**ОТЧЕТ**

**о выполнении долгосрочного домашнего задания по**

**дисциплине «Прикладные информационные технологии»**

Выполнили: Буркин Д.В Никитин В.С.

Проверил: Лемешкин Ю.В.

# Цель работы

Целью данного домашнего задания является развитие и закрепление навыков и умений по разработке и сопровождению ПО с применением реляционных баз данных и создания автоматизированных информационных систем. Результатом работы является законченный проект реляционной базы данных вкупе с сопроводительной документацией и ПО.

В ходе работы было создано графическое приложение, позволяющее организовать работу библиотеки. Т.е приложение, способное хранить и редактировать данные о книгах и читателях в систематизированном виде (добавлять и удалять данные, сортировать списки по различным критериям, отслеживать выдачу книг и т.д.)

**Архитектура приложения**

Приложение работает с тремя базами данных:

1. База данных Книг (books)

Содержит информацию о названии, авторе, годе выхода, жанре, наличии. Первичный ключ – id, генерируется автоматически при добавлении новой книги

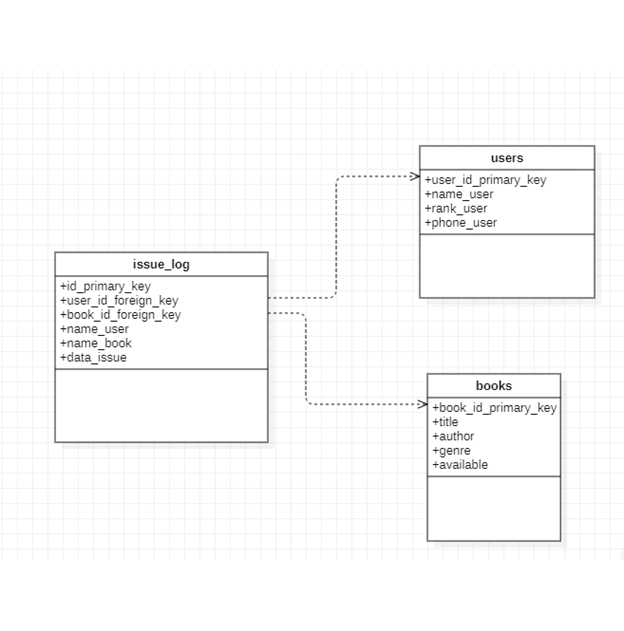
1. База данных Читатели (users)

Содержит информацию о фамилии, ранге, номере телефона. Ранг – целое число от одного до трёх, означает уровень доверия к данному читателю. Чем ниже ранг, тем больше доверие. Первичный ключ – id, генерируется автоматически при добавлении нового читателя

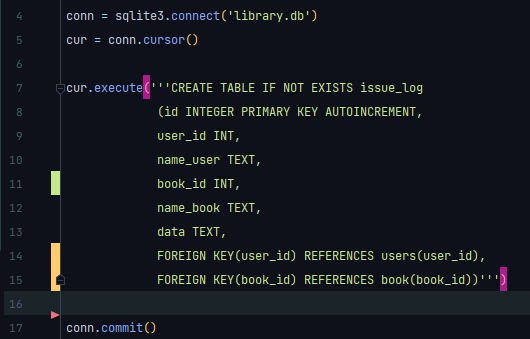
1. База данных Записи о выдаче (issue\_log)

Содержит информацию о названии выданной книги и её id(внешний ключ); о читателе, кому была выдана книга, и его id(внешний ключ), дате выдаче

