СОДЕРЖАНИЕ

[1. ПОСТАНОВКА ЗАДАЧИ 2](#_Toc511979988)

[1.1. Цель создания информационной системы 2](#_Toc511979989)

[1.1.1. Основание для разработки 2](#_Toc511979990)

[1.1.2. Назначение разработки 2](#_Toc511979991)

[1.1.3. Цели создания 3](#_Toc511979992)

[1.1.4. Требования к надежности 3](#_Toc511979993)

[1.1.5. Специальные требования 3](#_Toc511979994)

[1.1.6. Инструменты разработки 3](#_Toc511979995)

[1.2. Анализ предметной области 4](#_Toc511979996)

[1.2.1. Анализ деятельности организации 4](#_Toc511979997)

[1.2.2. Анализ существующих систем 4](#_Toc511979998)

[2. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ СИСТЕМЫ 15](#_Toc511979999)

[2.1. Определение состава базы данных 15](#_Toc511980000)

[2.2. Физическое проектирование базы данных 17](#_Toc511980001)

[2.3. Проектирование структуры системы 19](#_Toc511980002)

[3. РАЗРАБОТКА СТРУКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ БИБЛИОТЕКА 23](#_Toc511980003)

[3.1. Средства реализации 23](#_Toc511980004)

[3.2. Общая структура системы 24](#_Toc511980005)

[3.3. Руководство пользователя 36](#_Toc511980006)

[3.3.1. Системные требования программных и аппаратных средств 36](#_Toc511980007)

[3.3.2. Инсталляция системы 37](#_Toc511980008)

[3.3.3. Результаты тестирования с примерами экранных форм 38](#_Toc511980009)

[ЗАКЛЮЧЕНИЕ 38](#_Toc511980010)

# ПОСТАНОВКА ЗАДАЧИ

## Цель создания информационной системы

Целью информационной системы «Библиотека» является улучшение качества услуг и повышение удобства пользователей путем сокращения непроизводительных затрат и повторяющихся операций.

### Основание для разработки

Основанием для разработки приложения информационной системы «Библиотека» является выпускная квалификационная работа.

### Назначение разработки

Информационная система «Библиотека» предназначена для комплексного информационно-аналитического обеспечения процессов деятельности Библиотека.

Основным назначением приложения «Библиотека» является автоматизация следующих процессов:

1. Учет книг и их экземпляров;
2. Контроль выдачи и возврата книг;
3. Ведение личных дел сотрудников;
4. Ведение читательских билетов.

Приложение должно состоять из взаимосвязанных разделов с четко разделенными функциями, выполнять задачу по информационному обеспечению библиотеки и создавать благоприятные условия для выполнения определенных функций. Система должна обеспечивать:

* возможность добавления и редактирования книг/экземпляров, их хранение и отображение;
* ведение базы данных сотрудников и читателей;
* бронирование печатных изданий;
* возможность выдачи и возврата книг;
* разделение функциональных возможностей между пользователями с помощью аутентификации.

### Цели создания

Информационная система «Библиотека» создается с целью:

1. Повышения качества предоставляемых услуг в этой сфере, путем автоматизации информационных процессов;
2. Осуществления возможности учёта выдачи и возврата книг;
3. Возможности занесения, хранения и отражения всей необходимой информации в системе;
4. Повышения эффективности исполнения процессов, путем сокращения непроизводительных и дублирующих операций и операций, выполняемых «вручную»;
5. Возможности разграничения прав доступа с помощью разных режимов работы;
6. Повышения эффективности принятия управленческих решений за счет централизованности.

### Требования к надежности

Разрабатываемое программное обеспечение должно иметь:

* модуль авторизации для доступа к системе;
* ограничение доступа к информации базы данных;
* контроль вводимой информации и блокировку этой информации при некорректных действиях пользователя;
* разграничение пользовательских прав.

### Специальные требования

Интерфейс пользователя информационной системы должен обеспечивать интуитивно понятное и наглядное представление информационной структуры, размещенной на нем, переход к разделам и страницам приложения должен быть быстрый и логичный. Система должна отображать всю необходимую информацию.

### Инструменты разработки

Для проектирования выбран унифицированный язык моделирования CASE-средство Rational Rose Enterprise. Для создания клиентского приложения в качестве среды разработки была выбрана RAD Studio 10.2 и выбрана СУБД MSSQL.

## Анализ предметной области

### Анализ деятельности организации

Анализ предметной области позволяет тщательно изучить деятельность библиотеки с точки зрения выполняемых функций, с последующим обозначением пользователей будущей информационной системы.

Для глубокого анализа предметной области Библиотека необходимо определиться, что подпадает под это понимание. Библиотека – это учреждение культурного характера, осуществляющее сбор, хранение и выдачу с обязанностью возврата печатных изданий художественной, научной, исторической литературы и так далее.

Основными направлениями деятельности библиотеки являются: комплектование и организация печатного фонда, обслуживание читателей.

Рабочие процессы организации осуществляется несколькими категориями сотрудников: библиотекарь, выполняет свою основную задачу выдачи печатных изданий и оформление возврата печатных изданий; управляющий библиотекой, занимающийся формированием фонда печатных изданий, а также ведением личных дел сотрудников и читателей.

Организация литературного фонда включает вопросы учета, хранения, расстановки и доставки её читателю. Правильная организация фонда ускоряет работу всей структуры и облегчает библиотекарю выполнение читательских требований.

### Анализ существующих систем

На данный момент существует достаточно информационных систем для обеспечения программного обеспечения библиотек, основной целью которых автоматизация все возможных бизнес процессов библиотеки. Все данные системы обладают разными качественными и количественными характеристиками. Однако, как правило, часть систем имеют специализацию по конкретный тип библиотек. Было проанализировано множество информационных систем, среди которых подробнее будут рассмотрены ИС Нерюнгинской городской библиотеки, ИС для библиотеки имени Маяковского и ИС «1С: Библиотека».

**ИС для библиотеки имени Маяковского.**

Приведем описание структуры её деятельности и организации процессов:

В библиотеку приходит много книг из различных издательств. Каждой книге в библиотеке присваивают номер, а затем передают в различные отделы. При поступлении книги, учитывают такие данные как:

* номер поступившей книги,
* название книги,
* название издательства, из которого поступила книга,
* отдел, куда была передана книга;
* адреса издательств,
* названия издательства,
* название и расположение отделов библиотеки.

Предусматривается автоматизация учета выдачи и сдачи книг, записи новых книг и читателей, а также хранения информации об имеющихся в наличии книгах, данные о сотрудниках библиотеки, сотрудниках хранилища библиотеки и читателях.

Каждая книга, хранящаяся в библиотеке, имеет следующие параметры:

* автор,
* издание,
* год издания,
* жанр,
* ключевые слова,
* количество страниц.

Каждая книга может присутствовать в нескольких экземпляров.

Так же в библиотеке ведется картотека читателей. О каждом читателе заносятся следующие сведения:

* ФИО,
* номер паспорта,
* адрес,
* телефон.

Каждому читателю присваивается номер читательского билета.

В случае выдачи экземпляра книги в библиотеке остается вкладыш, в котором указана дата выдачи, дата предполагаемого возврата и номер читательского билета.

Можно продлить книгу по телефону книгу назвав номер читательского билета и уникальный номер книги.

При возврате книги во вкладыше отмечается срок возврата. При просрочке возврата книги, читатель получает предупреждение. Когда предупреждений у читателя накапливается больше определенного предела, он лишается правом пользования библиотекой на определенный срок.

При потере книги читатель лишается правом пользования библиотекой на определенный срок, независимо от количества предупреждений.

В библиотеке есть ряд ограничений: нельзя выдавать книги больше установленного периода, нельзя выдавать больше определенного количества книг.

При выдаче книг в базе данных необходимо фиксировать:

* название книги;
* дату выдачи;
* ФИО читателя, которому выдается книга;
* ФИО сотрудника библиотеки, выдающему книгу читателю;
* ФИО сотрудника хранилища, который передал книгу (непосредственно из хранилища) сотруднику библиотеки по заявке читателя;
* срок, на который выдается книга.

