



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 (ИУ)»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии (ИУ7)»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

«Система аренды книг в библиотеках»

Студент ИУ7-22М
(Группа)

(Подпись, дата)

И. А. Цветков
(И. О. Фамилия)

Руководитель курсовой работы

(Подпись, дата)

А. А. Ступников
(И. О. Фамилия)

2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Аналитическая часть	6
1.1 Разбор аналогов	6
1.2 Описание системы	8
1.3 Функциональные требования к системе с точки зрения пользо- вателя	9
1.4 Входные данные	10
1.5 Выходные параметры	11
1.6 Состав системы	14
1.6.1 Фронтенд	15
1.6.2 Сервис-координатор	15
1.6.3 Сервис регистрации и авторизации	16
1.6.4 Сервис библиотек	16
1.6.5 Сервис рейтинга	18
1.6.6 Сервис аренды	19
1.6.7 Сервис статистики	19
1.6.8 Сервис kafka	20
1.6.9 Сервис consumer	20
1.6.10 Сервис consumer	20
1.6.11 Сервис zookeeper	21
1.7 Требования к программной реализации	22
1.8 Функциональные требования к подсистемам	22

ВВЕДЕНИЕ

В современном мире, стремительное развитие технологий оказывает значительное влияние на повседневную жизнь человека. Одной из ключевых областей, где технологии играют важную роль, является доступ к информации и культурным ресурсам. Библиотеки, как центры знаний и культурного наследия, остаются важным элементом образовательной и интеллектуальной среды. Однако традиционная модель функционирования библиотек сталкивается с рядом вызовов: быстрый рост цифровых ресурсов, изменение пользовательских предпочтений, а также необходимость оптимизации процесса обслуживания и предоставления доступа к ним.

Одной из современных тенденций является популяризация аренды книг через цифровые платформы, что предоставляет удобство и упрощает доступ к библиотечным ресурсам. Это особенно актуально в условиях городской жизни, где время и доступность ресурсов играют важную роль. Разработка веб-сайтов и мобильных приложений для аренды книг может значительно улучшить взаимодействие пользователей с библиотеками, увеличивая их посещаемость и привлекая новые категории пользователей.

Примером успешного применения подобных технологий является проект цифровой платформы «Московская электронная библиотека», которая предоставляет доступ к сотням тысяч книг, журналов и других материалов. Пользователи могут брать книги в аренду онлайн и читать их как в электронном, так и в традиционном формате. Это показывает, что библиотеки не теряют своей актуальности, а, напротив, приспосабливаются к новым условиям и сохраняют свою роль в обществе.

Актуальность данной темы обусловлена также изменением культурных и потребительских привычек пользователей, которые все больше предпочитают онлайн-сервисы для удовлетворения своих потребностей. В условиях пандемии COVID-19 был зафиксирован резкий рост интереса к удаленным библиотечным услугам, что подчеркнуло необходимость развития и совершенствования цифровых решений для библиотек.

Данный курсовой проект нацелен на разработку веб-сайта аренды книг в библиотеках города, который будет решать задачи оптимизации работы библиотек, упрощения доступа к книгам. В рамках проекта будет создан интерфейс, удобный для поиска и аренды книг, а также для управления

личными данными пользователей и их библиотечными активностями.

1 Аналитическая часть

1.1 Разбор аналогов

Для успешной разработки веб-сайта аренды книг в библиотеках города важно проанализировать существующие аналогичные решения. Это позволит выделить сильные и слабые стороны уже реализованных проектов и использовать полученные данные для создания функционального и удобного ресурса. Рассмотрим несколько популярных примеров аналогов.

1. **ЛитРес: Библиотека** Это один из крупнейших российских онлайн-сервисов, предоставляющий доступ к электронной библиотеке, включая аренду книг для библиотек. Он активно сотрудничает с государственными библиотеками, предлагая пользователям доступ к бесплатным и платным материалам.

Плюсы:

- Широкий выбор книг. Пользователи могут арендовать книги из огромной коллекции, включая новинки.
- Удобный интерфейс. Сайт и мобильное приложение имеют интуитивно понятный дизайн и структуру.
- Синхронизация с библиотеками. Возможность брать книги в аренду через конкретные библиотеки.
- Многоформатность. Книги доступны как в электронном, так и в аудиоформатах.

