

# Курсовой проект на тему: «Создание приложения для общественной библиотеки»

Исполнитель:  
Гришин Егор Борисович, ИУ7-65Б

Научный руководитель:  
Волкова Лилия Леонидовна

Москва  
2021

# Цель и задачи

**Цель:** разработка базы данных и приложения для библиотеки.

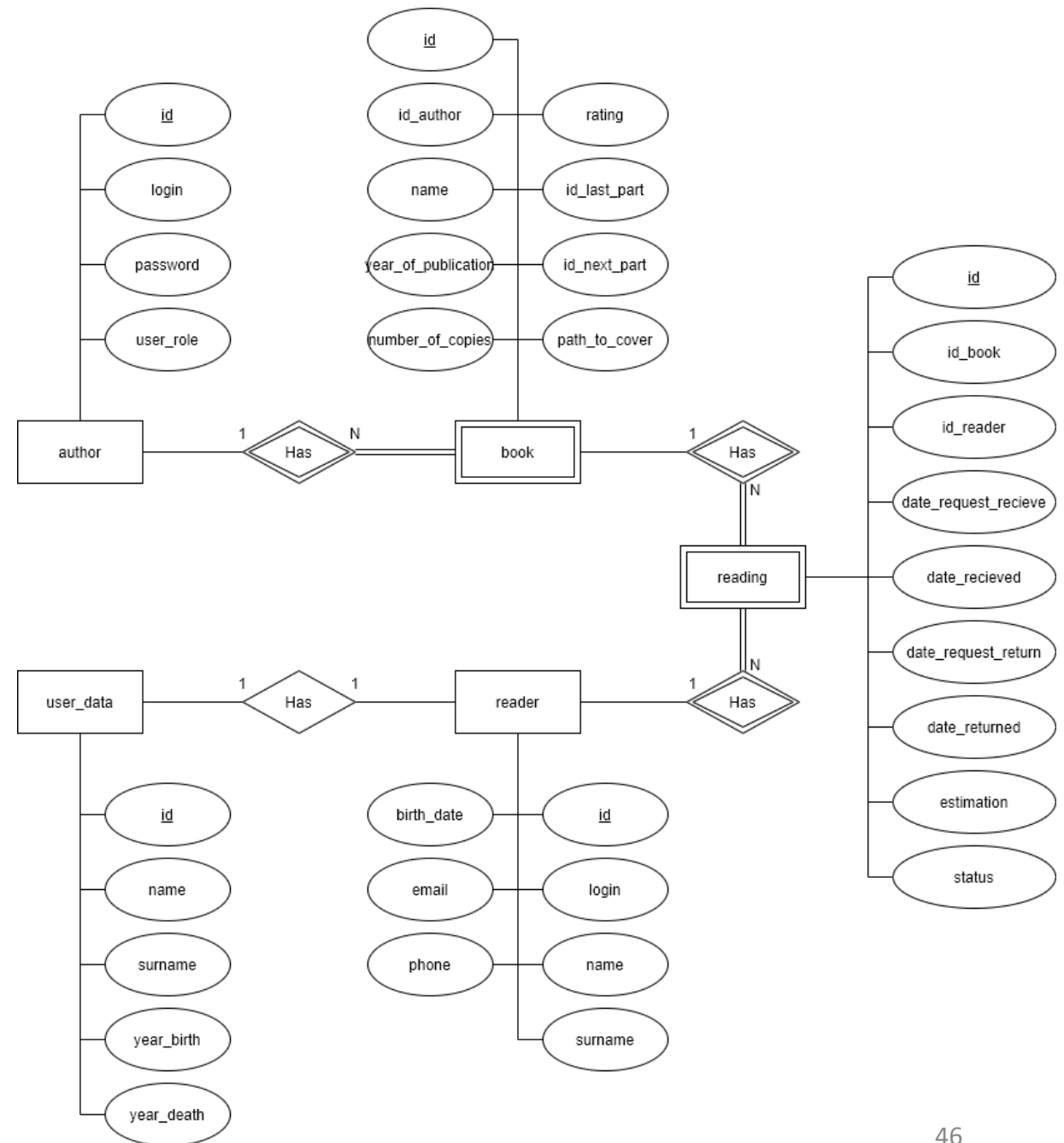
## **Задачи:**

- формализовать задание, выделив соответствующих акторов и их функционал;
- провести анализ СУБД и выбрать наиболее подходящую;
- спроектировать базу данных;
- спроектировать архитектуру приложения;
- разработать приложение.

