы управления (кратко): под-форма с полями для заполнения полей для описания книги, строка для сообщений об ошибках, два элемента input-file для выбора файла книги и файла обложки, кнопка «Добавить книгу»;

Методы:

*PageLoad()* – если пользователь не авторизован – перенаправляет его на страницу index, формирует выпадающие списки для полей «Категория», «Вид издания», «Язык издания».

*btnAddBook\_click()* – в нужном порядке выполняются следующие процедуры и функции:

- *AddingBookCheck* – проверка на корректность введенных значений в поля формы и другие проверки; В случае успеха – добавление книги на сервер; в случае неудачи – вывод сообщения об ошибках;
- *FileUpload* – загрузка файла на сервер либо содержимого файла в БД;
- *AddPub* – добавление (в случае необходимости) новой записи в БД в таблицу «Издательство»;

- *AddBook* – добавление новой записи в БД в таблицу «Книга»;
- *AuthorAdd* – добавление (в случае необходимости) новой записи в БД в таблицу «Автор»;
- *AddCover* – добавление (в случае наличия) содержимого файла обложки в БД в таблицу «Обложка»;
- *AddText* – добавление (в случае, если файл – текстовый) содержимого файла книги в БД в таблицу «Текст\_книги»;
- *AuthorCheck* – проверка на корректность заполнения поля «Автор(ы)» на форме;

Передаваемые параметры и их назначение: нет.

## BookAbout

Назначение web-страницы: предназначена для вывода полной информации о конкретной книге;

Размещенные пользовательские элементы управления: Volume, Menu, MostPopular, Categories, Calendar, LoginInfo;

Другие элементы управления (кратко): под-форма для вывода значений полей описания книги, кнопки «скачать», «редактировать», «удалить» (две последних доступны только для администратора, либо для пользователя, добавившего книгу, в течение часа с момента добавления);

### Методы:

*PageLoad()* – если пользователь не авторизован – перенаправляет его на страницу index, заполнение полей формы для вывода информации об интересующей книге;

*btnDownload\_click()* – предоставляет браузеру возможность осуществить механизм скачивания (с учетом специфики книги – наличия внешнего файла, либо содержимого файла в бинарном формате в БД);

*btnModify\_Click()* – перенаправляет пользователя (если он обладает правами) на страницу редактирования книги;

*btnDelete\_Click()* – производит действия по удалению книги (из БД и реального файла);

Передаваемые параметры и их назначение: [“bookid”] – идентификатор книги, подробную информацию о которой необходимо отобразить.

## BookModify

Назначение web-страницы: предоставляет возможность редактировать поля описания книги;

Размещенные пользовательские элементы управления: Volume, Menu, MostPopular, Categories, Calendar, LoginInfo;

Другие элементы управления (кратко): под-форма для редактирования значения полей описания книги, кнопка «Редактировать»;

### Методы:

*PageLoad()* – если пользователь не авторизован – перенаправляет его на страницу index, заполнение полей формы для вывода текущей информации об интересующей книге;

*btnModifyClick()* – подтверждение редактирования, выполняются аналогичные функции, описанные в классе AddBook, за той разницей, что в БД не добавляется новая запись, а редактируется уже существующая;

Передаваемые параметры и их назначение: [“bookid”] – идентификатор книги, поля которой необходимо отредактировать.

## Manage

Назначение web-страницы: т.н. страница «управления сайтом», предоставляет администратору возможность править категории упрощенной классификации: добавлять, удалять, изменять;

Размещенные пользовательские элементы управления: Volume, Menu, MostPopular, Categories, Calendar, LoginInfo;

Другие элементы управления (кратко): поле для ввода имени категории, выпадающие списки уже существующих категорий и УДК, кнопки «Добавить», «Установить соответствие с УДК», «Удалить»;

### Методы:

*PageLoad()* – если пользователь не обладает статусом «администратор» – перенаправляет его на страницу default;

*btnAddCategory\_Click(), btnChangeCategory\_Click(), btnDeleteCategory\_Click()* – функции, обращающиеся к БД, для выполнения выбранных действий;

Передаваемые параметры и их назначение: нет.