При сдаче книг в базе данных необходимо зафиксировать дату возврата сдаваемой читателем книги путем ввода ее названия в БД.

На каждого читателя заносятся следующие сведения:

* №читательского билета;
* ФИО читателя;
* Телефон;
* Адрес.

Каждая книга, хранящаяся в библиотеке, имеет следующие характеристики:

* Название книги;
* Уникальный шифр (ISBN);
* Библиотечно-библиографическая классификация (ББК);
* Раздел;
* Автор;
* Издательство;
* Место издания (город);
* Год издания.

Все книги различаются по своему уникальному шифру - ISBN.

Также в ИС организован учет посещаемости библиотеки следующим образом:

Учет посещений и обращений.

Единицей учёта посещений является одно посещение, зарегистрированное в документации или БД, принятых в библиотеке.

Единицей учёта посетителей мероприятий является лицо, присутствующее на мероприятии и зарегистрированное в документах, принятых в библиотеке.

Учет посещений производится путем суммирования посещений, учтённых каждым структурным подразделением библиотеки.

Единицей учета обращений к электронной библиотечно-информационной сети является одно обращение на сервер библиотеки.

Учет общего числа посещений проводится суммированием посещений (обращений), учтенных каждым структурным подразделением библиотеки.

Учет ведется в первой части Дневника работы библиотеки "Учет состава читателей и посещаемости". Обращения входят в общее число посещений.

Формирование: Библиотекарь делает запрос в базу данных о том, сколько читателей и в какой день посещали библиотеку, в результате выводится количественная информация по дням в рамках определенного периода, заданного библиотекарем. Например, за 2 квартал текущего года. Затем из этого списка либо таблицы делается график в виде диаграммы, которая детально отражает статистику посещений библиотеки читателями. По графику, например, можно зафиксировать в какой день 2-го квартала текущего года было наибольшее или наименьшее количество посещений.

Статистический учет, т.е. сбор исходных данных, производится для того, чтобы использовать собранный материал для выработки каких-либо оценок и управленческих решений. Без использования статистических показателей невозможно создание научно-обоснованных методик оценки деятельности библиотек. Статистике принадлежит определяющее место в системе управленческого учета, поскольку она выявляет "типичное, характерное, закономерное".

**ИС Нерюнгинской городской библиотеки**

Можно продлить книгу по телефону книгу назвав номер читательского билета и уникальный номер книги.

При возврате книги во вкладыше отмечается срок возврата. При просрочке возврата книги, читатель получает предупреждение. Когда предупреждений у читателя накапливается больше определенного предела, он лишается правом пользования библиотекой на определенный срок.

При потере книги читатель лишается правом пользования библиотекой на определенный срок, независимо от количества предупреждений.

В библиотеке есть ряд ограничений: нельзя выдавать книги больше установленного периода, нельзя выдавать больше определенного количества книг.

Объект исследования - библиотека.

Предмет исследования- процессы, происходящие в библиотеке, такие как:

процесс выдачи книг читателям;

процесс сдачи книг;

процесс поступления новых книг;

процесс записи новых читателей.

Субъекты - пользователи (читатели, администрация библиотеки, другие заинтересованные в получении информации лица)

Входная информация - информация:

* о новых поступивших книгах;
* о новых читателях, которые записываются в библиотеку;
* о процессе выдачи книг, включает в себя:
* - кто из сотрудников хранилища передал книгу сотруднику библиотеки;
* кто из сотрудников библиотеки передал книгу читателю;
* по заявке какого читателя была выполнена эта операция;
* название книги;
* дату выдачи книги;
* срок, на который выдается книга;

Выходная информация - информация:

* о статистике посещения клиентами библиотеки;
* о книгах, которые наиболее часто были выданы читателям, то есть рейтинг книг.

Основные статистические показатели:

Учет читателей. Показывает количество человек, которые пользуются документами библиотеки. Единицей учета служит формуляр читателя.

Учет выдачи книг. Показывает количество документов (по отраслям знаний), использованных читателями во всех структурных подразделениях библиотеки. Учёт выдачи изданий производится по числу выданных экземпляров, зарегистрированных в формуляре читателя, листке читательского требования, книжном формуляре или иной документации, принятой в библиотеке, а также в электронной базе данных (БД).

**ИС «1С: Библиотека»**

Продукт позволяет автоматизировать рабочие процессы библиотеки, в зависимости от ее назначения, типа, состава фондов, может быть интегрирован с другими типовыми решениями фирмы "1С".

С помощью системы «1С: Библиотека» может быть автоматизирована деятельность библиотек разных назначений и типов:

* универсальных - муниципальных, областных, региональных;
* учреждений образования - школ, детских садов, колледжей, вузов;
* отраслевых - библиотек предприятий, компаний, заводов, больниц, научно-исследовательских институтов, военных частей;
* специальных - музыкальных, театральных, библиотек музеев и академий художеств и пр.;
* частных (домашних).

Комплектование:

Поиск, формирование и оформление заказов в издательства/книготоргующие организации.

Контроль выполнения заказов.

Автоматическая сверка на дублетность.

Регистрация поступлений изданий и присвоение уникального номера каждому поступающему в библиотеку документу с использованием технологии штрихкодирования.

Каталогизация:

Описание любых видов изданий, включая документы в электронном формате, аудио-, видео-, нотные, картографические, изоиздания и пр.

Полная поддержка национального коммуникативного формата RUSMARC.

Обеспечение контроля вводимых данных на их соответствие требованиям стандартов.

Автоматическая индексация, на основе которой реализуется быстрый поиск по любым элементам библиографического описания и их сочетаниям.

Возможность использовать любые графические материалы (изображения титульных листов, артефактов, страниц текста и т.д.).

Экспорт/импорт данных в MARC-форматах.

Хранение неограниченного количества библиографических записей. Учет, актуализация и хранение фонда:

Поддержка процессов упорядоченного хранения и учета документов библиотеки любого формата в соответствии с ее типом и назначением.

Обеспечение процессов движения фонда.

Ведение индивидуального и суммарного учета документов (изданий) как по совокупному фонду библиотеки, так и по отдельным фондам (специализированным) с учетом специфики хранящихся в них изданий.

Возможность вводить в учетные документы не только обязательные, но и факультативные показатели с учетом специфики конкретной библиотеки.

Обслуживание читателей:

Регламентирование и дифференцированный доступ к документам фонда для различных групп читателей.

Быстрая идентификация читателей с помощью технологии штрихового кодирования читательских билетов.

Дневник работы библиотеки.

Виртуальный кабинет читателя:

Удаленный доступ читателя к электронному каталогу и поиск изданий по любым элементам библиографического описания.

Формирование электронной заявки по результатам поиска.

Удаленный доступ к данным о наличии свободных физических экземпляров изданий в фондах библиотеки.

Удаленный доступ читателя к своему формуляру.

Контроль читателя за процессом исполнения сформированных заказов.

Оповещение читателя о выполненных заказах и его задолженности.

Формирование и поддержка электронного библиотечного фонда

Прикрепление к библиографическим записям электронного каталога произвольного количества электронных материалов (изданий в электронном формате)

Поиск по содержимому электронных изданий

Возможность прикрепления обложек

Хранение и выдача читателю изданий в электронном формате

Статистика и отчетность

Формирование отчетных и учетных документов

КСУ

Инвентарная книга

Ведомость проверки фонда

Партии физических экземпляров изданий

Книговыдача

Книгообеспеченность

Статистический отчет

Поддержка печатных форм:

Формирование стандартного набора документов и сопутствующих документов для карточного библиотечного каталога.

Поддержка всех необходимых печатных форм в соответствии с типовыми инструкциями и методическими рекомендациями Министерства образования и науки РФ.

