# **1** **Постановка задачи**

## **Анализ предметной области**

Библиотека – это учреждение культуры, организующее сбор, хранение и общественное пользование произведениями печати и другими документами. Библиотеки систематически занимаются сбором, хранением и выдачей читателям произведений печати.

Основными направлениями работы любой библиотеки являются: комплектование и организация книжного фонда; обслуживание читателей.

Комплектование фондов библиотеки состоит из систематического выявления (путем просмотра библиографических источников и литературы) нужных для данной библиотеки изданий и приобретения их. От своевременности и полноты комплектования библиотеки в значительной мере зависит уровень обслуживания читателей.

Организация книжного фонда включает вопросы учета, расстановки, хранения литературы и доставки ее читателю.

Учет библиотечного фонда – это прием поступивших в библиотеку изданий, запись прибывших и исключение по тем или иным причинам непригодных для читателей изданий в документах библиотеки, которые дают точные сведения о наличном составе фонда и помогают обеспечить его сохранность.

Правильная организация фонда облегчает читателю пользование литературой, библиотекарю - быстрое выполнение читательских требований, а также обеспечивает сохранность фондов как общественной собственности.

В библиотеке ведется картотека читателей. О каждом читателе заносятся следующие сведения:

* фамилия;
* имя;
* отчество;
* дата рождения;
* адрес;
* мобильный телефон.

Обслуживание читателей библиотеки осуществляется различным путем выдачи литературы и помощи отдельным читателям в подборе необходимой им литературы.

Срок пользования литературой для различных категорий читателей и количество выдаваемых изданий на каждом абонементе определяется администрацией, исходя из вида литературы и категории читателя. Число выдаваемых книг не ограничивается.

Библиотечная деятельность связана с учетом большого количества операций, множество книг и читателей серьезно замедляют работу библиотекарей. Сложность поиска нужной книги в каталоге занимает длительное время, и целиком опирается на компетентность работников библиотеки.

Каждая книга может присутствовать в нескольких экземплярах.

Все книги различаются по своему уникальному номеру.

БД находится на специальном компьютере в электронном виде, постоянно редактируется и изменяется работниками библиотеки. С базой данных библиотеки могут работать как руководители (администрация) библиотеки, так и работники библиотеки, заинтересованные в получении необходимой информации. При работе с системой библиотекарь должен иметь возможность решать следующие задачи:

Принимать новые книги и регистрировать их в библиотеке.

Вести учет выданных книг читателям, при этом предполагается два режима работы: выдача книг читателю и прием от него возвращаемых им книг обратно в библиотеку. При выдаче книг фиксируется, когда и какая книга была выдана данному читателю и на какой срок выдается данная книга. При приеме книги, возвращаемой читателем, проверяется соответствие названия книги, и она ставится на свое старое место в библиотеке.

Администрация библиотеки должна иметь возможность получать сведения о должниках - читателях библиотеки, которые не вернули вовремя взятые книги.

Таким образом был произведен анализ предметной области работника сферы библиотекаря.

## **1.2 Обзор существующих аналогов**

На данный момент существует несколько аналогов, находящихся в свободном доступе.

Программа OPAC-Global обладает очень широким функционалом: ведение учета книг, ведение учета читателей и каталогизация, однако для того чтобы научится этим всем пользоваться, необходимо проходимо долгое и недешевое обучение у компании, выпускающей данный продукт. Помимо этого, заказчика не устраивает то, что программа использует внутреннюю базу данных, а заказчик хочет использовать уже существующую базу данных MySQL. Рабочее окно программы OPAC-Global представлено на рис. 1.1:

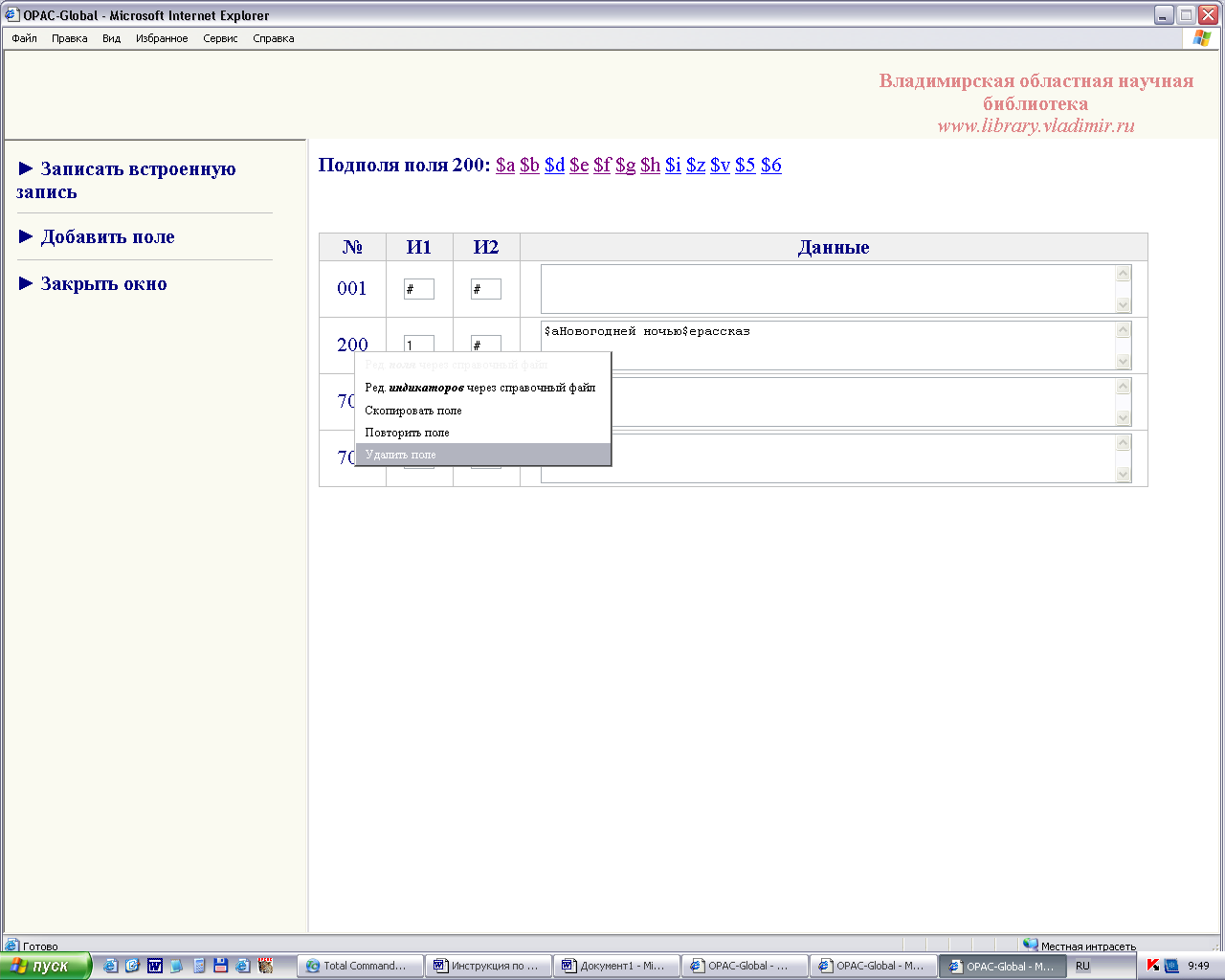


Рис. 1.1 – Рабочее окно программы OPAC-Global

Программа 1С:Библиотека имеет схожий функционал программы OPAC-Global однако из-за более простого интерфейса научиться пользоваться ей становится намного легче. Но даже эта программа не умеет подключаться к уже существующим базам данных MySQL, что делает эту программу плохим вариантом для заказчика.