## Credentials

Назначение web-страницы: страница просмотра и изменения регистрационных данных;

Размещенные пользовательские элементы управления: Volume, Menu, MostPopular, Categories, Calendar, LoginInfo;

Другие элементы управления (кратко): поля для просмотра и ввода информации, кнопка «Редактировать»;

### Методы:

*PageLoad()* – если пользователь не авторизован – перенаправляет его на страницу index, выводит информацию о пользователе;

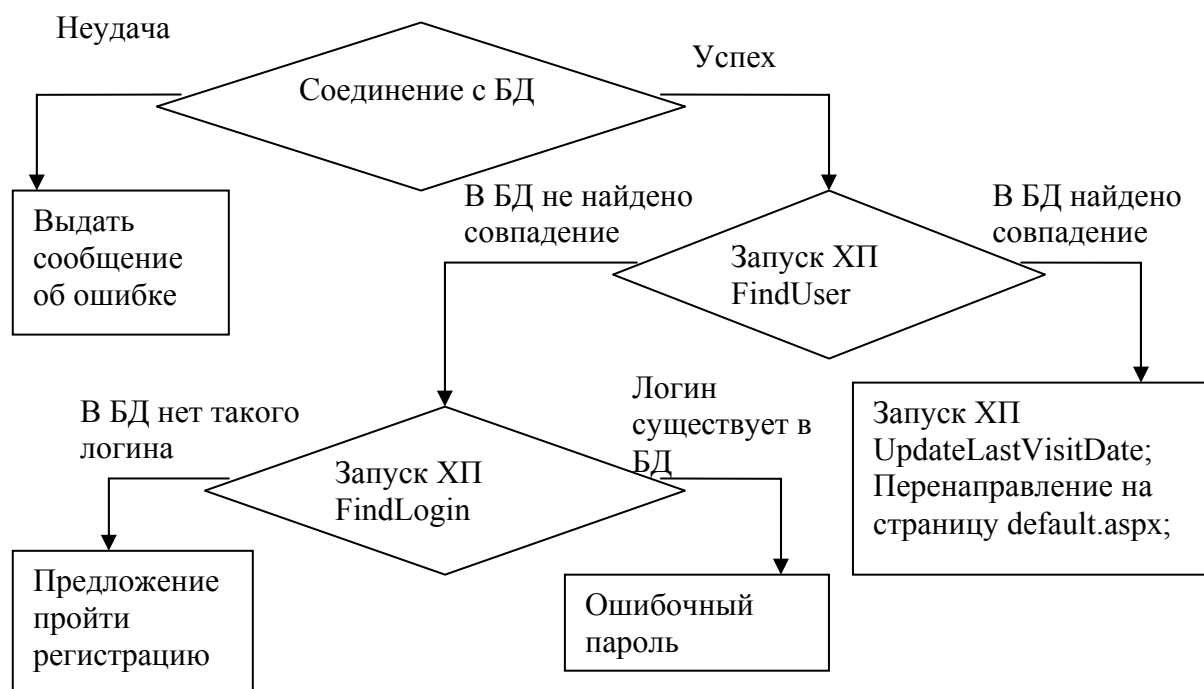
*btnModify\_Click()* – проверяет корректность введенных данных, обращается к БД и изменяет данные пользователя (такие, как пароль и e-mail);

Передаваемые параметры и их назначение: нет.

Рассмотрим группу классов, соответствующих пользовательским элементам управления.

## LoginControl

Пользовательский элемент управления LoginControl представляет из себя WEB-форму, с размещенными на ней полями для ввода логина и пароля, меткой для вывода сообщений и кнопкой «Вход». По нажатию на кнопку «Вход» выполняется процедура LoginProcedure() со следующей семантикой (см. рис. А.1)<sup>2</sup>.



**Рис. А.1 – Схема процедуры аутентификации LoginProcedure()**

### RegControl

Пользовательский элемент управления RegControl представляет из себя WEB-форму, с размещенными на ней полями для ввода регистрационных данных: логина и пароля (с предложением повтора пароля), e-mail; меткой для вывода сообщений, валидаторами (элементами для контроля вводимых данных), и кнопкой «Зарегистрироваться». По нажатию на кнопку «Зарегистрироваться» выполняется процедура RegProcedure() со следующей семантикой (см. рис. А.2).