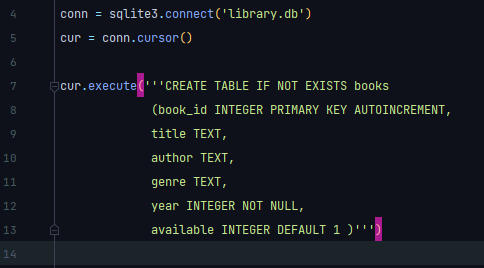
База данных library.db



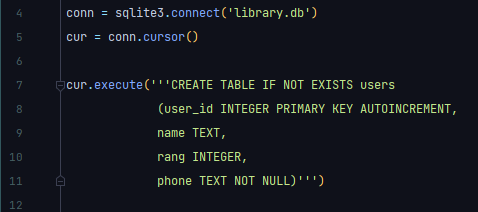
База данных выдачи/возврата книг



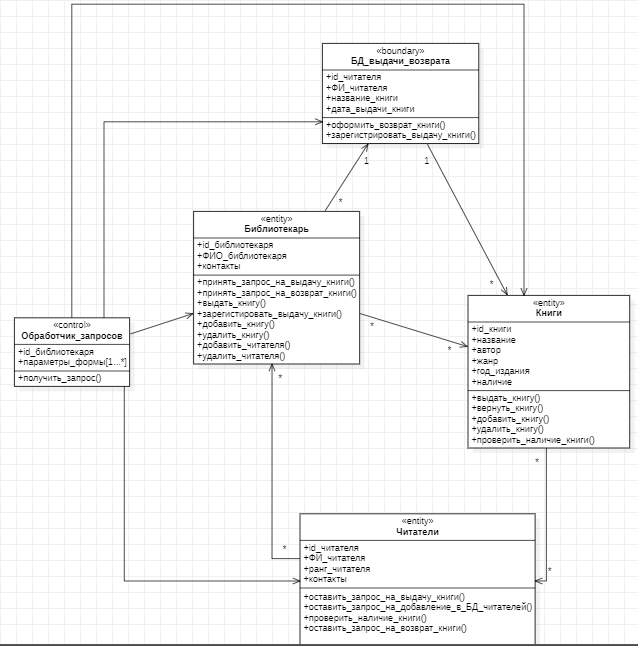
База данных Книги

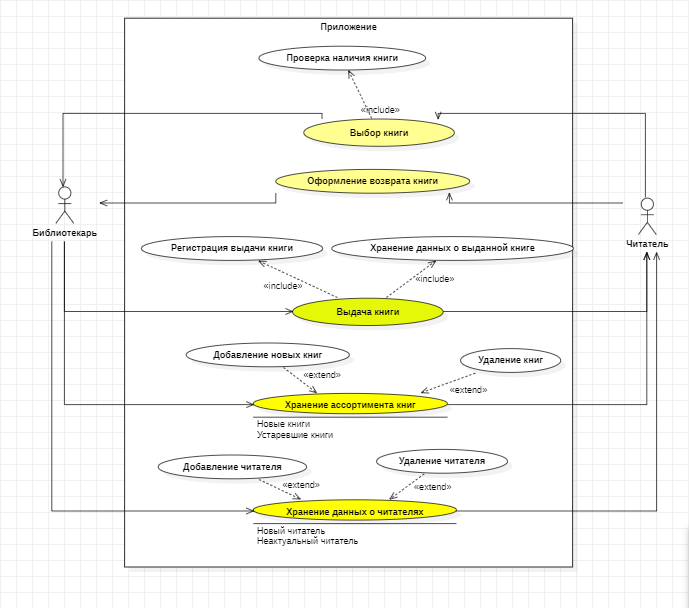


База данных Пользователи(читатели библиотеки)



Структура использования данного приложения изображена ниже на UML-диаграмме(UseCase)





НОРМАЛЬНЫЕ ФОРМЫ

**Таблица books**

1. Каждый столбец зависит от первичного ключа(book\_id), т.е. в таблице отсутствуют частичные зависимости. (условие 2НФ)
2. Все ее атрибуты являются простыми (неделимыми) (условие 1НФ)
3. В таблице отсутствуют транзитивные зависимости(условие 3 НФ), Транзитивные зависимости – это зависимости между атрибутами в таблице, когда один атрибут зависит от другого через третий атрибут. Например, если в таблице есть атрибуты "имя", "адрес" и "город", то "адрес" зависит от "города", а "имя" зависит от "адреса" через "город".

Данная таблица удовлетворяет 3НФ(нормальной форме)

**Таблица users**

Аналогично, 3НФ

**Таблица issue\_log**

Таблица issue\_log не является полностью второй нормальной формой (2НФ), так как атрибут name\_user зависит не только от первичного ключа id, но и от внешнего ключа user\_id

Таблица удовлетворяет условиям лишь первой нормальной формы.

В приложении реализованы три класса: MainWindow(главное окно), BooksWin(окно книг), UserWin(окно читателей). Как ясно из названий каждый класс реализует отдельное окно. Интерфейс каждого из них интуитивно понятен и не нуждается в описании.

Методы класса **MainWindow**:

openBooksWindow(self) – слот для кнопки btn\_books(‘Книги’), связанный с ней сигналом clicked. Открывает окно книг (класс BooksWin)

openUsersWindow(self) - слот для кнопки btn\_users(‘Читатели’). ), связанный с ней сигналом clicked. Открывает окно читателей (класс BooksWin)

issue\_book(self) – функция, осуществляющая выдачу книг. Получает название книги и id читателя из полей name\_book\_entry и name\_user\_entry.

Вносит соответствующие сведенья в базу данных issue\_log. Также меняет значение атрибута available (books) на 0 (книга выдана). Грамотно реагирует на ввод пустых значений и попытку выдать уже выданную книгу. Сообщает пользователю об этом.

return\_book(self) – функция, возвращающая книгу обратно в библиотеку. Получает название книги из поля name\_book\_entry. Удаляет запись о выдаче из базы данных issue\_log. Присваевает значению атрибута available для соотвеющей книги значение 1 (в наличии). Грамотно реагирует на ввод пустых имён и попытку вернуть уже возвращённую книгу.

output\_book(self) – функция, показывающая все записи о выданных на данный момент книгах. Если таковых нет, пишет “все книги возвращены”

Методы класса **BooksWin**:

add\_book(self) – функция добавления в базу данных книги.

Получает на с полей ввода название(title), автора(author), год издания(year) и жанр. Корректно обрабатывает возникшие ошибки при вводе данных.

show\_books(self) – функция вывода всех книг из таблицы books. При это пользователю предлагается выбрать способ сортировки при выводе книг(по умолчанию – как в таблице БД, по году издания, по авторам, по название)

delete\_book(self) – функция удаления книги по id.

Класс **UserWin**:

add\_reader(self) – функция добавления читателя в таблицу user.

Принимает на вход имя(Иванов И), ранг(от 1 до 3) и телефон(11 цифр).(id присваивается как по возрастанию номеров в таблице). Корректно обрабатывает возникшие ошибки при вводе.

delete\_reader() – функция удаления читателя из БД по id.

show\_readers(self) – функция вывода всех читателей, занесенных в базу данных.

sort\_user\_rang(self) – функция сортировки читателей из БД по рангу и вывод в консоль.