Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА ИУ7 «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

КУРСОВАЯ РАБОТА

НА ТЕМУ:

Разработка базы данных для хранения и обработки данных библиотечной картотеки

Студент	**ИУ7-62Б**		**Новиков**
	(группа)	(подпись, дата)	(И.О. Фамилия)
Руководитель курсового проекта			**Кузнецова О.В.**
		(подпись, дата)	(И.О. Фамилия)
Консультант			
		(подпись, дата)	(И.О. Фамилия)

2025 г.

РЕФЕРАТ

Расчетно-пояснительная записка к курсовой работе содержит 48 страниц, 12 рисунков, 15 таблиц, 20 источников, 1 приложения.

Целью работы является разработка базы данных для хранения и обработки данных библиотечной картотеки.

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей расчетно-пояснительной записке применяют следующие сокращения и обозначения.

АПУ — алфавитно-предметный указатель

ББК — библиотечно-библиографическая классификация

БД — база данных

РГБМ — Российская государственная библиотека для молодежи

СУБД — система управления базами данных

ISBN — International Standard Book Number, международный стандартный книжный номер

ВВЕДЕНИЕ

Издревле библиотеки служили важнейшим источником информации. До появления интернета, а также в эпоху его раннего развития, именно библиотеки часто были единственным местом, где можно было найти ответ на интересующий вопрос.

Однако и в цифровую эпоху, когда в интернете доступно огромное количество информации, библиотеки остаются ценным ресурсом. Они востребованы как людьми, предпочитающими работать с физическими носителями, так и теми, кому нужна информация, которую невозможно найти в оцифрованном виде. Кроме того, библиотеки предоставляют доступ не только к книгам, но и к газетам, журналам, плакатам, дискам и другим печатным и электронным материалам.

Появление интернета не уничтожило библиотеки, но существенно изменило их работу. Теперь нет необходимости хранить в бумажном виде информацию об изданиях, имеющихся в наличии, читателях и выдачах книг: все это переносится в цифровой формат, где информация становится проще в обработке и доступе.

Целью данной курсовой работы является разработка базы данных для хранения и обработки данных библиотечной картотеки.

Для достижения поставленной цели необходимо решить следующие задачи:

— провести анализ существующих библиотечных сервисов;

— определить требования к базе данных и программному обеспечению;

— спроектировать сущности базы данных и задать их ограничения;

— выбрать инструменты для реализации базы данных и программного обеспечения;

— реализовать базу данных и программное обеспечение для работы с ней;

— провести исследование на основе разработанной базы данных.

1 Аналитический раздел

В данном разделе проводится анализ предметной области, рассматриваются существующие решения в области электронных библиотек и сервисов для работы с книгами. Формулируются основные требования к проектируемой системе, описываются данные, которые необходимо хранить, а также определяются категории пользователей будущего приложения. На основе проведенного анализа описывается диаграмма вариантов использования и диаграмма сущность-связь проектируемой базы данных.

1.1 Анализ предметной области

Современные библиотеки давно перестали ограничиваться только хранением печатных изданий. В их деятельность входят организация учета большого количества экземпляров книг, газет, журналов и других носителей, ведение информации о читателях и контроль за процессом выдачи и возврата литературы. При традиционном подходе многие операции выполняются вручную: на каждого читателя заводится читательский билет, а библиотекари при резервировании, выдаче и возврате книги делают пометки в соответствующих листах. Такой способ работы трудоемок и подвержен ошибкам.

Использование информационных систем позволяет автоматизировать эти процессы и существенно повысить удобство как сотрудников, так и посетителей библиотеки, свести к минимуму ошибки при работе с формулярами и ускорить процесс обслуживания. Однако далеко не все библиотеки могут позволить себе дорогостоящее оборудование. Поэтому возникает потребность в разработке доступных решений, которые могут работать на обычных мобильных устройствах без специализированного оборудования.

1.2 Анализ существующих аналогов

Для анализа были рассмотрены следующие сервисы электронных библиотек: сайт Российской государственной библиотеки для молодежи (РГБМ) [1], сайт библиотеки МГТУ им. Н.Э. Баумана [2], сайт библиотек Москвы [3].

Для сравнения решений выделим критерии:

— КР1 — наличие возможности просматривать подробную информацию о книге;

— КР2 — наличие поиска по авторам, ключевым словам и жанрам;

— КР3 — наличие возможности бронировать книги;

— КР4 — наличие возможности добавлять понравившиеся книги в избранное;

— КР5 — наличие возможности вставать в очередь на книгу.

В таблице 1.1 приводится сравнение существующих решений по приведенным критериям.

Таблица 1.1 – Сравнение существующих решений

Решение	**КР1**	**КР2**	**КР3**	**КР4**	**КР5**
РГБМ	-	+	+	-	-
Библиотека МГТУ	+	+	+	+	-
Библиотеки Москвы	+	+	+	+	-

Из приведенной таблицы видно, что в наибольшей степени заданным критериям соответствуют библиотека МГТУ и библиотеки Москвы. Однако даже они не предоставляют пользователю возможности вставать в очередь на книгу, все экземпляры которой заняты, вынуждая самостоятельно регулярно проверять сайт либо использовать систему уведомлений.

1.3 Формализация задачи

В рамках курсовой работы необходимо спроектировать базу данных для хранения информации о пользователях, книгах, авторах, издателях, классификаторе, алфавитном указателе, избранных книгах, бронированиях и выдачах, очередях на книгу. Также требуется разработать приложение, предоставляющее графический интерфейс к базе данных с возможностью просмотра, добавления, редактирования и удаления информации из нее. Необходимо рассмотреть несколько пользовательских ролей с разным набором функция и различными правами: гость, читатель, библиотекарь, модератор.

1.4 Формализация данных

Выделим следующие сущности, информация о которых должна быть представлена в базе данных:

— пользователь;

— книга;

— автор;

— издатель;

— библиотечно-библиографическая классификация (ББК);

— алфавитно-предметный указатель (АПУ);

— бронь;

— выдача;

— избранное;

— очередь.

Какие сведения должна содержать каждая сущность, приведено в таблице 1.2

Таблица 1.2 – Сущности и информация о них

Сущность	**Информация**
Пользователь	Фамилия, имя, отчество, телефон, электронная почта, пароль, роль пользователя в системе
Книга	Название, аннотация, автор, издатель, ISBN, формат выпуска, ББК-код, объем, год издания, язык, язык оригинала, количество экземпляров, количество свободных экземпляров
Автор	Имя
Издатель	Название, описание, телефон, электронная почта
ББК	Код, расшифровка
АПУ	Термин, соответствующий ББК-код
Бронь	Пользователь, книга, дата бронирования, дата отмены бронирования
Выдача	Пользователь, книга, дата выдачи, крайний срок возврата, количество продлений
Избранное	Пользователь, книга
Очередь	Пользователь, книга, дата постановки в очередь

На рисунке 1.1 приведена диаграмма сущность-связь в нотации Чена, построенная на основе представленных сведениях об информации, которую должна содержать каждая сущность.