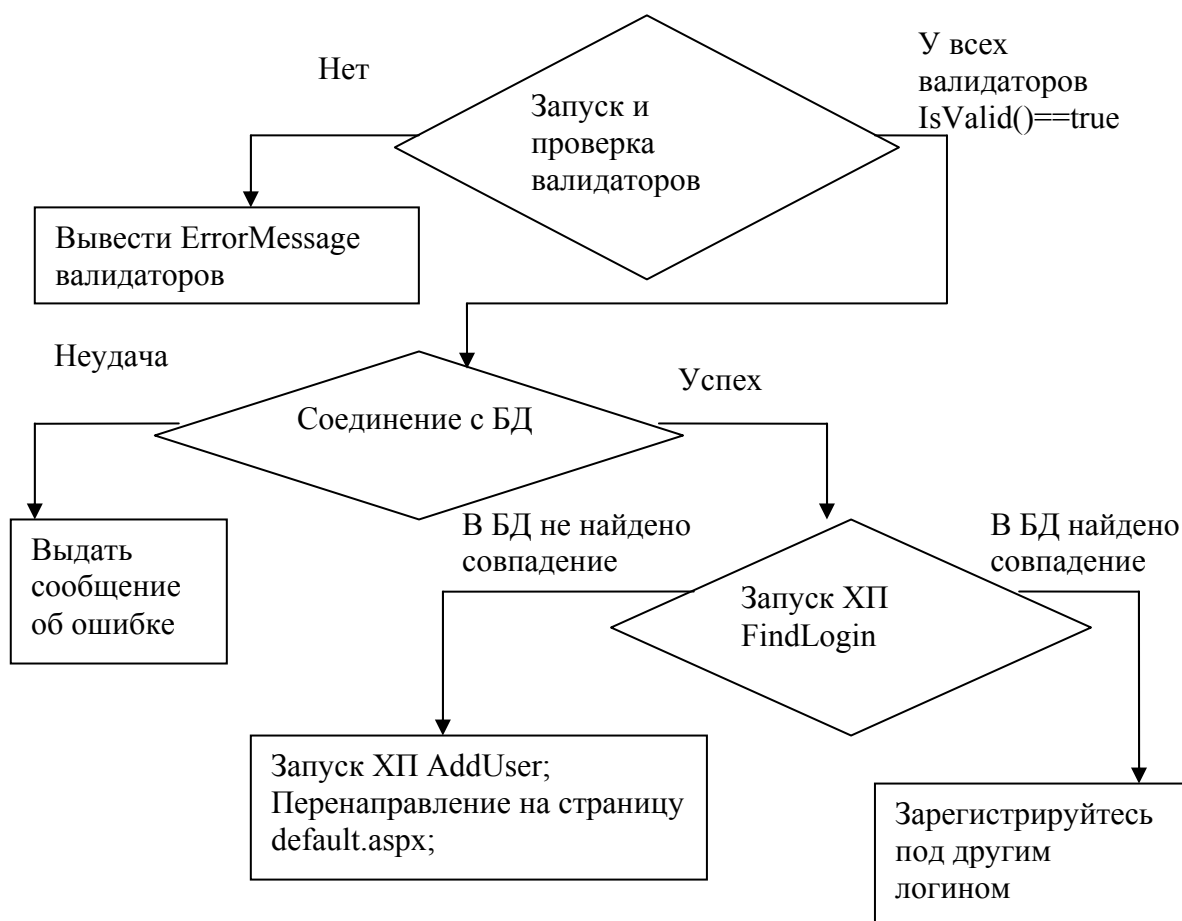
### MenuControl

Пользовательский элемент управления MenuControl содержит 4 ссылки (asp-компоненты LinkButton) на соответствующие web-страницы. Переход осуществляется вызовом метода Redirect(<имя страницы>) класса Response.

### BookRepeater

Основное назначение пользовательского элемента управления BookRepeater состоит в отображении блоков с краткой информацией о книгах, отображенных в соответствии с критерием выборки.

