



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

ФАКУЛЬТЕТ _____ Информатика и системы управления _____

КАФЕДРА _____ Программное обеспечение ЭВМ и информационные технологии _____

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

Библиотечная система для бронирования книг

Студент ИУ7-23М
(Группа)

(Подпись, дата) Н. И. Иксарица
(И.О.Фамилия)

Руководитель курсовой работы

(Подпись, дата) _____
(И.О.Фамилия)

Консультант

(Подпись, дата) _____
(И.О.Фамилия)

2021 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1. Аналитический раздел	6
1.1. Существующие аналоги	6
1.2. Описание системы	6
1.3. Назначение разработки.....	7
1.4. Общие требования к системе	7
1.5. Требования к функциональным характеристикам	8
1.6. Функциональные требования с точки зрения пользователя	8
1.7. Описание входных данных	9
1.8. Описание выходных данных	10
1.9. Требования к программной реализации.....	11
1.10. Требования по подготовке к вводу системы в эксплуатацию	12
1.11. Топология системы	13
1.12. Общие требования к подсистемам	14
1.13. Функциональные требования к сервисам	15
1.14. Требования к надежности	19
2. Конструкторский раздел	21
2.1. Концептуальный дизайн	21
2.2. Сценарии функционирования системы.....	22
2.2.1. Регистрация нового пользователя с ролью «Администратор».....	22
2.2.2. Регистрация нового пользователя с ролью «Читатель»	23
2.2.3. Авторизация зарегистрированного пользователя.....	24
2.2.4. Добавление новой библиотеки	24
2.2.5. Удаление существующей библиотеки	25
2.2.6. Добавление новой книги	26
2.2.7. Удаление существующей книги	27
2.2.8. Добавление нового бронирования.....	28
2.2.9. Просмотр сведений о читателях	29

2.2.10. Возврат книги.....	29
2.2.11. Получение сведений о библиотеке.....	30
2.2.12. Получение сведений о книге.....	30
2.3. Диаграммы прецедентов.....	31
2.3.1. Спецификации сценариев для неавторизованного пользователя	32
2.3.2. Спецификации сценариев для пользователя с ролью «Администратор».....	34
2.3.3. Спецификации сценариев для пользователя с ролью «Читатель».....	43
2.4. Логический дизайн	46
2.5. Спецификации таблиц.....	47
2.6. Структура сервисов.....	51
2.7. Диаграммы последовательности действий	51
2.8. Диаграмма потоков данных.....	54
2.9. Архитектура системы.....	55
3. Технологический раздел.....	57
3.1. Выбор и обоснование технологий разработки	57
3.2. Обеспечение отказоустойчивости	58
ЗАКЛЮЧЕНИЕ	60
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	61

ВВЕДЕНИЕ

За последние тридцать лет развитие библиотек во многом определялся прогрессом в области вычислительных систем и информационных технологий. Электронно-вычислительные машины (ЭВМ) и технологии автоматизированной обработки информации сразу нашли применение в библиотеках, которые имели значительные ресурсы и множество рутинных операций обработки данных, подлежащих частичной или полной формализации. Возникли первые автоматизированные библиотечные информационные системы (АБИС). АБИС первого поколения базировались на больших и средних ЭВМ и не получили широкого распространения. В начале 80-х годов в библиотеках начали создавать АБИС на базе персональных компьютеров. А с развитием Интернета новые системы стали предлагать пользователям онлайн-каталоги и графический интерфейс, доступный из браузера.

Системы охватывают или часть операций, или весь традиционный библиотечный цикл, в который входит: каталогизация, книговыдача, общедоступный электронный каталог и интерфейс для пользователей.

Целью данной работы является создание программного продукта, предоставляющего возможности просмотра и ведения реестра библиотек с их книжными фондами, а также создания читателями бронирований книг для очного получения и последующего ведения заказов администраторами библиотек.

Для достижения поставленной цели необходимо выполнить следующее:

- спроектировать библиотечную систему с использованием сервис-ориентированного подхода;
- определить модели данных, подходящих к поставленной задаче;

- выбрать и обосновать технологии реализации;
- разработать веб-приложение, реализующее заявленную функциональность;
- выполнить развёртывание системы.

1. Аналитический раздел

1.1. Существующие аналоги

Среди аналогов можно отметить АБИС «ИРБИС64», «МегаПро», «Руслан», «МАРК-SQL» [1]. Данный проект должен иметь следующие преимущества перед существующими аналогами:

- реализация функций (поиск и бронирование книг), интересующих в первую очередь посетителей библиотек, а не работников;
- кроссплатформенность ядра системы;
- пользовательский интерфейс, реализованный в виде web-приложения, доступный не только в локальной сети библиотеки, но и в сети Интернет;
- возможность формирования рейтинга читателя.

1.2. Описание системы

Система должна позволять пользователю искать интересующие его книги в библиотеке и бронирования этих книг для того, чтобы забрать их позднее в удобное время. Если пользователя уже взял определённое количество книг, то он не может взять новую до тех пор, пока не сдал старые. Если пользователь возвращает книги в хорошем состоянии и сдает их в срок, то максимальное количество книг у него на руках увеличивается.

В систему включается сервис статистики, собирающий данные о выполнении операций. По собранным данным необходимо строить отчет, доступ к которому предоставляется только пользователю с ролью «Администратор».

Общая схема рассматриваемой предметной области приведена ниже на рисунке 1.

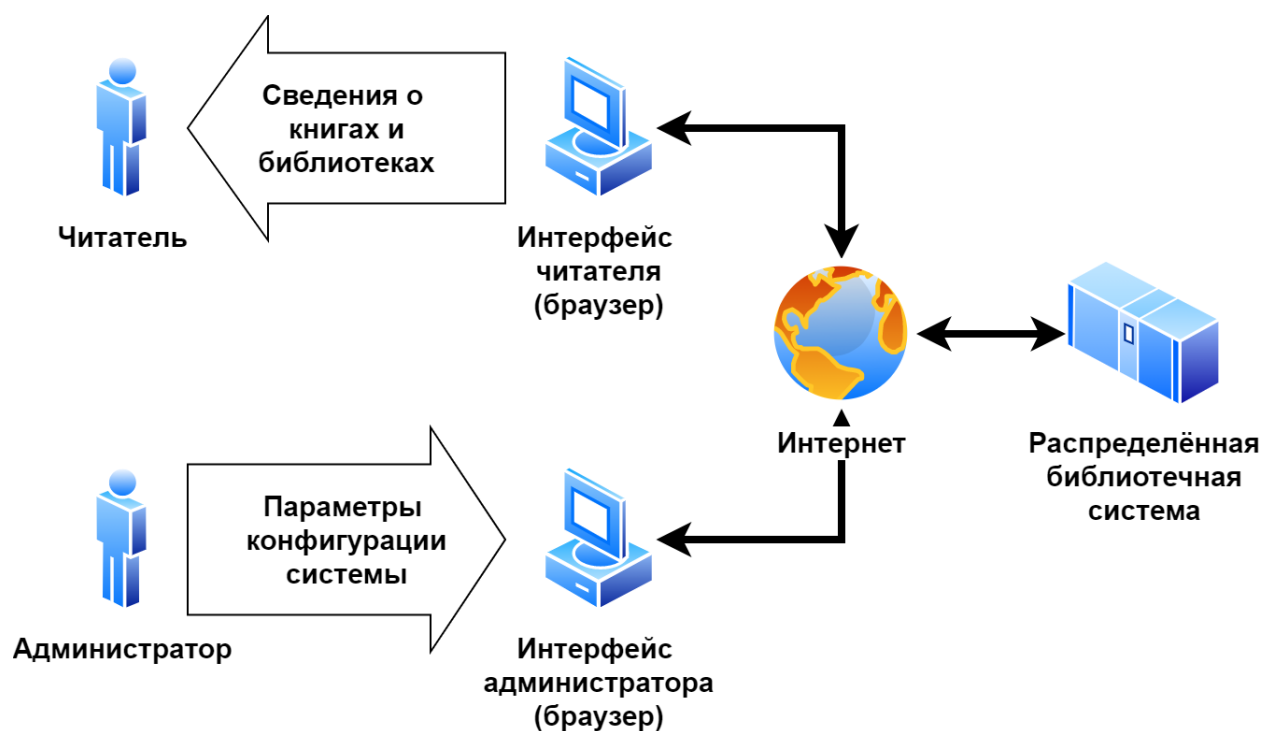


Рисунок 1 – Общая схема предметной области

1.3. Назначение разработки

Назначение разработки – привлечение в библиотеку новых читателей и сохранение старых, путём предоставления им удобного сервиса для дистанционного поиска, выбора и бронирования книг. Наличие такого сервиса может быть привлекательным для представителей молодого поколения и людей, уверенно пользующихся Интернетом, но не знающих устройство классических библиотек, либо испытывающих затруднения при посещении этих библиотек.

1.4. Общие требования к системе

Можно выделить следующие общие требования к системе:

- 1) Система должна состоять из двух основных компонентов – front-end'a и back-end'a, последний представляет собой совокупность взаимодействующих между собой независимых подсистем.
- 2) Время восстановления системы после сбоя не должно превышать 15 минут.
- 3) Каждый узел должен автоматически восстанавливаться после сбоя.
- 4) Обеспечить безопасность работоспособности системы за счет отказоустойчивости узлов.

1.5. Требования к функциональным характеристикам

- 1) По результатам работы модуля сбора статистики медиана времени отклика системы на запросы пользователя на получение информации не должна превышать 5 секунд без учета латентности географического расположения узла.
- 2) По результатам работы модуля сбора статистики медиана времени отклика системы на запросы, добавляющие или изменяющие информацию в системе, не должна превышать 7 секунд без учета латентности географического расположения узла.

1.6. Функциональные требования с точки зрения пользователя

- 1) Система должна обеспечивать возможность регистрации новых пользователей с предварительной проверкой вводимых ими данных.
- 2) Система должна обеспечивать идентификацию, аутентификацию и авторизацию зарегистрированных пользователей.

3) Система должна обеспечивать разделение зарегистрированных пользователей на следующие роли:

- читатель;
- администратор.

4) Система должна предоставлять читателю следующие функции:

- просмотр информации обо всех книгах;
- просмотр информации обо всех библиотеках, доступных в системе;
- бронирование книги;
- просмотр информации своего профиля в системе, а также читательского рейтинга;
- просмотр собственного лимита для бронирования книг.

5) Администратору системы должны быть предоставлены следующие функции:

- просмотр информации обо всех зарегистрированных в системе пользователях;
- возможность изменения и удаления данных зарегистрированных в системе читателей;
- возможность добавления, изменения и удаления данных о книгах;
- возможность добавления, изменения и удаления данных о библиотеках;
- возможность отметить книгу как выданную пользователю;
- возможность отметить книгу как возвращённую в библиотеку;
- просмотр отчётов со статистикой по всем операциям для каждого сервиса системы.

1.7. Описание входных данных

Входные данные читателя:

- Электронный адрес, не более 256 символов.
- Пароль, не более 256 символов.
- ФИО, не более 256 символов.

Входные данные администратора:

- Электронный адрес, не более 256 символов.
- Пароль, не более 256 символов.
- ФИО, не более 256 символов.

Входные данные книги:

- Название, не более 1024 символов.
- Авторы, не более 256 символов на каждую запись.
- Жанры, не более 256 символов на каждую запись.
- Описание, не более 100000 символов.
- Стоимость книги.
- Количество экземпляров для каждой библиотеки, владеющей книгой.

Входные данные библиотеки:

- Название, не более 1024 символов.
- Адрес, не более 1024 символов.
- График работы по дням недели и праздникам, по 256 символов на каждую запись.