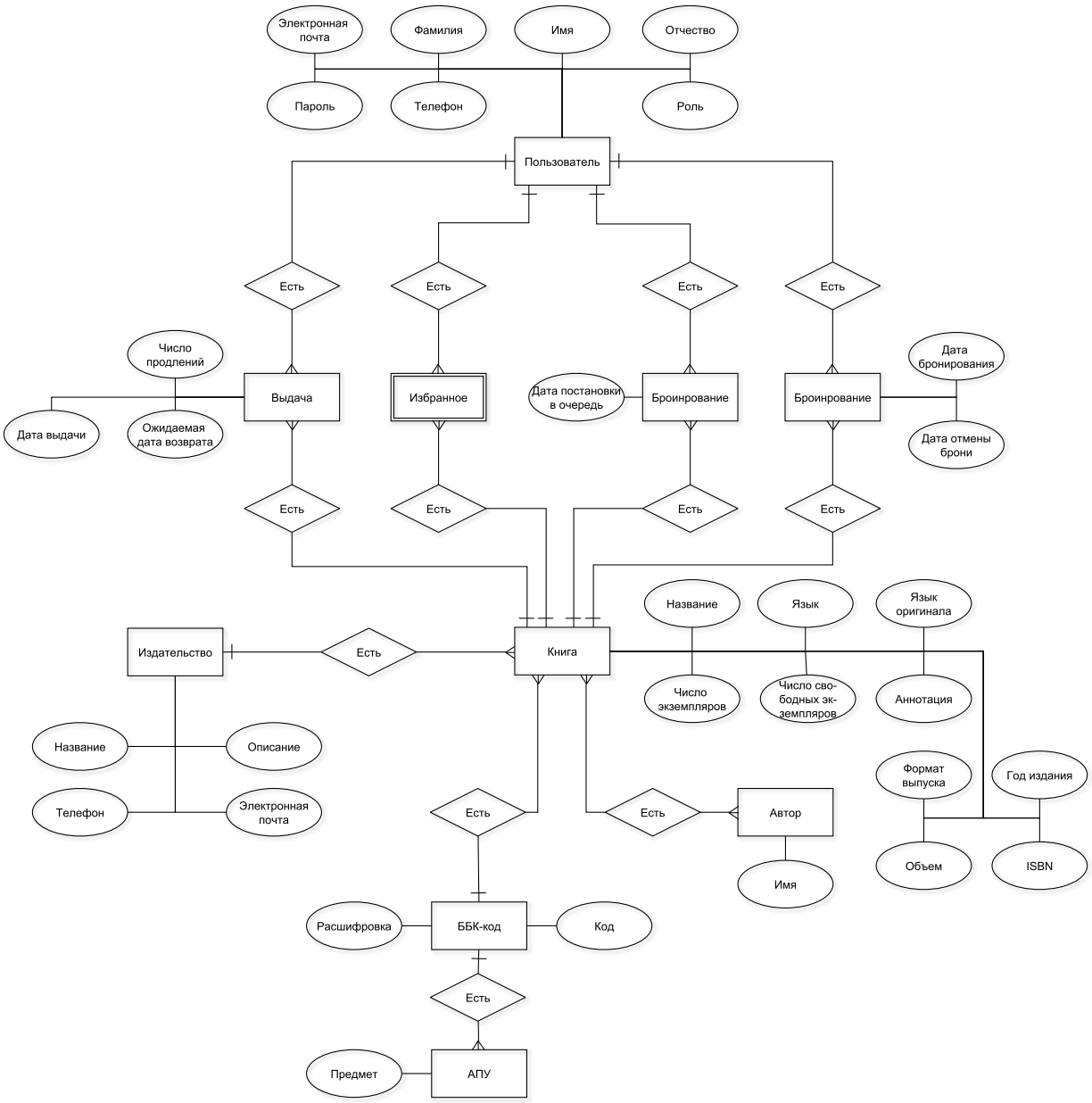

Рисунок 1.1 – Диаграмма сущность-связь в нотации Чена

1.5 Формализация ролей пользователей

Исходя из предметной области задачи выделим следующие группы пользователей разрабатываемой системы:

— гость — не авторизованный пользователь, который может зарегистрироваться или авторизоваться, а также выполнять поиск по ключевым словам, авторам, жанрам;

— читатель — пользователь, который может выполнять поиск книг, а также добавлять их в избранное, бронировать, вставать в очередь на

них. Также читатель должен иметь возможность просматривать свои списки избранного, броней, выдачей и очередей;

— библиотекарь — пользователь, который может изменять статус брони книги, выдавать книги и принимать их обратно;

— модератор — пользователь, который может изменять редактировать информацию о книгах, авторах, издателях, ББК и АПУ. Также модератор может изменить роль другого пользователя в системе.

На рисунке 1.2 приведена Use-case диаграмма выделенных ролей.

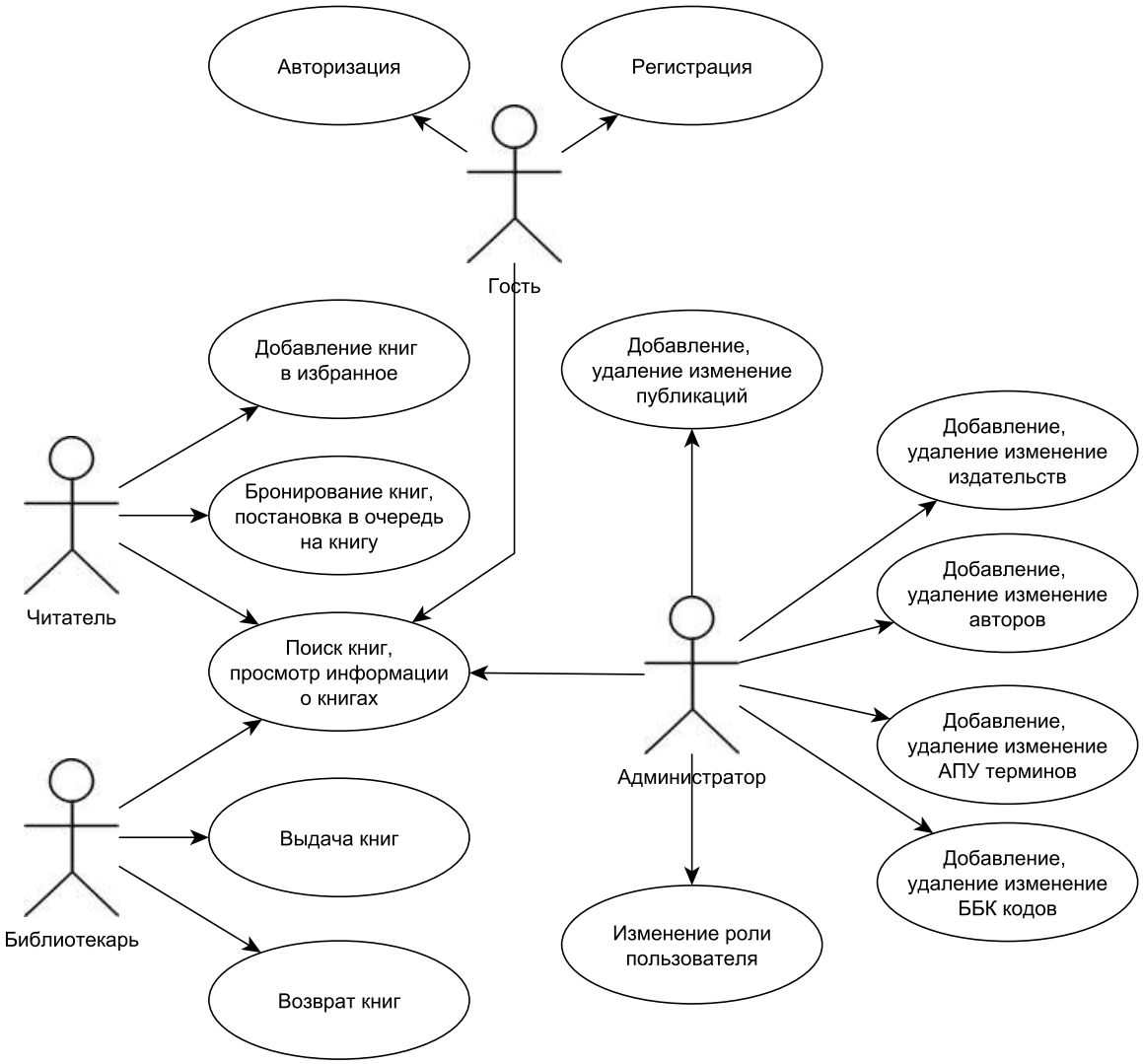

Рисунок 1.2 – Use-case диаграмма

1.6 Модели баз данных

База данных — это совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объектов и их взаимосвязей в рассматриваемой предметной области [4].

Модель представления данных — представление структуры данных и ограничений целостности, описанных на формальном языке [5].

Рассмотрим три основные модели баз данных:

— дореляционные;

— реляционные;

— постреляционные.

Дореляционные модели данных

К дореляционным моделям баз данных относят иерархическую и сетевую модели.

В иерархической модели база данных представляется в виде древовидной структуры, состоящей из объектов различных уровней. Между объектами устанавливаются связи типа «предок–потомок»: один объект может включать несколько объектов более низкого уровня, при этом у каждого потомка всегда есть только один предок. Объекты, имеющие одного и того же предка, называются братьями (или близнецами). Основными достоинством иерархической модели данных является простота понимания и использования, к недостаткам можно отнести сложности при моделировании связей «многие ко многим» и необходимость дублирования данных.

Для решения проблем иерархической модели была придумана сетевая модель. В ней запись может иметь несколько предков и потомков, благодаря чему уменьшается дублирование данных. Однако сетевые модели обладают сложной структурой, что утяжеляет их разработку и поддержку.

Общим недостатком дореляционных моделей является их зависимость от физической организации данных: изменение структуры часто влечет необходимость переработки прикладной программы.

Реляционные модели данных

Реляционная модель основана на представлении информации в виде таблиц, отношений. Каждая таблица содержит строки и столбцы: строки соответствуют отдельным записям, экземплярам объектов, а столбцы — их свойствам, атрибутам. Ключевая особенность реляционной модели заключается в том, что данные и связи между ними описываются только с помощью таблиц, а для манипулирования данными используется формальный язык, например SQL. Такая модель обеспечивает высокий уровень абстракции, гибкость при работе с информацией и независимость логического описания данных от способа их физического хранения.

Постреляционные модели данных

Постреляционная модель представляет собой развитие реляционной модели и направлена на снятие ее главного ограничения — требования атомарности данных в таблицах. В отличие от классического подхода, где каждая ячейка хранит только одно значение, в постреляционной модели допускаются многозначные поля: внутри записи может содержаться набор подзначений, который фактически образует вложенную таблицу.