<sup>2</sup> Примечание: на рис. А.1 и рис. А.2 ХП – хранящая процедура (сокращ. автора)



**Рис. А.2 – Схема процедуры регистрации RegProcedure()**

Зачастую книг, удовлетворяющих критерию выборки, слишком много, и отображать *все* результаты выборки на одной web-странице нецелесообразно. Поэтому все множество отобранных книг разбивается на подмножества и отображается по частям. Количество книг в одном таком подмножестве определяется глобальной переменной NumBooksOnPage класса Functions. В соответствии со значением этой переменной и количеством книг, удовлетворяющих условию выборки, определяется общее количество «подстраниц».

#### Методы:

*PageLoad()* - происходит считывание значений параметров, переданных странице default, в зависимости от значений параметров формируется строка запроса к БД на выборку книг. Считается количество «подстраниц», и посредством двух элементов типа Repeater отображаются ссылки на «подстраницы» и сами блоки с информацией о книгах. В блоке с информацией о книгах, название книги само является ссылкой на web-страницу bookabout.

*String CreateStringSql(string controlStr)* – функция формирует строку запроса на полнотекстовый поиск по содержанию к БД в случае, когда в поисковом выражении отсутствуют спецсимволы.

*string CreateStringSQLLogogram(string controlStr)* – аналогичная функция, в случае, когда в поисковом выражении присутствуют спецсимволы.



## Calendar

Пользовательский элемент управления Calendar предназначен для удобного выбора даты добавления книги, как критерия отбора.

### Метод:

*Calendar1\_SelectionChanged()* – происходит перенаправление на страницу default с выбранным значением параметра [“date”].

## Categories

Данный элемент предназначен для вывода названий категорий из упрощенной классификации. Он предоставляет возможность выбрать категорию, тем самым задать значение параметра [“catid”] и отобразить книги, принадлежащие к данной категории. Для этого используется компонент Repeater.

### Метод:

*PageLoad()* – компонент Repeater заполняется значениями полей «Имя категории» из таблицы «Категория» в том порядке, чтобы прослеживались отношения «родитель-потомок».

## MostPopular

Элемент предназначен для вывода списка десяти самых скачиваемых книг (выводит название книги и счетчик – сколько раз она была скачена). Предоставляет возможность, щелкнув на названии книги, перейти к просмотру подробной информации о книге. Использует компонент Repeater.

### Метод:

*PageLoad()* – поступает запрос к БД, результаты которого отображаются посредством элемента Repeater.

## UDK

Элемент представляет собой DropDownList, связанный с БД с таблицей «УДК». Выбор элемента из списка формирует значение параметра [“udk”], который передается странице default, что приводит к отображению книг, относящихся к выбранному классу УДК (запрос к таблице «УДК-Категория»).

*PageLoad()* – связывает DropDownList с БД с таблицей «УДК»;

*ddlUDK\_SelectedIndexChange()* – формирует параметр [“udk”] и перенаправляет на страницу default.

## Volume

Предоставляет информацию о количестве книг в библиотеке, количестве зарегистрированных пользователей и количестве пользователей (в том числе и незарегистрированных) online.

*PageLoad()* – формирует запрос к БД к таблицам «Пользователь» и «Книга», отображает счетчики.

Количество открытых сессий сохраняется в переменной “sessionCount” сохраненной в классе Application. Данная переменная увеличивается при открытии новой сессии, и уменьшается при закрытии сессии на единицу (процедуры *SessionStart()* и *SessionEnd()* класса Global.asax).

## LoginInfo

Данный элемент размещен на каждой web-странице с целью отображения информации: авторизован пользователь или нет.

### Компоненты и методы:

Label (lblLogin) – отображает логин авторизованного пользователя (Context.User.Identity.Name), либо статус «Гость»;

*lblLogin\_Click()* – переход на страницу Credentials для просмотра/правки информации о пользователе;

LinkButton (btnExit) – кнопка предлагает выйти из статуса «авторизован»;

*btnExit\_Click()* – вызывает метод SignOut() класса FormsAuthentication, перенаправляет на страницу index.