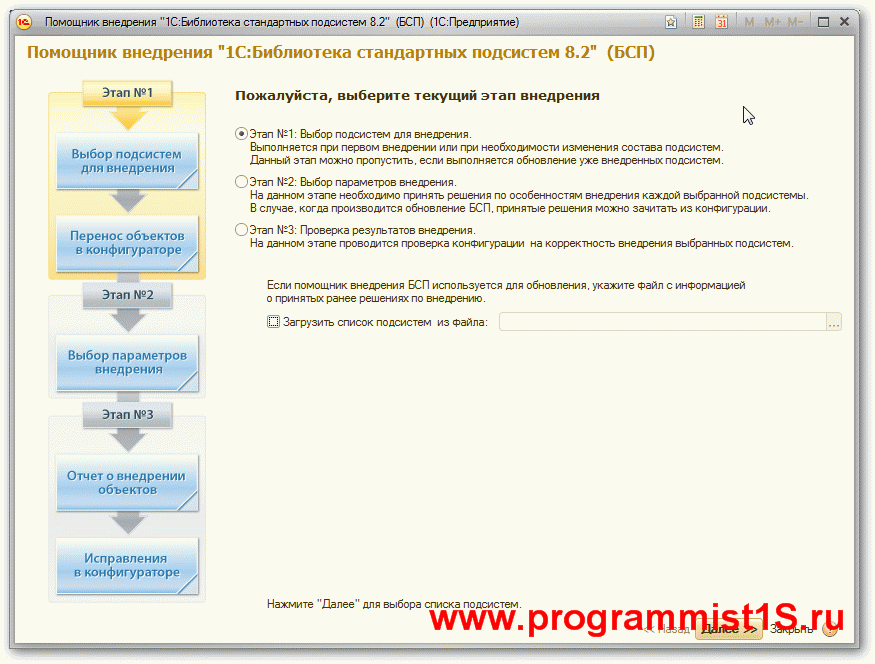


Рис.1.2.2.1. Внедрение ИС «1С: Библиотека»

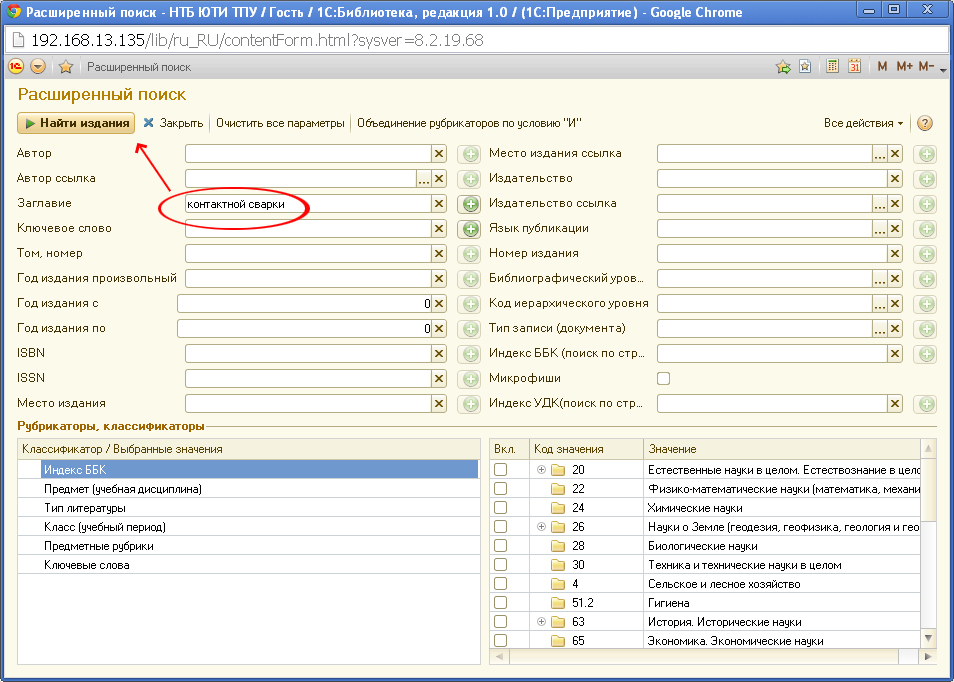


Рис.1.2.2.2. Форма поиска ИС «1С: Библиотека»

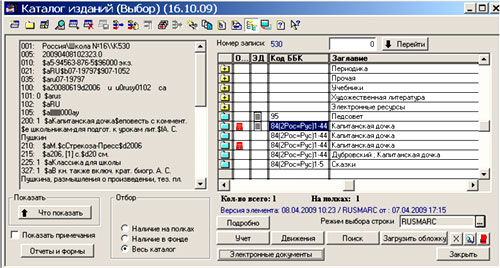


Рис.1.2.2.3. Каталог ИС «1С: Библиотека»

# ПРОЕКТИРОВАНИЕ СТРУКТУРЫ СИСТЕМЫ

## Определение состава базы данных

Спроектируем инфологическую модель системы, предназначенной для хранения информации о книгах, их выдачи и возврата. Разработку модели начнем с выделения основных сущностей.

Прежде всего, существует сущность «Книги»; каждая книга имеет уникальный шифр, который является ее ключом, и ряд атрибутов, которые взяты из описания предметной области. Множество экземпляров сущности определяет множество книг, которые хранятся в библиотеке. Каждый экземпляр сущности «Книги» соответствует не конкретной книге, стоящей на полке, а описанию некоторой книги, которое дается обычно в предметном каталоге библиотеке. Каждая книга может присутствовать в нескольких экземплярах, и это как раз те конкретные книги, которые стоят на полках библиотеки. Чтобы отразить это, следует ввести сущность «Экземпляры», которая должна содержать описания всех экземпляров книг, которые хранятся в библиотеке. Каждый экземпляр сущности «Экземпляры» соответствует конкретной книге на полке. Каждый экземпляр имеет уникальный инвентарный номер, однозначно определяющий конкретную книгу.

Между сущностями «Книги» и «Экземпляры» существует связь (1:М), обязательная с двух сторон. Данный тип связи определяется тем, что каждая книга может присутствовать в библиотеке в нескольких экземплярах, поэтому — связь 1:М. При этом если в библиотеке нет ни одного экземпляра данной книги, то мы не будем хранить ее описание, поэтому если книга описана в сущности «Книги», то по крайней мере один экземпляр этой книги присутствует в библиотеке. Это означает, что со стороны книги связь обязательная. Что касается сущности «Экземпляры», то не может существовать в библиотеке ни одного экземпляра, который бы не относился к конкретной книге, поэтому и со стороны «Экземпляры» связь тоже обязательная.

Теперь необходимо определить, как в системе будет представлен читатель. Естественно предложить ввести для этого сущность «Читатели», каждый экземпляр которой будет соответствовать конкретному читателю. В библиотеке каждому читателю присваивается уникальный номер читательского билета, который однозначно идентифицирует читателя. Номер читательского билета будет ключевым атрибутом сущности «Читатели». Кроме того, в сущности «Читатели» должны присутствовать дополнительные атрибуты, которые требуются для решения поставленных задач; этими атрибутами будут: «Фамилия Имя Отчество», «Адрес читателя», «Телефон» и «Адрес».

Каждый экземпляр книги может находиться либо в библиотеке, либо на руках у некоторого читателя. Каждый читатель может держать на руках несколько экземпляров книг. В том случае, когда книга находится у читателя на руках необходимо дополнительно хранить информацию о дате взятия книги читателем, дата предполагаемого и фактического возврата книги. Для этого выделим сущность «Лист выдачи».

Узнать, какая книга у данного читателя можно по дополнительной связи между сущностями «Экземпляры» и «Книги», и эта связь каждому экземпляру ставит в соответствие одну книгу, поэтому всегда можно однозначно определить, какие книги находятся на руках у читателя, хотя связываем с читателем только инвентарные номера взятых книг.

Между сущностями «Читатели» и «Выдача книг» установлена связь 1:М, и при этом она не обязательная с двух сторон. Читатель в данный момент может не держать ни одной книги на руках, а с другой стороны, данный экземпляр книги может не находиться ни у одного читателя, а просто стоять на полке в библиотеке.