Такой подход позволяет более естественно отражать сложные структуры данных и упрощает работу с ними. В результате одна постреляционная таблица может совмещать в себе функции нескольких связанных реляционных таблиц, что делает представление информации более наглядным и ускоряет ее обработку.

ВЫВОД

В данном разделе были рассмотрены рассмотрены существующие решения библиотечных сервисов и проведен их анализ по выделенным критериям. Также были формализованы задача, роли пользователей и данные. Были рассмотрены модели баз данных. Была выбрана реляционная модель, так как данные в библиотечной системе легко описываются в виде таблиц, а между таблицами устанавливаются четкие связи.

2 Конструкторский раздел

В данном разделе приводится диаграмма разрабатываемой базы данных. Каждой сущности, описанной в аналитическом разделе, ставится в соответствие таблица, а информация о них соотносится с полями этих таблиц. Рассматриваются разрабатываемые функции, триггеры, а также ролевая модель.

2.1 Проектирование базы данных

В соответствии с таблицей 1.2 и диаграммой приведенной на рисунке 1.1 в разрабатываемой базе данных выделены следующий таблицы:

— user — таблица пользователей;

— book — таблица книг;

— author — таблица авторов;

— book_author — таблица-связь книг и авторов;

— publisher — таблица издателей;

— bbk — таблица ББК;

— apu — таблица АПУ;

— reservation — таблица броней;

— issuance — таблица выдач;

— favorite — таблица избранного;

— queue — таблица очередей.

В таблицах 2.1 — 2.11 определены типы и ограничения для каждого столбца, перечисленных выше таблиц.

В таблице 2.4 пара book_id, author_id образуют первичный ключ. В таблице 2.9 пара user_id, book_id образуют первичный ключ.

Таблица 2.1 – Информация о таблице user

Столбец	**Значение**	**Тип**	**Ограничение**
id	Идентификатор	UUID	Не ноль, первичный ключ
surname	Фамилия	Строка	Не ноль
name	Имя	Строка	Не ноль
second_name	Отчество	Строка	
phone_number	Номер телефона	Строка	Не ноль, уникальный
email	Электронная почта	Строка	
password	Пароль	Строка	Не ноль
role	Роль в системе	Строка	Не ноль

Таблица 2.2 – Информация о таблице book

Столбец	**Значение**	**Тип**	**Ограничение**
id	Идентификатор	UUID	Не ноль, первичный ключ
title	Название	Строка	Не ноль
annotation	Аннотация	Строка	
publisher_id	Идентификатор издателя	UUID	Внешний ключ
publication_year	Год публикации	Целое число	Больше нуля, меньше текущего года
ISBN	ISBN	Строка	
bbk_id	Идентификатор ББК	UUID	Не ноль, внешний ключ
media_type	Формат выпуска	Строка	
volume	Объем	Строка	
language	Язык	Строка	
original_language	Язык оригинала	Строка	
copies	Количество копий	Целое число	Неотрицательный
avaliable_copies	Количество свободных копий	Целое число	Неотрицательно, меньше или равно copies

Таблица 2.3 – Информация о таблице author

Столбец	**Значение**	**Тип**	**Ограничение**
id	Идентификатор	UUID	Не ноль, первичный ключ
name	Имя	Строка	Не ноль, уникальный

Таблица 2.4 – Информация о таблице book_author

Столбец	**Значение**	**Тип**	**Ограничение**
book_id	Идентификатор книги	UUID	Не ноль, внешний ключ
author_id	Идентификатор автора	UUID	Не ноль, внешний ключ

Таблица 2.5 – Информация о таблице publisher

Столбец	**Значение**	**Тип**	**Ограничение**
id	Идентификатор	UUID	Не ноль, первичный ключ
name	Название	Строка	Не ноль, уникальный
description	Описание	Строка	
email	Электронная почта	Строка	
phone_number	Номер телефона	Строка	

Таблица 2.6 – Информация о таблице bbk

Столбец	**Значение**	**Тип**	**Ограничение**
id	Идентификатор	UUID	Не ноль, первичный ключ
code	ББК-код	Строка	Не ноль, уникальный
description	Расшифровка кода	Строка	Не ноль

Таблица 2.7 – Информация о таблице apu

Столбец	**Значение**	**Тип**	**Ограничение**
id	Идентификатор	UUID	Не ноль, первичный ключ
term	Ключевое слово	Строка	Не ноль, уникальный
bbk_id	Идентификатор ББК	UUID	Не ноль, внешний ключ

Таблица 2.8 – Информация о таблице reservation

Столбец	**Значение**	**Тип**	**Ограничение**
id	Идентификатор	UUID	Не ноль, первичный ключ
book_id	Идентификатор книги	UUID	Не ноль, внешний ключ
user_id	Идентификатор пользователя	UUID	Не ноль, внешний ключ
reservation_date	Дата бронирования	Дата	Не ноль
cancel_date	Дата отмены бронирования	Дата	Не ноль, больше reservation_date

Таблица 2.9 – Информация о таблице issuance

Столбец	**Значение**	**Тип**	**Ограничение**
id	Идентификатор	UUID	Не ноль, первичный ключ
book_id	Идентификатор книги	UUID	Не ноль, внешний ключ
user_id	Идентификатор пользователя	UUID	Не ноль, внешний ключ
issuance_date	Дата выдачи	Дата	Не ноль
return_date	Крайняя дата возврата	Дата	Не ноль, больше issuance_date

Таблица 2.10 – Информация о таблице favorite

Столбец	**Значение**	**Тип**	**Ограничение**
user_id	Идентификатор пользователя	UUID	Не ноль
book_id	Идентификатор книги	UUID	Не ноль

Таблица 2.11 – Информация о таблице queue

Столбец	**Значение**	**Тип**	**Ограничение**
id	Идентификатор	UUID	Не ноль, первичный ключ
book_id	Идентификатор книги	UUID	Не ноль, внешний ключ
user_id	Идентификатор пользователя	UUID	Не ноль, внешний ключ
created_time	Время постановки в очередь	Дата и время	Не ноль

На рисунке 2.1 изображена диаграмма разрабатываемой базы данных, построенная в соответствии с рассмотренными таблицами.