## Приложение Б. Руководство пользователя

Для того, чтобы воспользоваться услугами Томской Электронной Библиотеки, необходимо иметь доступ к сети Интернет. Запустите браузер Microsoft Internet Explorer (или другой), в адресной строке браузера введите URL: [www.infonet.tsu.ru](http://www.infonet.tsu.ru). (рис. Б.1.)



Рис. Б.1. - Ввод URL-адреса

После загрузки в окне вашего браузера должна отобразиться стартовая страница библиотеки (рис. Б. 2.). Здесь вы можете ознакомиться с некоторой общей информацией о библиотеке.

The screenshot shows the homepage of the Tomsk Electronic Library. The header features the library's name and a logo of an open book. On the right, it indicates the user is a "Guest" (Гость) with a "Logout" (Выход) link. The main content area is divided into several sections: a "Menu" (МЕНЮ) with links to Home, Add book, Catalog, and Site management; a "Greeting" (ПРИВЕТСТВИЕ) section with a welcome message and instructions for new and returning users; a "Most downloaded" (САМЫЕ СКАЧИВАЕМЫЕ) list of books like "Plato" and "Harry Potter"; a "Book categories" (КАТЕГОРИИ КНИГ) list including "Unsorted" and "Computers"; and a calendar for June 2006. On the right side, there are sections for "Authentication" (АУТЕНТИФИКАЦИЯ) with login and password fields, and "Registration" (РЕГИСТРАЦИЯ) with fields for login, password, email, and a "Register" button. A footer at the bottom provides contact information.

Рис. Б.2. – Стартовая страница (index)

Пока ваш статус не авторизован – вы не сможете зайти на другие страницы, поэтому необходимо зарегистрироваться (под-форма «Регистрация»), или авторизоваться (под-форма «Аутентификация»).

В случае успешной аутентификации/регистрации вы будете перенаправлены на основную страницу сайта (см. рис. Б. 4.), а в случае ошибки (например, некорректного ввода пароля) вы будете извещены об этом (например, см. рис. Б. 3).

**РЕГИСТРАЦИЯ**

Логин: \*  
 Не указан логин

Пароль:

Еще раз пароль:

E-mail:  
 sya1401@mail.ru

**Зарегистрироваться**

Введенные пароли не совпадают!

Рис. Б. 3. – Ошибки регистрации

**ТОМСКАЯ ЭЛЕКТРОННАЯ БИБЛИОТЕКА**

Всего книг: 12  
 Читателей: 7  
 Online читателей и гостей: 1

**МЕНЮ**

- Главная
- Добавить книгу
- Каталог книг
- Управление сайтом

**САМЫЕ СКАЧИВАЕМЫЕ**

- » Платон (5)
- » Магия дня недели (3)
- » Маятник Фуко (2)
- » Компьютерные сети (2)
- » Ночной дозор (1)
- » Стена (1)
- » Гарри Поттер и философский камень (1)
- » Алхимик (1)
- » Законы магии (0)
- » Test (0)

**КАТЕГОРИИ КНИГ**

- » Неотсортированное
- » компьютеры
- » --> операционные системы
- » --> программирование
- » --> базы данных
- » --> веб-дизайн
- » фантастика
- » философия

**СТРАНИЦЫ**

12

**Алхимик**  
 Категория: Неотсортированное  
 Автор(ы): Коэльо Пауло ;  
 Добавил: anet  
 Дата: 04.06.2006 16:02:00  
 Скачена раз: 1  
 Язык: русский  
 Степень соответствия: 1000

**Гарри Поттер и философский камень**  
 Категория: Неотсортированное  
 Автор(ы): Роулинг Джоанна ;  
 Добавил: anet  
 Дата: 04.06.2006 15:50:49  
 Скачена раз: 1  
 Язык: русский  
 Степень соответствия: 1000

**Законы магии**  
 Категория: Неотсортированное  
 Автор(ы): Неизвестен  
 Добавил: admin  
 Дата: 03.06.2006 13:17:57  
 Скачена раз: 0  
 Язык: русский  
 Степень соответствия: 1000

**Test**  
 Категория: Неотсортированное  
 Автор(ы): Неизвестен  
 Добавил: admin  
 Дата: 30.05.2006 11:09:31  
 Скачена раз: 0  
 Язык: русский  
 Степень соответствия: 1000