1.8. Описание выходных данных

Выходными данными системы для пользователя являются веб-страницы, содержащие в зависимости от запроса следующую информацию:

- список всех библиотек, зарегистрированных в системе;
- сведения о конкретной библиотеке;
- список всех книг, удовлетворяющих поисковому запросу (по жанру, по названию, по авторам);
- сведения о конкретной книге;
- список всех читателей и администраторов;
- сведения о конкретном читателе или администраторе;
- сведения о статистике бронирования и возврата книг читателями;
- сведения о статистике бронирования конкретной книги;
- сведения о статистике использования системы.

1.9. Требования к программной реализации

1) Требуется использовать СОА (сервис-ориентированную архитектуру) для реализации системы.

2) Система состоит из микросервисов; каждый сервис реализует свою функциональность и взаимодействует с другими сервисами по протоколу HTTP в нотации RESTful, либо через очередь.

3) Все запросы к back-end'у от front-end'а также осуществляются по протоколу HTTP в нотации RESTful.

4) Данные сервисов должны храниться в базе данных; каждый сервис взаимодействует только со своей схемой данных.

5) Выделить отдельный сервис авторизации, который хранит в себе информацию о читателях и администраторах и используется для их авторизации и аутентификации; использовать авторизацию на основе JWT-токенов; пароли хранить в хэшированном виде.

6) Реализовать межсервисную авторизацию для запросов между сервисами.

7) Выделить Gateway Service для маршрутизации запросов между остальными сервисами, который будет единой точкой входа в систему.

8) Реализовать пользовательский веб-интерфейс с использованием HTML и CSS; запросы от него могут быть адресованы только к Gateway Service или сервису авторизации.

9) Реализовать проверку входных данных как на front-end'е, так и на back-end'е.

10) Выделить сервис статистики, агрегирующий информацию по всем операциям; обращение к сервису статистики от остальных микросервисов в системе построить на основе очереди.

11) При недоступности какого-либо из компонентов системы должна осуществляться деградация функциональности или выдача пользователю сообщения об ошибке.

12) Для запросов, выполняющих обновление данных на нескольких узлах распределенной системы, в случае недоступности одной из систем, необходимо выполнять полный откат транзакции.

13) Приложение должно поддерживать возможность горизонтального и вертикального масштабирования за счет увеличения количества функционирующих узлов и совершенствования технологий реализации компонентов и всей архитектуры системы.

1.10. Требования по подготовке к вводу системы в эксплуатацию

Минимальные требования к аппаратному обеспечению для развертывания каждого из компонентов back-end'a:

- Процессор – четырехядерный, частота не менее 2 ГГц;
- Оперативная память – не менее 8 Гб;

- Свободное пространство на жестком диске – не менее 1000 Гб;
- Наличие Ethernet адаптера, обеспечивающего передачу данных со скоростью не менее 100 Мб/с.

Для развертывания front-end'a требуется:

- Процессор – одноядерный, частота не менее 2 ГГц;
- Оперативная память – не менее 8 Гб;
- Свободное пространство на жестком диске – не менее 10 Гб;
- Наличие Ethernet адаптера, обеспечивающего передачу данных со скоростью не менее 100 Мб/с.

1.11. Топология системы

Топология системы приведена на рисунке 2

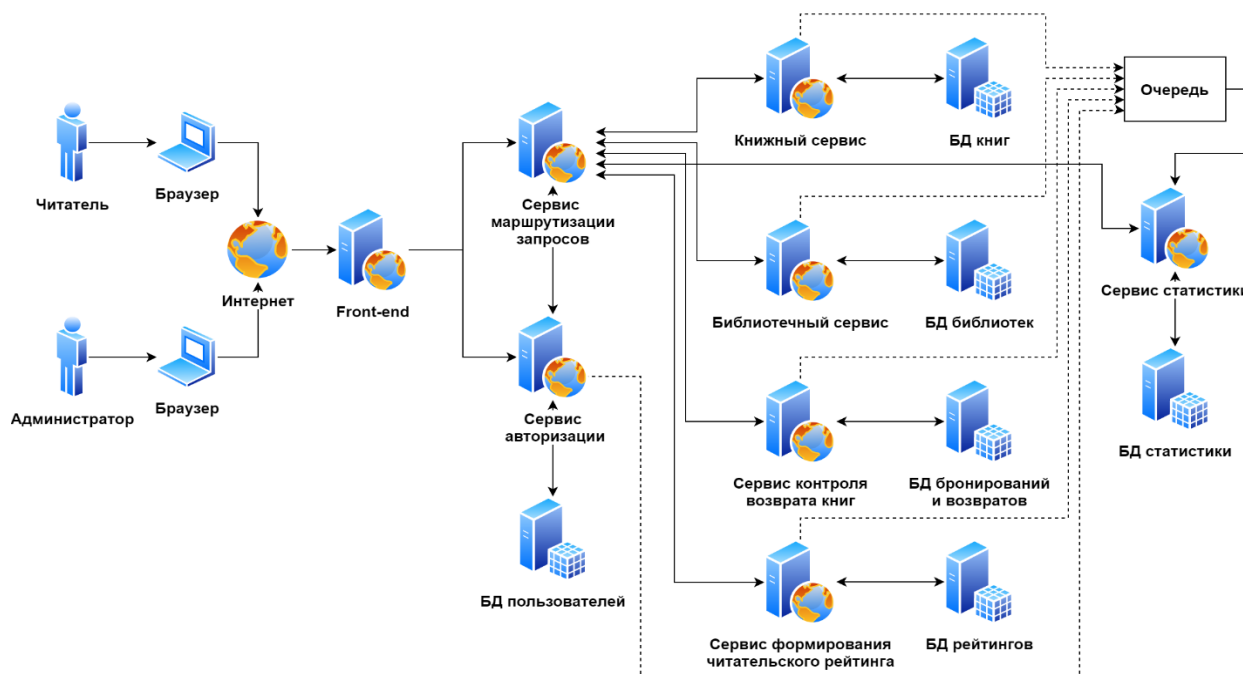


Рисунок 2 – Топология системы.

Разрабатываемое программное обеспечение включает front-end и back-end. Последний включает в себя 7 подсистем:

- сервис маршрутизации запросов;
- сервис авторизации;
- книжный сервис;
- библиотечный сервис;
- сервис контроля возврата книг;
- сервис формирования читательского рейтинга;
- сервис статистики.

1.12. Общие требования к подсистемам

1) Front-end представляет собой серверное приложение, при разработке которого необходимо учесть следующие требования:

- Front-end должен принимать запросы по протоколу HTTP и формировать ответ пользователю в формате HTML-страниц;
- Front-end является посредником между пользователями и back-end'ом, передавая их запросы последовательно на сервис маршрутизации запросов.

2) К реализации сервисов back-end'а должны быть предъявлены следующие требования:

- Прием и возврат всех данных, должен происходить в формате JSON по протоколу HTTP в соответствии с REST;
- Все сервисы изолированы друг от друга, т. е. развертываются и исполняются в разных средах (на разных виртуальных и/или физических машинах).

1.13. Функциональные требования к сервисам

1. **Сервис маршрутизации запросов** – единая точка входа в систему, через которую проходят все запросы пользователя кроме запроса на авторизацию. Данный компонент предоставляет внешний API back-end'a и реализует следующие функции:

- регистрация нового пользователя (читателя или администратора);
- авторизация зарегистрированного пользователя;
- изменение зарегистрированного пользователя;
- получение списка всех зарегистрированных пользователей;
- удаление зарегистрированного пользователя;
- получение списка библиотеки;
- получение списка книг;
- получение подробной информации о книге;
- удаление книги;
- изменение информации о книге;
- получение списка информации обо всех библиотеках в системе;
- получение информации о библиотеке;
- удаление информации о библиотеке;
- добавление информации о библиотеке;
- добавление книги в фонд библиотеки;
- удаление книги из фонда библиотеки;
- изменение числа экземпляров книги в фонде библиотеке;
- бронирование книги;
- возврат книги;
- добавление заказа в историю;
- создание пустой истории для нового пользователя;
- удаление существующего пользователя и его заказов из истории;

- изменение лимита книг на руках пользователя;
- получение книжного лимита пользователя;
- создание базового рейтинга для нового пользователя;
- изменение рейтинга пользователя;
- получение рейтинга пользователя;
- удаление рейтинга пользователя;
- просмотр статистики по всем запросам, поступающим в систему.

2. Сервис авторизации – выполняет идентификацию, аутентификацию и авторизацию зарегистрированных пользователей, а также регистрацию новых пользователей. Реализует следующие функции:

- регистрация нового пользователя;
- авторизация зарегистрированного пользователя;
- изменение зарегистрированного пользователя;
- получение списка всех зарегистрированных пользователей;
- удаление зарегистрированного пользователя.

Хранимая в базе данных сущность, ассоциированная с данным сервисом, включает следующие обязательные поля:

- идентификатор пользователя;
- ФИО пользователя;
- электронный адрес пользователя;
- захешированный пароль пользователя;
- роль пользователя.

3. Книжный сервис – отвечает за обработку данных по всем книгам в системе. Реализует следующие функции:

- получение списка книг;

- получение подробной информации о книге;
- удаление книги;
- изменение информации о книге.

В базе данных сервиса хранится следующая сущность, ассоциированная с сервисом:

- идентификатор книги;
- наименование книги;
- жанры книги;
- авторы книги;
- описание книги.

4. Библиотечный сервис – управляет хранением данных о библиотеках и хранящихся в них книгах. Реализует следующие функции:

- получение списка информации обо всех библиотеках в системе;
- получение информации о библиотеке;
- удаление информации о библиотеке;
- добавление информации о библиотеке;
- добавление книги в фонд библиотеки;
- удаление книги из фонда библиотеки;
- изменение числа экземпляров книги в фонде библиотеке;
- бронирование книги;
- возврат книги;

В базе данных сервиса хранятся следующие сущности:

- Библиотека:
 - идентификатор;
 - название;

- адрес;
- график работы.
- Библиотечные фонды:
 - идентификатор библиотеки;
 - идентификатор книги;
 - доступное количество книг.
- Заказ:
 - идентификатор;
 - идентификатор библиотеки;
 - идентификатор книги;
 - идентификатор пользователя;
 - дата бронирования;
 - дата выдачи;
 - дата возврата;
 - состояние книги после возврата.

5. Сервис контроля возврата книг – хранит историю выдачи и возврата книг пользователем. Реализует следующие функции:

- добавление заказа для изменения книжного лимита;
- удаление заказа для изменения книжного лимита;
- создание пустой истории для нового пользователя;
- удаление существующего пользователя и его истории;
- изменение лимита книг на руках пользователя;
- получение книжного лимита пользователя.

Хранимая в базе данных сущность, ассоциированная с данным сервисом, включает следующие обязательные поля:

- идентификатор пользователя;

- идентификаторы заказов;
- лимит книг на руках.

6. Сервис формирования читательского рейтинга. Реализует следующие функции:

- создание базового рейтинга для нового пользователя;
- изменение рейтинга пользователя;
- получение рейтинга пользователя;
- удаление рейтинга пользователя.

Сущность базы данных, ассоциированная с сервисом, включает следующий перечень полей:

- идентификатор пользователя;
- пользовательский рейтинг.

7. Сервис статистики – выполняет сбор статистики по всем операциям в системе. Реализует следующие функции:

- просмотр статистики по всем запросам, поступающим в систему.

Хранимая в базе данных сущность, ассоциированная с данным сервисом, включает следующие обязательные поля:

- идентификатор операции;
- идентификатор сервиса, на котором была произведена операция;
- тип содержимого статистической записи;
- содержимое статистической записи.

1.14. Требования к надежности

Система должна работать в соответствии с данным техническим заданием без перезапуска.

Для всех сервисов в системе должен быть настроен автоматический перезапуск в случае сбоя. На время сбоя все сервисы, обеспечивающие доступ к отказавшему узлу, должны выполнять деградацию функциональности или возвращать сообщение с соответствующей ошибкой. Автоматический перезапуск в случае сбоя должен быть также предусмотрен для клиентского компонента системы.