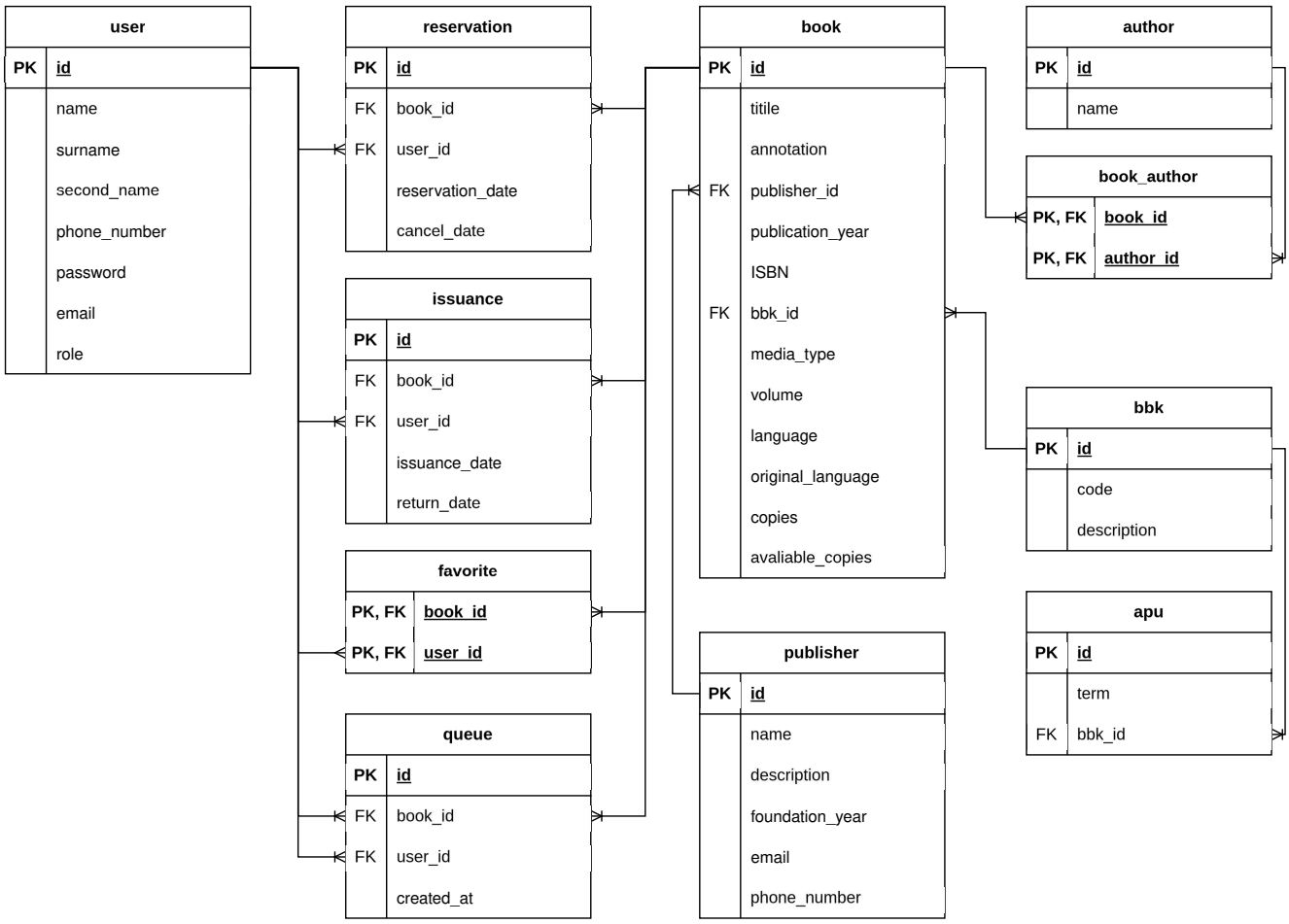

Рисунок 2.1 – Диаграмма разрабатываемой базы данных

2.2 Описание проектируемых функций

Для корректной работы с базой данных были разработаны функции. Функция process_book_return обрабатывает возврат книги после брони или выдачи. Принимает на вход идентификатор обрабатываемой книги. Функция обновляет поле avaliable_copies в таблице book и создает новую бронь в таблице reservation, если есть пользователь в очереди на эту книгу. Схема алгоритма функции представлена на рисунке 2.2.

Функция process_book_loan обрабатывает создание брони или выдачи на книгу. Функция генерирует исключение, если avaliable_copies меньше или равно нулю. Схема алгоритма функции представлена на рисунке 2.3.

Функция get_queue_number возвращает позицию в очереди пользователя на книгу. Схема алгоритма функции представлена на рисунке 2.4.

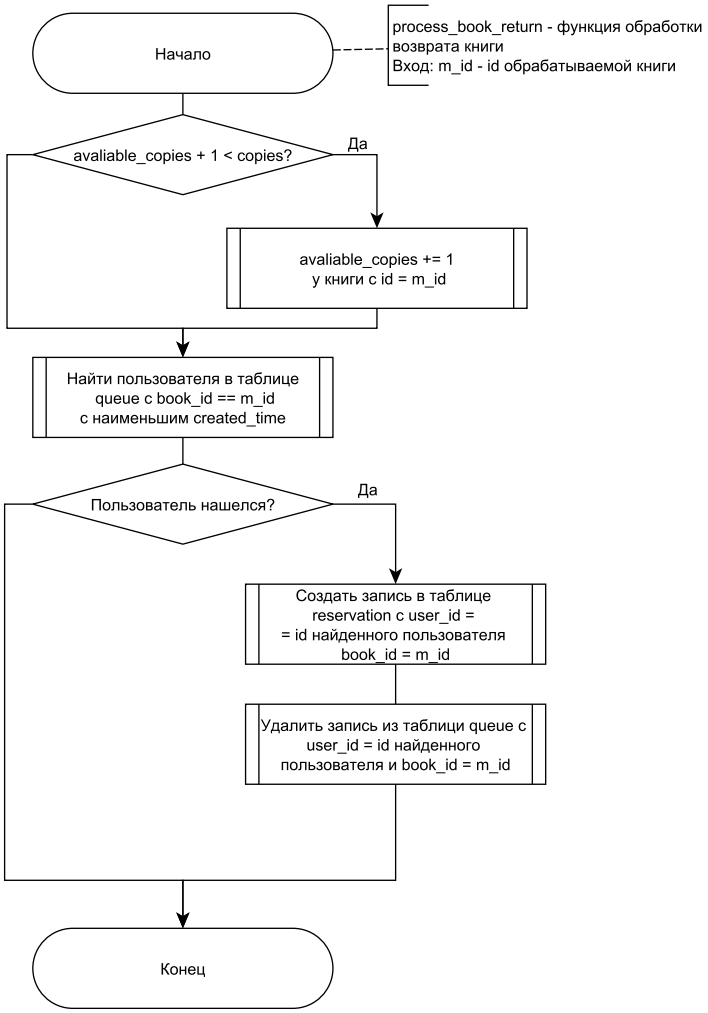

Рисунок 2.2 – Схема алгоритма функции process_book_return

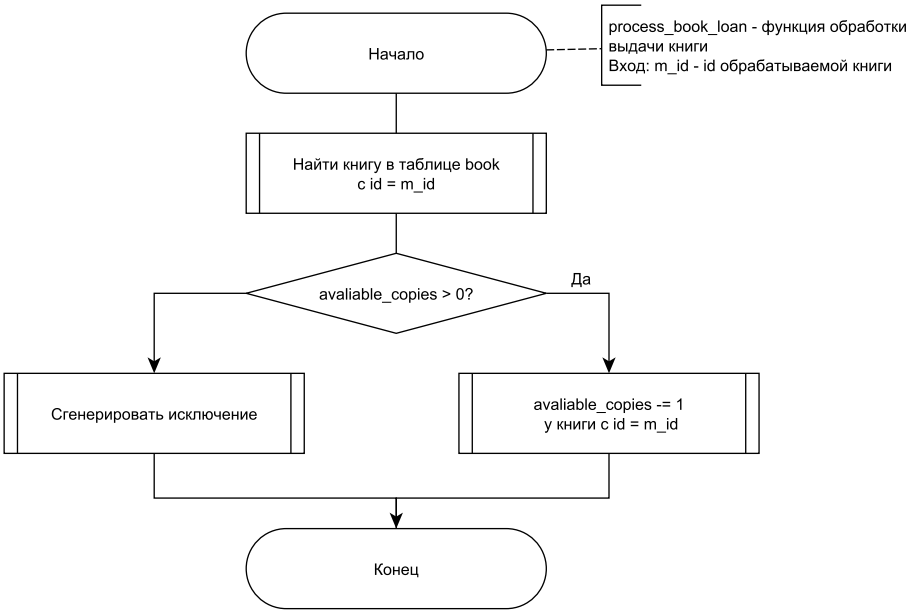

Рисунок 2.3 – Схема алгоритма функции process_book_loan

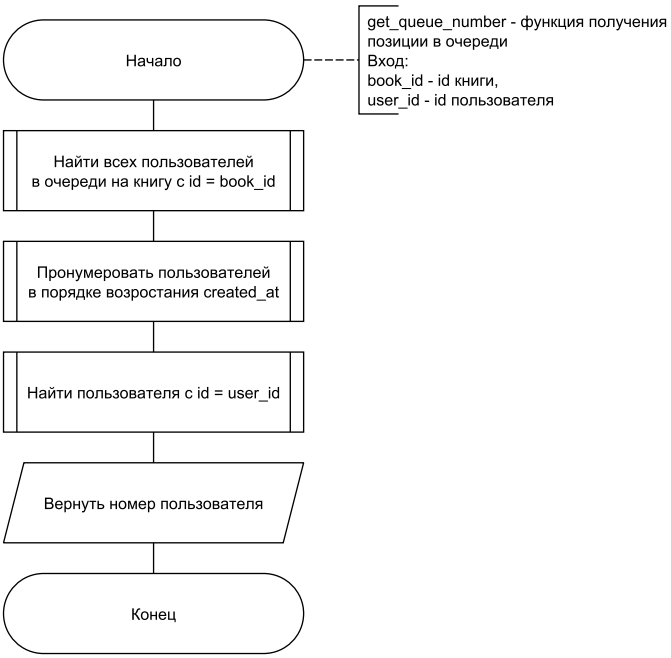

Рисунок 2.4 – Схема алгоритма функции get_queue_number

2.3 Описание проектируемых триггеров

Так как поле avaliable_copies в таблице book зависит как от количества броней, так и от количества выдач книги, были разработаны по два триггера на каждую из таблиц reservation и issuance. При добавлении записи в одну из них вызывается триггер, который в свою очередь вызывает функцию process_book_loan. При удалении записи из этих таблиц вызывается триггер, который вызывает функцию process_book_return.

2.4 Описание ролевой модели

В аналитическом разделе были описаны четыре категории пользователей, которые могут взаимодействовать с приложением: гость, читатель, библиотекарь, модератор. Для каждой из этих категорий требуется создать роль в базе данных.

1. Гость может просматривать таблицы book, author, bbk, apu, book_author, некоторые столбцы таблицы publisher. Так как гость имеет возможность зарегистрироваться или авторизироваться, то у него должна быть возможность просматривать и добавлять записи в таблицу user;

2. Читатель может просматривать таблицы book, author, bbk, apu,

book_author, favorite, issuance, user, reservation, некоторые столбцы таблицы publisher. Так как читатель должен иметь возможность получать свой номер в очереди на книгу, то у него должен быть полный доступ на чтение к таблице queue. Также читатель может добавлять и удалять записи в таблицах favorite, reservation, queue;

3. Библиотекарь может просматривать таблицы book, author, bbk, apu, book_author, publisher, просматривать и удалять записи из таблицы reservation, просматривать, добавлять и удалять записи в таблице issuance;

4. Модератор имеет полный доступ ко всем таблицам.