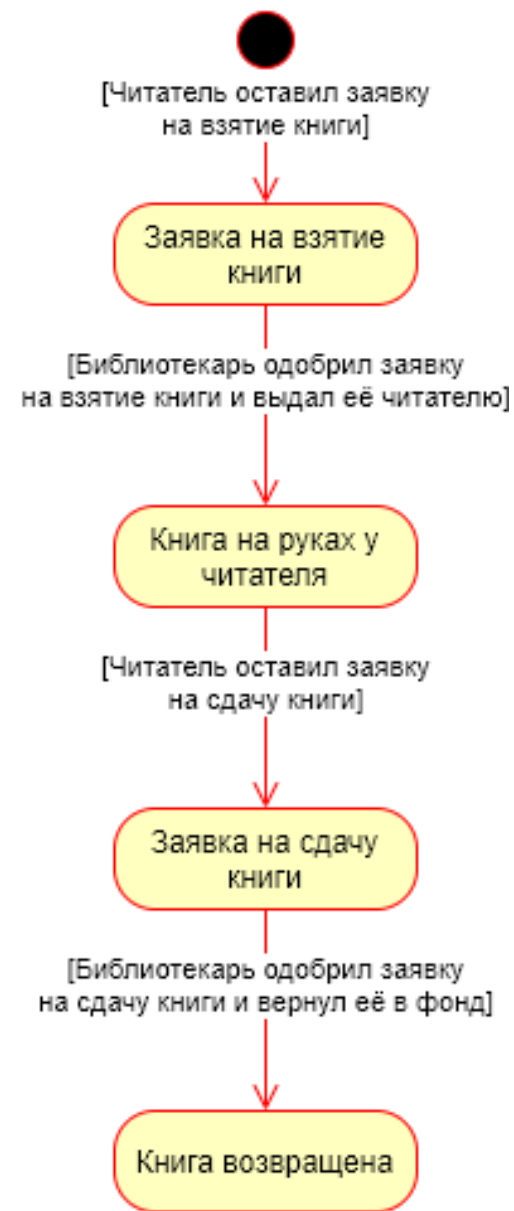
# ER-диаграмма

Выделенные сущности:

- книга - book;
- автор - author;
- формуляр - reading;
- читатель - reader;
- данные пользователя системы - user\_data.