2. Конструкторский раздел

2.1. Концептуальный дизайн

Концептуальный дизайн включает описание разрабатываемой системы с точки зрения пользователей. Для создания функциональной модели портала, отражающей его основные функции и потоки информации, используется IDEF0-модель и графические модели, входящие в нее. На рисунке 3 отображена IDEF0-диаграмма верхнего уровня, обеспечивающая наиболее общее и абстрактное описание работы системы. Данный вид диаграмм позволяет формализовать описание запросов пользователя и ответов системы на данные запросы, отобразив ее в виде «черного ящика».

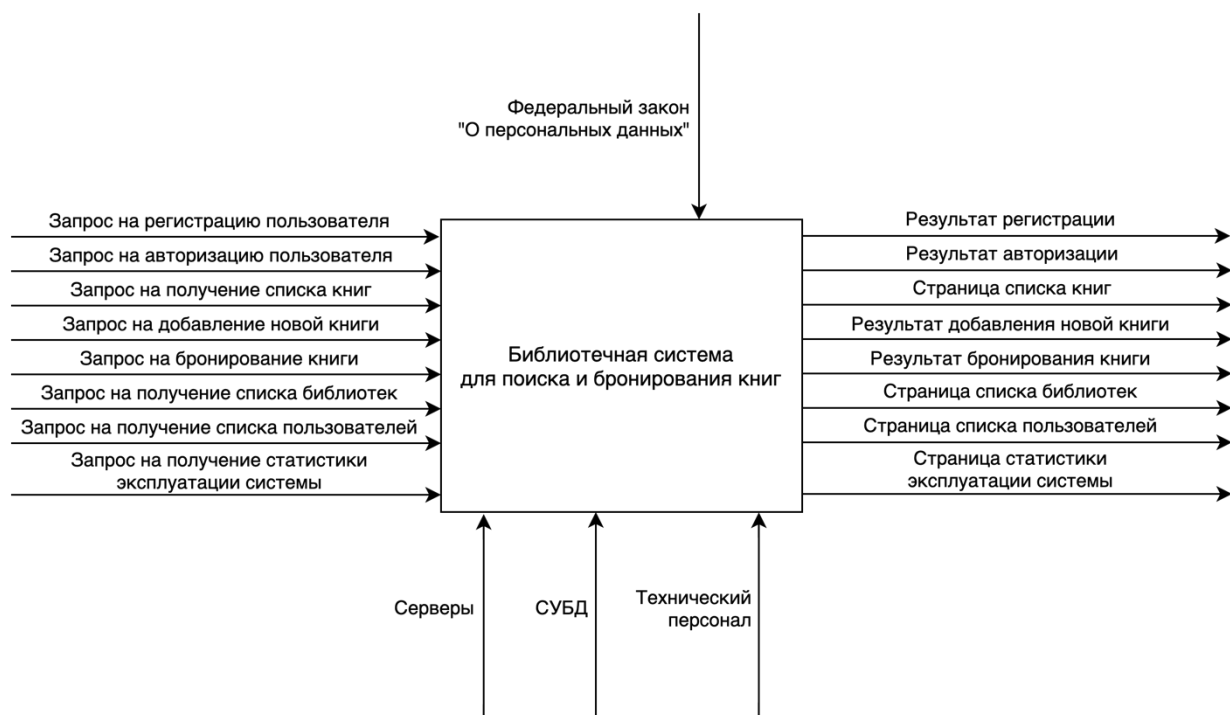


Рисунок 3 – Концептуальное описание системы в нотации IDEF0

На рисунке 4 приведена детализированная схема процесса бронирования книги.

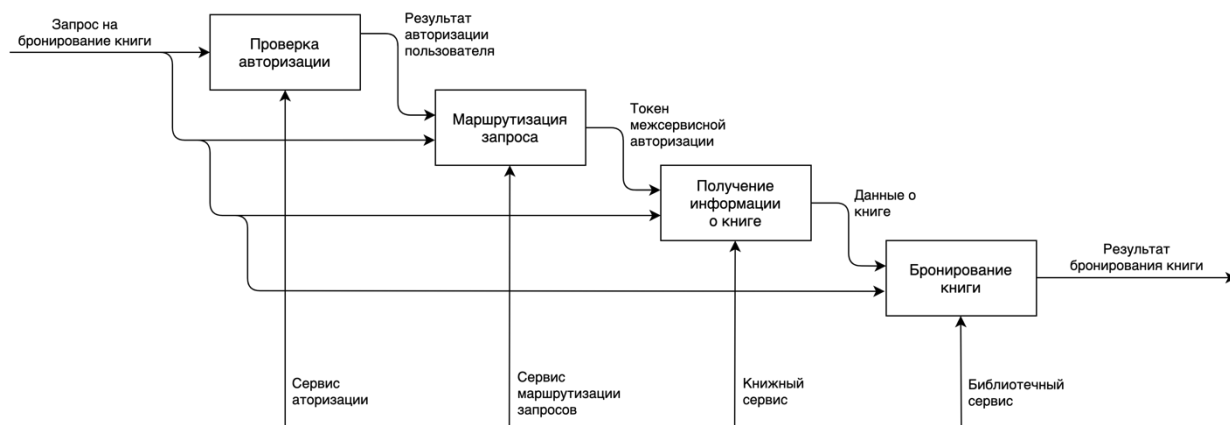


Рисунок 4 – Детализированная схема процесса бронирования книги в нотации IDEF0

2.2. Сценарии функционирования системы

Сценарии функционирования системы описывают конкретную последовательность действий, иллюстрирующую поведение пользователя при работе с приложением. Далее приведены подробные сценарии основных возможных действий пользователя.

2.2.1. Регистрация нового пользователя с ролью «Администратор»

Действие может быть выполнено только авторизованным пользователем с ролью «Администратор» из его графического интерфейса и состоит из следующих шагов:

- Администратор нажимает на кнопку «Пользователи» на главной странице системы.
- Администратор перенаправляется на страницу зарегистрированных пользователей системы.

- Администратор нажимает на кнопку «Добавить нового администратора» на странице зарегистрированных пользователей.
- Администратор перенаправляется на страницу регистрации, содержащую поля данных нового пользователя.
- Администратор вводит данные нового администратора в графический интерфейс и нажимает кнопку «Регистрация».
- В случае корректности введенных данных администратор перенаправляется на страницу зарегистрированных пользователей, которая уже содержит только что добавленного пользователя.
- В случае ошибки во введенных данных или внутренней ошибки системы на странице регистрации выводится соответствующее сообщение об ошибке, а также предложение повторить ввод.

2.2.2. Регистрация нового пользователя с ролью «Читатель»

Действие может быть выполнено только пользователем с ролью «Читатель» из его интерфейса и состоит из следующих шагов:

- Читатель нажимает на кнопку «Регистрация» в графическом интерфейсе.
- Читатель перенаправляется на страницу регистрации, которая содержит поля данных нового пользователя.
- Читатель вводит данные в форму и для завершения регистрации нажимает на кнопку «Регистрация», тем самым подтверждая верность своих данных, а также согласие на их обработку и хранение.
- В случае ошибки во введенных данных или внутренней ошибки системы на странице регистрации выводится соответствующее сообщение об ошибке, а также предложение повторить ввод.
- При успешной регистрации читатель перенаправляется на страницу своего профиля в системе.

2.2.3. Авторизация зарегистрированного пользователя

Действие может быть выполнено любым неавторизованным пользователем и включает следующие шаги:

- Пользователь нажимает кнопку «Войти» в графическом интерфейсе на странице входа в систему.
- Пользователь перенаправляется на страницу входа в систему, содержащую поля для ввода логина и пароля.
- Пользователь вводит логин и пароль в поля графического интерфейса страницы входа и нажимает на кнопку «Вход».
- В случае корректности указанных учетных данных пользователь перенаправляется на главную страницу системы, содержащую данные авторизованного пользователя, а также ссылки для перехода на остальные страницы интерфейса системы.
- В случае ошибки на странице входа выводится сообщение об ошибке и предложение повторить ввод.

2.2.4. Добавление новой библиотеки

Действие может быть выполнено только авторизованным пользователем с ролью «Администратор» из его интерфейса и состоит из следующих шагов:

- Администратор нажимает на кнопку «Библиотеки» на главной странице системы.
- Администратор перенаправляется на страницу зарегистрированных библиотек.
- На странице зарегистрированных библиотек администратор нажимает на кнопку «Добавить библиотеку».

- Администратор перенаправляется на страницу добавления новой библиотеки, содержащей поля ввода данных библиотеки.
- Администратор вводит данные новой библиотеки и нажимает на кнопку «Добавить».
- В случае корректности введенных данных и успешного добавления сведений в систему выводится уведомление об успешном добавлении новой библиотеки, и администратор перенаправляется на страницу зарегистрированных библиотек, которая теперь содержит новую библиотеку.
- В случае ошибки во введенных данных или внутренней ошибки системы выводится уведомление о соответствующей ошибке и предложение повторить ввод.

2.2.5. Удаление существующей библиотеки

Действие может быть выполнено только авторизованным пользователем с ролью «Администратор» из его интерфейса и состоит из следующих шагов:

- Администратор нажимает на кнопку «Библиотеки» на главной странице системы.
- Администратор перенаправляется на страницу зарегистрированных библиотек.
- На странице зарегистрированных библиотек администратор выбирает библиотеку для удаления и нажимает на кнопку «Удалить» рядом с выбранной библиотекой.
- Администратор перенаправляется на страницу с подтверждением удаления выбранной библиотеки.

- Администратор вводит свой пароль в поле ввода пароля на странице подтверждения удаления и нажимает на кнопку «Удалить».
- В случае успеха администратору выводится уведомление об успешном выполнении операции, и он перенаправляется на страницу зарегистрированных библиотек, на которой теперь отсутствует удаленная библиотека.
- В случае ошибки ввода пароля на странице подтверждения выводится сообщение с предложением повторить ввод.
- В случае успешного ввода пароля и последующего возникновения внутренней ошибки системы выводится уведомление с сообщением об ошибке, и администратор также перенаправляется обратно на страницу зарегистрированных библиотек.

2.2.6. Добавление новой книги

Действие может быть выполнено только авторизованным пользователем с ролью «Администратор» из его интерфейса и состоит из следующих шагов:

- Администратор нажимает на кнопку «Книги» на главной странице системы.
- Администратор перенаправляется на страницу зарегистрированных книг.
- На странице зарегистрированных книг администратор нажимает на кнопку «Добавить книгу».
- Администратор перенаправляется на страницу добавления новой книги, содержащую поля ввода данных книги.
- Администратор вводит данные новой книги в графический интерфейс страницы добавления новой книги и нажимает на кнопку «Добавить».

- В случае корректности введенных данных и успешного добавления сведений в систему выводится уведомление об успешном добавлении новой книги, и администратор перенаправляется на страницу зарегистрированных книг, которая теперь содержит новую книгу.
- В случае ошибки во введенных данных или внутренней ошибки системы выводится уведомление о соответствующей ошибке и предложение повторить ввод.

2.2.7. Удаление существующей книги

Действие может быть выполнено только авторизованным пользователем с ролью «Администратор» из его интерфейса и состоит из следующих шагов:

- Администратор нажимает на кнопку «Книги» на главной странице системы.
- Администратор перенаправляется на страницу зарегистрированных книг.
- На странице зарегистрированных книг администратор выбирает книгу для удаления и нажимает на кнопку «Удалить» рядом с выбранной книгой.
- Администратор перенаправляется на страницу с подтверждением удаления выбранной книги.
- Администратор вводит свой пароль в поле ввода пароля на странице подтверждения удаления и нажимает на кнопку «Удалить».
- В случае успеха администратору выводится уведомление об успешном выполнении операции, и он перенаправляется на страницу зарегистрированных книг, на которой теперь отсутствует удаленная книга.