ВЫВОД

В данном разделе была спроектирована база данных. Описаны все таблицы и их поля, предъявлены ограничения к ним, описаны две разрабатываемые функции и триггеры. Также была разобрана ролевая модель базы данных, описаны ограничения на каждую из четырех ролей.

3 Технологический раздел

В данном разделе рассматриваются различные СУБД реляционных баз данных, проводится их сравнение по выделенным критериям, представлены средства реализации программного обеспечения, демонстрируется процесс создания таблиц, функций, триггеров и ролей базы данных. Демонстрируется работа программного обеспечения.

3.1 Выбор СУБД

Наиболее распространенными СУБД реляционных баз данных являются:

— PostgreSQL [6];

— MySQL [7];

— Microsoft SQL Server [8];

— Oracle DB [9].

Выделим критерии для сравнения приведенных СУБД:

— КР1 — бесплатное распространение;

— КР2 — поддержка ролевой модели;

— КР3 — производительность [10; 11];

В таблице 3.1 приводится сравнение рассматриваемых СУБД, по выделенным критериям.

Таблица 3.1 – Сравнение СУБД

Решение	**КР1**	**КР2**	**КР3**
PostgreSQL	+	+	1
MySQL	+	+	4
Microsoft SQL Server	-	+	3
Oracle DB	-	+	2

По результатам сравнения в качестве используемой СУБД был выбран PostgreSQL, так как он удовлетворяет всем поставленным критериям и является наиболее быстрым из всех.

3.2 Выбор средств реализации ПО

В качестве языка программирования был выбран Kotlin [12], поскольку разработка ориентирована на мобильные устройства, большинство которых функционирует под управлением Android [13]. Для работы с базой данных был выбран фреймворк Exposed [14], который позволяет использовать DSL для описания запросов, а также поддерживает объектно-ориентированный подход к доступу данным. Для реализации серверной части был выбран фреймфорк Ktor [15]. Для написания клиентского приложения и интерфейса пользователя был использован фреймворк Jetpack Compose [16].

3.3 Архитектура приложения

Для реализации приложения был выбран подход MVVM (Model-View-ViewModel) [17]. В нём приложение разделяется на три слоя:

— Model отвечает за бизнес-логику и доступ к данным;

— View представляет собой пользовательский интерфейс и отвечает только за отображение данных;

— ViewModel выступает в качестве посредника между Model и View, обрабатывая пользовательские события, управляя состоянием интерфейса и предоставляя данные в удобной для отображения форме.

3.4 Создание таблиц базы данных

В листингах 3.1 — 3.11 представлено создание таблиц и описание соответствующих ограничений к ниимммм.

Листинг 3.1 – Создание таблицы book

```
1 CREATE TABLE IF NOT EXISTS book (
2     id UUID PRIMARY KEY,
3     title VARCHAR(255) NOT NULL,
4     annotation TEXT,
5     publisher_id UUID REFERENCES publisher(id) ON DELETE CASCADE,
6     publication_year INT,
7     ISBN VARCHAR(50),
8     bbk_id UUID REFERENCES bbk(id) ON DELETE RESTRICT NOT NULL,
```

```
    media_type VARCHAR(100),
    volume TEXT,
    language VARCHAR(100),
    original_language VARCHAR(100),
    copies INT DEFAULT 0 NOT NULL,
    available_copies INT DEFAULT 0 NOT NULL CHECK
        (available_copies <= copies)
);
```

Листинг 3.2 – Создание таблицы author

```
CREATE TABLE IF NOT EXISTS author (
    id UUID PRIMARY KEY,
    name VARCHAR(255) NOT NULL
);
```

Листинг 3.3 – Создание таблицы связки book_author

```
CREATE TABLE IF NOT EXISTS book_author (
    book_id UUID REFERENCES book(id) ON DELETE CASCADE,
    author_id UUID REFERENCES author(id) ON DELETE CASCADE,
    PRIMARY KEY (book_id, author_id)
);
```

Листинг 3.4 – Создание таблицы publisher

```
CREATE TABLE IF NOT EXISTS publisher (
    id UUID PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    description VARCHAR(255),
    email VARCHAR(50),
    phone_number VARCHAR(20)
);
```

Листинг 3.5 – Создание таблицы bbk

```
CREATE TABLE IF NOT EXISTS bbk (
    id UUID PRIMARY KEY,
    code VARCHAR(16) NOT NULL,
    description VARCHAR(255) NOT NULL
);
```

Листинг 3.6 – Создание таблицы apu

```
CREATE TABLE IF NOT EXISTS apu (
    id UUID PRIMARY KEY,
    term VARCHAR(100) NOT NULL,
    bbk_id UUID REFERENCES bbk(id) ON DELETE CASCADE NOT NULL
);
```

Листинг 3.7 – Создание таблицы user

```
CREATE TABLE IF NOT EXISTS "user" (
    id UUID PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    surname VARCHAR(50) NOT NULL,
    second_name VARCHAR(50),
    email VARCHAR(50),
    password VARCHAR(100) NOT NULL,
    phone_number VARCHAR(20) NOT NULL,
    role VARCHAR(20) DEFAULT 'READER' NOT NULL
);
```

Листинг 3.8 – Создание таблицы issuance

```
CREATE TABLE IF NOT EXISTS issuance (
    id UUID PRIMARY KEY,
    book_id UUID REFERENCES book(id) ON DELETE CASCADE NOT NULL,
    user_id UUID REFERENCES "user"(id) ON DELETE CASCADE NOT NULL,
    issuance_date DATE NOT NULL,
    return_date DATE NOT NULL CHECK (return_date > issuance_date)
);
```

Листинг 3.9 – Создание таблицы reservation

```
CREATE TABLE IF NOT EXISTS reservation (
    id UUID PRIMARY KEY,
    book_id UUID REFERENCES book(id) ON DELETE CASCADE NOT NULL,
    user_id UUID REFERENCES "user"(id) ON DELETE CASCADE NOT NULL,
    reservation_date DATE NOT NULL,
    cancel_date DATE NOT NULL CHECK (cancel_date >
       reservation_date)
);
```

Листинг 3.10 – Создание таблицы favorite

```
CREATE TABLE IF NOT EXISTS user_book (
    user_id UUID REFERENCES "user"(id) ON DELETE CASCADE,
    book_id UUID REFERENCES book(id) ON DELETE CASCADE,
    PRIMARY KEY (user_id, book_id)
);
```

Листинг 3.11 – Создание таблицы queue

```
CREATE TABLE IF NOT EXISTS queue (
    id UUID PRIMARY KEY,
    book_id UUID REFERENCES book(id) ON DELETE CASCADE NOT NULL,
    user_id UUID REFERENCES "user"(id) ON DELETE CASCADE NOT NULL,
    created_at TIMESTAMP NOT NULL DEFAULT NOW()
);
```

3.5 Создание функций базы данных

В листингах 3.12 — 3.14 представлено создание функций базы данных рассмотренных в конструкторском разделе.

Функции process_book_return и process_book_loan предполагается вызывать из триггеров, поэтому они объявлены как триггерные функции [18].

Листинг 3.12 – Создание функции process_book_return

```
CREATE OR REPLACE FUNCTION process_book_return()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
DECLARE
    next_user uuid;
    book_uuid uuid;
    total_copies int;
    free_copies int;
BEGIN
    book_uuid := OLD.book_id;

    SELECT copies, available_copies
    INTO total_copies, free_copies
    FROM book
    WHERE id = book_uuid;
```

```
17
18      IF free_copies < total_copies THEN
19          UPDATE book
20          SET available_copies = available_copies + 1
21          WHERE id = book_uuid;
22      END IF;
23
24      SELECT user_id INTO next_user
25      FROM queue
26      WHERE book_id = book_uuid
27      ORDER BY created_at
28      LIMIT 1;
29
30      IF next_user IS NOT NULL THEN
31          INSERT INTO reservation(id, book_id, user_id,
                reservation_date, cancel_date)
32          VALUES (gen_random_uuid(), book_uuid, next_user, NOW(),
                NOW() + INTERVAL '3 days');
33
34          DELETE FROM queue
35          WHERE user_id = next_user AND book_id = book_uuid;
36      END IF;
37
38      RETURN OLD;
39  END;
40  $$;
```

Листинг 3.13 – Создание функции process_book_loan

```
1  CREATE OR REPLACE FUNCTION process_book_loan()
2  RETURNS TRIGGER
3  LANGUAGE plpgsql
4  AS $$
5  DECLARE
6      available integer;
7  BEGIN
8      SELECT available_copies
9      INTO available
10     FROM book
11     WHERE id = NEW.book_id
12     FOR UPDATE;
13
```

```
14      IF available <= 0 THEN
15          RAISE EXCEPTION 'Нет доступных экземпляров книги с id=%',
                NEW.book_id;
16      END IF;
17
18      UPDATE book
19      SET available_copies = available_copies - 1
20      WHERE id = NEW.book_id;
21
22      RETURN NEW;
23  END;
24  $$;
```

Листинг 3.14 – Создание функции get_queue_number

```
1  CREATE OR REPLACE FUNCTION get_queue_number(p_book_id UUID,
       p_user_id UUID)
2  RETURNS INT
3  LANGUAGE plpgsql
4  AS $$
5  BEGIN
6      RETURN (
7          SELECT position
8          FROM (
9              SELECT user_id,
10                 ROW_NUMBER() OVER (PARTITION BY book_id ORDER BY
                       created_at) AS position
11             FROM queue
12             WHERE book_id = p_book_id
13         ) q
14         WHERE q.user_id = p_user_id
15     );
16 END;
17 $$;
```

3.6 Создание триггеров базы данных

В листингах 3.15 — 3.16 представлено создание триггеров базы данных.