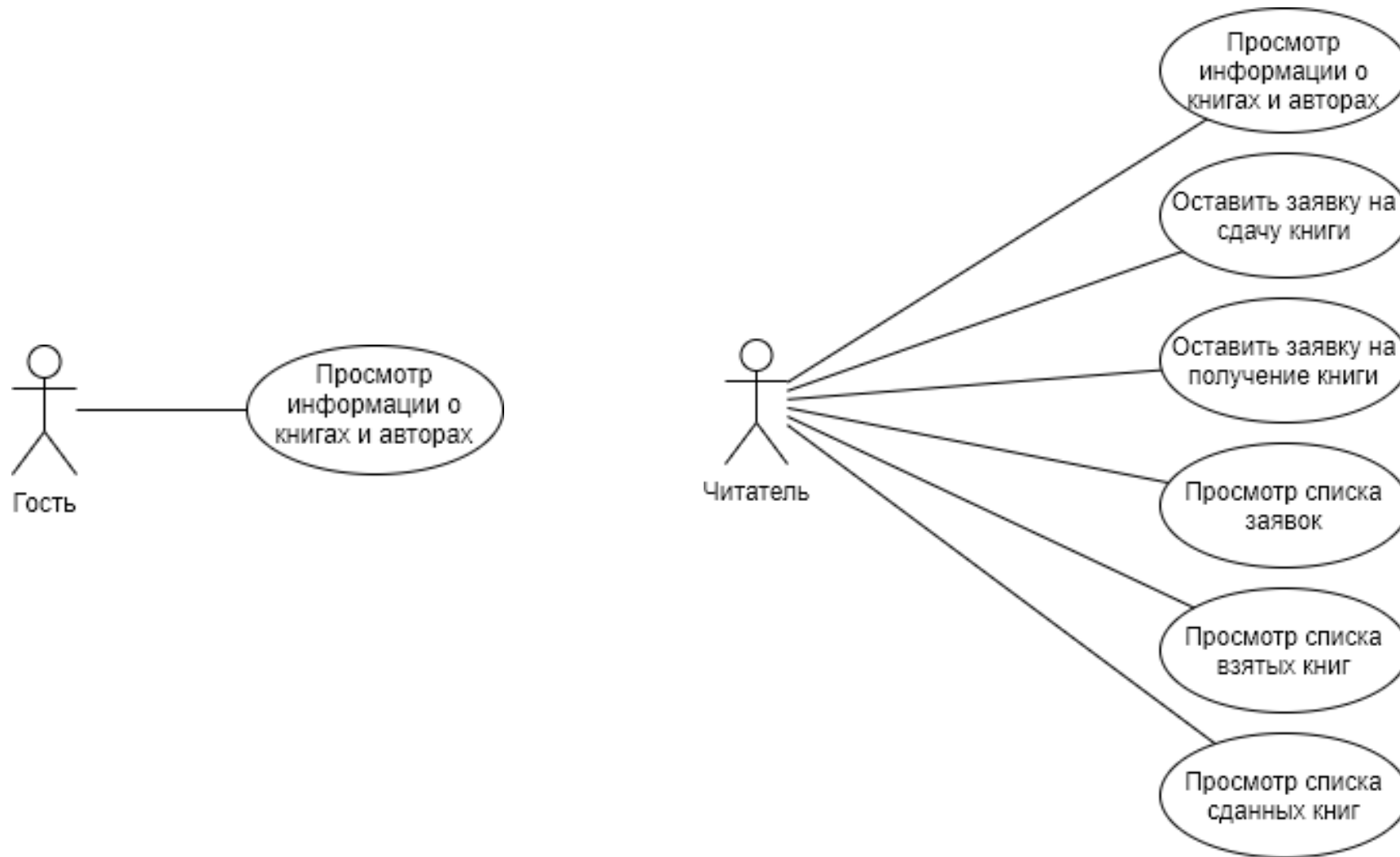
# Диаграмма состояний формуляра



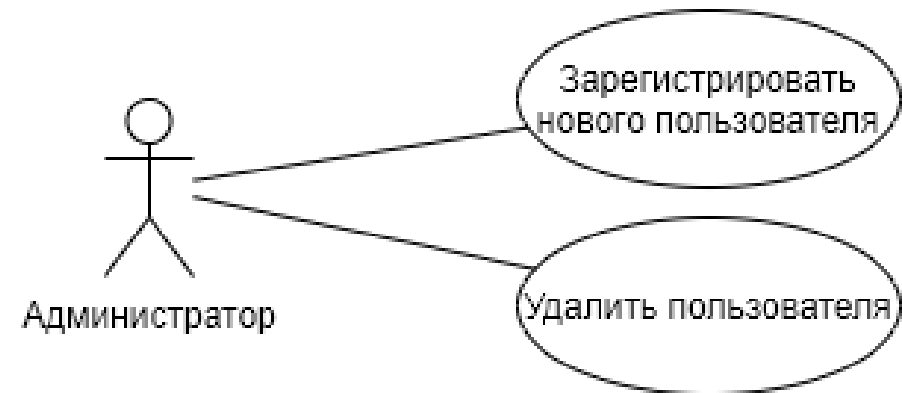
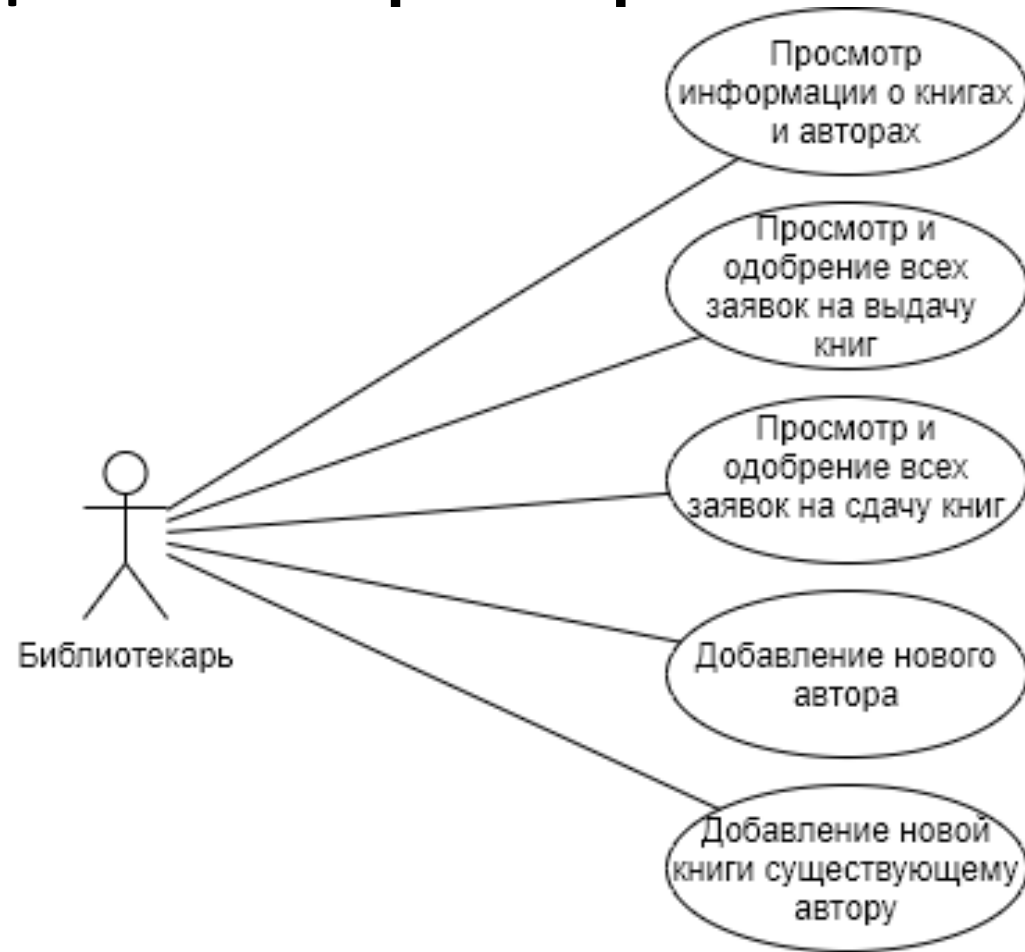
# Пользователи системы

- Гость — это неавторизованный пользователь;
- Читатель - это авторизованный пользователь. Он может подать заявку на взятие книги, если она у него не на руках, и на её сдачу в противном случае. Также ему доступны списки книг, на которые он оставил заявки, которые у него на руках и которые он уже прочел и сдал;
- Библиотекарь - это авторизованный пользователь, контролирующий выдачу книг. Он одобряет заявки читателей, и добавляет новые книги и авторов в БД;
- Администратор является авторизованным пользователем с повышенным уровнем полномочий — он может регистрировать новых пользователей и удалять существующих.