**БЫСТРЫЙ ПОИСК**

☒ в авторах  
☒ в названии  
☒ в описании

**ПОЛНОТЕКСТОВЫЙ ПОИСК**

Правила формирования строки для полнотекстового поиска по содержанию

Синтаксис	Что означает оператор	Пример
&	Логическое И	Программист & Тестировщик
	Логическое ИЛИ	Программист   Тестировщик
+	Обязательное наличие слова	Программист +Тестировщик
-	Обязательное отсутствие слова (И НЕ)	Программист - Тестировщик
" "	Поиск фразы (цитаты)	"Я к вам пишу"
()	Группирование слов	(Программист Тестировщик)
~	Слово должно идти после предыдущего	Программист ~Тестировщик

**УДАК**

000 Общий класс

Рис. Б. 4. – Основная страница (default)

После того, как ваш статус авторизован, на любой странице в правом верхнем углу вы можете увидеть свой логин и кнопку «Выход». По нажатию на кнопку «Выход» ваш статус перестанет быть авторизованным, вас перенаправят на главную страницу и в правом верхнем углу будет отображено слово «Гость».

После авторизации вы можете просмотреть и/или изменить свои данные. Для этого щелкните в правом верхнем углу на строчку с вашим логином, так вы будете перенаправлены на страницу credentials (см. рис. Б. 5.).

Томская  
ЭЛЕКТРОННАЯ  
БИБЛИОТЕКА

Вы вошли как:  
**Гость**  
Выход

Всего книг: 12  
Читателей: 7  
Online читателей и гостей: 1

**МЕНЮ**

Главная  
Добавить книгу  
Каталог книг  
Управление сайтом

**САМЫЕ СКАЧИВАЕМЫЕ**

»Платон (5)  
»Магия дня недели (3)  
»Маятник Фуко (2)  
»Компьютерные сети (2)  
»Ночной дозор (1)  
»Стена (1)  
»Гарри Поттер и философский камень (1)  
»Алхимик (1)  
»Законы магии (0)

**Ваши данные**

Логин: Mouse  
Пароль:   
Еще раз пароль:   
E-mail:   
Статус: читатель  
Последний визит: 6.06.06 16:06:06  
Дата регистрации: 27.04.06 18:25:12  
Закачал книг: 3

**Рис. Б.5. – Страница с личными данными пользователя (credentials)**

Навигация между страницами осуществляется с помощью выбора пунктов меню:

- Главная – перенаправление на главную страницу;
- Добавить книгу – перенаправление на страницу добавления книги;
- Каталог книг – перенаправление на основную страницу;
- Управление сайтом – доступна лишь для пользователей со статусом администратор (см. ПРИЛОЖЕНИЕ В. Руководство администратора).

По умолчанию на основной странице (default) представлены книги библиотеки, упорядоченные по дате добавления (вы можете, используя календарь в левом нижнем углу, выбрать интересующую вас дату). Естественно, что книг может быть очень много, поэтому книги отображаются «блоками». Для перехода к следующему «блоку» нужно выбрать номер под-страницы (например, на рис. Б.4. возможен переход ко второй под-странице, где будут отображены другие книги).

Вы можете задавать критерий выборки книг, которые будут представлены на основной странице. Для этого вы можете выбрать интересующую категорию, или класс УДК (см. рис. Б.6).

Также вы можете воспользоваться информационным поиском, который предоставляет ТЭБ. Вы можете ввести поисковое выражение в поле под-формы «Быстрый поиск» (по полям описания книги) или под-формы «Полнотекстовый поиск» (по содержанию книг, представленных в текстовом формате). Нажмите на кнопку «Искать» и вы получите результаты отбора. Обратите внимание на поле «Степень соответствия» - это

поле может принимать значения от 0 до 1000 (0 – самая маленькая степень соответствия критерию выборки, 1000 – самая большая).



Рис. Б.6. – Выборка книг по категориям и по УДК

Итак, на странице default представлены небольшие блоки с краткой информацией о книгах, удовлетворяющих критерию выборки (см. рис. Б.7.). Вы можете щелкнуть по названию книги и перейти на страницу bookabout, где будет представлена более полная информация о книге и возможность ее скачать (см. рис. Б.8.).