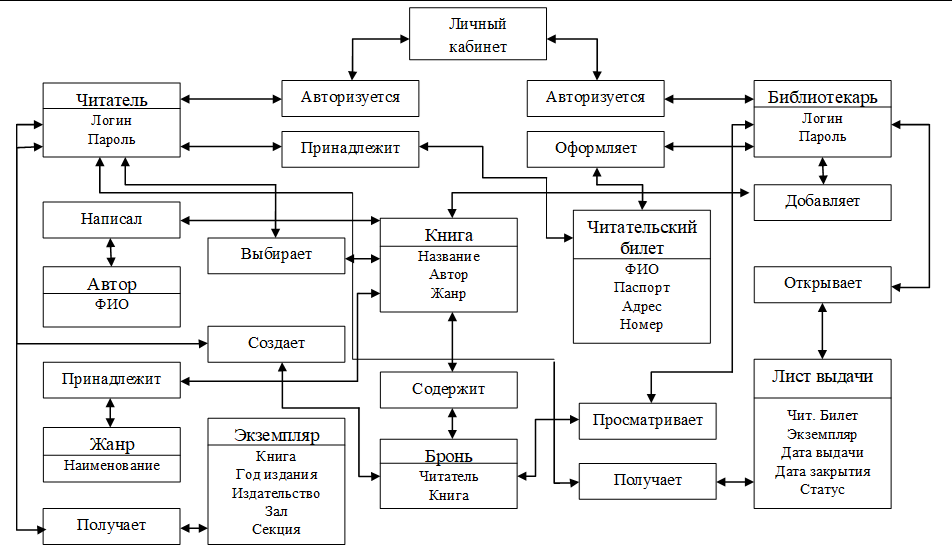


Рис.2.1.1. ER-диаграмма

## Физическое проектирование базы данных

Схема реляционной базы данных показывает таблицы и их взаимодействия в базе данных. Пример схемы реляционной базы данных "Библиотека" отображен на рис. 2.2.1.

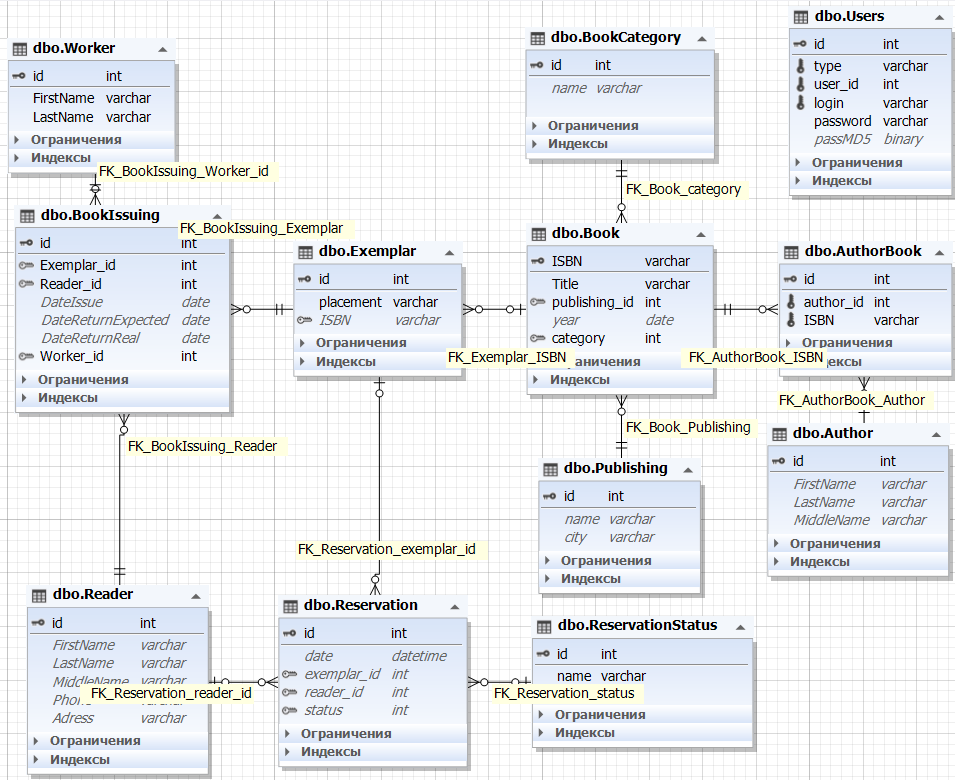


Рис. 2.2.1. Физическая схема базы данных

## Проектирование структуры системы

Данные, полученные на основе анализа деятельности Библиотека, помогают спроектировать структуру системы, описать основные роли и их функции.

На основе полученных данных осуществляется этап проектирования информационной системы с использованием объектно-ориентированного подхода. В данном подходе используются языки UML (Унифицированный язык моделирования), которые являются специальными языками для представления, определения, документирования и проектирования технических систем.

Для разработки диаграмм системы использовалось CASE-средство Rational Rose Enterprise, которое является инструментом моделирования, упрощает рабочую среду и имеет возможность генерации кода.

Вначале строится диаграмма вариантов использования, она показывает функциональные требования к проектируемой системе с точки зрения пользователя, здесь выделяются действующие лица, взаимодействующие с системой, к тому же указываются все варианты использования для действий, производимых выделенными пользователями.



Рис. 2.3.1. Диаграмма вариантов использования



Рис. 2.3.2. Общая диаграмма классов

Диаграмма классов определяет структуру классов системы и статические связи между ними, описываются операции, накладываемые на связи. На диаграмме классов для данной системы (Рис. 2.3.2) отражены действующие лица, которым доступны определенные формы для их взаимодействия с системой.

# РАЗРАБОТКА СТРУКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ БИБЛИОТЕКА

## Средства реализации

В качестве средства разработки использовалась интегрированная среда разработки RAD Studio 10.2 и СУБД MS SQL Server 2008.

RAD Studio — среда быстрой разработки приложений (Rapid Application Development), функционирующая на платформе Windows. Основным преимуществом ее использования является то, что использование объектно-ориентированной (ОО) технологии в визуальном проектировании позволяет получить качественный продукт в короткие сроки. Постепенное и предсказуемое развитие проектов обеспечивается благодаря сочетанию возможностей быстрого прототипирования приложений с технологиями предприятия. Основным залогом успеха проектов, разрабатываемых с использованием Delphi является следование стандартам индустрии и возможность взаимодействия с любыми готовыми решениями.

MS SQL Server продукт компании Microsoft который ориентирован на операционные системы Windows. По современным меркам, минимальные требования предъявляемый для функционирования системы MS SQL Server 2008 не велики и определяются нагрузкой на базу данных. Сервер в тоже время обладает высокой производительностью и лёгок для администрирования. Основной используемый язык запросов — Transact-SQL. К основным преимуществам MS SQL Server можно перечислить: простоту визуального проектирования баз данных, легкость управления данными, а также наглядность и что не мало важно высокий уровень безопасности.

Основной инструмент для доступа к MS SQL Server 2008 это интегрированная среда SQL Server Management Studio. Она включает в себя включает скриптовый и графический редактор, а также программу, работающую с объектами и настройками сервера.

## Общая структура системы

Структура проекта:

Файлы с разрешением \*.dpr – это текстовые файлы, которые используется для хранения информации о формах и модулях. В них содержатся операторы инициализации и запуска программ на выполнение.

Файлы с разрешением \*.dfm –двоичные или текстовые файлы, которые создается для хранения информации о формах. Каждому файлу формы соответствует файл модуля (.pas)

Каждой создаваемой форме соответствует текстовый файл модуля с разрешением \*.pas, используемый для хранения кода. Многие из функций и процедур Delphi хранятся в модулях.

Подробнее рассмотрим структуру модулей.

UnitAuth.pas– cодержит в себе обработчики формы логику авторизации пользователей в системе.

UnitSuperUser.pas– содержит в себе логику для работы пользователя с ролью заведующий библиотекой.

UnitLibrarianAccount.pas– содержит в себе обработчик формы для пользователя библиотекарь.

UnitReaderAccount.pas– содержит в себе обработчик формы для пользователя читатель.

UnitExemplar.pas – Файл, содержащий форму для добавления и редактирования информации о экземпляре книги.

UnitBook.pas – Файл, содержащий форму для создания, редактирования и просмотра информации о книгах.