Листинг 3.15 – Создание тригеров для таблицы reservation

```
1  CREATE TRIGGER trg_return_book_from_reservation
```

```
2 AFTER DELETE ON reservation
3 FOR EACH ROW
4 EXECUTE FUNCTION process_book_return();
5
6 CREATE TRIGGER trg_loan_book_to_reservation
7 BEFORE INSERT ON reservation
8 FOR EACH ROW
9 EXECUTE FUNCTION process_book_loan();
```

Листинг 3.16 – Создание тригеров для таблицы issuance

```
1 CREATE TRIGGER trg_return_book_from_reservation
2 AFTER DELETE ON issuance
3 FOR EACH ROW
4 EXECUTE FUNCTION process_book_return();
5
6 CREATE TRIGGER trg_loan_book_to_reservation
7 BEFORE INSERT ON issuance
8 FOR EACH ROW
9 EXECUTE FUNCTION process_book_loan();
```

3.7 Создание ролей базы данных

В листингах 3.17 — 3.20 представлено создание ролей базы данных и выдача соответствующих прав каждой из них.

Листинг 3.17 – Создание роли гостя

```
1  CREATE ROLE guest
2      NOSUPERUSER
3      NOCREATEDB
4      NOCREATEROLE
5      NOINHERIT
6      NOREPLICATION
7      NOBYPASSRLS
8      LOGIN
9      PASSWORD 'guest';
10
11 GRANT SELECT ON book, author, bbk, apu, book_author TO guest;
12 GRANT SELECT (id, name) ON publisher TO guest;
13 GRANT SELECT, INSERT ON "user" TO guest;
```

Листинг 3.18 – Создание роли читателя

```
CREATE ROLE reader
    NOSUPERUSER
    NOCREATEDB
    NOCREATEROLE
    NOINHERIT
    NOREPLICATION
    NOBYPASSRLS
    LOGIN
    PASSWORD 'reader';

    GRANT SELECT ON book, author, bbk, apu, book_author TO reader;
    GRANT SELECT (id, name) ON publisher TO reader;
    GRANT SELECT, UPDATE ON "user" TO reader;
    GRANT SELECT ON favorite, queue, issuance, reservation TO
        reader;
    GRANT INSERT, DELETE ON favorite, reservation, queue TO
        reader;
```

Листинг 3.19 – Создание роли библиотекаря

```
CREATE ROLE librarian
    NOSUPERUSER
    NOCREATEDB
    NOCREATEROLE
    NOINHERIT
    NOREPLICATION
    NOBYPASSRLS
    LOGIN
    PASSWORD 'librarian';

GRANT SELECT ON book, author, bbk, apu, book_author, publisher TO
    librarian;
GRANT SELECT, DELETE ON reservation TO librarian;
GRANT SELECT, INSERT, DELETE ON issuance TO librarian;
```

Листинг 3.20 – Создание роли модератора

```
CREATE ROLE moderator
    SUPERUSER
    CREATEDB
    CREATEROLE
```

```
5       INHERIT
6       REPLICATION
7       BYPASSRLS
8       LOGIN
9       PASSWORD 'moderator';
```

3.8 Демонстрация работы программы

На рисунках 3.1 — 3.5 изображены экраны приложения, на которых демонстрируется функционал читателя, библиотекаря, модератора.