# Use-Case диаграммы гостя и читателя



# Use-Case диаграммы библиотекаря и администратора

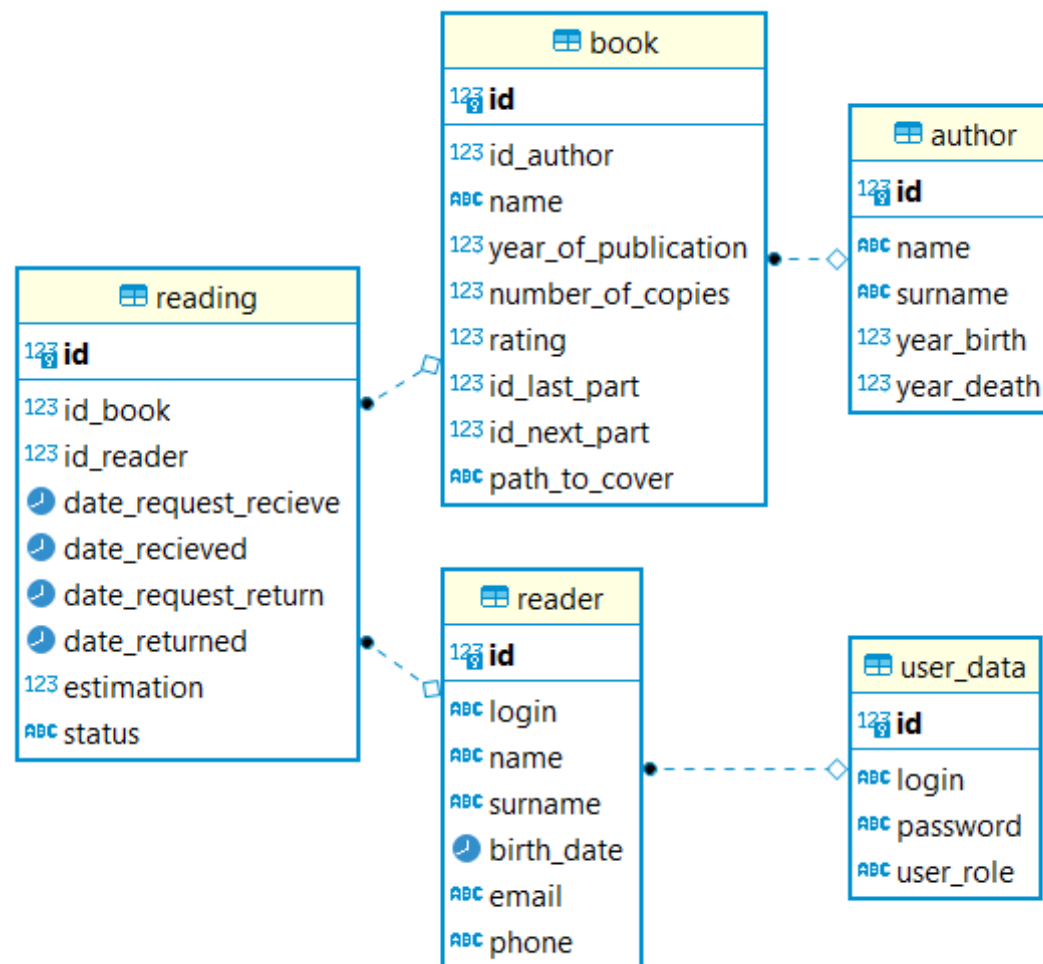


# Выбор СУБД

	Oracle	MySQL	Microsoft SQL Server	PostgreSQL
Простота в использовании	+	+	+	+
Бесплатная	-	+	+	+
Безопасность данных	+	-	+	+
Поддержка стандарта SQL	+	+	+	+
Поддержка хранимых процедур и триггеров	+	+	+	+
Кроссплатформенность	+	+	-	+



# Диаграмма БД



# Спроектированные триггеры

- Триггер поддержания серийности книг;
- Триггер поддержания оценки книг;
- Триггер поддержания количества книг в фонде.