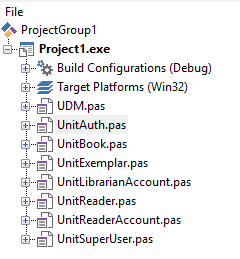


Рисунок. 3.2.1. Структура модулей

Рисунок. 3.2.2. Структура взаимосвязей модулей

Единственной не визуальной формой является DataModule. Она представляет собой форму-компонент для размещения на нем компонентов, предназначенных для соединения с базой данных.

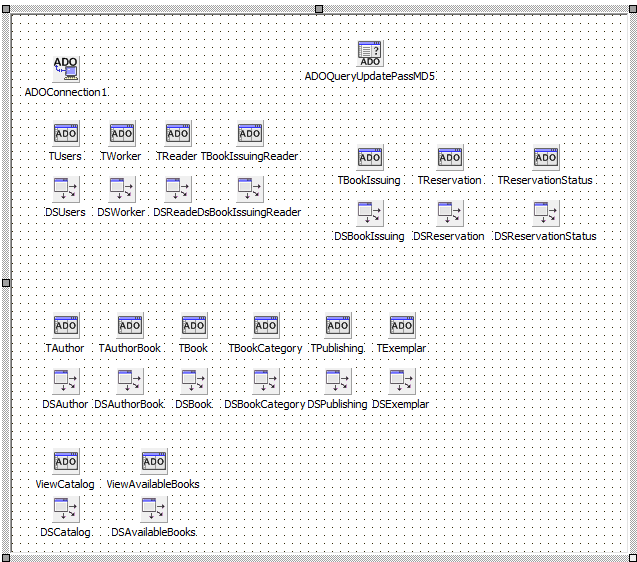


Рисунок. 3.2.2. Data Module

Схема соединения приложения с базой данных изображена на рисунке 3.2.3.

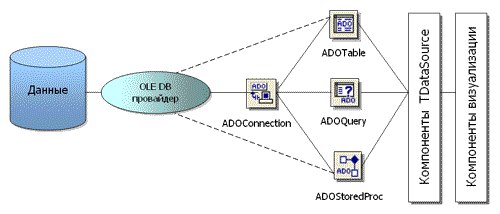


Рисунок. 3.2.4. Схема соединения

ADOConnection является компонентом, предназначенным для соединения с базой данных. Свойство компонента строка соединения формируется автоматически. В качестве поставщика данных используется «Microsoft OLEDB Providerfor SQL Server». В наименовании сервера указывается именованный экземпляр, созданный при установке MSSQLSQLServer, либо сервер, используемый по умолчания на локальной машине – (local). Для входа на сервер базы данных используется средство встроенной безопасности Window NT. Имя базы данных «LibraryDB».

После установления соединения с сервером базы данных необходимо настроить компоненты для отображения информации получаемой и базы данных.

ADOTable является компонентом, отвечающим за получение данных. Компонент из базы данных получает полную копию таблицы и хранит ее на всем протяжении работы приложения, а также по мере изменений содержания таблицы, синхронизирует ее состояние с состоянием реальной в базе данных.

Компонентом для отображения информации из набора данных является DBGrid. Однако для его соединения с набором данных необходима связь. Эту задачу решает компонент DataSource, он является посредником между наборами данных и конечными визуальными компонентами. Все конечные визуальные компоненты содержат свойство DataSource.

Что бы настроить корректное отображение информации в таблице достаточно в свойстве DataSource указать необходимый набор данных.

В качестве компонентов для визуалиазции: DBGrid, DBEdit,DBLookupComboBox. В этих компонентах уже заложена логика отображения информации из базы данных, достаточно только указать им нужный источник данных. Для добавления новых записей в набор данных с данными компонентами используются такие функции как Insert/Addend, для редактирования записи Editи для сохранения изменений Post.

По мере возрастания сложности запросов, появления больших межтабличных запросов, встроенного функционала стандартных компонентов ADOTable становится недостаточно и появляется необходимость использовать компонент ADOQuery.В отличие от ADOTable, запрос не содержит свойств для выбора таблицы, которую он будет отображать. Вместо этого он имеет поле SQL, которое должно содержать текст SQL-запроса к БД для выбора необходимых данных. Запросы, которые содержали в себе значительный объем текста были сохранены в БД в представлении.

Представление (VIEW) - объект данных который не содержит никаких данных его владельца. Это - тип таблицы, чье содержание выбирается из других таблиц с помощью выполнения запроса. Поскольку значения в этих таблицах меняются, то автоматически, их значения могут быть показаны представлением. Поэтому работа с представлениями с представлениями не отличается от работы с обычной таблицей в понимании использования компонентов ADOTable.

Основные процедуры:

Модуль «UnitFormAuth» предназначен для авторизации пользователя в системе и последующего доступа к ней в зависимости от роли авторизованного пользователя.

procedure TFormAuth.BitBtn1Click(Sender: TObject);

begin

ADOQuery1.Close;

// передача в параметры sql запроса данных введенных пользователем

ADOQuery1.Parameters.ParamByName('l').Value := EditLogin.Text;

ADOQuery1.Parameters.ParamByName('p').Value := EditPass.Text;

// выполнение запроса

ADOQuery1.Open;

if (ADOQuery1.RecordCount > 0) then

begin

if ADOQuery1.Fields[1].AsString = 'reader' then

begin

dm.TReader.Locate('id', ADOQuery1.Fields[0].AsInteger,

[loCaseInsensitive, loPartialKey]);

FormReader.Show;

end;

if ADOQuery1.Fields[1].AsString = 'librarian' then

begin

dm.TWorker.Locate('id', ADOQuery1.Fields[0].AsInteger,

[loCaseInsensitive, loPartialKey]);

FormLibrary.Show;

end;

if ADOQuery1.Fields[1].AsString = 'admin' then

begin

dm.TWorker.Locate('id', ADOQuery1.Fields[0].AsInteger,

[loCaseInsensitive, loPartialKey]);

FormSuper.Show;

end;

hide;

end

else

ShowMessage('Неверный логин или пароль');

end;

В компоненте ADOQuery1 в свойстве SQL содержится следующий текст запроса:

SELECT

user\_id

,type

FROM LibraryDB.dbo.Users

WHERE login =:l

AND passMD5 = CAST(HASHBYTES('MD5', :p) AS BINARY(16))

Здесь следует пояснить последнюю строчку. Парили в забе данных хранятся в зашифрованном виде с использованием шифромания MD5. Оператор HASHBYTES (Transact-SQL) – возвращает хэш MD2, MD4, MD5, SHA1 или SHA2 входного значения в SQL Server.

Модуль «UnitLibrarianAccount» содержит в себе обработчики формы личного кабинета библиотекаря. Здесь реализован весь функционал из перечня задач библиотекаря. Особенность данного модуля заключается в удобном поиске и просмотре информации из разных таблиц базы данных. Запросы используемые для отображения информации в данном модуле были довольно большие по объему, поэтому часть из них была сохранена на стороне базы данных в виде представления (view). Далее информация из данного представления использовалась как обычная таблица базы данных и подключалась через компонент ADOTable. Ниже приведен код содержимого представления «Catalog», предназначенного для отображения информации о всех книгах в библиотеке с подсчетом суммарного количества экземпляров каждой книги.

SELECT

b.ISBN ,bc.name AS 'category' ,b.Title ,FirstName + ' ' + SUBSTRING(a.LastName, 1, 1) + '.' + SUBSTRING(a.MiddleName, 1, 1) + '.' AS 'Author' ,p.name ,p.city ,b.year ,COUNT(\*) AS 'Count'

FROM LibraryDB.dbo.Exemplar e

LEFT JOIN Book b

ON e.ISBN = b.ISBN

LEFT JOIN BookCategory bc

ON b.category = bc.id

LEFT JOIN Publishing p

ON b.publishing\_id = p.id

LEFT JOIN AuthorBook ab

ON b.ISBN = ab.ISBN

LEFT JOIN Author a

ON ab.author\_id = a.id

GROUP BY b.ISBN

,bc.name,b.Title, FirstName + ' ' + SUBSTRING(a.LastName, 1, 1) + '.' + SUBSTRING(a.MiddleName, 1, 1) + '.' ,p.name ,p.city ,b.year

Следующее представление «AvailableBooks» - предназначено для отображения сводной информации об экземплярах книг, доступных для бронирования и выдачи. В таковым относятся те, которые в данный момент времени находятся не на руках у читателя, а в библиотеке, а также книги, которые не были ранее забронированы читателями и ожидающие их выдачич.