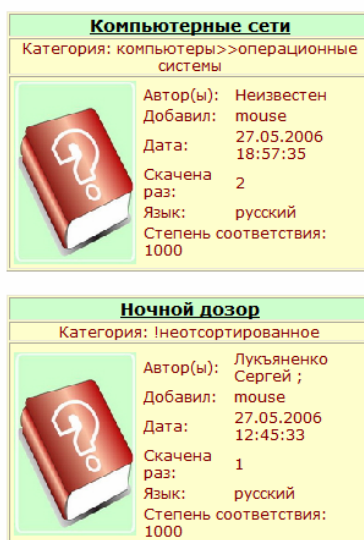


Рис. Б.7. – Блоки с краткой информацией о книгах



Вы вошли как: **mouse**  
[Выход](#)

Всего книг: 12  
 Читателей: 7  
 Online читателей и гостей: 1

**МЕНЮ**  
[Главная](#)  
[Добавить книгу](#)  
[Каталог книг](#)  
[Управление сайтом](#)

**САМЫЕ СКАЧИВАЕМЫЕ**  
[»Платон \(5\)](#)  
[»Магия дня недели \(3\)](#)  
[»Маятник Фуко \(2\)](#)  
[»Компьютерные сети \(2\)](#)  
[»Ночной дозор \(1\)](#)  
[»Стена \(1\)](#)  
[»Гарри Поттер и философский камень \(1\)](#)  
[»Алхимик \(1\)](#)  
[»Законы магии \(0\)](#)

**Компьютерные сети**  
 Автор(ы): Неизвестен  
 Категория: компьютеры>>операционные системы  
 УДК: 004 Вычислительная техника  
 Издатель: Не указано  
 Дата издания:  
 Место издания:  
 Вид издания: книга  
 Код ISBN:  
 Язык: русский  
 Сведения о количестве страниц: 30  
 Номер тома/выпуска:  
 Добавил(а): mouse 27.05.2006 18:57:35  
 Число скачиваний: 2  
 Размер файла: 564 Кб  
 Тип файла: pdf

[Скачать](#)  
[Редактировать](#)  
[Удалить](#)

Рис. Б.8. – Страница с полной информацией о книге (bookabout)

Нажмите на кнопку «Скачать» и браузер предложит открыть документ, либо сохранить его на вашем компьютере.

Кнопки «Редактировать» и «Удалить» будут доступны вам лишь в том случае, если вы являетесь администратором, либо если вы сами добавили выбранную книгу не ранее, чем час назад. Нажатие на кнопку «Редактировать» перенаправит вас на страницу редактирования bookmodify, где вы можете отредактировать поля книги (см. рис. Б.9.)

Вы вошли как: **admin**  
[Выход](#)

Всего книг: 12  
 Читателей: 7  
 Online читателей и гостей: 2

**МЕНЮ**  
[Главная](#)  
[Добавить книгу](#)  
[Каталог книг](#)  
[Управление сайтом](#)

**САМЫЕ СКАЧИВАЕМЫЕ**  
[»Платон \(5\)](#)  
[»Компьютерные сети \(4\)](#)  
[»Магия дня недели \(3\)](#)  
[»Маятник Фуко \(2\)](#)  
[»Ночной дозор \(1\)](#)  
[»Стена \(1\)](#)  
[»Гарри Поттер и философский камень \(1\)](#)  
[»Алхимик \(1\)](#)  
[»Законы магии \(0\)](#)  
[»Test \(0\)](#)

**КАТЕГОРИИ КНИГ**

Основное заглавие: Ночной дозор  
 Автор(ы): Лукьяненко Сергей  
 Категория: !неотсортированное  
 Язык: неотсортированное  
 Год издания (выпуска): компьютеры  
 Издательство: -->операционные системы  
 Место издания: -->программирование  
 Вид издания: -->базы данных  
 Код ISBN: -->веб-дизайн  
 Сведения о количестве страниц: фантастика  
 № тома/выпуска: философия  
 Обложка: [Browse...](#)  
 Краткое описание:

95  
 Старое краткое описание книги

[Изменить поля](#)

Рис. Б.9. – Страница редактирования полей описания книги

Также, любой зарегистрированный пользователь библиотеки может добавлять в нее книги. Для этого нужно перейти на страницу добавления книги и заполнить необходимые поля под-формы (см. рис. Б.10.)