Минусы:

- Ограниченный бесплатный доступ. Большая часть контента платная, что может ограничивать круг пользователей.
- Отсутствие персонализации. Недостаточно индивидуальных рекомендаций и настроек под пользователя.
- Нет работы с физическими книгами. Платформа ориентирована в основном на электронные ресурсы, что не всегда удовлетворяет потребности пользователей, предпочитающих бумажные издания.

2. **Московская электронная библиотека** Государственный проект, предоставляющий жителям Москвы бесплатный доступ к библиотечным фондам, включая электронные книги и периодику.

Плюсы:

- Бесплатный доступ. Пользователи могут бесплатно брать в аренду книги, что делает услугу доступной для всех.
- Интеграция с городскими библиотеками. Прямое взаимодействие с реальными библиотеками города, что упрощает получение информации о наличии книг.
- Широкий выбор. Библиотека содержит множество произведений различной тематики, включая учебную литературу.

Минусы:

- Устаревший интерфейс. Сайт выглядит морально устаревшим, что затрудняет навигацию и снижает пользовательский опыт.
- Малый функционал. Нет удобных инструментов для личного кабинета, отсутствуют продвинутые функции поиска и рекомендации.
- Отсутствие интеграции с физическими книгами. Хотя проект предоставляет электронные ресурсы, нет механизма аренды бумажных книг.

3. **OverDrive (международный аналог)** Это популярная платформа для аренды книг, используемая библиотеками по всему миру. Она позволяет пользователям брать в аренду как электронные, так и аудиокниги через местные библиотеки.

Плюсы:

- Интернациональность. Поддержка множества языков и сотрудничество с библиотеками по всему миру.
- Широкий ассортимент книг. Включает не только художественную литературу, но и учебные и исследовательские работы.

- Интеграция с мобильными устройствами. Поддержка мобильных приложений для различных платформ, что упрощает доступ к арендованным материалам.

Минусы:

- Ограниченное количество копий. Каждая библиотека ограничена количеством цифровых копий, что может привести к дефициту книг.
- Требование регистрации в местных библиотеках. Пользователю нужно быть зарегистрированным в библиотеке, чтобы получить доступ к ресурсам.
- Проблемы с доступом к физическим книгам. Платформа ориентирована на электронные и аудиокниги, а не на физические издания.

Изучив плюсы и минусы существующих решений, можно выделить несколько ключевых моментов для создания более эффективного веб-сайта аренды книг в библиотеках города.

1. Современный и удобный интерфейс. Нужно избегать устаревших интерфейсов и предложить интуитивно понятный и современный дизайн.
2. Бесплатный доступ и интеграция с библиотеками. Необходимо предоставить пользователям возможность арендовать книги бесплатно через городские библиотеки.

Эти элементы помогут создать ресурс, способный конкурировать с существующими решениями и удовлетворить требования современных пользователей.

1.2 Описание системы

Разрабатываемый портал должен представлять собой систему для аренды книг в библиотеках города.

Если пользователь хочет оформить заказ, то ему нужно пройти регистрацию, указав следующую информацию: ф. и., номер телефона, адрес электронной почты, пароль. Для неавторизованных пользователей доступен только просмотр общей информации сайта: списка библиотек и книг в них.

1.3 Функциональные требования к системе с точки зрения пользователя

Портал должен обеспечивать реализацию следующих функций.

1. Система должна обеспечивать регистрацию и авторизацию пользователей с валидацией вводимых данных.
2. Аутентификация пользователей.
3. Разделение всех пользователей на 3 роли:
 - неавторизованный пользователь (гость);
 - авторизованный пользователь (пользователь);
 - администратор.
4. Предоставление возможностей **гостю, пользователю, администратору** представленных в таблице 1.1.

Таблица 1.1 – Функции пользователей

Гость	<ol style="list-style-type: none"> 1. Просмотр списка библиотек (включая фильтрацию по городу); 2. Просмотр списка книг в выбранной библиотеке; 3. Регистрация в системе; 4. Авторизация в системе.
Пользователь	<ol style="list-style-type: none"> 1. Авторизация в системе; 2. Просмотр списка библиотек (включая фильтрацию по городу); 3. Просмотр списка книг в выбранной библиотеке; 4. Получение информации о данных текущего аккаунта; 5. Просмотр всех своих арендованных книг во всех библиотеках с фильтрацией по статусу аренды 6. Получение детальной информации по конкретной аренде на имя текущего пользователя; 7. Оформление аренды книги на имя авторизованного пользователя; 8. Отмена аренды (возврат) на имя авторизованного пользователя.
Администратор	<ol style="list-style-type: none"> 1. Функции пользователя; 2. Просмотр статистики по сайту.