- В случае ошибки ввода пароля на странице подтверждения выводится сообщение с предложением повторить ввод.
- В случае успешного ввода пароля и последующего возникновения внутренней ошибки системы выводится уведомление с сообщением об ошибке, и администратор также перенаправляется обратно на страницу зарегистрированных книг.

2.2.8. Добавление нового бронирования

Действие может быть выполнено только авторизованным пользователем с ролью «Читатель» из его интерфейса и состоит из следующих шагов:

- Читатель нажимает на кнопку «Книги» на главной странице системы.
- Читатель перенаправляется на страницу зарегистрированных книг.
- На странице зарегистрированных книг читатель выбирает книгу для бронирования и нажимает на кнопку «Бронировать» рядом с выбранной книгой.
- Читатель перенаправляется на страницу бронирования книги, содержащую поля ввода данных бронирования.
- Читатель вводит данные нового бронирования в графический интерфейс страницы добавления нового бронирования и нажимает на кнопку «Забронировать».
- В случае корректности введенных данных и успешного бронирования книги в систему выводится уведомление об успешном бронировании книги, и читатель перенаправляется на страницу бронированных им книг, которая теперь содержит новую забронированную книгу.

- В случае ошибки во введенных данных или внутренней ошибки системы выводится уведомление о соответствующей ошибке и предложение повторить ввод.

2.2.9. Просмотр сведений о читателях

Действие может быть выполнено только авторизованным пользователем с ролью «Администратор» из его интерфейса и состоит из следующих шагов:

- Администратор нажимает на кнопку «Читатели» на главной странице системы.
- Администратор перенаправляется на страницу читателей.

2.2.10. Возврат книги

Действие может быть выполнено только авторизованным пользователем с ролью «Администратор» из его интерфейса и состоит из следующих шагов:

- Администратор нажимает на кнопку «Заказы» на главной странице системы.
- Администратор перенаправляется на страницу заказов.
- На странице заказов администратор выбирает заказ читателя, осуществляющего возврат книги, и нажимает на кнопку «Возврат» рядом с выбранной книгой.
- Администратор вводит свой пароль в поле ввода пароля на странице подтверждения возврата и нажимает на кнопку «Возврат»;

- В случае успеха администратору выводится уведомление об успешном выполнении операции, и он перенаправляется на страницу заказов, на которой теперь отсутствует возвращённая книга;
- В случае ошибки ввода пароля на странице подтверждения выводится сообщение с предложением повторить ввод;
- В случае успешного ввода пароля и последующего возникновения внутренней ошибки системы выводится уведомление с сообщением об ошибке, и администратор также перенаправляется обратно на страницу заказов.

2.2.11. Получение сведений о библиотеке

Действие может быть выполнено любым авторизованным пользователем и включает следующие шаги:

- Пользователь нажимает на кнопку «Библиотеки» на главном экране системы.
- Пользователь перенаправляется на страницу зарегистрированных библиотек.
- Пользователь выбирает библиотеку из списка и нажимает на кнопку «Сведения» рядом с выбранной библиотекой.
- Пользователь перенаправляется на страницу выбранной библиотеки, содержащую информацию о библиотеке.

2.2.12. Получение сведений о книге

Действие может быть выполнено любым авторизованным пользователем и включает следующие шаги:

- Пользователь нажимает на кнопку «Книги» на главном экране системы.
- Пользователь перенаправляется на страницу зарегистрированных книг.
- Пользователь выбирает книгу из списка и нажимает на кнопку «Сведения» рядом с выбранной книгой.
- Пользователь перенаправляется на страницу сведений о выбранной книге.
- Пользователь выбирает экземпляр оборудования в списке развернутых экземпляров для данной модели оборудования и нажимает на кнопку «Сведения»;
- Пользователь перенаправляется на страницу выбранной книги, содержащую информацию о книге.

2.3. Диаграммы прецедентов

Графически сценарии функционирования системы можно представить при помощи диаграмм прецедентов. В системе выделены две основные роли: читатель и администратор, диаграммы прецедентов для этих ролей изображены на рисунках 5, 6 и 7.

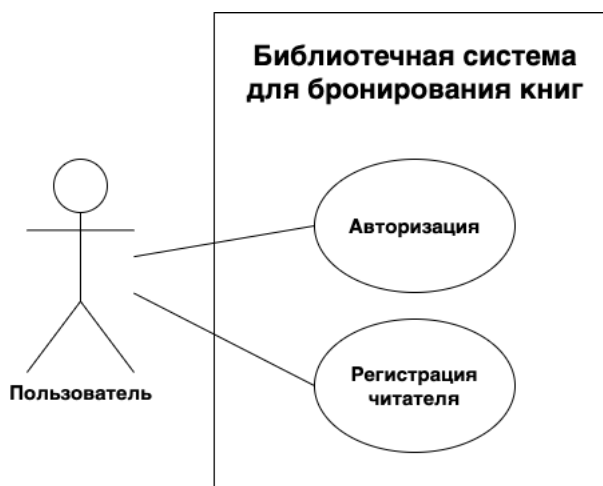


Рисунок 5 – Диаграмма прецедентов для неавторизованного пользователя

2.3.1. Спецификации сценариев для неавторизованного пользователя

Спецификация сценария «Авторизация»

Нормальный ход сценария	
Действие пользователя	Отклик системы
Пользователь нажимает кнопку «Войти»	Открывается страница для ввода данных
Пользователь вводит данные в поля и нажимает кнопку «Войти»	Открывается главная страница системы
Альтернативный ход сценария	
Действие пользователя	Отклик системы
Пользователь нажимает кнопку «Войти»	Открывается страница для ввода данных
Пользователь вводит данные в поля и нажимает кнопку «Войти»	Выводится сообщение об ошибке во введенных данных и предложение повторить ввод
Альтернативный ход сценария	
Действие пользователя	Отклик системы
Пользователь нажимает кнопку «Войти»	Открывается страница для ввода данных
Пользователь вводит данные в поля и нажимает кнопку «Войти»	Выводится сообщение о внутренней ошибке сервера

Спецификация сценария «Регистрация читателя»

Нормальный ход сценария	
Действие пользователя	Отклик системы

Пользователь нажимает кнопку «Регистрация»	Открывается страница для ввода данных
Пользователь вводит данные в поля и нажимает кнопку «Зарегистрироваться»	Выводится сообщение об успешной регистрации, и пользователь перенаправляется на главную страницу
Альтернативный ход сценария	
Действие пользователя	Отклик системы
Пользователь нажимает кнопку «Регистрация»	Открывается страница для ввода данных
Пользователь вводит данные в поля и нажимает кнопку «Зарегистрироваться»	Выводится сообщение об ошибке во введенных данных и предложение повторить ввод
Альтернативный ход сценария	
Действие пользователя	Отклик системы
Пользователь нажимает кнопку «Регистрация»	Открывается страница для ввода данных
Пользователь вводит данные в поля и нажимает кнопку «Зарегистрироваться»	Выводится сообщение о внутренней ошибке сервера

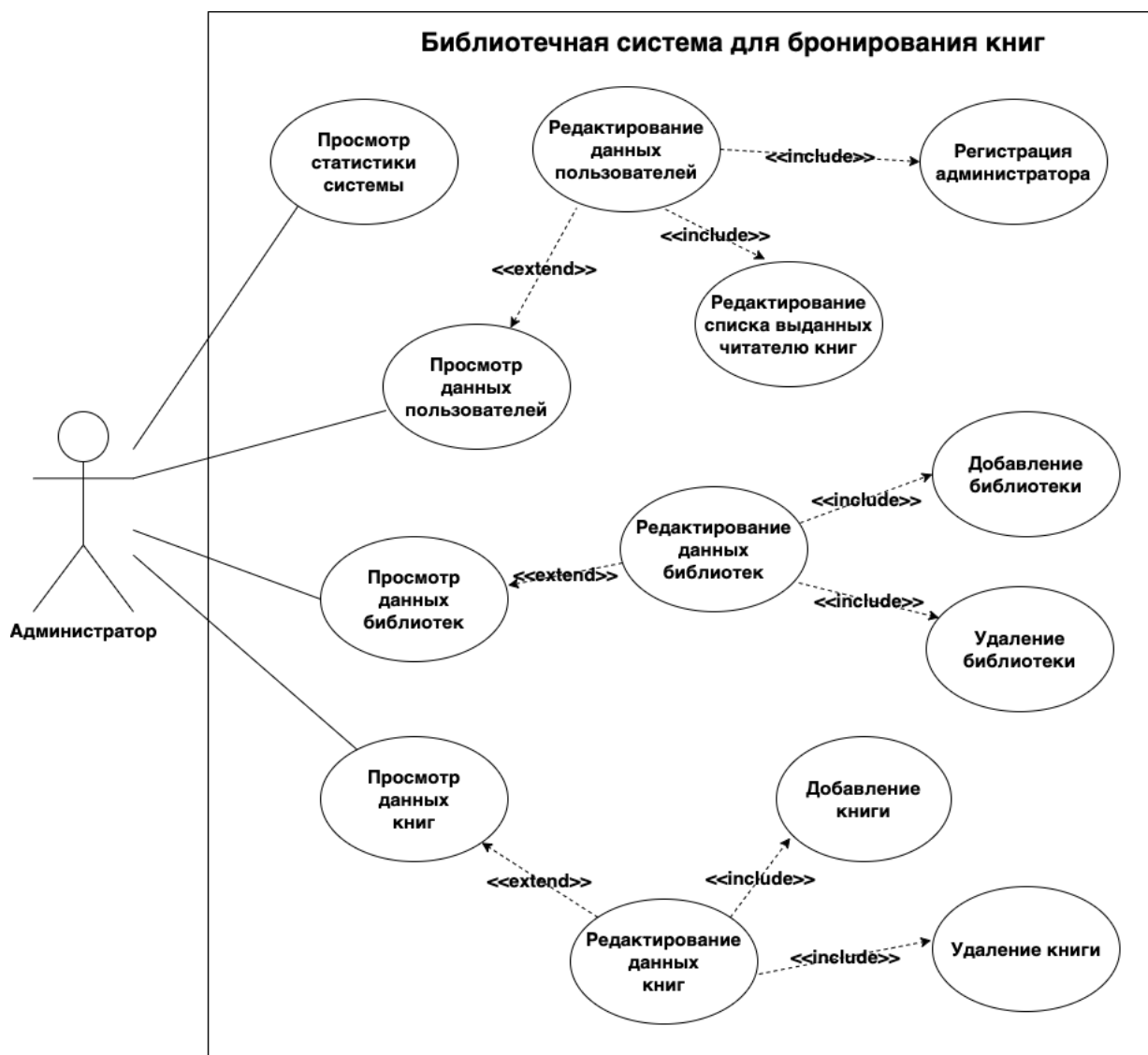


Рисунок 6 – Диаграмма прецедентов для пользователя с ролью «Администратор»

2.3.2. Спецификации сценариев для пользователя с ролью «Администратор»

Спецификация сценария «Просмотр данных пользователей»

Нормальный ход сценария	
Действие администратора системы	Отклик системы
Пользователь выбирает пункт	Открывается страница со списком

«Пользователи» в главном меню	зарегистрированных пользователей
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь выбирает пункт «Пользователи» в главном меню	Выводится уведомление с сообщением о внутренней ошибке сервера

Спецификация сценария «Регистрация администратора»

