

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

# Проектная работа

По дисциплине «Системный анализ и системное проектирование»  
Текстовый отчёт по проекту

*Выполнili:*

Студент группы Р3306

Михайлов Дмитрий

Андреевич

Студент группы Р3312

Малышев Никита

Александрович

*Преподаватель:*

Маркина Татьяна

Анатольевна



Санкт-Петербург  
2025 год

# Оглавление

1	Введение . . . . .	2
2	Бизнес-требования . . . . .	3
3	Анализ и проектирование . . . . .	4
3.1	Итерация 1: Карта стейкхолдеров и план выявления требований . . . . .	4
3.2	Итерация 2: BPMN-диаграмма ключевого процесса . . . . .	6
3.3	Итерация 3: Use Case Diagram и детализация прецедентов . . . . .	7
3.4	Итерация 4: Диаграммы архитектуры, классов и последовательностей . . . . .	22
4	Нефункциональные требования . . . . .	29
4.1	Производительность . . . . .	29
4.2	Надёжность и доступность . . . . .	29
4.3	Безопасность . . . . .	29
4.4	Удобство использования (Usability) . . . . .	30
4.5	Масштабируемость . . . . .	30
4.6	Поддерживаемость . . . . .	31
5	Заключение . . . . .	32

# 1 Введение

Данный проект посвящён разработке информационной системы для управления библиотекой. Система предназначена для автоматизации процессов выдачи и возврата книг, управления каталогом библиотечных материалов, бронирования и продления сроков выдачи, а также для предоставления читателям удобного доступа к библиотечным ресурсам.

Система решает проблему неэффективного управления библиотечными процессами, связанную с ручной обработкой операций выдачи и возврата книг, отсутствием централизованного каталога и сложностью отслеживания состояния библиотечных материалов. Автоматизация этих процессов позволит повысить эффективность работы библиотеки, улучшить качество обслуживания читателей и обеспечить более точный учёт библиотечного фонда.

Основными пользователями системы являются:

- **Читатели** — пользователи библиотеки, которые могут искать книги в каталоге, бронировать и продлевать сроки выдачи, просматривать свою историю выдачи;
- **Библиотекари** — сотрудники библиотеки, которые управляют каталогом, обрабатывают операции выдачи и возврата, управляют пользователями;
- **Администраторы** — управляют системой, настраивают параметры, формируют отчёты.

## 2 Бизнес-требования

Бизнес-требования к системе сформулированы в формате SMART (Specific, Measurable, Achievable, Relevant, Time-bound):

- Увеличить среднее количество бронирований на пользователя на **15%** через персональные рекомендации и напоминания.
- Время ответа поискового запроса не более **2 секунд** для **95%** запросов.
- Доля успешных резерваций не менее **99%** от впервые отправленных запросов.
- Предоставлять ежемесячные дашборды по выдаче, продлениям, популярности жанров и периоду пиковых нагрузок.
- Внедрить автоматизированные уведомления за **3 дня** до просрочки, чтобы заполучить лояльность пользователей.

### 3 Анализ и проектирование

#### 3.1 Итерация 1: Карта стейкхолдеров и план выявления требований

##### Карта стейкхолдеров

На первом этапе проекта была проведена идентификация и анализ стейкхолдеров системы. Карта стейкхолдеров представлена в таблице 1.

Стейкхолдер	Роль	Внутренний/внешний	Влияние	Интерес
Директор библиотеки	Заказчик, руководитель	Внутренний	Высокое	Высокий
Читатель (пользователь)	Пользователь системы	Внешний	Высокое	Высокий
Библиотекарь	Администратор, оператор	Внутренний	Среднее	Высокий
IT-администратор	Технический специалист	Внутренний	Среднее	Средний
Разработчик	Технический специалист	Внутренний	Высокое	Средний
Бухгалтер	Финансовый специалист	Внутренний	Среднее	Средний
Городская администрация	Регулятор / инвестор	Внешний	Высокое	Средний
Провайдер платежей	Сервис-партнёр	Внешний	Среднее	Низкий

Таблица 1: Стейкхолдеры проекта: роль, контекст взаимодействия и интересы.