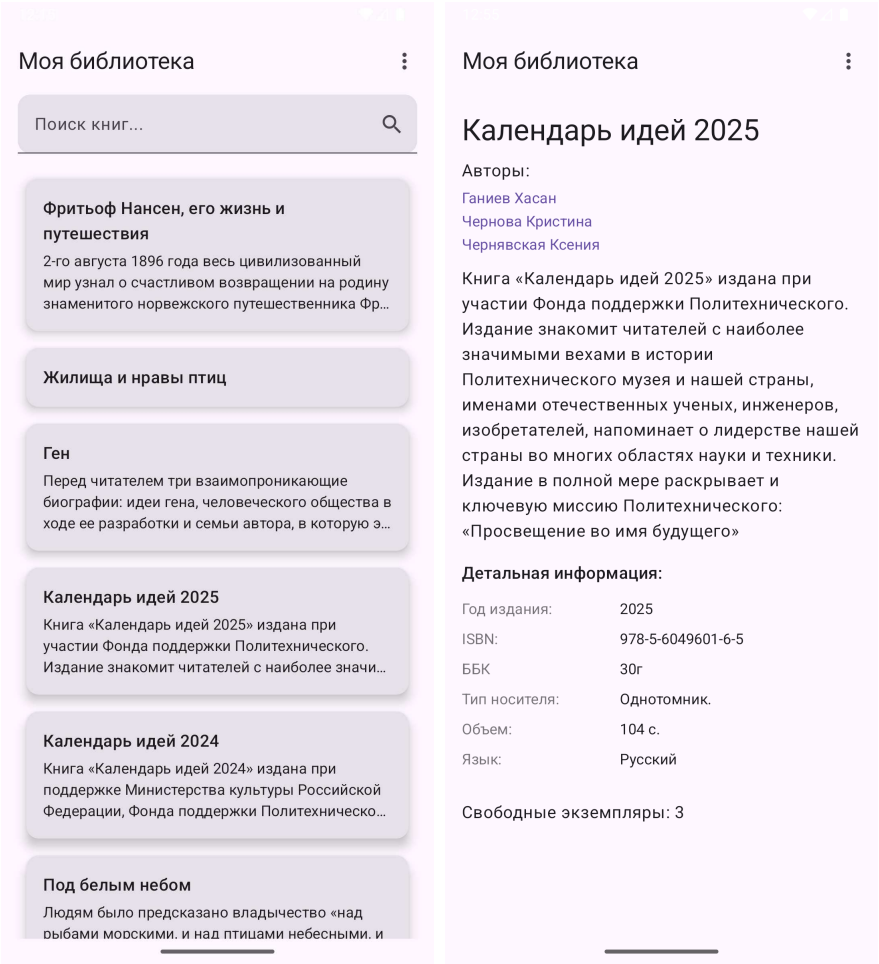

Рисунок 3.1 – Домашний экран приложения

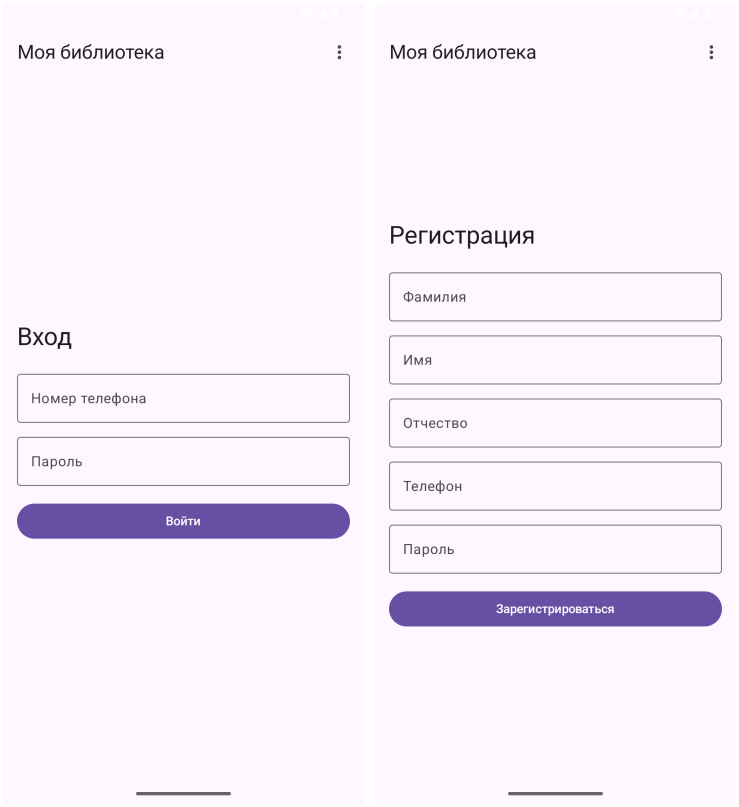

Рисунок 3.2 – Экраны авторизации и регистрации

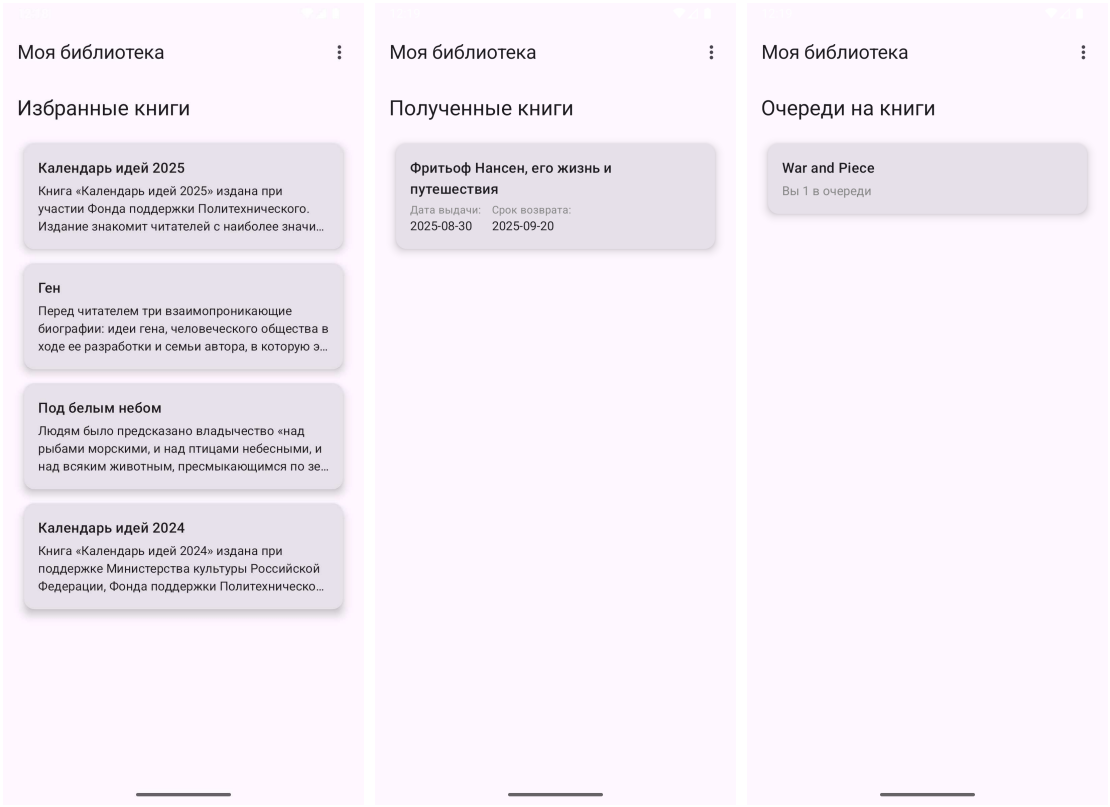

Рисунок 3.3 – Экраны читателя

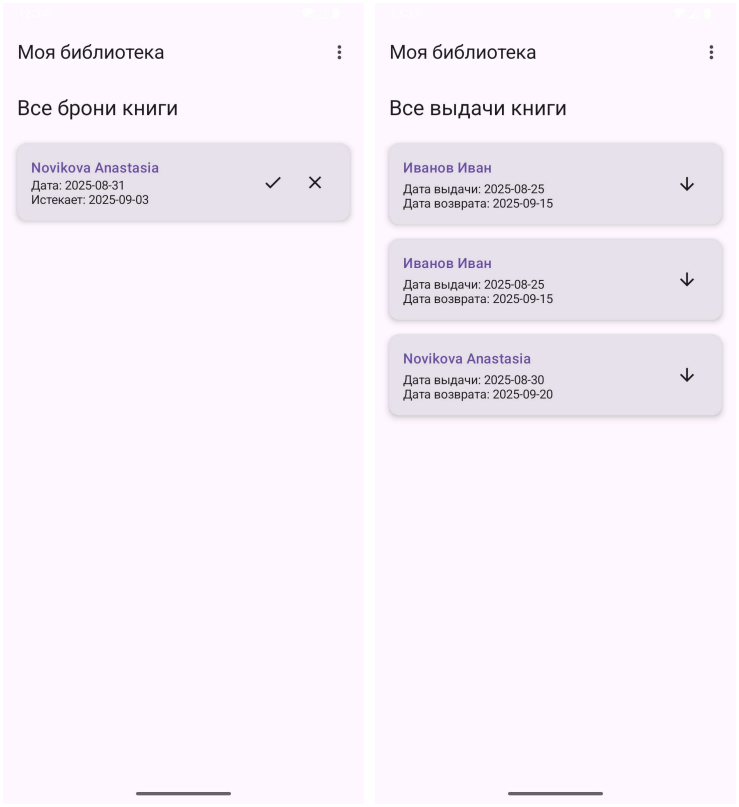

Рисунок 3.4 – Экраны библиотекаря

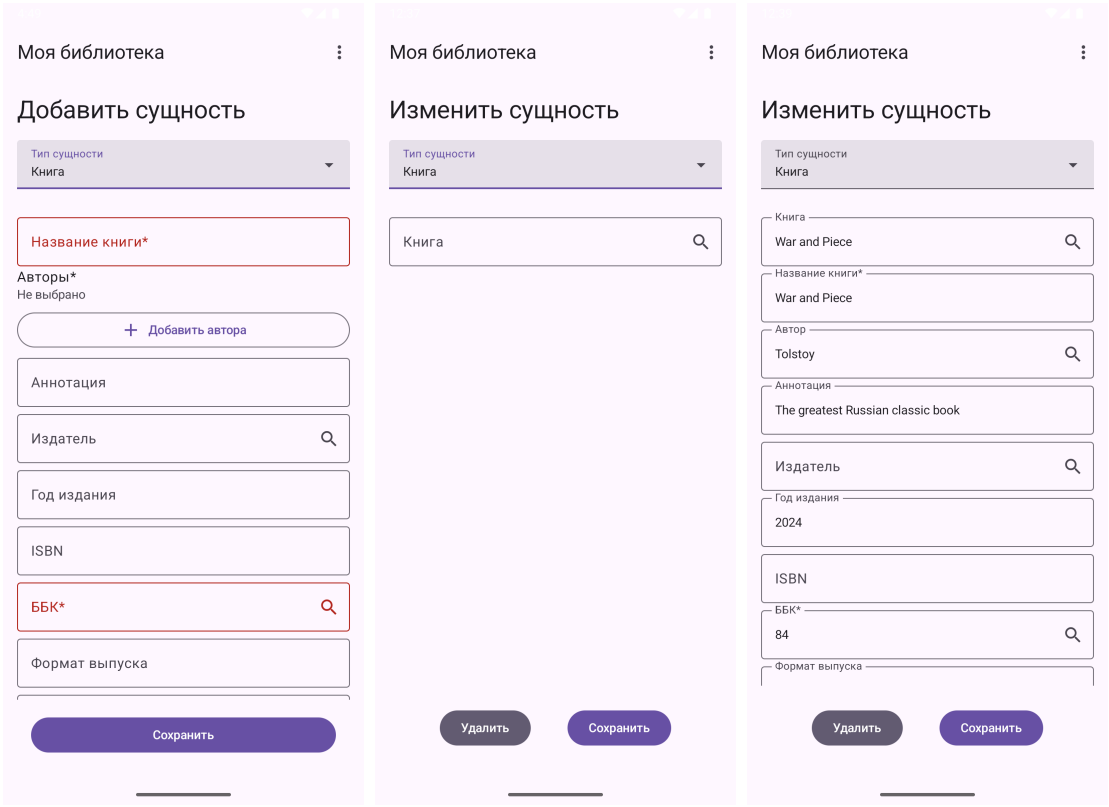

Рисунок 3.5 – Экраны модератора

ВЫВОД

В данном разделе было проведено сравнение различных СУБД по выделенным критериям, описаны средства реализации ПО, представлено создание таблиц, функций, триггеров и ролей базы данных, а также продемонстрирована работа приложения.

4 Исследовательский раздел

В данном разделе приведены характеристики используемого устройства, описано проводимое исследование, демонстрируются результаты данного исследования и их анализ.

4.1 Технические характеристики

Технические характеристики используемого устройства:

— операционная система — Ubuntu Linux x86_64 [19];

— память — 16 Гб;

— процессор — AMD Ryzen 5 5500U (6x2.10 ГГц) [20].

