



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 \_\_\_\_\_

**Техническое задание  
на реализацию курсового проекта  
по теме «Библиотечная система для бронирования  
книг» по курсу «Распределённые системы  
обработки информации»**

Студент \_\_\_\_\_ ИУ7-23М \_\_\_\_\_  
(Группа)  
(И.О.Фамилия)

\_\_\_\_\_ Иксарица Н.И. \_\_\_\_\_

Преподаватель \_\_\_\_\_

\_\_\_\_\_ (И.О.Фамилия)

2021 г.

## ТЕХНИЧЕСКОЕ ЗАДАНИЕ

### Глоссарий

Термин	Определение
Front-end	Клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса
Back-end	Программно-аппаратная часть сервиса
REST (RESTful)	архитектурный стиль взаимодействия компонентов распределённого приложения в сети, основанный на протоколе HTTP
Веб-интерфейс	Веб-страница или совокупность веб-страниц, предоставляющая пользовательский интерфейс для взаимодействия с сервисом или устройством посредством протокола HTTP и веб-браузера
Сервис-ориентированная архитектура	Модульный подход к разработке программного обеспечения, основанный на использовании распределённых, слабо связанных заменяемых компонентов, оснащённых стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам
Автоматизированные библиотечные информационные системы (АБИС)	Системы планирования ресурсов предприятий для библиотеки, которые используются для отслеживания библиотечных фондов, от их заказа и приобретения до выдачи посетителям библиотек.

### Введение

Данное техническое задание составлено для разработки курсового проекта «Библиотечная система для бронирования книг» по дисциплине «Распределённые системы обработки информации». Техническое задание выполнено на основе ГОСТ 19.201—78 «ЕСПД. Техническое задание. Требования к содержанию и оформлению».

### Краткое описание предметной области

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.

Данное техническое задание определяет требования к разработке АБИС для поиска и бронирования книг в библиотеках.

### **Существующие аналоги**

Среди аналогов можно отметить АБИС «ИРБИС64», «МегаПро», «Руслан», «МАРК-SQL» [2]. 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.