1.4 Входные данные

Входные параметры системы представлены в таблице 1.2.

Таблица 1.2 – Входные данные

Сущность	Входные данные
Регистрация пользователя	<ol style="list-style-type: none"> 1. <i>фамилия</i> не более 256 символов; 2. <i>имя</i> не более 256 символов; 3. <i>логин</i> не более 256 символов; 4. <i>пароль</i> не более 128 символов; 5. <i>номер телефона</i> в формате (+7XXXXXXXXXX); 6. <i>роль</i> администратор или пользователь; 7. <i>электронная почта</i> в формате (*@*.*).
Аутентификация пользователя	<ol style="list-style-type: none"> 1. <i>логин</i> не более 256 символов; 2. <i>пароль</i> не более 128 символов.
Аренда книги	<ol style="list-style-type: none"> 1. <i>идентификатор книги</i>; 2. <i>идентификатор библиотеки</i>; 3. <i>дата возврата</i> в формате ДД/ММ/ГГГГ.
Возврат книги	<ol style="list-style-type: none"> 1. <i>идентификатор аренды</i>; 2. <i>состояние книги</i> прекрасное / хорошее / плохое; 3. <i>дата возврата</i> в формате ДД/ММ/ГГГГ.
Фильтр и пагинация библиотек	<ol style="list-style-type: none"> 1. <i>город</i> не более 256 символов; 2. <i>номер страницы</i> не менее 1; 3. <i>объектов на странице</i> не менее 1.
Фильтр и пагинация книг	<ol style="list-style-type: none"> 1. <i>показать закончившиеся</i> да / нет; 2. <i>номер страницы</i> не менее 1; 3. <i>объектов на странице</i> не менее 1.
Фильтр и пагинация аренд	<ol style="list-style-type: none"> 1. <i>статус аренды</i> в аренде / возвращена вовремя / возвращена после срока; 2. <i>номер страницы</i> не менее 1; 3. <i>объектов на странице</i> не менее 1.

1.5 Выходные параметры

Выходными параметрами системы являются web-страницы. В зависимости от запроса и текущей роли пользователя они содержат следующую информацию (таблица 1.3).

Таблица 1.3 – Выходные параметры

Гость	<p>1. Список библиотек:</p> <ul style="list-style-type: none"> • <i>название;</i> • <i>адрес;</i> • <i>город.</i>
	<p>2. Список книг:</p> <ul style="list-style-type: none"> • <i>название;</i> • <i>автор;</i> • <i>жанр;</i> • <i>состояние;</i> • <i>количество в библиотеке.</i>
	<p>3. О сайте:</p> <ul style="list-style-type: none"> • <i>общая информация о сайте;</i> • <i>правила сайта;</i> • <i>контактная информация поддержки.</i>
Пользователь	<p>1. Список библиотек:</p> <ul style="list-style-type: none"> • <i>название;</i> • <i>адрес;</i> • <i>город.</i>
	<p>2. Список книг:</p> <ul style="list-style-type: none"> • <i>название;</i> • <i>автор;</i> • <i>жанр;</i> • <i>состояние;</i> • <i>количество в библиотеке.</i>
	<p>3. О сайте:</p> <ul style="list-style-type: none"> • <i>общая информация о сайте;</i> • <i>правила сайта;</i> • <i>контактная информация поддержки.</i>

Администратор	<p>4. Детальная информация о пользователе, вошедшем в систему;</p> <ul style="list-style-type: none"> ● фамилия; ● имя; ● логин; ● номер телефона; ● рейтинг число от 0 до 100, характеризующее количество книг, которое может взять пользователь; ● электронная почта.
	<p>5. Список взятых в аренду книг пользователя, вошедшего в систему:</p> <ul style="list-style-type: none"> ● библиотека, в которой арендована книга в соответствии с пунктом 1; ● арендованная книга в соответствии с пунктом 2; ● дата взятия книги в аренду; ● дата возврата книги; ● статус аренды.
	<p>1. Список библиотек:</p> <ul style="list-style-type: none"> ● название; ● адрес; ● город.
	<p>2. Список книг:</p> <ul style="list-style-type: none"> ● название; ● автор; ● жанр; ● состояние; ● количество в библиотеке.
	<p>3. О сайте:</p> <ul style="list-style-type: none"> ● общая информация о сайте; ● правила сайта; ● контактная информация поддержки.