Нормальный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Добавить администратора» на странице «Пользователи»	Открывается страница для ввода данных нового пользователя
Пользователь вводит данные в поля и нажимает кнопку «Регистрация»	Выводится уведомление об успешной регистрации, и пользователь возвращается на страницу «Пользователи»
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Добавить администратора» на странице «Пользователи»	Открывается страница для ввода данных нового пользователя
Пользователь вводит данные в поля и нажимает кнопку «Регистрация»	Выводится сообщение об ошибке во введенных данных
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Добавить администратора» на	Открывается страница для ввода данных нового пользователя

странице «Пользователи»	
Пользователь вводит данные в поля и нажимает кнопку «Регистрация»	Выводится сообщение о внутренней ошибке сервера, и пользователь возвращается на страницу «Пользователи»

Спецификация сценария «Удаление пользователя»

Нормальный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Удалить пользователя» для выбранной записи на странице «Пользователи»	Открывается страница для подтверждения удаления
Пользователь вводит пароль для подтверждения и нажимает на кнопку «Удалить»	Выводится сообщение об успешном удалении, и пользователь возвращается на страницу «Пользователи»
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Удалить пользователя» для выбранной записи на странице «Пользователи»	Открывается страница для подтверждения удаления
Пользователь вводит пароль для подтверждения и нажимает на кнопку «Удалить»	Выводится сообщение о неверном пароле и предложение повторить ввод
Альтернативный ход сценария	
Действие администратора системы	Отклик системы

Пользователь нажимает кнопку «Удалить пользователя» для выбранной записи на странице «Пользователи»	Открывается страница для подтверждения удаления
Пользователь вводит пароль для подтверждения и нажимает на кнопку «Удалить»	Выводится сообщение о внутренней ошибке сервера, и пользователь возвращается на страницу «Пользователи»

Спецификация сценария «Редактирование списка выданных читателю книг»

Нормальный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Выданные книги» в главном меню	Открывается страница «Выданные книги»
Пользователь нажимает кнопку «Возврат» для выбранной записи на странице «Выданные книги»	Открывается страница для подтверждения возврата
Пользователь вводит пароль для подтверждения и нажимает на кнопку «Возврат»	Выводится сообщение об успешном возврате, и пользователь возвращается на страницу «Выданные книги»
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Выданные книги» в главном меню	Открывается страница «Выданные книги»
Пользователь нажимает кнопку «Возврат» для выбранной записи на	Открывается страница для подтверждения возврата

странице «Выданные книги»	
Пользователь вводит пароль для подтверждения и нажимает на кнопку «Возврат»	Выводится сообщение о неверном пароле и предложение повторить ввод
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Выданные книги» в главном меню	Открывается страница «Выданные книги»
Пользователь нажимает кнопку «Возврат» для выбранной записи на странице «Выданные книги»	Открывается страница для подтверждения возврата
Пользователь вводит пароль для подтверждения и нажимает на кнопку «Возврат»	Выводится сообщение о внутренней ошибке сервера, и пользователь возвращается на страницу «Выданные книги»
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Выданные книги» в главном меню	Открывается страница «Выданные книги»
Пользователь нажимает кнопку «Возврат» для выбранной записи на странице «Выданные книги»	Выводится уведомление с сообщением о внутренней ошибке сервера
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Выданные книги» в главном меню	Выводится уведомление с сообщением о внутренней ошибке сервера

Спецификация сценария «Просмотр статистики системы»

Нормальный ход сценария	
Действие администратора системы	Отклик системы
Пользователь выбирает пункт «Статистика системы» в главном меню	Открывается страница статистики со списком сервисов с возможностью просмотра данных по всем операциям для каждого сервиса
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь выбирает пункт «Статистика системы» в главном меню	Выводится уведомление с сообщением о внутренней ошибке сервера

Спецификация сценария «Добавление библиотеки»

Нормальный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Добавить библиотеку» на странице «Библиотеки»	Открывается страница с полями ввода данных новой библиотеки
Пользователь вводит данные библиотеки и нажимает на кнопку «Добавить»	Выводится сообщение об успешном добавлении библиотеки, и пользователь возвращается на страницу «Библиотеки»
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Добавить библиотеку» на странице «Библиотеки»	Открывается страница с полями ввода данных новой библиотеки

Пользователь вводит данные библиотеки и нажимает на кнопку «Добавить»	Выводится сообщение об ошибке при добавлении библиотеки и предложение повторить ввод
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Добавить библиотеку» на странице «Библиотеки»	Открывается страница с полями ввода данных новой библиотеки
Пользователь вводит данные библиотеки и нажимает на кнопку «Добавить»	Выводится сообщение о внутренней ошибке сервера, и пользователь возвращается на страницу «Библиотеки»

Спецификация сценария «Удаление библиотеки»

Нормальный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Удалить библиотеку» для выбранной библиотеки на странице «Библиотеки»	Открывается страница с подтверждением удаления
Пользователь вводит пароль для подтверждения и нажимает на кнопку «Удалить»	Выводится сообщение об успешном удалении, и пользователь возвращается на страницу «Библиотеки»
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Удалить библиотеку» для выбранной библиотеки на странице «Библиотеки»	Открывается страница с подтверждением удаления

Пользователь вводит пароль для подтверждения и нажимает на кнопку «Удалить»	Выводится сообщение о неверном пароле и предложение повторить ввод
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Удалить библиотеку» для выбранной библиотеки на странице «Библиотеки»	Открывается страница с подтверждением удаления
Пользователь вводит пароль для подтверждения и нажимает на кнопку «Удалить»	Выводится сообщение о внутренней ошибке сервера, и пользователь возвращается на страницу «Библиотеки»

Спецификация сценария «Добавление книги»

Нормальный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Добавить книгу» на странице «Книги»	Открывается страница с полями ввода данных новой книги
Пользователь вводит данные книги и нажимает на кнопку «Добавить»	Выводится сообщение об успешном добавлении книги, и пользователь возвращается на страницу «Книги»
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Добавить книгу» на странице «Книги»	Открывается страница с полями ввода данных новой книги
Пользователь вводит данные книги и	Выводится сообщение об ошибке при

нажимает на кнопку «Добавить»	добавлении книги и предложение повторить ввод
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Добавить книгу» на странице «Книги»	Открывается страница с полями ввода данных новой книги
Пользователь вводит данные книги и нажимает на кнопку «Добавить»	Выводится сообщение о внутренней ошибке сервера, и пользователь возвращается на страницу «Книги»

Спецификация сценария «Удаление книги»

Нормальный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Удалить книгу» для выбранной книги на странице «Книги»	Открывается страница с подтверждением удаления
Пользователь вводит пароль для подтверждения и нажимает на кнопку «Удалить»	Выводится сообщение об успешном удалении, и пользователь возвращается на страницу «Книги»
Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Удалить книгу» для выбранной книги на странице «Книги»	Открывается страница с подтверждением удаления
Пользователь вводит пароль для подтверждения и нажимает на кнопку «Удалить»	Выводится сообщение о неверном пароле и предложение повторить ввод

Альтернативный ход сценария	
Действие администратора системы	Отклик системы
Пользователь нажимает кнопку «Удалить книгу» для выбранной книги на странице «Книги»	Открывается страница с подтверждением удаления
Пользователь вводит пароль для подтверждения и нажимает на кнопку «Удалить»	Выводится сообщение о внутренней ошибке сервера, и пользователь возвращается на страницу «Книги»



Рисунок 7 – Диаграмма прецедентов для пользователя с ролью «Читатель»

2.3.3. Спецификации сценариев для пользователя с ролью «Читатель»

Спецификация сценария «Просмотр данных своего профиля»

Нормальный ход сценария

Действие пользователя «Читатель»	Отклик системы
Пользователь выбирает пункт «Профиль» в главном меню	Открывается страница с данными профиля пользователя
Альтернативный ход сценария	
Действие пользователя «Читатель»	Отклик системы
Пользователь выбирает пункт «Профиль» в главном меню	Выводится уведомление с сообщением о внутренней ошибке сервера

Спецификация сценария «Просмотр данных библиотек»

Нормальный ход сценария	
Действие пользователя «Читатель»	Отклик системы
Пользователь выбирает пункт «Библиотеки» в главном меню	Открывается страница со списком зарегистрированных пользователей
Альтернативный ход сценария	
Действие пользователя «Читатель»	Отклик системы
Пользователь выбирает пункт «Библиотеки» в главном меню	Выводится уведомление с сообщением о внутренней ошибке сервера

Спецификация сценария «Бронирование книг»

Нормальный ход сценария	
Действие пользователя	Отклик системы

«Читатель»	
Пользователь нажимает кнопку «Бронировать» для выбранной книги на странице «Найденные книги»	Открывается страница с полями ввода данных бронирования
Пользователь вводит данные бронирования и нажимает на кнопку «Забронировать»	Выводится сообщение об успешном завершении бронирования, и пользователь перенаправляется на страницу «Мои бронированные книги»
Альтернативный ход сценария	
Действие пользователя «Читатель»	Отклик системы
Пользователь нажимает кнопку «Бронировать» для выбранной книги на странице «Найденные книги»	Открывается страница с полями ввода данных бронирования
Пользователь вводит данные бронирования и нажимает на кнопку «Забронировать»	Выводится сообщение об ошибке при бронировании книги и предложение повторить ввод
Альтернативный ход сценария	
Действие пользователя «Читатель»	Отклик системы
Пользователь нажимает кнопку «Бронировать» для выбранной книги на странице «Найденные книги»	Открывается страница с полями ввода данных бронирования
Пользователь вводит данные бронирования и нажимает на кнопку «Забронировать»	Выводится сообщение о внутренней ошибке сервера, и пользователь возвращается на страницу «Найденные книги»

2.4. Логический дизайн

В данном разделе представлена логическая организация элементов системы и их взаимодействие между собой. На основе функциональных требований к выделенным подсистемам, а также объектов, о которых необходимо хранить данные в системе, была разработана схема данных приложения. Результат ее проектирования отображен на ER-диаграмме, представленной на рисунке 8. На данной схеме прямоугольниками обозначены ключевые сущности, а ромбами – связи между ними. Участие сущности в отношении с другой сущностью отмечается линией, соединяющей их. Число, располагающееся около линии, означает тип связи между соединенными сущностями.