Программа 1С:Библиотека, рабочее окно которой представлено на рис. 1.2:

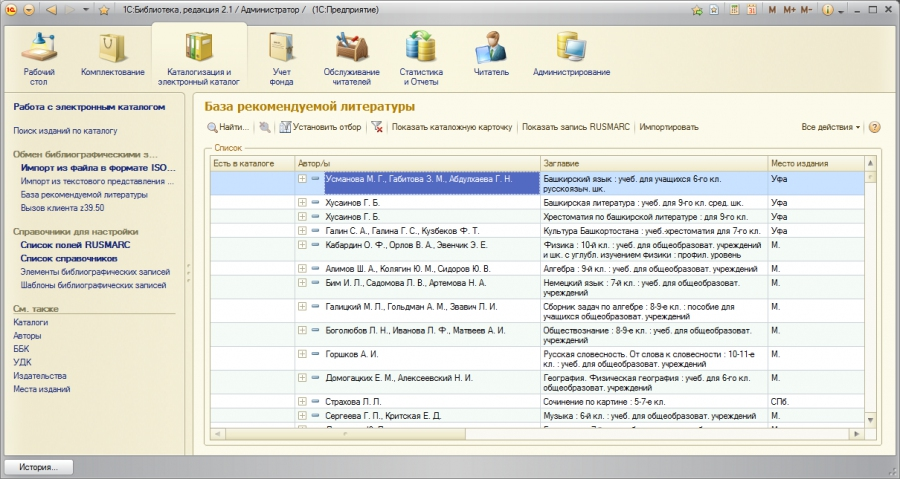


Рис. 1.2 Рабочее окно программы 1С:Библиотека

Таким образом на основе приведенных аналогов, можно сделать вывод о том, какими преимуществами аналогов должно обладать разрабатываемое средство и какие недостатки должны в нем отсутствовать. К достоинствам программного средства будут относится возможность работы с уже существующей базой данных MySQL и простой интерфейс.

## **1.3 Информационная база задачи**

Информационная база задачи состоит из входной, выходной и постоянной информации.

Входные данные:

* данные о книгах;
* данные о читателях;
* данные об условиях выдачи книги;
* критерии и данные для поиска;
* данные о датах выдачи книг.

Выходными данные:

* данные о взятых книгах читателями;
* отфильтрованные читатели по заданным критериям;
* отфильтрованные книги по заданным критериям;
* данные о должниках, не отдавших книги в срок.

Постоянными данными в программе является информация о библиотеке.

## **1.4 Функциональное назначение**

Данное приложение предназначено для хранения, редактирования, создания информации о книгах, писателях, жанрах, издательствах, языках, городах, обложках, а также возможности выдачи их читателям и учета выданных книг.

Приложения должно содержать следующие вкладки: книги, должники, читатели, авторы, языки, жанры, издатели, города, обложки. А также предусмотреть вспомогательные окна для создания и редактирования каждой из выше представленных сущностей.

Так же приложение должно реализовывать следующие функции:

* добавление, редактирование и удаление читателей;
* добавление, редактирование и удаление книг;
* возможность выдачи книги читателю;
* поиск по читателям по различным критериям;
* поиск по книгам по различным критериям;
* возможность продлить дату возврата книги читателя;
* возможность вернуть книгу читателю;
* контроль сроков возврата книги путем сравнения текущей даты и даты предположительной сдачи книги.

Программа представляет собой стандартное оконное Windows-приложение. Осуществляет взаимодействие с пользователем при помощи различных элементов управления: командных кнопок, текстовых полей для ввода значений, графических кнопок.

## **1.5 Выбор и обоснование средств разработки**

Выбор среды разработки и языка программирования всегда очень важен. Их нужно выбирать исходя из своих возможностей и возможностей среды разработки и языка, которые должны обеспечить соответствие программного средства с требованиями пользователя.

Для разработки этого программного средства нужны различные инструменты и средства, которые более подробно будут описаны ниже.

Для разрабатываемого программного продукта используются следующие средства:

* Visual Studio 2019;
* MySQL;
* С#.

В качестве среды разработки был выбран Visual Studio 2019.

Microsoft Visual Studio – бесплатная, полнофункциональная и расширяемая интегрированная среда разработки программного обеспечения, и ряд других инструментальных средств для создания современных приложений для Windows. Данная среда разработки предоставляет:

* мощные инструменты программирования, которые обеспечивают легкость написания кода, перемещения по нему и устранение ошибок;
* отладка для языка С#.

В качестве базы данных использовался MySQL.

MySQL характеризуется своей надежностью и простотой, гибкой поддержкой форматов чисел, строк переменной длины и меток времени, быстрой работой, масштабируемостью.