<p>4. Детальная информация о пользователе, вошедшем в систему;</p> <ul style="list-style-type: none"> • фамилия; • имя; • логин; • номер телефона; • рейтинг число от 0 до 100, характеризующее количество книг, которое может взять пользователь; • роль в системе; • электронная почта.
<p>5. Список взятых в аренду книг пользователя, вошедшего в систему:</p> <ul style="list-style-type: none"> • библиотека, в которой арендована книга в соответствии с пунктом 1; • арендованная книга в соответствии с пунктом 2; • дата взятия книги в аренду; • дата возврата книги; • статус аренды.
<p>6. Статистика по portalу, собранная через сервис статистики:</p> <ul style="list-style-type: none"> • идентификатор; • метод запроса GET/POST/PATCH/DELETE/OPTIONS; • url запроса; • числовой статус выполнения запроса; • время выполнения запроса в формате ДД/ММ/ГГГГ (ЧЧ:ММ).

1.6 Состав системы

Система будет состоять из фронтенда и 9 подсистем:

- сервис-координатор;
- сервис регистрации и авторизации;
- сервис библиотек;
- сервис рейтинга;
- сервис аренды;
- сервис статистики;
- сервис kafka;

- сервис consumer;
- сервис zookeeper.

1.6.1 Фронтенд

Фронтенд – принимает запросы от пользователя по протоколу HTTP и возвращает ответ в виде HTML страниц, файлов стилей и TypeScript.

1.6.2 Сервис-координатор

Сервис-координатор – сервис, который отвечает за координацию запросов внутри системы. Все сервисы портала (кроме сервиса регистрации и авторизации) должны взаимодействовать друг с другом через сервис-координатор, запросы с фронтенда в том числе сначала должны приходить на сервис-координатор, а затем перенаправляться на нужный сервис. При этом сервис-координатор отвечает за следующие действия.

1. получения списка библиотек в городе с пагинацией от сервиса библиотек 1.6.4;
2. получения списка книг в библиотеке с пагинацией от сервиса библиотек 1.6.4;
3. получения списка книг, арендованных пользователем с пагинацией от сервисов библиотек 1.6.4 и аренды 1.6.6;
4. получения рейтинга пользователя от сервиса рейтинга 1.6.5;
5. оформление аренды книги через сервисы библиотек 1.6.4 и аренды 1.6.6 с учетом рейтинга пользователя из сервиса рейтинга 1.6.5 (книг в аренде может быть не больше, чем рейтинг пользователя);
6. возврат книги через сервисы библиотек 1.6.4 и аренды 1.6.6 и с изменением данных в сервисе рейтинга 1.6.5 (если книга возвращена в более плохом состоянии и/или позже заявленного срока возврата, то за каждое снимается 10 рейтинга, иначе прибавляется 1 рейтинг).

1.6.3 Сервис регистрации и авторизации

Сервис регистрации и авторизации отвечает за следующие действия.

1. Регистрацию нового пользователя;
2. Аутентификацию пользователя;
3. Авторизацию пользователя;
4. Получение данных пользователей;
5. Изменение данных о пользователе;
6. Удаление пользователя.

Взаимодействие сервиса регистрации и авторизации с остальными сервисами должно осуществляться по протоколу OpenID Connect. Сервис регистрации и авторизации в своей работе используют базу данных, которая хранит следующую информацию:

- Пользователь:
 - *уникальный идентификатор*;
 - *логин*;
 - *имя*;
 - *фамилия*;
 - *захешированный пароль*;
 - *номер телефона*;
 - *электронная почта*;
 - *роль*.

1.6.4 Сервис библиотек

Сервис библиотек реализует следующие функции.

1. Получение списка всех библиотек с фильтрацией и пагинацией;

2. Получение информации о конкретной библиотеке;
3. Создание библиотеки;
4. Изменение библиотеки;
5. Удаление библиотеки;
6. Получение списка всех книг с фильтрацией и пагинацией;
7. Получение информации о конкретной книге;
8. Создание книги;
9. Изменение книги;
10. Удаление книги;
11. Получение списка всех связей книг и библиотек с фильтрацией и пагинацией;
12. Получение информации о конкретной связи библиотеки и книги;
13. Создание связи библиотеки и книги
14. Изменение связи библиотеки и книги
15. Удаление связи библиотеки и книги.