SELECT

ex.id ,b.ISBN ,b.Title ,b.publishing\_id ,p.name AS 'publishing'

,b.year

,b.category AS 'category\_id'

,bc.name AS 'category'

,ex.placement

FROM LibraryDB.dbo.Exemplar ex

LEFT JOIN Book b

ON ex.ISBN = b.ISBN

LEFT JOIN publishing p

ON b.publishing\_id = p.id

LEFT JOIN BookCategory bc

ON b.category = bc.id

WHERE ex.id NOT IN (SELECT

bi.Exemplar\_id

FROM BookIssuing bi

WHERE bi.DateReturnReal IS NULL)

AND ex.id NOT IN (SELECT

r.Exemplar\_id

FROM Reservation r

LEFT JOIN ReservationStatus rs

ON r.status = rs.id

WHERE rs.name IN ('Забронирован', 'Ожидание выдачи'))

В данном модуле практически у каждой таблицы есть возможность поиска по ней. Однако от таблицы к таблице суть этого поиска не изменяется – везде применяется стандартное свойство компонента ADOTable - Filtered. Свойство Filter - наиболее часто используемый способ фильтрации записей, имеет тип String. Для фильтрации таблицы вначале задается условие фильтрации в этом свойстве, затем логическому свойству Filtered присваивается значение True, после чего таблица будет отфильтрована. Общий обработчик фильтрации по заданному полю имеет следующий вид:

procedure TDM.DoIncrementalFilter(DataSet: TDataSet;

const FieldName, SearchTerm: string);

begin

Assert(Assigned(DataSet), 'No dataset is assigned');

if SearchTerm = '' then

DataSet.Filtered := False

else

begin

DataSet.Filter := FieldName + ' LIKE ' + QuotedStr(SearchTerm + '\*');

DataSet.Filtered := True;

end;

end;

Для его использования достаточно задать набор данных для фильтрации, столбец по которому будет происходить поиск, а также искомое значение. Так, например, для поиска книг на форме был создан выпадающий список с предварительно заданным содержимым (свойство Items): Название, Категория, Автор, Издательство. Добавлен компонент TSearchBox и создан следующий обработчик поиска на заданному пользователем полю:

procedure TFormLibrary.SearchBoxCatalogInvokeSearch(Sender: TObject);

var

FieldName, SearchTerm: string;

begin

case ComboBoxBookSearch.ItemIndex of

0: FieldName := 'Title';

1: FieldName := 'category';

2: FieldName := 'Author';

3: FieldName := 'name';

end;

SearchTerm := SearchBoxCatalog.Text;

DM.DoIncrementalFilter(DM.ViewCatalog, FieldName, SearchTerm);

end;

Изначально определяется по какому полю происходит поиск, затем название FieldName записывается название столбца, по которому будет происходить поиск и в ранее описанную процедуру DoIncrementalFilter передаются все необходимые параметры. После этого набор данных будет отфильтрован и DBGrid отобразятся записи, удовлетворяющие условию фильтра.

Помимо просмотра и поиска информации роль рядового библиотекаря подразумевает и регистрацию читателя в библиотеке, а также выдачу и возврат книги.

Далее рассмотрим процедуру, позволяющую создать нового читателя в системе.

//

procedure TFormLibrary.CreateReaderClick(Sender: TObject);

begin

DM.TReader.Insert; //открываем набор данных для вставки записи

case FormAddReadTicket.ShowModal of

mrOk:

//отображает модально форму в которой пользователю необходимо будет заполнить поля и нажать на кнопку «Сохранить»

begin

DM.TReader.Post; //сохраним введенную информацию о читателе

DM.TUsers.Insert; // откроем набор данных для вставки записи

DM.TUsers.FieldByName('type').Value := 'reader'; //устанавливаем тип пользователя

DM.TUsers.FieldByName('user\_id').Value := DM.TReader.Fields[0].Value;

DM.TUsers.FieldByName('login').Value :=

FormAddReadTicket.EditLogin.Text;

DM.TUsers.FieldByName('password').Value :=

FormAddReadTicket.EditPass.Text;

DM.TUsers.Post;

// обновим пароль MD5

DM.ADOQueryUpdatePassMD5.ExecSQL;

end;

mrCancel:

begin

DM.TReader.Cancel;

DM.TUsers.Cancel;

end;

end;

end;

Приступим к рассмотрению одной из важных функций библиотекаря – выдача книги. Перед выполнением необходимо перевести набор данных в режим редактирования. Со стороны пользователя произойдет эт после нажатия на кнопку «Открыть лист выдачи». После того как состояние набора данных DM.TBookIssuing переведено в режим редактирования, и пользователь ввел всю необходимую информацию о выдаче, такую как код экземпляра, читатель и дата выдачи, необходимо применить сохраненный изменения вызовом метода Post у набора данных DM.TBookIssuing.

procedure TFormLibrary.ButtonBookIssueSaveClick(Sender: TObject);

begin

if DM.TBookIssuing.State in [dsInsert] then

begin

DM.TBookIssuing.FieldByName('DateIssue').Value := DateTimePicker1.DateTime;

DM.TBookIssuing.FieldByName('DateReturnExpected').Value :=

DateTimePicker2.DateTime;

DM.TBookIssuing.FieldByName('Worker\_id').Value := DM.TWorker.Fields

[0].Value;

DM.TBookIssuing.Post;

GroupBoxBookIssue.Enabled := false;

end;

end;

## Руководство пользователя

### Системные требования программных и аппаратных средств

В состав технических средств обеспечения системы должны входить:

* + Рабочие станции;
  + серверы БД;
  + серверы приложений;
  + источники бесперебойного питания;
  + средства для построения ЛВС;
  + принтер.

Минимальные системные требования, предъявляемые к аппаратным средствам сервера программного комплекса:

* + Размер оперативной памяти не менее 1024 МБ;
  + Процессор с тактовой частотой 2,3 ГГц;
  + Несъемный носитель информации (жесткий диск) 100 Гб.

Минимальные системные требования, предъявляемые к программным средствам сервера программного комплекса:

* + операционная система WindowsServer 2012;
  + версия СУБД MS SQL Server R2;

Минимальные системные требования аппаратного обеспечения, предъявляемые к клиентским рабочим местам программного комплекса:

* + Размер оперативной памяти не менее 512 МБ;
  + Процессор с тактовой частотой – 1,7 ГГц и более;
  + видеоадаптер – совместимый с DirectX 9.0c.

Минимальные системные требования программного обеспечения, предъявляемые к клиентским рабочим местам программного комплекса:

* + операционная система – Windows XP/Vista/7/8/10;
  + версия СУБД MS SQL Server R2;

По технике безопасности, в соответствии с ГОСТ 12.1.030-81 все внешние элементы, которые находятся под напряжением, в обязательном порядке должны иметь защиту от случайного прикосновения. Также обязательным требованием к обеспечению безопасности является, заземление всех элементов технического комплекса системы.

Для обеспечения полной работоспособности системы во время бросков напряжения и защиты аппаратуры, необходимо использовать сетевые фильтры, а также источники бесперебойного питания. Также рекомендуется обеспечить рабочие помещения резервными источниками питания, например, генераторы электрического тока.

При возникновении сбоев в сети, перегрузках напряжения или коротких замыканиях электрическое питание должно обеспечить защищенное, а также аварийное или ручное отключение.

Общие требования, предъявляемые к пожарной безопасности должны соответствовать всем нормам на бытовое электрооборудование, и в случае возникновения воспламенений не должно выделяться ядовитых газов и дымов.

### Инсталляция системы

Процесс инсталляции программного обеспечения можно разделить на 2 этапа: установка СУБД и создание базы данных с последующей установкой ее резервной копии.