Поддерживаемые типы значений: TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT, FLOAT, DOUBLE, DOUBLE, REAL, DECIMAL, NUMERIC, DATE, DATETIME, TIMESTAMP, TIME, YEAR, CHAR, VARCHAR, TINYBLOB, TINYTEXT, BLOB, BLOB, TEXT, BLOB, MEDIUMBLOB, MEDIUMTEXT, BLOB, LONGBLOB, LONGTEXT, BLOB, ENUM и SET. Так же поддерживается специальное значение NULL.

Каждое значение в любом поле любой записи может быть любого из этих типов, независимо от типа, указанного при объявлении полей таблицы. Указанный при объявлении поля тип хранится для справки в его исходном написании, и используется в качестве основы для выбора предпочтений (так называемое «type affinity»: это подход, редко встречающийся в других СУБД) при выполнении неявных преобразований типов на основании похожести этого названия типа на что-либо, знакомое MySQL. В этот алгоритм зашит обширный перечень практикуемых в других СУБД вариантов названий типов данных. Если безопасного преобразования записываемого значения в предпочитаемый тип не получается, MySQL записывает значение в его исходном виде. Для получения значений из базы есть ряд функций для каждого из типов, и, если тип хранимого значения не соответствует запрашиваемому, оно тоже, по возможности, преобразуется.

В качестве языка программирования для реализации данного проекта был выбран С# – компилируемый статически типизированный язык программирования общего назначения. Именно этот язык наиболее подходит для написания системы учета книги библиотеки, так как С# успешно используется во многих областях приложения, далеко выходящих за указанные рамки, и позволяет создавать как простые приложения и утилиты, так и сложные системы.

Поддерживает такие парадигмы программирования как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование, обеспечивает модульность, раздельную компиляцию, обработку исключений, абстракцию данных, объявление типов (классов) объектов, виртуальные функции. Стандартная библиотека включает, в том числе, общеупотребительные контейнеры и алгоритмы. C# сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником – языком C++, – наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования.

C# широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений (игр).

C# – чрезвычайно мощный язык, содержащий средства создания эффективных программ практически любого назначения, от низкоуровневых утилит и драйверов до сложных программных комплексов самого различного назначения.

В частности, поддерживаются различные стили и технологии программирования, включая традиционное директивное программирование, ООП; имеется возможность работы на низком уровне с памятью, адресами, портами; возможность создания обобщённых алгоритмов для разных типов данных, их специализация и вычисления на этапе компиляции, используя шаблоны.

# **2 Проектирование программного средства**

## **2.1 Инфологический этап проектирования**

Инфологическое проектирование – построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Для описания семантической модели я использовал диаграмму сущность-связь.

Для программы системы учета книг в библиотеке были выделены следующие главные сущности: автор, книга, издательство, библиотекарь, администратор, читатель.

У сущности «автор» присутствуют следующие атрибуты: имя, фамилия, отчество. Сущность «издательство» содержит атрибуты город и название. Сущность «читатель» состоит из атрибутов имя, фамилия, отчество, адрес, номер телефона и дата рождения. У сущностей «администратор» и «библиотекарь» присутствуют атрибуты имя, фамилия и хэш пароля. У сущности «книга» имеются атрибуты название, автор, жанр, тип обложки, размер книги, шрифт текста и вид литературы.

Диаграмма сущность связь представлена на рис. 2.1:

Diagram

Description automatically generated

Рис. 2.1 Диаграмма сущность-связь

Взаимоотношения между сущностями иллюстрируются с помощью связей. В данной диаграмме имеются следующие связи: связь один-ко-многим между «автором» и «книгой». Связь один-ко-многим между «издательством» и «книгой». Эта связь показывает, что экземпляр книги может быть издан только одним издательством, а издательство может издать сколько угодно экземпляров книг. Связь один-ко-многим между «библиотекарем» и «администратором». Она показывает, что библиотекарь может быть создан только одним администратором, но администратор может создать сколько угодно библиотекарей. Связь многие-ко-многим между «читателем» и «книгой». Она показывает, что книга за свое существование могла находиться у многих читатели, а у читателей, за свое существование, могло быть много книг. У этой связи так же атрибуты даты взятия, даты возврата, а также библиотекаря, который выдал книгу читателю.