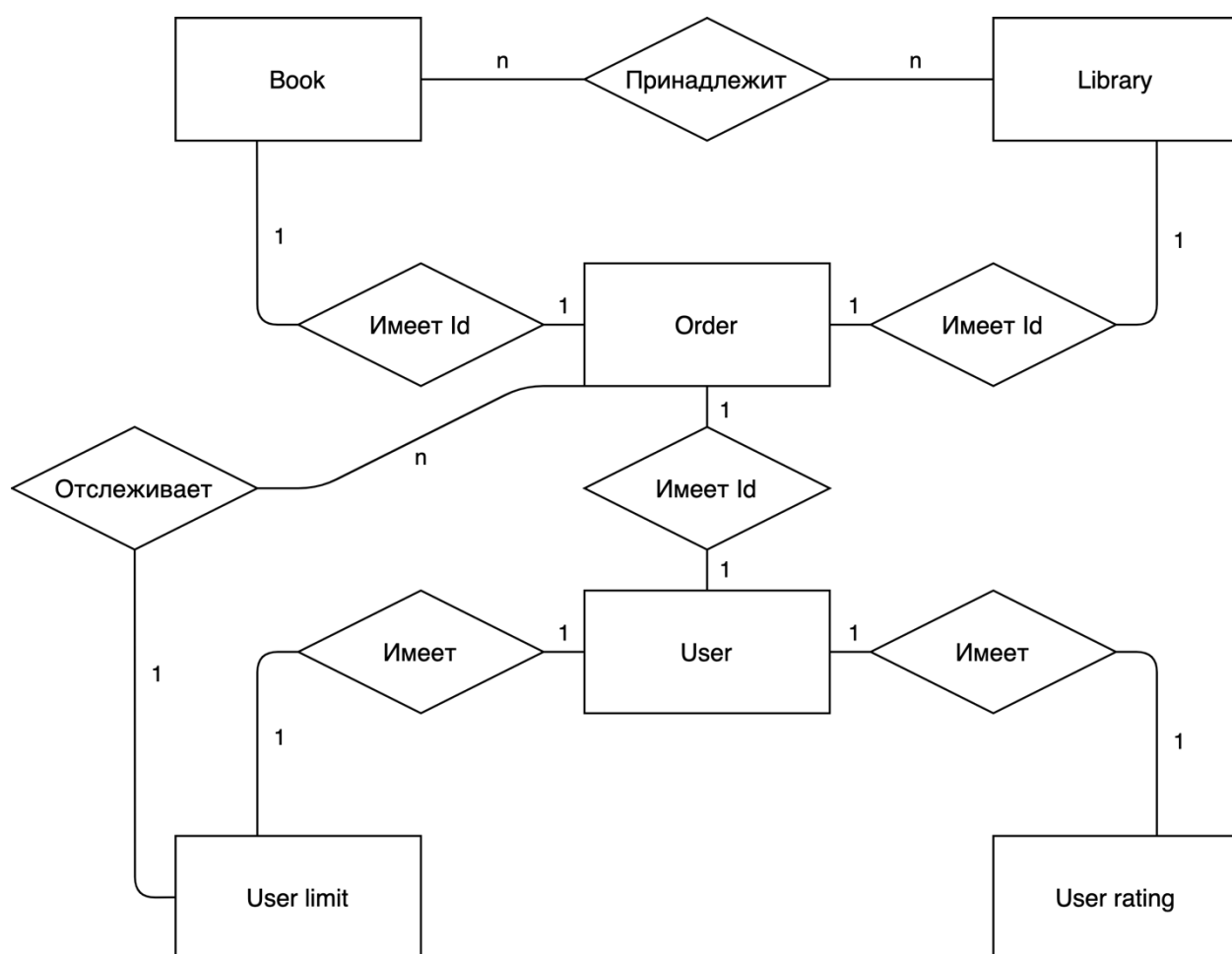


Рисунок 8 – ER-диаграмма концептуальной модели данных системы

На следующей стадии проектирования, добавив в схему данных атрибуты сущностей, получаем схему базы данных, которая изображена на рисунке 9.

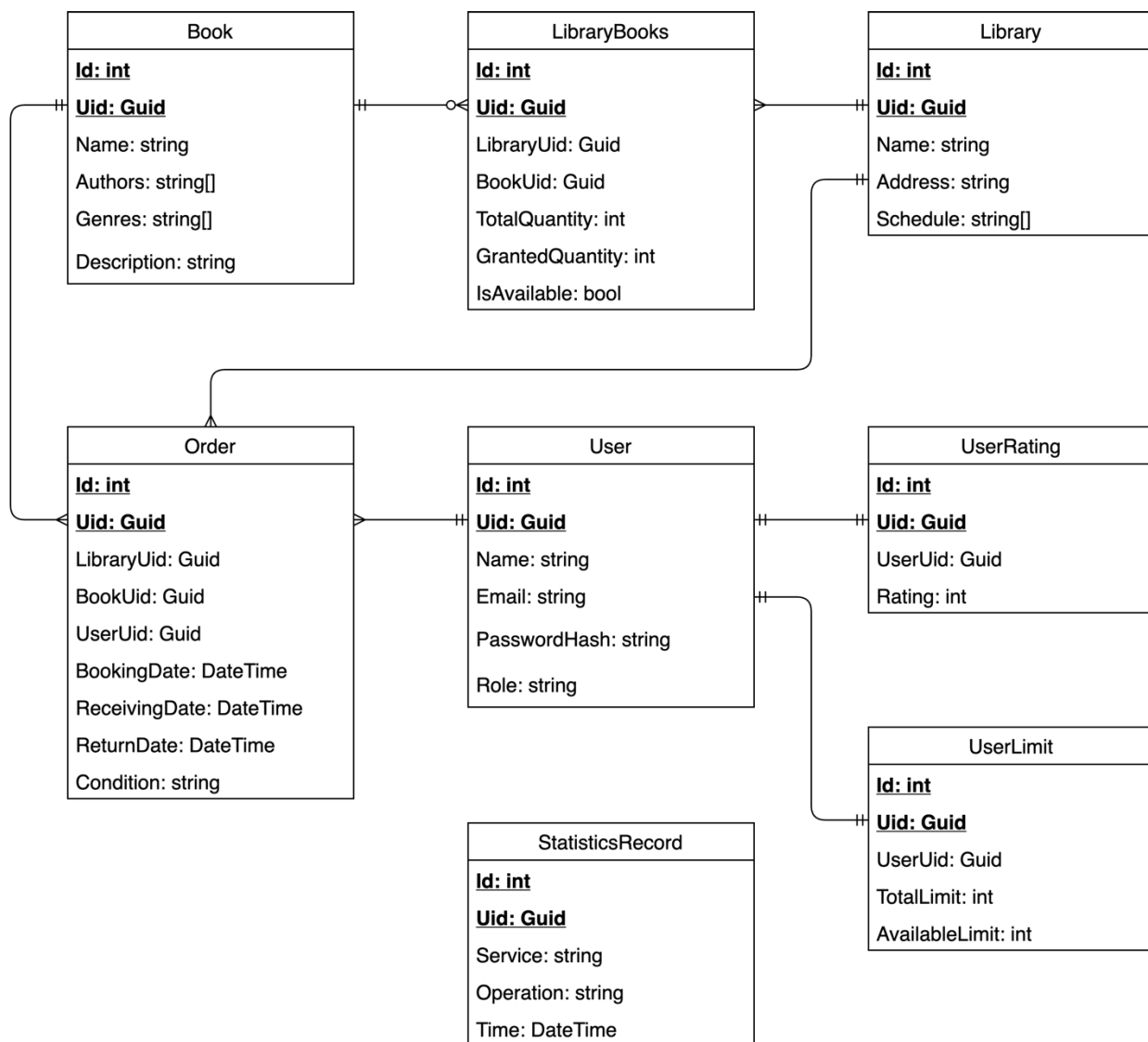


Рисунок 9 – ER-диаграмма базы данных системы

2.5. Спецификации таблиц

Спецификация таблицы User

Таблица User предназначена для хранения информации о пользователях.

Имя атрибута	Тип атрибута	Описание атрибута
Id	Public: int	Внутренний идентификатор пользователя
Uid	Public: Guid	Внешний идентификатор пользователя
Name	Public: string	Имя пользователя
Email	Public: string	Электронная почта пользователя
PasswordHash	Public: string	Хэш пароля пользователя
Role	Public: string	Роль пользователя (читатель или администратор)

Спецификация таблицы Book

Таблица Book предназначена для хранения информации о книгах.

Имя атрибута	Тип атрибута	Описание атрибута
Id	Public: int	Внутренний идентификатор книги
Uid	Public: Guid	Внешний идентификатор книги
Name	Public: string	Название книги
Authors	Public: string[]	Авторы книги
Genres	Public: string[]	Жанры книги
Description	Public: string	Описание книги

Спецификация таблицы Library

Таблица Library предназначена для хранения информации о библиотеках.

Имя атрибута	Тип атрибута	Описание атрибута
Id	Public: int	Внутренний идентификатор библиотеки
Uid	Public: Guid	Внешний идентификатор библиотеки
Name	Public: string	Название библиотеки
Address	Public: string	Адрес библиотеки
Schedule	Public: string[]	Расписание работы библиотеки

Спецификация таблицы LibraryBooks

Таблица LibraryBooks предназначена для хранения информации о библиотечных фондах.

Имя атрибута	Тип атрибута	Описание атрибута
Id	Public: int	Внутренний идентификатор сущности
Uid	Public: Guid	Внешний идентификатор сущности
LibraryUid	Public: Guid	Идентификатор, ассоциированный с конкретной библиотекой
BookUid	Public: Guid	Идентификатор, ассоциированный с конкретной книгой
TotalQuantity	Public: int	Количество экземпляров конкретной книги на балансе конкретной библиотеки
GrantedQuantity	Public: int	Количество бронированных и/или выданных экземпляров конкретной книги в конкретной библиотеке
IsAvailable	Public: bool	Флаг, указывающий на доступность книги для бронирования

Спецификация таблицы Order

Таблица Order предназначена для хранения информации о бронированиях и возвратах.

Имя атрибута	Тип атрибута	Описание атрибута
Id	Public: int	Внутренний идентификатор заказа
Uid	Public: Guid	Внешний идентификатор заказа
LibraryUid	Public: Guid	Идентификатор, ассоциированный с конкретной библиотекой
BookUid	Public: Guid	Идентификатор, ассоциированный с конкретной книгой
UserId	Public: Guid	Идентификатор, ассоциированный с конкретным пользователем
BookingDate	Public: DateTime	Время совершения бронирования
ReceivingDate	Public: DateTime	Время получения бронированной книги на руки

ReturnDate	Public: DateTime	Время возврата книги в библиотеку
Condition	Public: string	Состояние книги после возврата

Спецификация таблицы UserRating

Таблица UserRating предназначена для хранения информации о пользовательском рейтинге.

Имя атрибута	Тип атрибута	Описание атрибута
Id	Public: int	Внутренний идентификатор сущности
Uid	Public: Guid	Внешний идентификатор сущности
UserUid	Public: Guid	Идентификатор, ассоциированный с конкретным пользователем
Rating	Public: int	Рейтинг пользователя

Спецификация таблицы UserLimit

Таблица UserLimit предназначена для хранения информации о лимите книг, доступных пользователю для бронирования/получения на руки

Имя атрибута	Тип атрибута	Описание атрибута
Id	Public: int	Внутренний идентификатор сущности
Uid	Public: Guid	Внешний идентификатор сущности
UserUid	Public: Guid	Идентификатор, ассоциированный с конкретным пользователем
TotalLimit	Public: int	Максимальное число книг, доступных пользователю для бронирования/получения на руки
AvailableLimit	Public: int	Число книг, доступное пользователю в данный момент для бронирования/получения

Спецификация таблицы StatisticsRecord

Таблица **StatisticsRecord** предназначена для хранения статистики функционирования системы.

Имя атрибута	Тип атрибута	Описание атрибута
Id	Public: int	Внутренний идентификатор записи операции
Uid	Public: Guid	Внешний идентификатор записи операции
Service	Public: string	Название сервиса, ассоциированного с данной записью
Operation	Public: string	Операция, ассоциированная с данной записью
Time	Public: DateTime	Время совершения операции

2.6. Структура сервисов

На основании разработанной схемы данных в данном разделе устанавливается соответствие сущностей и сервисов, отвечающих за работу с этими сущностями. Были установлены следующие соответствия:

- Сущность **User** – сервис авторизации **AuthService**;
- Сущности **Book** – книжный сервис **BookService**;
- Сущности **Library**, **LibraryBooks**, **Order** – библиотечный сервис **LibraryService**;
- Сущность **UserRating** – сервис формирования читательского рейтинга **RatingService**;
- Сущность **UserLimit** – сервис контроля возврата книг **ReturnService**;
- Сущность **StatisticsRecord** – сервис статистики **StatisticsService**.

2.7. Диаграммы последовательности действий

Для описания поведения компонентов системы на единой оси времени

используются диаграммы последовательности действий, при помощи которых можно описать последовательность действий для каждого прецедента, необходимую для достижения цели. Ниже приведены диаграммы для следующих действий в системе:

- рисунок 10 – регистрация нового администратора другим администратором;
- рисунок 11 – удаление существующей книги администратором системы;
- рисунок 12 – добавление нового бронирования книги читателем;
- рисунок 13 – закрытие существующего заказа администратором.