Сервис использует в своей работе базу данных:

— Библиотека:

- *уникальный идентификатор;*
- *название;*
- *город;*
- *адрес.*

— Книга:

- *уникальный идентификатор;*

- *название*;
 - *автор*;
 - *жанр*;
 - *состояние*.
- Связь библиотеки и книги:
- *уникальный идентификатор*;
 - *уникальный идентификатор библиотеки*;
 - *уникальный идентификатор книги*;
 - *количество* книг в библиотеке по связи.

1.6.5 Сервис рейтинга

Сервис рейтинга реализует следующие функции.

1. Получение списка рейтингов всех пользователей с фильтрацией и пагинацией;
2. Получение информации о конкретной рейтинге;
3. Создание рейтинга;
4. Изменение рейтинга;
5. Удаление рейтинга.

Сервис использует в своей работе базу данных:

- Рейтинг:
- *уникальный идентификатор*;
 - *логин пользователя*;
 - *рейтинг* число от 0 до 100.

1.6.6 Сервис аренды

Сервис аренды реализует следующие функции.

1. Получение списка аренд всех пользователей с фильтрацией и пагинацией;
2. Получение информации о конкретной аренде;
3. Создание аренды;
4. Изменение аренды;
5. Удаление аренды.

Сервис использует в своей работе базу данных:

- Аренда:
 - *уникальный идентификатор;*
 - *логин пользователя;*
 - *уникальный идентификатор библиотеки;*
 - *уникальный идентификатор книги;*
 - *статус в аренде / возвращена в срок / возвращена после срока;*
 - *дата взятия книги в аренду;*
 - *дата возврата книги.*

1.6.7 Сервис статистики

Сервис статистики – сервис, который отвечает за запись событий сервиса координатора в базу данных для осуществления возможности быстрого обнаружения, локализации и воспроизведения ошибки в случае её возникновения. Дает возможность получить статистику с пагинацией.

Сервис использует в своей работе базу данных:

- Статистика:
 - *уникальный идентификатор;*
 - *метод запроса GET/POST/PATCH/DELETE/OPTIONS;*

- *url запроса;*
- *числовой статус выполнения запроса;*
- *время выполнения запроса.*

1.6.8 Сервис kafka

Сервис kafka – сервис, который необходим для сервиса статистики для сбора и обработки данных в реальном времени, что позволяет анализировать пользовательскую активность. Kafka поддерживает высокие объёмы данных и легко масштабируется, обеспечивая надёжную работу даже при значительных нагрузках. Благодаря встроенной отказоустойчивости и гарантии доставки сообщений, система статистики не потеряет важные данные при сбоях.

1.6.9 Сервис consumer

Сервис consumer – сервис, который нужен kafka для получения, обработки и анализа данных, поступающих от producer в реальном времени. Kafka действует как посредник, обеспечивая доставку сообщений между различными сервисами, что позволяет consumer-серверам асинхронно получать данные и обрабатывать их по мере поступления. Это важно для поддержания высокой производительности и отказоустойчивости, так как Kafka распределяет нагрузку между несколькими consumer-серверами, помогая избежать перегрузки. Также Kafka гарантирует надёжную доставку сообщений, что позволяет consumer корректно обрабатывать каждое сообщение без риска потери данных. Наконец, она обеспечивает возможность параллельной обработки данных, что ускоряет анализ больших объёмов информации.

1.6.10 Сервис consumer

Сервис consumer – сервис, который нужен kafka для получения, обработки и анализа данных, поступающих от producer в реальном времени. Kafka действует как посредник, обеспечивая доставку сообщений между различными сервисами, что позволяет consumer-серверам асинхронно получать данные и обрабатывать их по мере поступления. Это важно для поддержания высокой производительности и отказоустойчивости, так как Kafka распределяет нагрузку между несколькими consumer-серверами, помогая избежать

перегрузки. Также Kafka гарантирует надёжную доставку сообщений, что позволяет consumer корректно обрабатывать каждое сообщение без риска потери данных. Наконец, она обеспечивает возможность параллельной обработки данных, что ускоряет анализ больших объёмов информации.