4.2 Описание исследования

Целью исследования является проверить зависимость времени выполнения запроса от наличия индексации на различном количестве записей в таблице.

Для проведения исследования была выбрана таблица book_author, которая служит таблицей-связкой для авторов и книг. Так как необходимо иметь возможность быстро получать все книги определенного автора, то для индексации был выбран столбец author_id. Замеры проводились на количестве записей от 10 до 2500.

В листинге 4.1 демонстрируется рассматриваемый запрос, на котором будут проводиться замеры. Значение author_id берется из таблицы В листинге 4.2 демонстрируется создание индекса в таблице book_author.

Листинг 4.1 – Рассматриваемый запрос

```
1 SELECT b.*
2 FROM book_author ba
3 JOIN book b ON ba.book_id = b.id
4 WHERE ba.author_id = author_id;
```

Листинг 4.2 – Создание роли модератора

```
1 CREATE INDEX idx_book_author_author_id ON book_author(author_id);
```

4.3 Результаты исследования

В таблице 4.1 приведены результаты исследования. Для каждого размера было проведено 1000 замеров, значение в таблице взято как среднее арифметическое от них.

Таблица 4.1 – Результаты исследования

Количество записей	**Без индекса, мс**	**С индексом, мс**
10	0.017	0.01145
30	0.018	0.01150
60	0.021	0.01151
100	0.022	0.01152
300	0.040	0.01153
600	0.057	0.01142
1000	0.066	0.01149
1500	0.092	0.01140
2000	0.105	0.01151
2500	0.129	0.01154

Графическая интерпретация результатов представлена на рисунке 4.1.

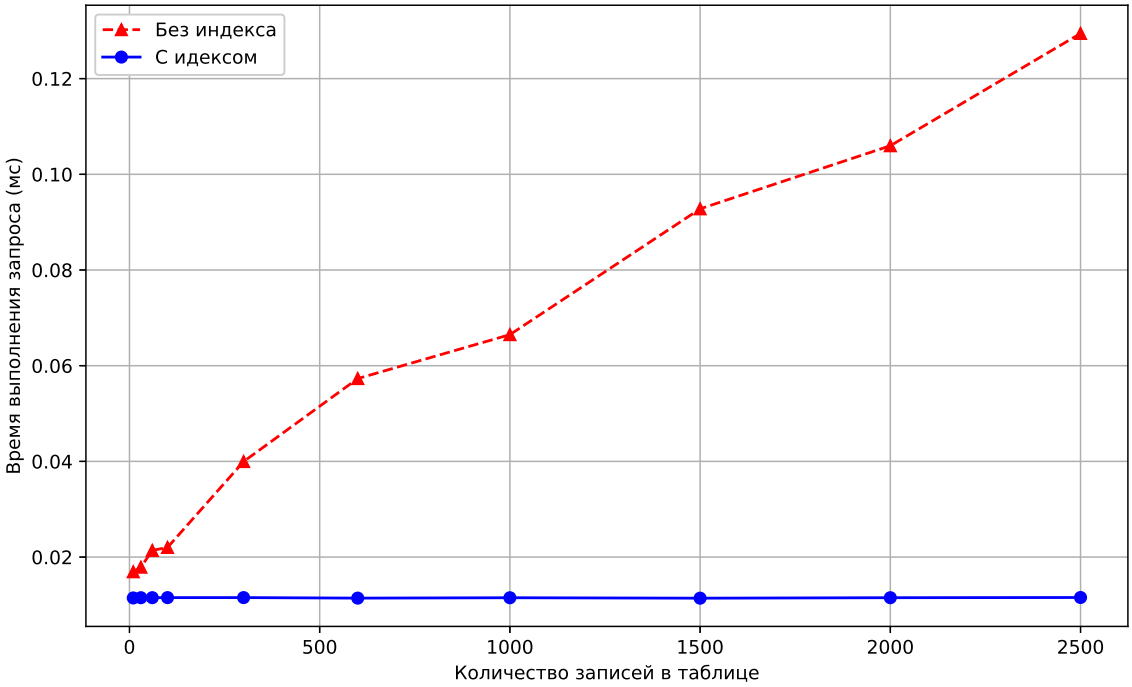

Рисунок 4.1 – Зависимость времени выполнения запроса от количества записей в таблице

Из полученных результатов видно, что наличие индексации в базе данных дает выигрыш по времени даже при 10 записях, тогда как при 100 время выполнения запроса с индексом сокращается приблизительно в 2 раза. При 2500 записей время выполнения запроса сокращается более чем в 10 раз. Также из полученных данных следует, что время выполнения запроса без индекса с увеличением количества записей возрастает, тогда как время выполнения запроса с индексом остается приблизительно одинаковым.

ВЫВОД

В данном разделе были рассмотрены технические характеристики устройства, было описано исследование зависимости скорости выполнения запроса от наличия индексации на разном количестве записей в базе данных.

Результаты показали, что в отсутствие индекса время выполнения запроса увеличивается по мере роста числа записей, тогда как при наличии индекса оно остается практически неизменным. Причем даже при малом объеме данных запрос без индекса выполняется дольше: при 100 записях — приблизительно в два раза медленнее, при 2500 — более чем в десять раз.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы была разработана база данных для хранения и обработки данных библиотечной картотеки, а также приложение для доступа к ней.

В процессе работы были решены следующие задачи:

— проведен анализ существующих библиотечных сервисов;

— определены требования к базе данных и программному обеспечению;

— спроектированы сущности базы данных и заданы их ограничения;

— выбраны инструменты для реализации базы данных и программного обеспечения;

— реализованы база данных и программное обеспечение для работы с ней;

— проведено исследование на основе разработанной базы данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Российская государственная библиотека для молодежи [Электронный ресурс]. — URL: `https://rgub.ru/` (дата обращения 05.03.2025).
2. Научно-техническая библиотека МГТУ им. Н.Э. Баумана [Электронный ресурс]. — URL: `https://library.bmstu.ru/` (дата обращения 05.03.2025).
3. Библиотеки Москвы [Электронный ресурс]. — URL: `https://www.mos.ru/knigi/` (дата обращения 05.03.2025).
4. *Хомоненко А.*, *Цыганков В.*, *Мальцев М.* Базы данных. — 2009.
5. *Стружкин Н. П.*, *Годин В. В.* Базы данных: проектирование. — 2017.
6. PostgreSQL [Электронный ресурс]. — URL: `https://www.postgresql.org/docs/` (дата обращения 19.03.2025).
7. MySQL [Электронный ресурс]. — URL: `https://dev.mysql.com/doc/` (дата обращения 19.03.2025).
8. Microsoft SQL Server [Электронный ресурс]. — URL: `https://learn.microsoft.com/ru-ru/sql/?view=sql-server-ver17` (дата обращения 19.03.2025).
9. Oracle DB [Электронный ресурс]. — URL: `https://docs.oracle.com/en/database/oracle/oracle-database/` (дата обращения 19.03.2025).
10. *Ismail Hossain M.*, *Mahmud S.*, *Santa T.* Oracle, MySQL, PostgreSQL, SQLite, SQL Server: Performance based competitive analysis // Daffodil International University. — 2019.
11. *Solarz A.*, *Szymczyk T.* Oracle 19c, SQL Server 2019, Postgresql 12 and MySQL 8 database systems comparison // Journal of Computer Sciences Institute. — 2020. — Т. 17. — С. 373—378.
12. Kotlin [Электронный ресурс]. — URL: `https://kotlinlang.org/docs/home.html` (дата обращения 02.04.2025).
13. StatCounter Global Stats [Электронный ресурс]. — URL: `https://gs.statcounter.com/os-market-share/mobile/worldwide` (дата обращения 03.04.2025).

14. Exposed [Электронный ресурс]. — URL: https://www.jetbrains.com/help/exposed/home.html (дата обращения 02.04.2025).

15. Ktor [Электронный ресурс]. — URL: https://ktor.io/docs/welcome.html (дата обращения 02.04.2025).

16. Compose [Электронный ресурс]. — URL: https://developer.android.com/develop/ui/compose/documentation?hl=ru (дата обращения 02.04.2025).

17. *Anderson C.* The model-view-viewmodel (mvvm) design pattern // Pro Business Applications with Silverlight 5. — Springer, 2012. — С. 461—499.

18. *Фишер А. В.*, *Дьяченко Р. А.*, *Лоба И. С.* Организация хранения хронологических данных в базах данных систем мониторинга и прогнозирования // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. — 2012. — № 79. — С. 271—280.

19. Ubuntu technical documentation for developers and IT pros [Электронный ресурс]. — URL: https://ubuntu.com/tutorials (дата обращения 10.06.2025).

20. AMD Ryzen 5 5500U Processor [Электронный ресурс]. — URL: https://www.amd.com/en/products/apu/amd-ryzen-5-5500u (дата обращения 10.06.2025).

ПРИЛОЖЕНИЕ А

Презентация к курсовой работе состоит из 15 слайдов.