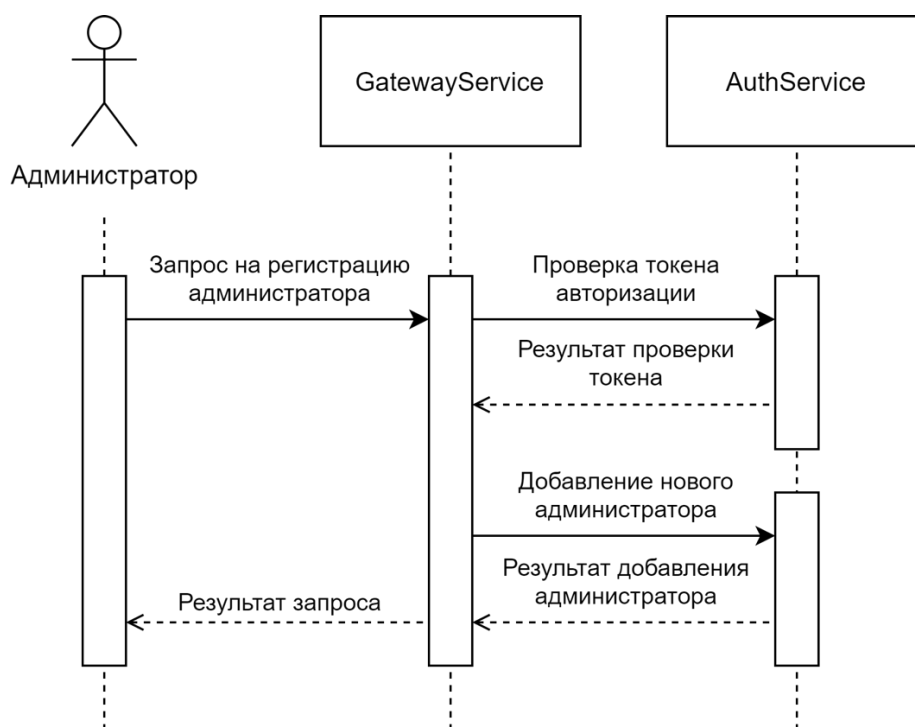


Рисунок 10 – Диаграмма последовательности действий при регистрации нового администратора другим администратором

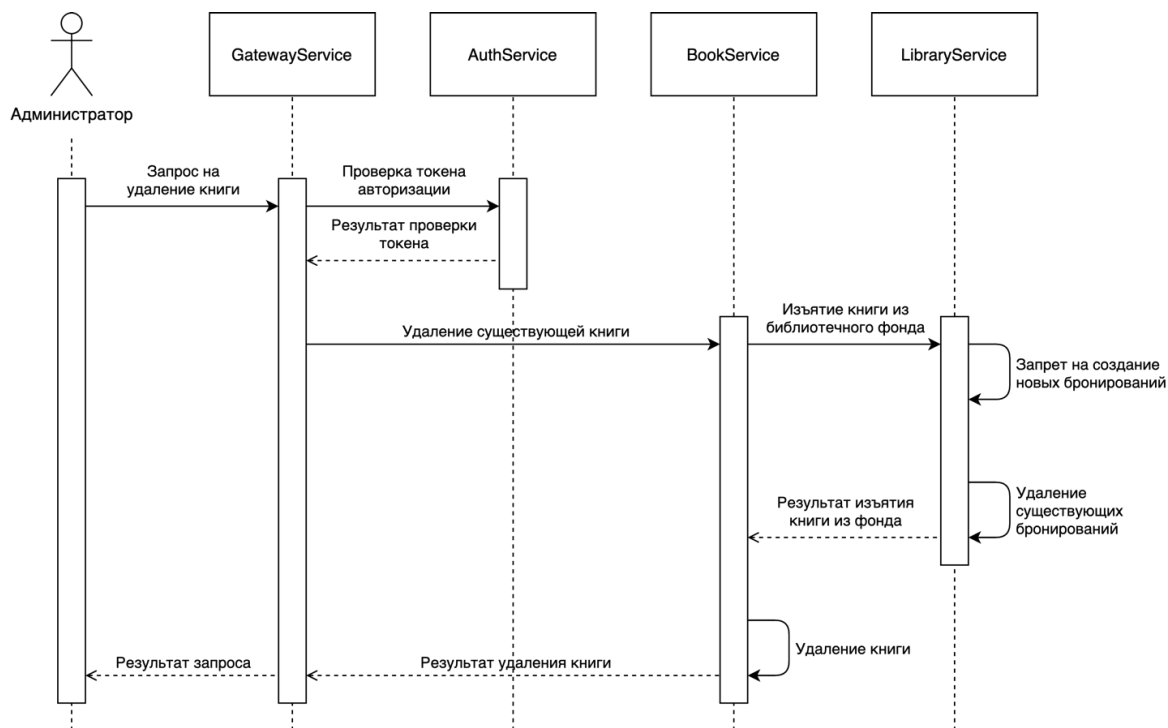


Рисунок 11 – Диаграмма последовательности действий при удалении существующей книги администратором системы

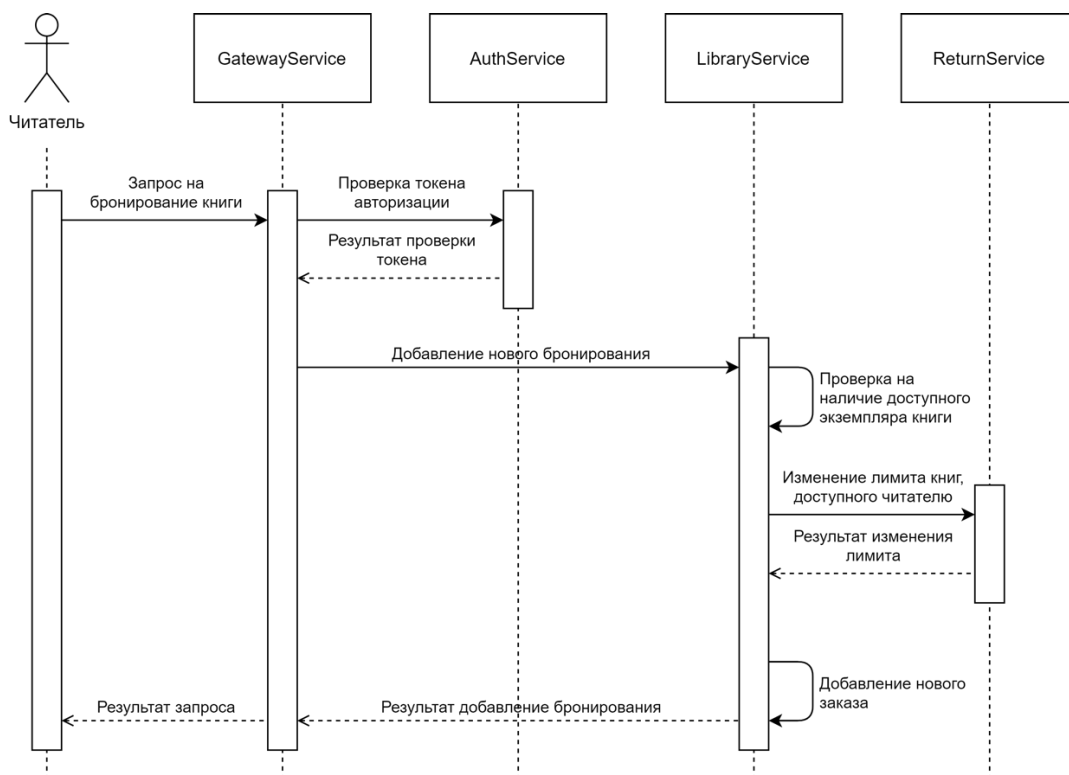


Рисунок 12 – Диаграмма последовательности действий при добавлении нового бронирования книги читателем

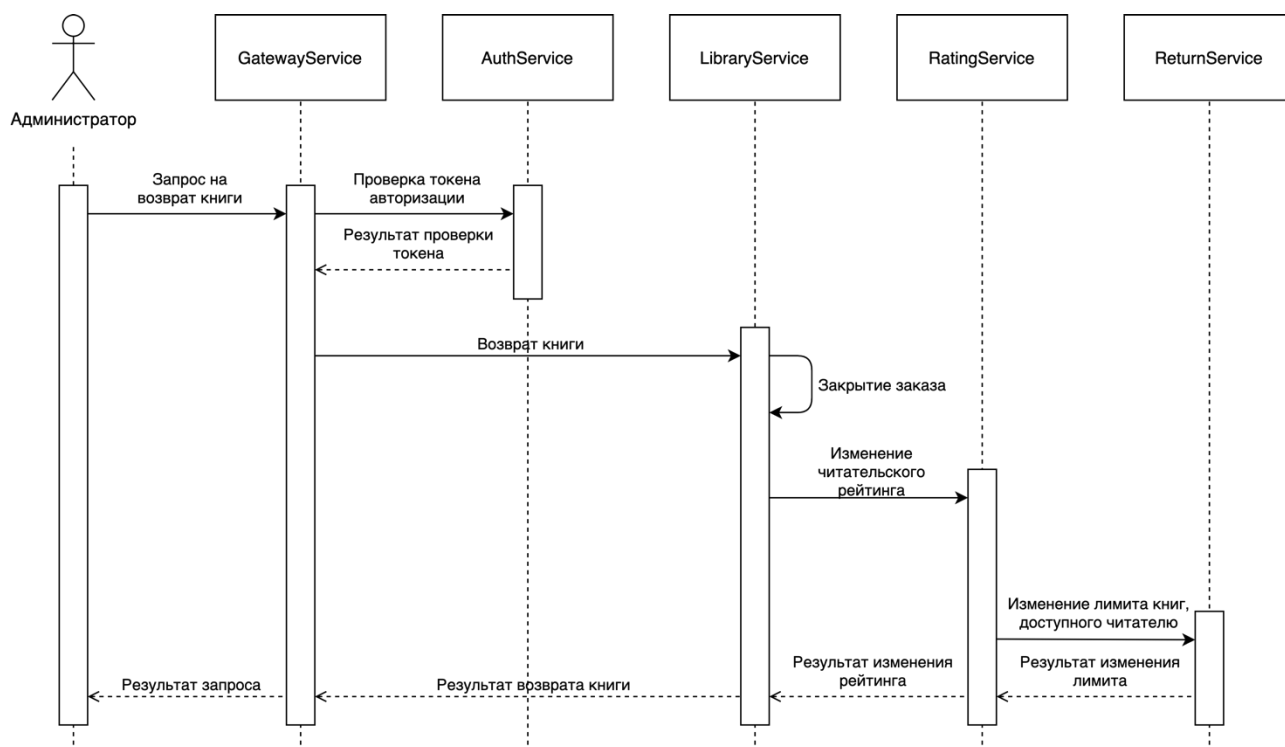


Рисунок 13 – Диаграмма последовательности действий при закрытии существующего заказа администратором

2.8. Диаграмма потоков данных

Рассматриваемая система предполагает распределенное хранение данных. Все данные системы предполагают хранение в единой базе данных, хранилищами данных являются таблицы. Диаграмма потоков данных, представленная на рисунке 14, отображает модель информационной системы с точки зрения хранения, передачи и обработки данных во время обработки запроса читателя на бронирование книги.