На первом этапе для работы приложения необходимо установить пакет MS SQL Server 2008. Приложение распространяется на бесплатной основе.Для установки пакета MS SQL Server 2008, его необходимо скачать с официального сайта Microsoft в разделе загрузки. После загрузки нужно запустить исполняемый файл и следовать указаниям мастера установки. Помимо MS SQL Server нужно аналогичным способом установить MS SQL Management Studio для возможности управления базой данных.

После того как СУБД окончательно установлена необходимо убедиться в корректности установки. Для этого нужно запустить MS SQL Management Studio и проверить соединение с сервером (local). В случае успешного соединения можно приступать к следующему этапу инсталляции программного обеспечения.

На втором этапе необходимо добавить новую базу данных. После того как база данных создана, нужно запустить восстановление резервной копии базы данных (бэкап). В Managenent Studio необходимо выбрать созданную базу данных и выполнить SQL скрипт, поставляемый с приложением. После выполнения SQL скрипта установку можно считать завершенной, а программное обеспечение информационной системы готовыми к использованию.

### Результаты тестирования с примерами экранных форм

Приложение предназначено для трёх категорий пользователей: библиотекарь, управляющий библиотекой, читатель.

При запуске приложения открывается окно авторизации пользователя (Рис.3.3.3.1). Необходимо ввести в поля ввода логин и пароль и кликнуть «ОК».

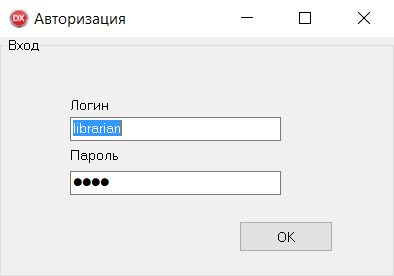


Рис.3.3.3.1 Окно авторизации

Далее разбор тестирования будет происходить в соответствии ролей пользователей.

Библиотекарь.

Форма библиотекаря сформирована из основных вкладок: каталог, читатель, бронь, лист выдачи.

Поиск определенного печатного издания производиться на вкладке Каталог/Все книги в верхней части окна. Поиск можно осуществлять по трём признакам: категориям, автору, названию. Для этого необходимо выбрать один из трёх признаков и ввести в поле поиска ключевое слово, по которому будет происходить поиск, кликнуть на специальный символ «лупа», как на Рис.3.3.3.2. После указатель слева в виде стрелки покажет на нужную запись в таблице, которая расположена в конце окна.

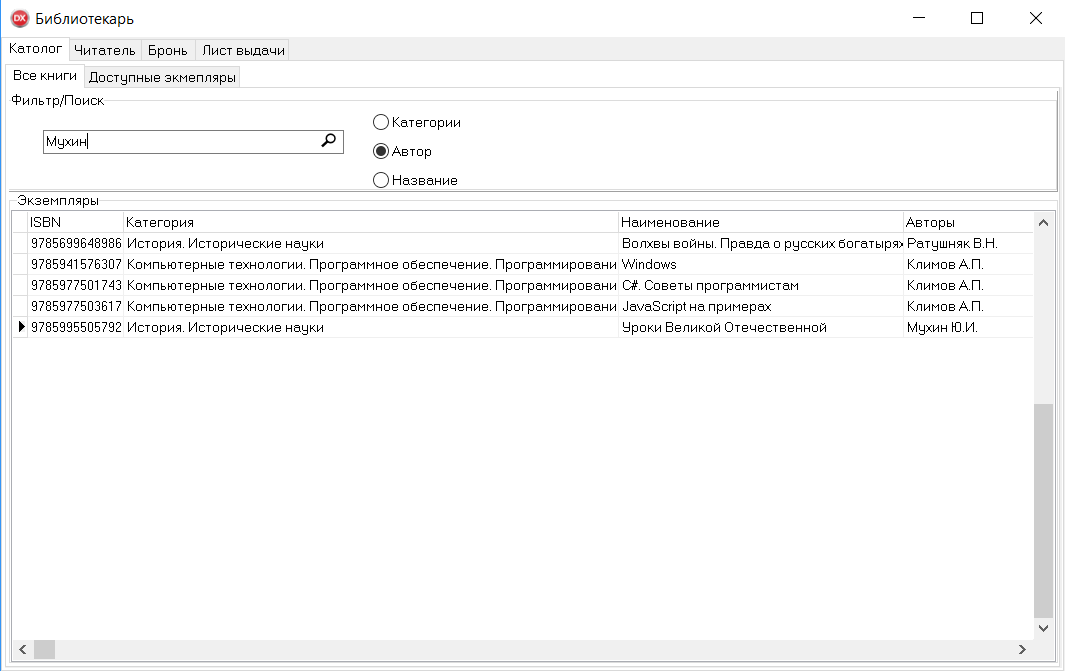


Рис.3.3.3.2 Окно «Все книги»

Для того, чтобы просмотреть доступные печатные издания нужно перейти на вкладку Каталог/Доступные экземпляры (Рис.3.3.3.2).

По аналогии, описанной выше, можно также осуществить поиск.

Замечание: а поле поиска можно вводить неполное название ключевого слова.

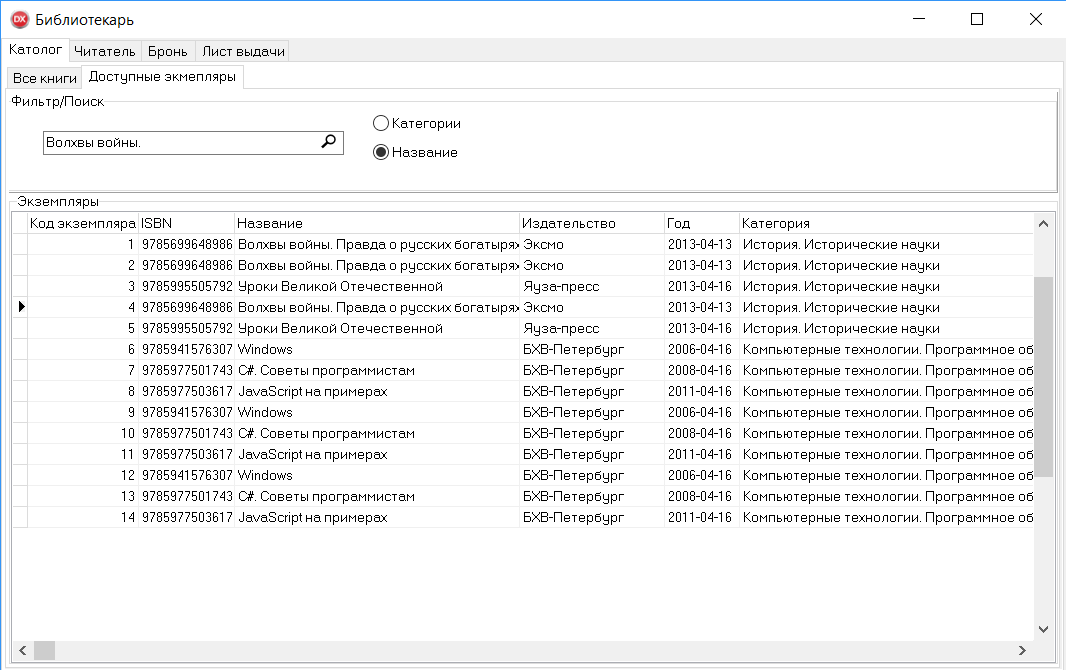


Рис.3.3.3.2 Окно «Доступные экземпляры»

Чтобы просмотреть данные читателей и оформленные на них листы выдачи, нужно перейти на вкладку Читатель (Рис.3.3.3.3).

Для просмотра списка листов выдачи определенного читателя нужно этого читателя найти, введя любую информацию о читателе в поле поиска в верхней части окна. Стрелка слева укажет на искомого читателя в таблице, а его список листов выдачи можно увидеть в нижней таблице окна.

Как-либо отредактировать список читателей можно нажав на определенную кнопку создать/редактировать/удалить в верхней правой части окна приложения. Перед редактированием или удалением читателя необходимо его найти в верхней таблице и выбрать его.