1.6.11 Сервис zookeeper

Сервис zookeeper – сервис, который нужен kafka для управления и координации различных компонентов в своей распределённой системе. Вот ключевые задачи, которые решает Zookeeper в Kafka.

1. Координация кластеров: Zookeeper помогает координировать работу брокеров (серверов Kafka) внутри кластера, отслеживая их состояние. Он сообщает Kafka о том, какие узлы доступны и активно работают, обеспечивая бесперебойное взаимодействие между ними.
2. Управление метаданными: Zookeeper хранит важную информацию о топиках, партициях и распределении лидеров партиций среди брокеров. Это нужно для того, чтобы потребители (consumers) и производители (producers) могли эффективно взаимодействовать с нужными данными в кластере.
3. Обнаружение лидера: Zookeeper определяет лидера для каждой партиции Kafka, который отвечает за запись и чтение данных. В случае сбоя одного из брокеров Zookeeper автоматически выбирает нового лидера для партиции, чтобы поддерживать непрерывную работу.
4. Отказоустойчивость: Zookeeper обеспечивает высокую доступность и надёжность кластера Kafka, помогая восстанавливать компоненты после сбоев и поддерживать согласованное состояние всех узлов системы. Управление доступом: Zookeeper управляет доступом клиентов к Kafka и координирует изменения конфигурации, обеспечивая стабильность и безопасность работы кластера.
5. Таким образом, Zookeeper является критически важным компонентом для обеспечения координации, отказоустойчивости и управления Kafka-кластером.

1.7 Требования к программной реализации

1. Требуется использовать СОА (сервис-ориентированную архитектуру) для реализации системы.
2. Система состоит из микросервисов. Каждый микросервис отвечает за свою область логики работы приложения и должны быть запущены изолированно друг от друга.
3. При необходимости, каждый сервис имеет своё собственное хранилище, запросы между базами запрещены.
4. При разработке базы данных необходимо учитывать, что доступ к ней должен осуществляться по протоколу TCP.
5. Необходимо реализовать один web-интерфейс для фронтенда. Интерфейс должен быть доступен через тонкий клиент (браузер).
6. Для межсервисного взаимодействия использовать HTTP (придерживаться RESTful).
7. Выделить Gateway Service как единую точку входа и межсервисной коммуникации. В системе не должно осуществляться горизонтальных запросов.
8. Необходимо предусмотреть авторизацию пользователей через интерфейс приложения.
9. Код хранить на Github, для сборки использовать Github Actions.
10. Каждый сервис должен быть завернут в docker.

1.8 Функциональные требования к подсистемам

Фронтенд – серверное приложение, предоставляет пользовательский интерфейс и внешний API системы, при разработке которого нужно учитывать следующее:

- должен принимать запросы по протоколу HTTP и формировать ответы пользователям в формате HTML;

- в зависимости от типа запроса должен отправлять последовательные запросы в соответствующие микросервисы;
- запросы к микросервисам необходимо осуществлять по протоколу HTTP;
- данные необходимо передавать в формате JSON;
- целесообразно использовать Tailwind для упрощения написания стилей.

Сервис-координатор – это серверное приложение, которое должно отвечать следующим требованиям по разработке:

- обрабатывать запросы в соответствии со своим назначением, описанным в топологии системы;
- принимать и возвращать данные в формате JSON по протоколу HTTP;
- использовать очередь для отложенной обработки запросов (например, при временном отказе одного из сервисов);
- осуществлять деградацию функциональности в случае отказа некритического сервиса (зависит от семантики запроса);
- уведомлять сервис статистики о событиях в системе.

Сервис регистрации и авторизации, сервис библиотек, сервис рейтинга, сервис аренды, сервис статистики – это серверные приложения, которые должны отвечать следующим требованиям по разработке:

- обрабатывать запросы в соответствии со своим назначением, описанным в топологии системы;
- принимать и возвращать данные в формате JSON по протоколу HTTP;
- осуществлять доступ к СУБД по протоколу TCP.

Сервис kafka, сервис consumer, сервис zookeeper – это серверное приложение, которое должно отвечать следующим требованиям по разработке:

- обрабатывать запросы в соответствии со своим назначением, описанным в топологии системы;
- принимать и возвращать данные в формате JSON по протоколу HTTP.