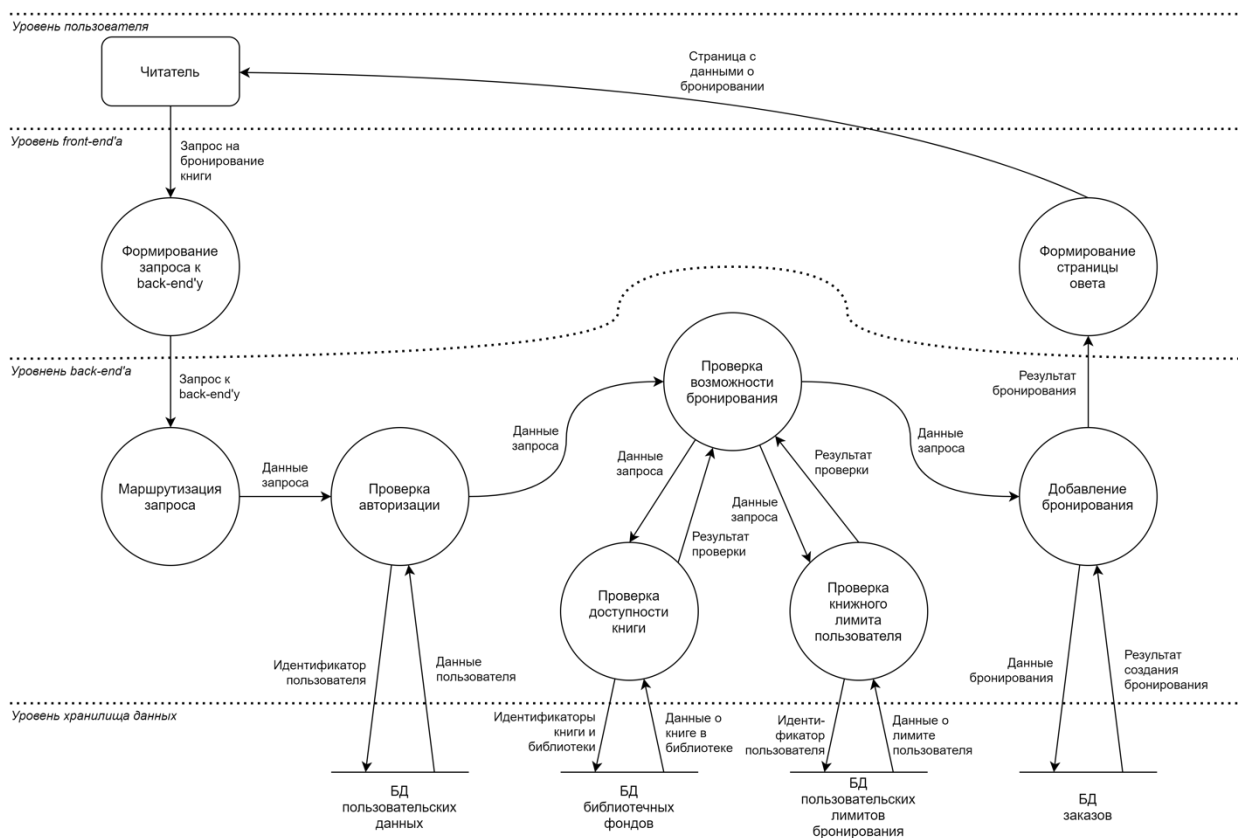


Рисунок 14 – Диаграмма потоков данных при создании читателем нового бронирования

2.9. Архитектура системы

Основополагающей идеей построения программной архитектуры является идея снижения сложности системы путём абстракции и разграничения полномочий. В данном проекте каждая функциональная область реализована посредством собственного микросервиса. Этот подход позволяет бороться со сложностью современных систем. Архитектура системы призвана показать способ развертывания системы во внешних средах. На рисунке 15 представлена архитектура системы, которая показывает размещение элементов системы на физических носителях и способах их взаимодействия, то есть, указаны протоколы, по которым происходит информационный обмен.

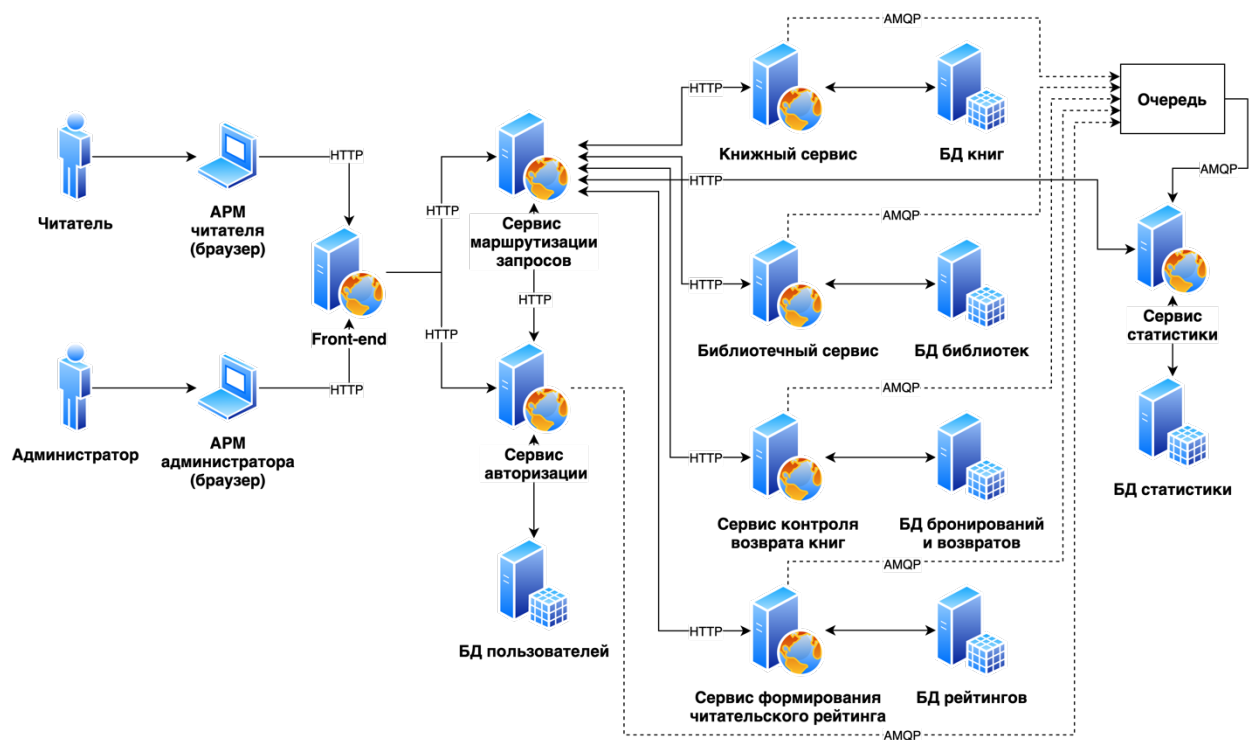


Рисунок 15 – Архитектура разрабатываемой системы

3. Технологический раздел

В этом разделе описаны технические средства реализации программного обеспечения.

3.1. Выбор и обоснование технологий разработки

Для разработки системы был выбран современный диалект языка Lisp – Clojure и его реализация для веб-программирования на стороне клиента – ClojureScript. Этот выбор обусловлен следующими причинами:

- философия языка «код как данные» (гомоиконность), позволяющая использовать развитую систему лисп-макросов для модификации исходного кода как на стороне клиента, так и на стороне сервера;
- динамическая, интерактивная разработка в REPL-цикле;
- минималистичный синтаксис и строго ограниченный набор абстракций;
- функциональный язык программирования;
- расширенный набор стандартных структур данных относительно языка Lisp;
- иммутабельность всех основных структур данных;
- возможность переиспользования программного кода между клиентом и сервером;
- поддержка edn-нотации [2] для представления данных, существенно более мощной, чем json, и более гибкой, чем yaml.

В качестве СУБД использовалась свободно распространяемая PostgreSQL.

Для тестирования Web API использовался Postman.

В качестве системы управления версиями использовался Git.

Для контейнеризации использовался Docker.

Для документации серверного API использовался стандарт OpenAPI версии 2.

API сервера способно принимать запросы и отдавать ресурсы в форматах json и edn. Межсервисное взаимодействие осуществляется в родном для Clojure формате edn, что снижает накладные расходы на сериализацию ресурсов.

3.2. Обеспечение отказоустойчивости

Для обеспечения отказоустойчивости применялись следующие подходы:

- 1) Повтор запросов (retry pattern) – повтор запроса ресурса в случае получения ошибки.
- 2) Паттерн автоматический выключатель [3] (Circuit Breaker pattern) – шаблон **Circuit Breaker** обеспечивает стабильность, пока система восстанавливается после сбоя и снижает влияние на производительность.
- 3) Деградация функциональности [4] – функциональной деградацией и называется снижение ее функциональных возможностей без увеличения числа полностью отказавших модулей системы.

Паттерн **Circuit Breaker** предотвращает попытки приложения выполнить операцию, которая скорее всего завершится неудачно, что позволяет продолжить работу дальше не тратя важные ресурсы, пока известно, что проблема не устранена. Приложение должно быстро принять сбой операции и обработать его.

Он также позволяет приложению определять, была ли устранена неисправность. Если проблема устранена, приложение может попытаться вызвать операцию снова.

В паттерне **Circuit Breaker** [3] (автоматический выключатель) различают 3 состояния:

1) **Closed:** Запрос приложения перенаправляется на операцию. Прокси-сервер ведет подсчет числа недавних сбоев, и если вызов операции не завершился успешно, прокси-сервер увеличивает это число. Если число недавних сбоев превышает заданный порог в течение заданного периода времени, прокси-сервер переводится в состояние **Открытый**. На этом этапе прокси-сервер запускает таймер времени ожидания, и по истечении времени этого таймера прокси-сервер переводится в состояние **Half-Open**. Цель применения этого паттерна — дать системе время на исправление ошибки, которая вызвала сбой, прежде чем разрешить приложению попытаться выполнить операцию еще раз. Назначение таймера — дать сервису время для решения проблемы, прежде чем разрешить приложению попытаться выполнить операцию еще раз.

2) **Open:** запрос от приложения немедленно завершает с ошибкой и исключение возвращается в приложение.

3) **Half-Open:** Ограниченному числу запросов от приложения разрешено проходить через операцию и вызывать ее. Если эти запросы выполняются успешно, предполагается, что ошибка, которая ранее вызывала сбой, устранена, а автоматический выключатель переходит в состояние **Закрытый** (счетчик сбоев сбрасывается). Если какой-либо запрос завершается со сбоем, автоматическое выключение предполагает, что неисправность все еще присутствует, поэтому он возвращается в состояние **Открытый** и перезапускает таймер времени ожидания, чтобы дать системе дополнительное время на восстановление после сбоя. Состояние **Half-Open** помогает предотвратить быстрый рост запросов к сервису. Т.к. после начала работы сервиса, некоторое время он может быть способен обрабатывать ограниченное число запросов до полного восстановления.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы был создан программный продукт, предоставляющий возможности просмотра и ведения реестра библиотек и их книжных фондов, а также создания читателями бронирований книг для очного получения и последующего ведения заказов администраторами библиотек.

Были выполнены все поставленные цели, а именно:

- спроектирована библиотечная система с использованием сервис-ориентированного подхода;
- определена модель данных, подходящая к поставленной задаче;
- выбрана и обоснована технология реализации;
- разработано веб-приложение, реализующее заявленную функциональность;
- выполнить развёртывание системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Обзор автоматизированных библиотечных систем. [Электронный ресурс] URL: <https://lala.lanbook.com/obzor-avtomatizirovannyh-bibliotechnyh-sistem-4-ochen-raznyh-produkta>. (Дата обращения: 20.04.2021)
2. Extensible data notation. [Электронный ресурс] URL: <https://github.com/edn-format/edn>. (Дата обращения: 20.04.2021)
3. Circuit Breaker паттерн. [Электронный ресурс] URL: <https://bool.dev/blog/detail/circuit-breaker-pattern>. (Дата обращения: 20.04.2021)
4. А. А. Тарасов. Стратегии функциональной перестройки отказоустойчивых информационных систем при различных видах деградации. [Электронный ресурс] URL: <https://bit.mephi.ru/index.php/bit/article/download/461/466>. (Дата обращения: 20.04.2021)