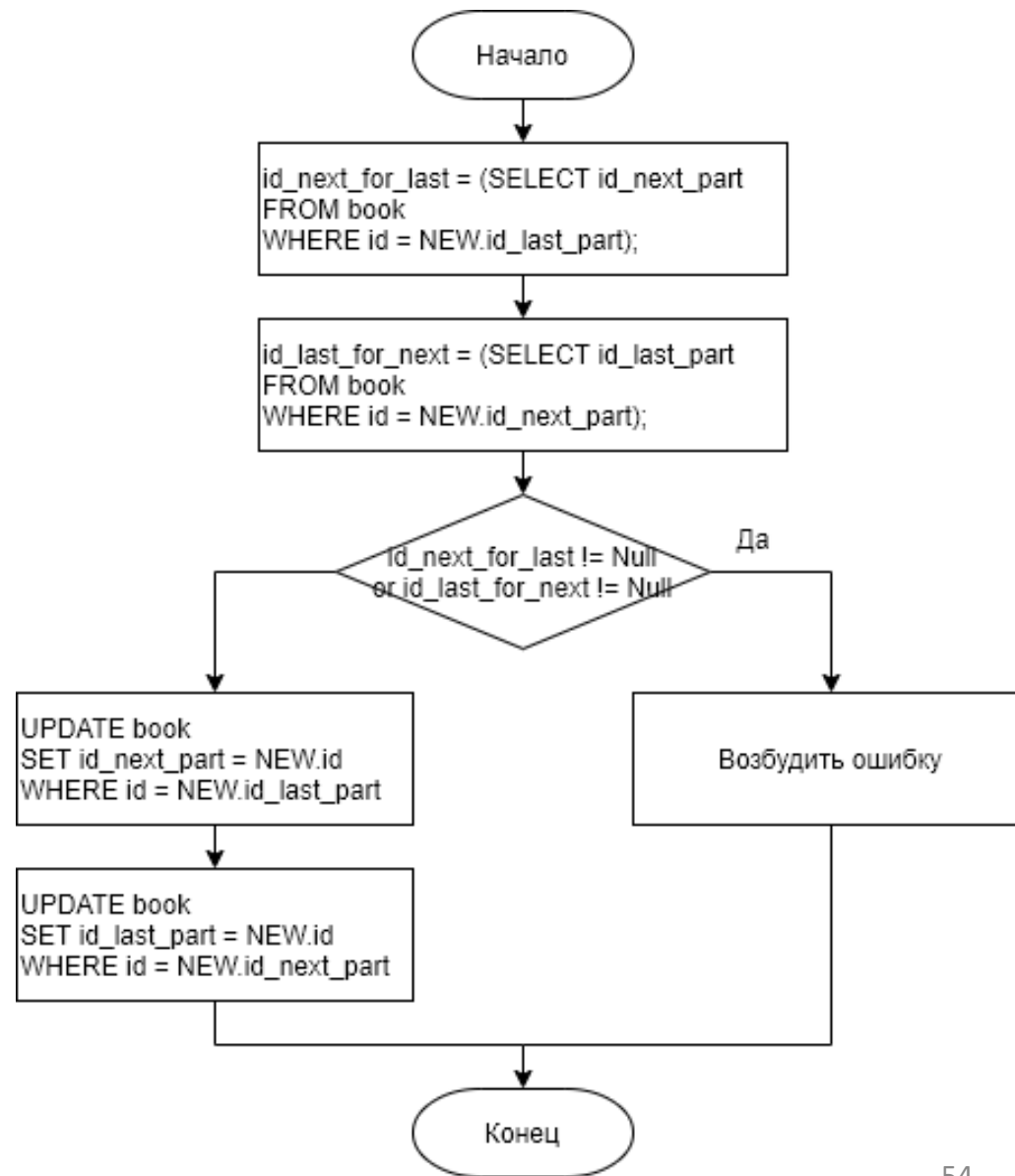
# Триггер поддержания серийности книг

При добавлении новой книги срабатывает данный триггер.

Триггер обеспечивает целостность связи частей серии.

Библиотекарю будет показано сообщение об ошибке, ему нужно будет повторить попытку добавления книги, проставив верные ссылки.