##### План выявления требований

Для каждого стейкхолдера был определён метод выявления требований и цель опроса. План выявления требований представлен в таблице 2.

<b>Стейкхолдер</b>	<b>Метод</b>	<b>Цель опроса</b>
Директор библиотеки	Интервью	Выяснить стратегические цели сервиса, приоритеты по развитию, бюджетные ограничения, требования к отчетности и КПИ, регуляторные и аудиторские требования
Читатель (пользователь)	Анкеты / Наблюдение	Понять, какие функции и удобства нужны читателю
Библиотекарь	Анкеты	Выяснить операционные требования: процессы выдачи/возврата, учёт материалов, каталогизация, проверки доступа, уведомления, отчётность
IT-администратор	Интервью	Определить требования к инфраструктуре, безопасности, резервному копированию, доступности сервисов, мониторингу и обновлениям, интеграциям
Разработчик	Мозговой штурм	Понять требования к API, интеграциям, архитектуре, используемым технологиям, ограничениям по времени релиза и качеству кода
Бухгалтер	Анкеты	Понять требования к учёту, финансовым операциям, интеграции с бухгалтерскими системами, отчётности
Городская администрация	Интервью	Определить требования к соответствию регуляторным нормам, взаимодействию с госструктурами, доступности услуг, инфраструктурным требованиям и вопросам открытых данных
Провайдер платежей	Интервью / Наблюдение	Понять требования к интеграции платежной инфраструктуры

Таблица 2: План выявления требований.

### 3.2 Итерация 2: BPMN-диаграмма ключевого процесса

На втором этапе была разработана BPMN-диаграмма ключевого бизнес-процесса системы — процесса выдачи книги читателю. Диаграмма отражает последовательность действий библиотекаря и системы при обработке запроса на выдачу книги.