### **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.

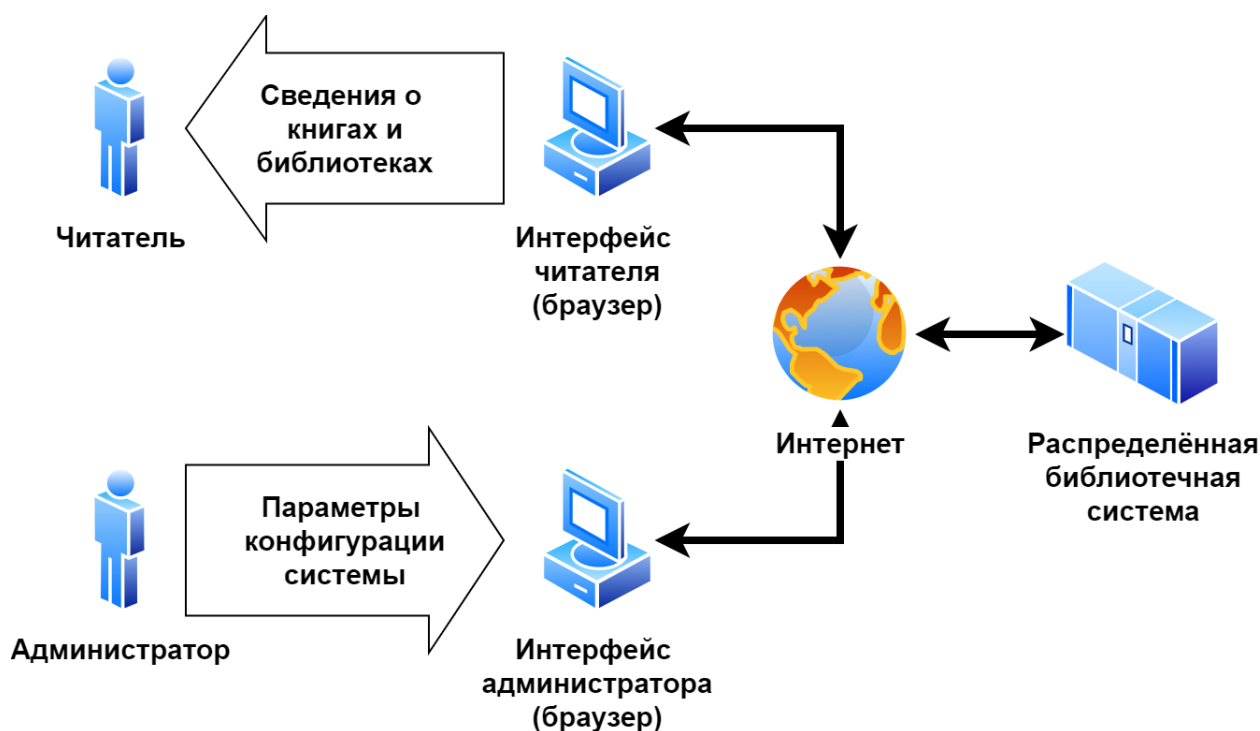


Рисунок 1 – Общая схема предметной области.

## Основания для разработки

Разработка ведется в рамках выполнения лабораторных работ по курсу «Методология программной инженерии» на кафедре «Программное обеспечение ЭВМ и информационные технологии» факультета «Информатика и системы управления» МГТУ им. Н. Э. Баумана.

## 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.

## 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.

### **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.

### **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.

## **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.

## **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.

## **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.

## 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.

## Топология системы

Топология системы приведена на рисунке 2.

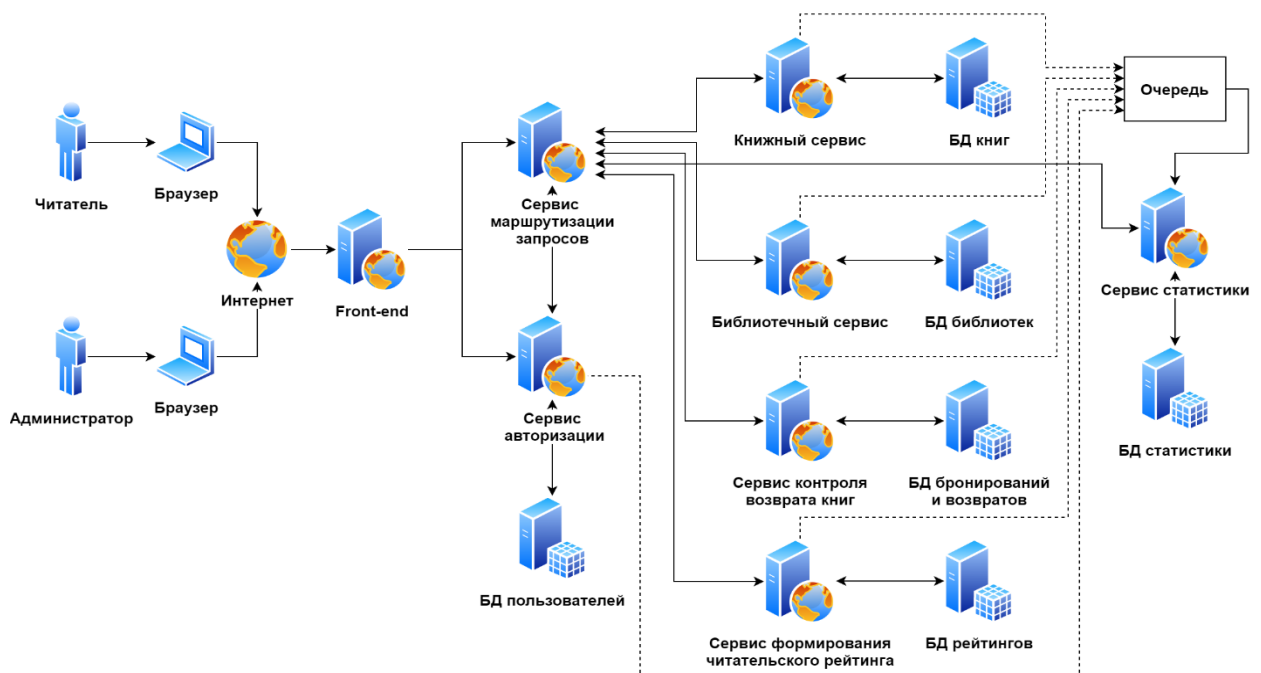


Рисунок 2 – Топология системы.



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.

## **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).

## **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'а 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.

## **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.