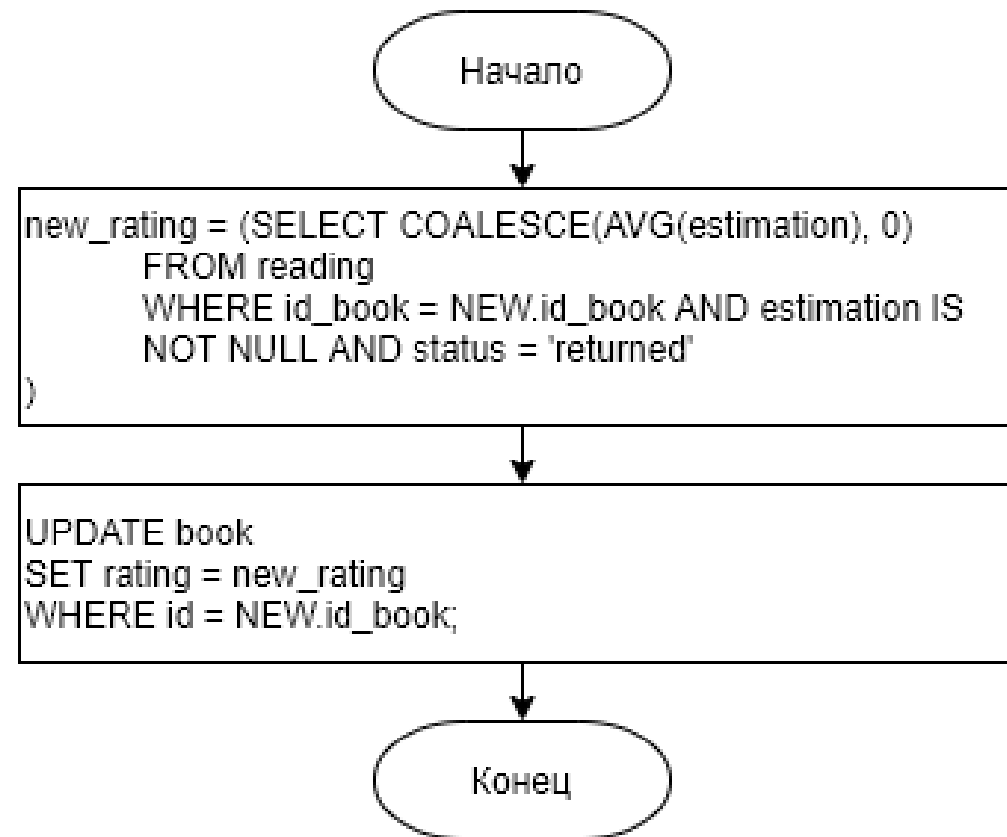
Ограничение триггера: книги добавляются только в начало или в конец серии.



# Триггер поддержания оценки книг

При возврате книги, в соответствующей записи в таблице библиотечных карточек reading полю status присваивается значение 'returned' и срабатывает данный триггер.

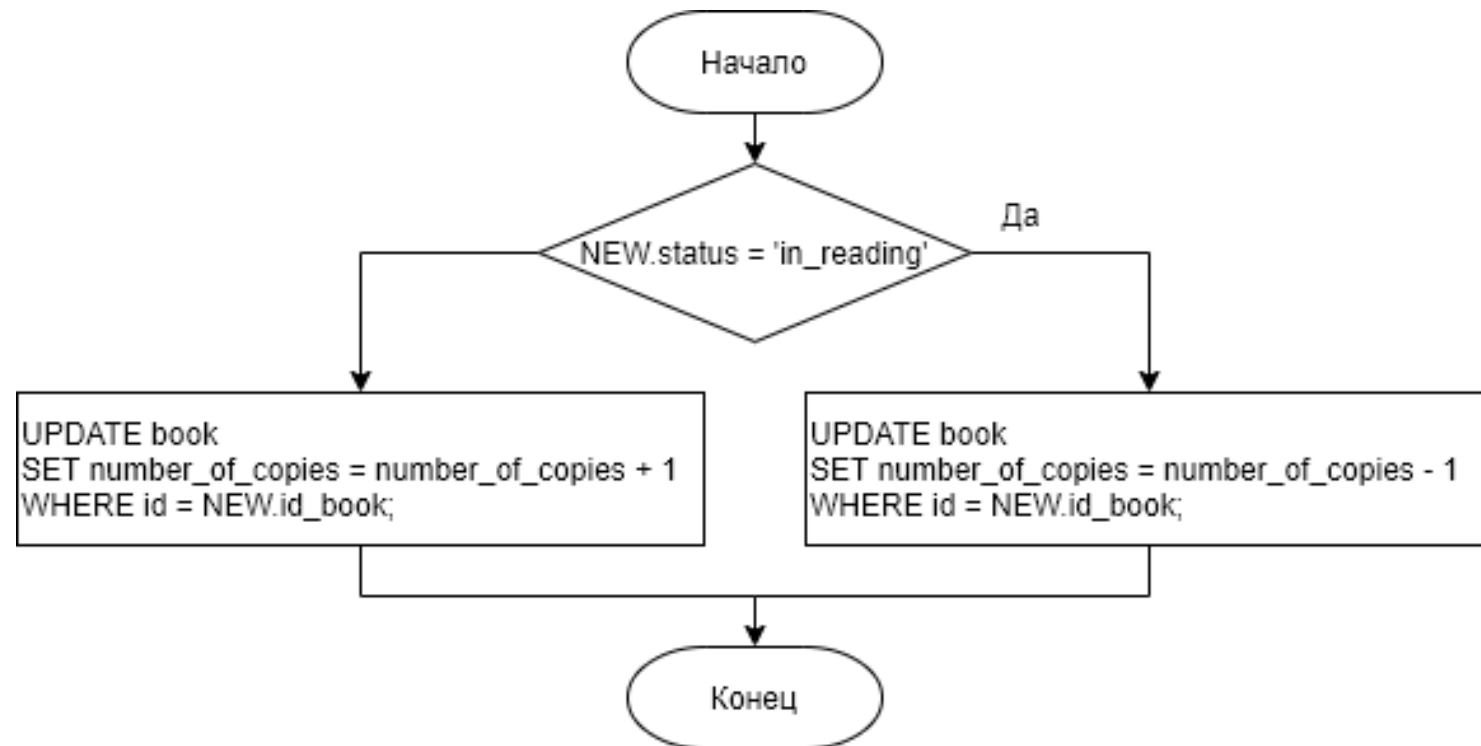
При возврате книги ей может быть дана оценка - estimation. На основании всех данных оценок триггер вычисляет рейтинг книги, обеспечивая его актуальность.