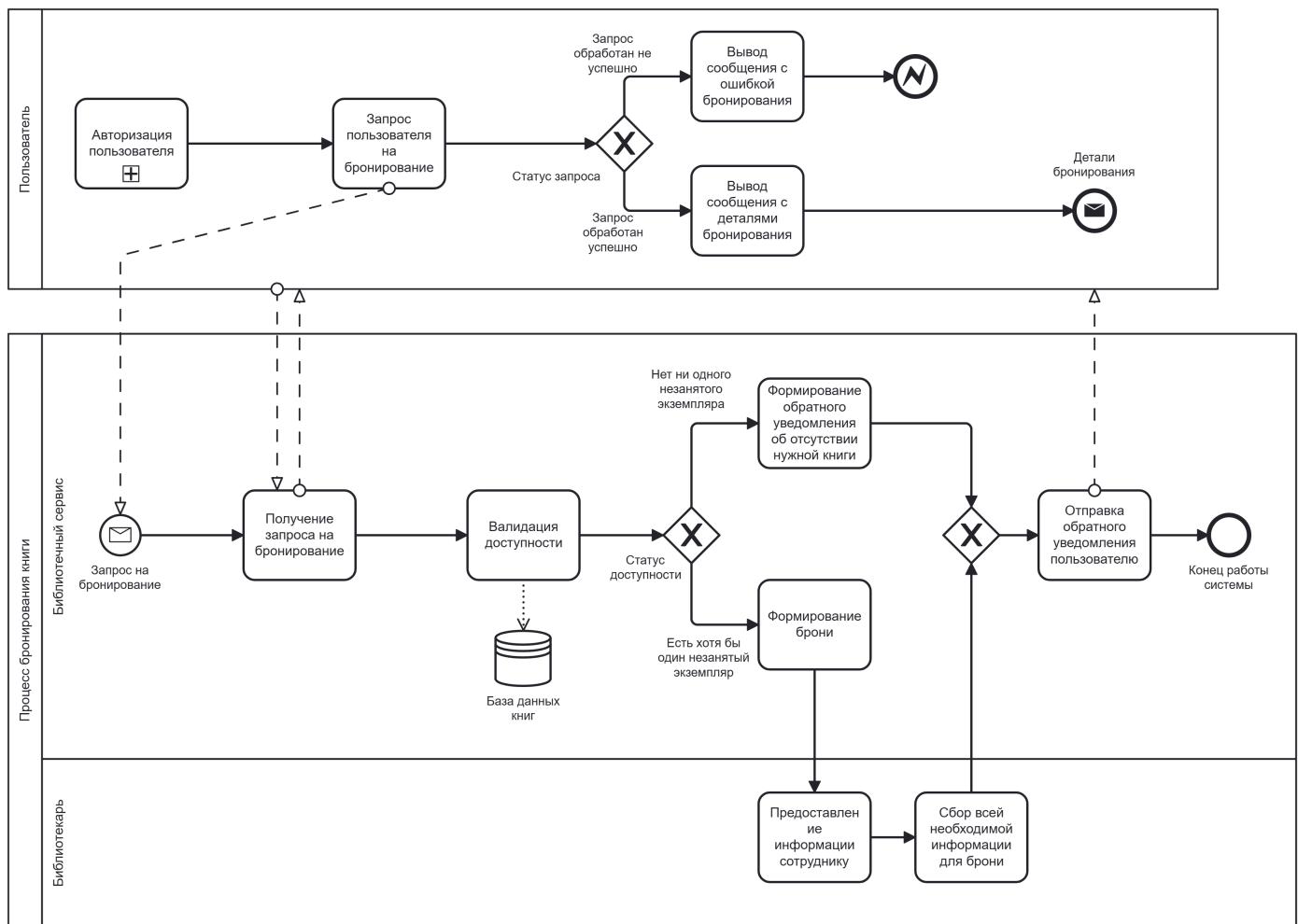


Рис. 1: BPMN-диаграмма процесса выдачи книги.

Также была разработана BPMN-диаграмма процесса возврата книги, которая показывает процедуру возврата книги читателем и обработки этой операции библиотекарем.