Таким образом была спроектирована концептуальная модель БД.

## **2.2 Логический этап проектирования**

Логическое проектирование — создание схемы базы данных на основе конкретной модели данных. Для реляционной модели данных даталогическая модель — набор схем отношений, обычно с указанием первичных ключей, а также связей между отношениями, представляющих собой внешние ключи.

Будем рассматривать логическое проектирование БД для реляционной модели данных, так как современные СУБД – реляционные.

Разрабатываемая схема БД будет содержать следующие таблицы:

* таблица «reader»;
* таблица «author»;
* таблица «genre»;
* таблица «language»;
* таблица «city»;
* таблица «publishing\_house»;
* таблица «cover»;
* таблица «era»;
* таблица «type\_of\_literature»;
* таблица «book\_size»;
* таблица «font\_size»;
* таблица «book»;
* таблица «admin»;
* таблица «librarian»;
* таблица «record»;

Схема БД представлена на рис. 2.2:

Graphical user interface, application, Word

Description automatically generated

Рис. 2.2 Схема базы данных

Таблица «reader» хранит информацию о читателе, его данные: имя, фамилия, отчество, дата рождения, номер телефона и адрес.

Таблица «author» хранит информацию об авторе, его имя, фамилию, отчество.

Таблица «genre» хранит всевозможные жанры.

Таблица «language» хранит различные языки.

Таблица «language» хранит различные горда.

Таблица «publishing\_house» хранит разных издательства, их названия, а также эта таблица связана с таблицей «city». Так, у издательства есть и город.

Таблица «cover» хранит виды обложек.

Таблица «era» хранит список временных эпох, в которых была написана книга.

Таблица «type\_of\_literature» хранит национальность вида литературы.

Таблица «book\_size» хранит примерные размеры книги в мм, а так же общее название для такого размера.

Таблица «font\_size» хранит размер шрифта, использованного при печати, а также его название.

Таблица «book» соединена связью один-ко-многим с таблицами «author», «genre», «language», «publishing\_house», «cover», «era», «type\_of\_literature», «book\_size» и «font\_size», а также имеет с названием книги.

Таблица «admin» хранит имя, фамилию и хэш пароля администратора.

Таблица «librarian» хранит имя, фамилию и хэш пароля библиотекаря, а также она связана с таблицей «admin», указывая на администратора, создавшего библиотекаря.

Таблица «record» связана с таблицами «reader», «book» и «librarian», что показывает какому читателю каким библиотекарем была выдана какая книга. Также таблица содержит поля, хранящие дату взятия и дату возврата книги.

## **2.3 Функциональная модель**

Для описания функциональной модели я буду использовать диаграмму вариантов использования. Она представлена на рис. 2.3:

Diagram

Description automatically generated

Рис 2.3 Диаграмма вариантов использования

В системе учета книг для библиотеки существуют следующие актеры: администратор, библиотекарь и гость. Гость может только проходить процесс авторизации. В результате которого он станет либо администратором, либо библиотекарем, в зависимости от вида аккаунта, который принадлежит пользователю.

Библиотекарь может работать с книгами и читателями: редактировать, удалять, добавлять их. У библиотекаря есть возможность искать книги и читателей по различным критериям. Так же библиотекарь может выдавать книги читателям, искать должников, продлевать и возвращать книги, находящие у читателей.

Администратор может делать все то же самое, что и библиотекарь, только помимо этого, администратор может управлять списком библиотекарей. Он может добавлять, удалять и просматривать списки библиотекарей.

## **2.4 Конфигурация и состав ПО, требуемых для работы приложения**

Программное средство представляет собой оконное Windows приложение. Для запуска требуется, чтобы компьютер удовлетворял следующим требованиям:

* Процессор Intel Core i3 или более новые модели с тактовой частотой 2 ГГц;
* Microsoft Windows 10;
* 512 МБ ОЗУ (минимум);
* 20 МБ свободного места на диске;
* Монитор VGA разрешением 1024 x 768 с поддержкой полноцветного режима (минимальные требования);
* Мышь или другое указывающее устройство.

Так же, для запуска приложения, на компьютере должен быть установлен Microsoft .NET Framework версии 4.8.