**ТОМСКАЯ ЭЛЕКТРОННАЯ БИБЛИОТЕКА**

Вы вошли как: **admin**  
[Выход](#)

Всего книг: 12  
Читателей: 7  
Online читателей и гостей: 2

**МЕНЮ**

- [Главная](#)
- [Добавить книгу](#)
- [Каталог книг](#)
- [Управление сайтом](#)

**САМЫЕ СКАЧИВАЕМЫЕ**

- » Платон (5)
- » Компьютерные сети (4)
- » Магия дня недели (3)
- » Маятник Фуко (2)
- » Ночной дозор (1)
- » Стена (1)
- » Гарри Поттер и философский камень (1)
- » Алхимик (1)
- » Законы магии (0)
- » Test (0)

**КАТЕГОРИИ КНИГ**

- » Неотсортированное
- » компьютеры
- » --> операционные системы
- » --> программирование
- » --> базы данных
- » --> веб-дизайн
- » фантастика
- » философия

**ПРАВИЛА ДОБАВЛЕНИЯ КНИГУ**

Прежде чем добавить книгу на сервер, убедитесь, пожалуйста, что таковой книги там еще нет. Одинаковые книги будут удаляться в пользу первого закачавшего.  
Заполните необходимые поля на форме. Учтите, что допустимые расширения для файла обложки - jpg/jpeg, gif, bmp, а для файла книги - rar, zip, djv/djvu, pdf, doc, chm, txt.  
Обратите внимание на поле "Автор". Если у книги один автор, то заполнять это поле можно следующим образом:  
Фамилия Имя Отчество  
Фамилия Имя  
Фамилия И. О.  
Фамилия И.  
Фамилия  
Если авторов несколько - поле "Автор" заполняется по принципу:  
Автор(,Автор)  
Например:  
Иванов Иван; Петров; Сидоров В. И.

Основное заглавие:

Автор(ы):

Категория:

Язык:

Год издания (выпуска):

Издательство:

Место издания:

Вид издания:

Код ISBN:

Сведения о количестве страниц:

№ тома/выпуска:

Обложка:

Файл книги:

Краткое описание:

Рис. Б.10. – Страница добавления книги

В случае некорректного заполнения полей под-формы появится сообщение об ошибках (например, см. рис. Б.11)

**Недопустимый формат файла книги!**

Основное заглавие:

Автор(ы):

Категория:

Язык:

Год издания (выпуска):

Файл книги:  Не указано название книги

Рис. Б.11. – Сообщение об ошибке при добавлении книги



## Приложение В. Руководство администратора

Под администратором мы понимаем пользователя со статусом «администратор». Web-приложение предоставляет администратору все возможности и права обычного пользователя, а кроме этого следующее:

- Возможность редактировать поля описания и удалять любые книги (в любое время);
- Возможность добавлять, изменять и удалять категории из упрощенной классификации (см. рис. В.1).

ТОМЕКАЯ ЭЛЕКТРОННАЯ БИБЛИОТЕКА

Вы вошли как: **admin**  
[Выход](#)

Всего книг: 12  
Читателей: 7  
Online читателей и гостей: 2

**МЕНЮ**

- [Главная](#)
- [Добавить книгу](#)
- [Каталог книг](#)
- [Управление сайтом](#)

**САМЫЕ СКАЧИВАЕМЫЕ**

- » Платон (5)
- » Компьютерные сети (4)
- » Магия дня недели (3)

Имя новой категории:

Категория-родитель:

УДК:

Добавляет новую категорию, необходимо выбрать категорию-родителя и УДК для соответствия

Изменяет или устанавливает новое соответствие категории и УДК - необходимо выбрать категорию и УДК

Удаляет выбранную категорию, автоматически переносит все книги из этой категории в категорию "Неотсортированное"

Рис. В.1 – Страница управления (manage)

Для перехода на страницу управления необходимо нажать на ссылку в меню «Управление сайтом».