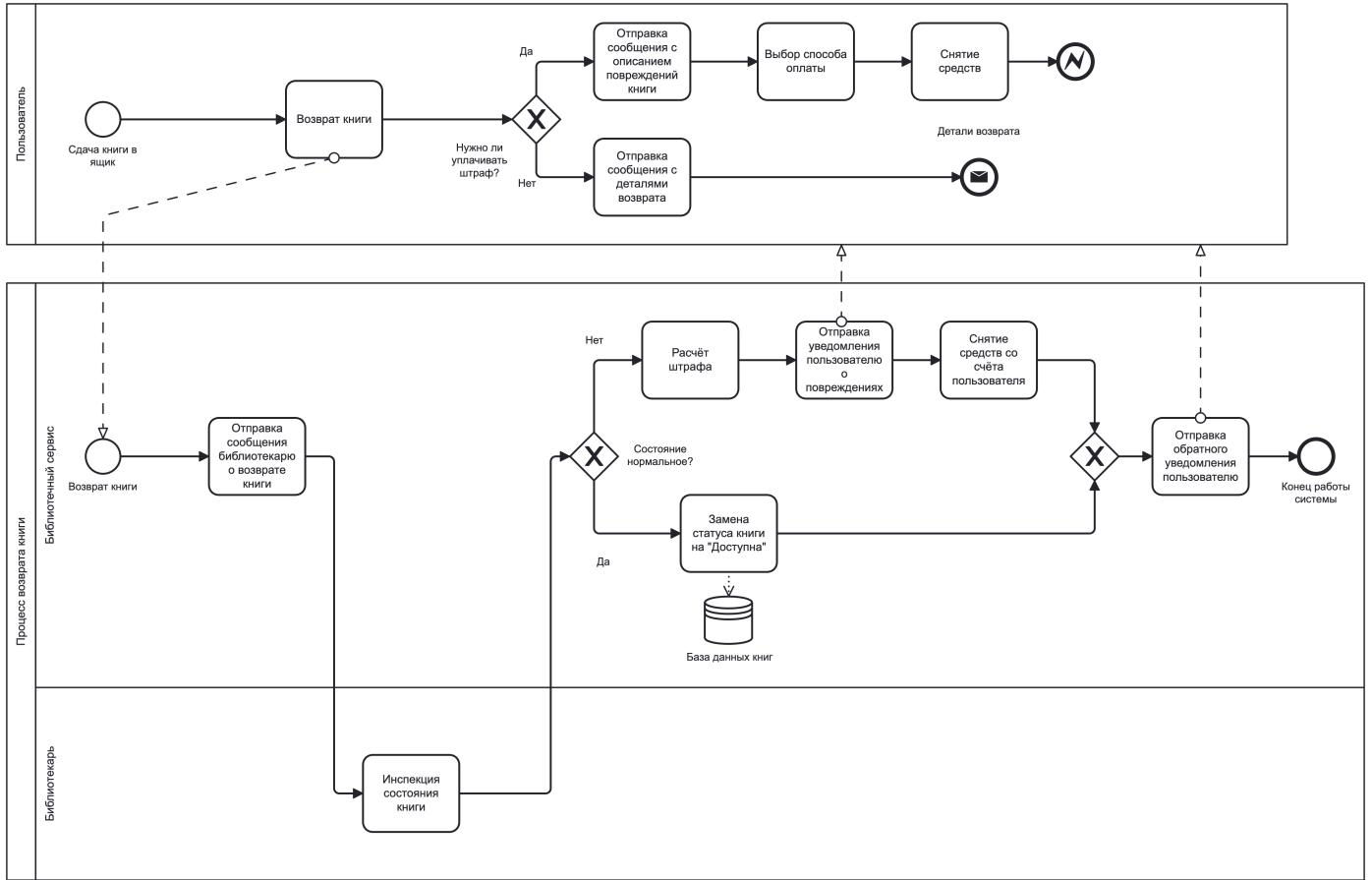


Рис. 2: BPMN-диаграмма процесса возврата книги.

### 3.3 Итерация 3: Use Case Diagram и детализация прецедентов

На третьем этапе был разработан Use Case Diagram, который описывает основные сценарии использования системы различными акторами. Также были детализированы ключевые прецеденты использования системы.