# Триггер поддержания количества книг в фонде

При выдаче или сдаче книги, в соответствующей записи в таблице reading полю status присваивается значение 'in\_reading' или 'returned' соответственно. При обоих событиях срабатывает этот триггер.

Триггер обеспечивает актуальность количества оставшихся в фонде экземпляров книги.



# Сложный запрос выбора всех книг серии

Запрос выдает информацию о  
всех книгах из серии, в которую  
входит книга с id = ID\_BOOK

```
WITH RECURSIVE last_parts AS (  
    SELECT *  
    FROM book  
    WHERE id = <ID_BOOK>  
    UNION ALL  
    SELECT book.*  
    FROM last_parts JOIN book ON last_parts.id_last_part =  
        book.id  
) , next_parts AS (  
    SELECT *  
    FROM book  
    WHERE id = <ID_BOOK>  
    UNION ALL  
    SELECT book.*  
    FROM next_parts JOIN book ON next_parts.id_next_part =  
        book.id  
) , res AS (  
    SELECT * FROM last_parts  
    UNION  
    SELECT * FROM next_parts  
)  
SELECT *  
FROM res  
WHERE id != <ID_BOOK>;
```

# Архитектура приложения

Для разработки программы выбран язык Python по причине совмещения нескольких парадигм программирования, а также из-за большого разнообразия представленных библиотек и фреймворков для создания веб-приложения.

Для создания данного проекта в качестве инструмента, который облегчит процесс создания веб-приложения, был выбран фреймворк Flask. Благодаря его простоте и гибкости, разработчик может сам выбрать способ реализации тех или иных задач.

# Пример обработки заявок библиотекарем

Для каждой заявки на взятие книги проверяется её наличие в фонде. Если книги в наличии нет, то библиотекарю не будет отображаться кнопка одобрения данной заявки.

## Заявки на выдачу книг

### Заявка №2

Читатель: Анна Шишихина

Книга: Организация переворота с помощью табакерки

Количество оставшихся экземпляров: 1

[Одобрить заявку](#)

### Заявка №1

Читатель: Егор Гришин

Книга: Организация переворота с помощью табакерки

Количество оставшихся экземпляров: 1

[Одобрить заявку](#)

## Заявки на выдачу книг

### Заявка №1

Читатель: Егор Гришин

Книга: Организация переворота с помощью табакерки

Количество оставшихся экземпляров: 0

Одобрить заявку временно не возможно, т.к. все экземпляры данной книги выданы



# Заключение

Цель курсовой работы достигнута: разработаны база данных и приложение для библиотеки. Все задачи решены:

- задание формализовано, акторы и их функционал выделены;
- проведен анализ СУБД и выбрана наиболее подходящая;
- спроектирована база данных;
- спроектирована архитектура приложения;
- разработано приложение.

Получен опыт разработки базы данных.