Создание читательского билета (читателя) осуществляется после нажатия на кнопку «ОК» в форме Рис.3.3.3.4. Она открывается после нажатия на кнопку «Создать». Аналогичная форма открывается при нажатии на кнопку «Редактировать».

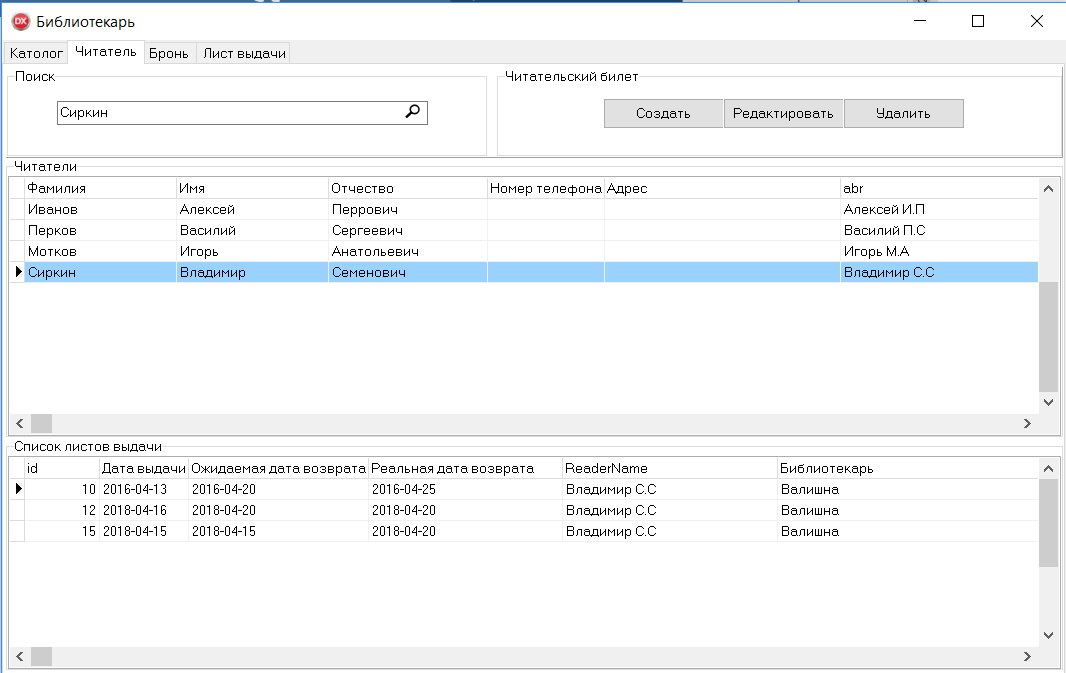


Рис.3.3.3.3 Окно «Читатель»

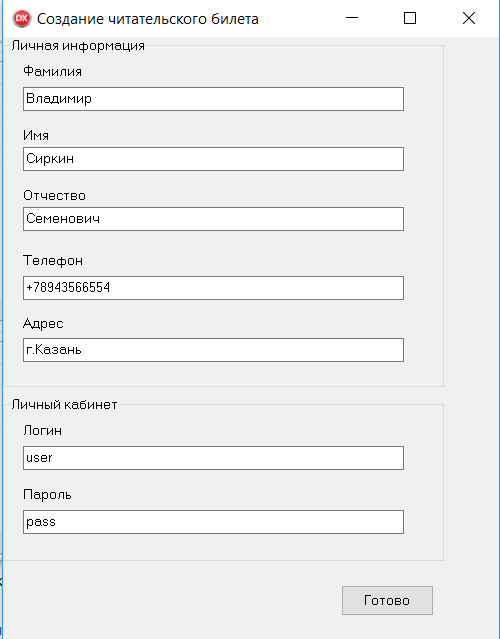


Рис.3.3.3.4 Окно «Создание читательского билета»

Просмотр брони на печатные издания осуществляется на форме «Бронь» (Рис.3.3.3.5.).

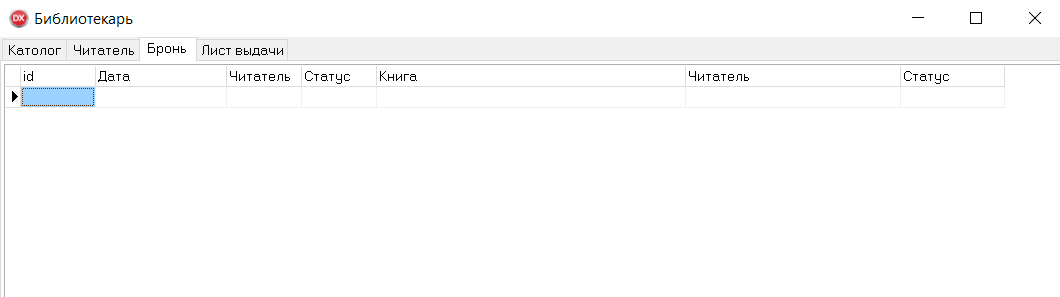


Рис.3.3.3.5 Окно «Бронь»

Чтобы просмотреть сформированные листы выдачи, нужно перейти на вкладку Лист выдачи (Рис.3.3.3.6).

Также, как было описано выше, можно производить поиск листа выдачи по определенным ключевым словам. Стрелка слева укажет на нужный лист выдачи в таблице.

Открыть или закрыть лист выдачи можно нажав на определенную кнопку открыть/закрыть в верхней правой части окна приложения. Перед закрытием листа выдачи необходимо его найти в верхней таблице и выбрать его.

Открытие листа выдачи осуществляется после заполнения полей в нижней части окна и нажатия на кнопку «ОК».

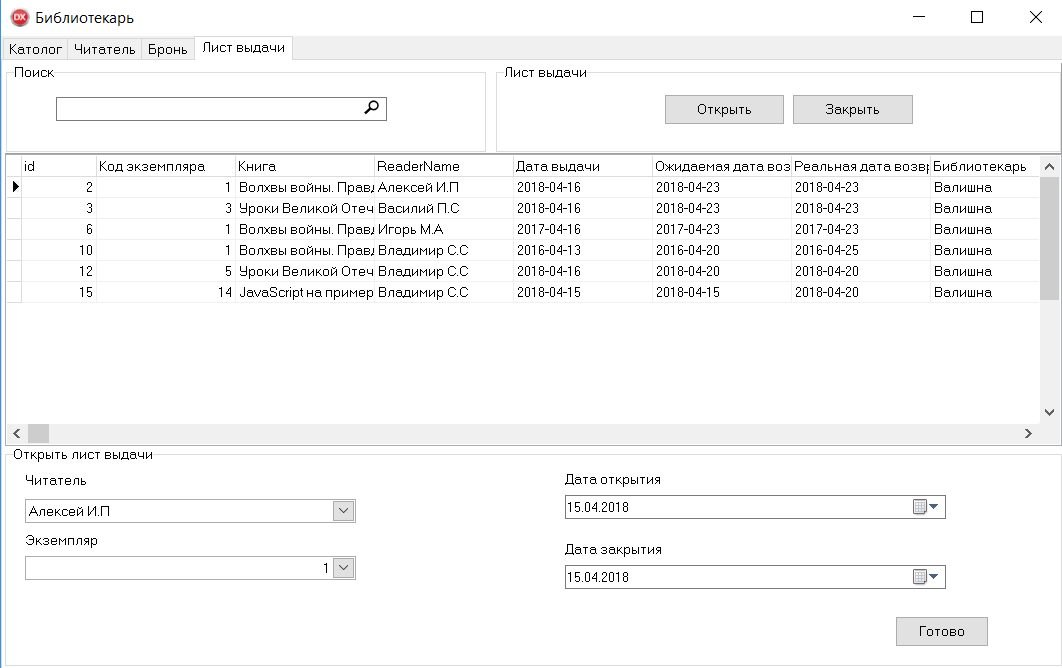


Рис.3.3.3.6 Окно «Лист выдачи»

Читатель.

Читатель получает доступ к приложению через стационарный компьютер, находящийся на территории библиотеки.

//тут будет описание работы приложения для читателя

//для управляющего библиотекой

# ЗАКЛЮЧЕНИЕ