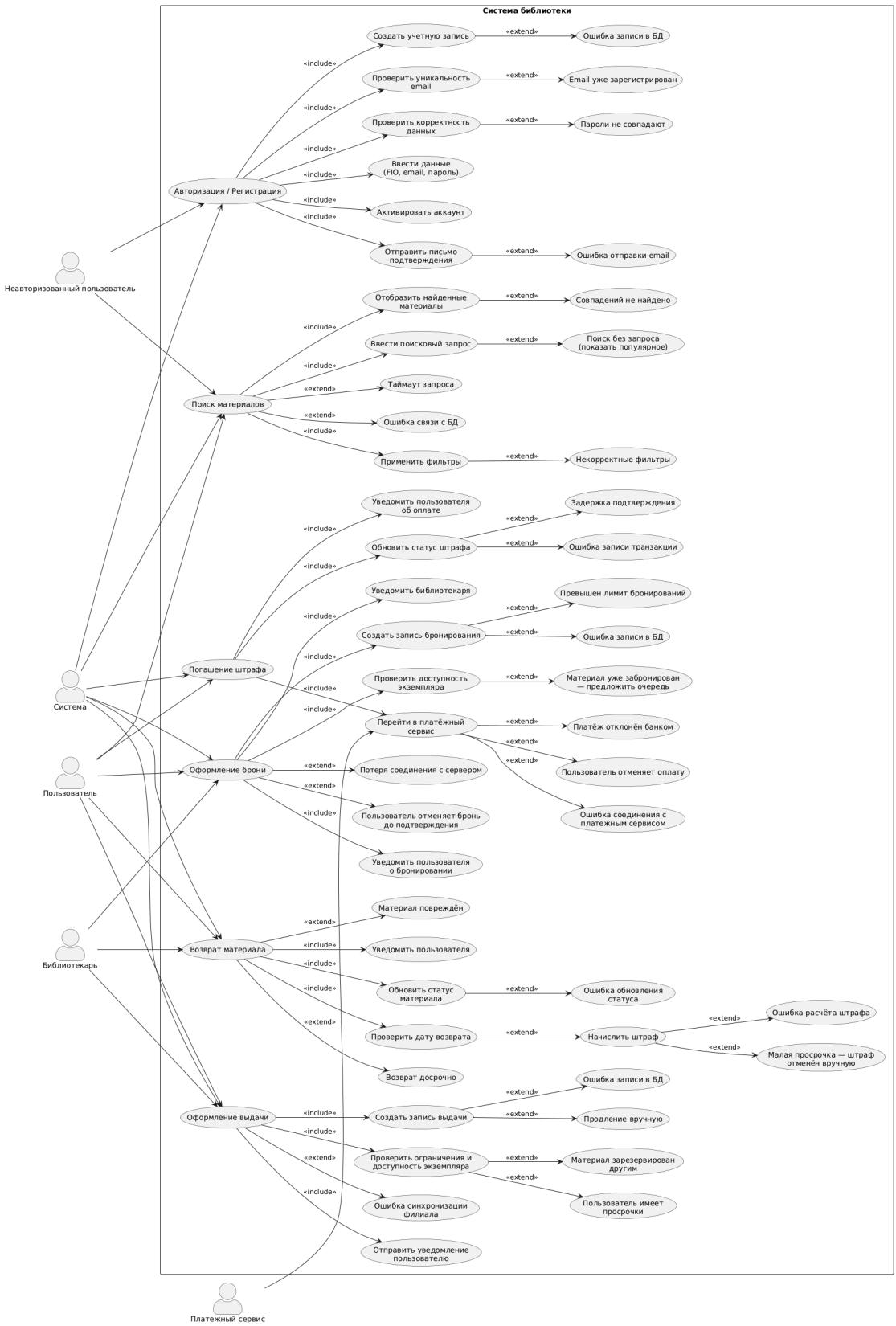


Рис. 3: Use Case Diagram системы управления библиотекой.

---

## Авторизация (регистрация и активация учётной записи)

---

**Описание** Определить, какие ресурсы и действия доступны конкретному пользователю в рамках системы, на основе его учётной записи, ролей и прав.

**Предусловия**

1. Пользователь не авторизован.

**Акторы** Неавторизованный пользователь, Система  
**(Участники)**

### Основной сценарий

<b>№ шага</b>	<b>Действие</b>
1	Неавторизованный пользователь инициирует процесс регистрации в системе.
2	Система запрашивает у неавторизованного пользователя регистрационные данные (ФИО, электронный адрес, пароль, подтверждение пароля).
3	Неавторизованный пользователь вводит запрошенные регистрационные данные.
4	Неавторизованный пользователь отправляет введённые регистрационные данные на обработку.
5	Система проверяет корректность введённых регистрационных данных.
6	Система проверяет уникальность указанного электронного адреса пользователя.
7	Система создаёт учётную запись пользователя в состоянии «ожидает подтверждения».
8	Система формирует запрос на подтверждение регистрации и направляет его пользователю (например, по электронной почте).
9	Пользователь подтверждает регистрацию в системе в ответ на запрос.
10	Система активирует учётную запись пользователя.

### Альтернативный сценарий: Несовпадающие пароли (шаг №5)

5.A.1	Система определяет, что значения пароля и подтверждения пароля не совпадают.
5.A.2	Система информирует неавторизованного пользователя о несоответствии паролей.
5.A.3	Система запрашивает повторный ввод пароля и его подтверждения.
5.A.4	Неавторизованный пользователь вводит новый пароль и его подтверждение.
5.A.5	Неавторизованный пользователь повторно отправляет регистрационные данные на обработку.
5.A.6	Система возвращается к шагу 5 основного сценария.

### Альтернативный сценарий: Электронный адрес уже зарегистрирован (шаг №6)

6.B.1	Система определяет, что указанная электронная почта уже зарегистрирована в системе.
6.B.2	Система информирует неавторизованного пользователя о том, что учётная запись с таким электронным адресом уже существует.
6.B.3	Система предлагает неавторизованному пользователю перейти к процедуре входа в существующую учётную запись.
6.B.4	Неавторизованный пользователь может инициировать процесс авторизации (входа) вместо регистрации. Альтернативный сценарий завершается без возврата к основному сценарию.

---

## Авторизация (регистрация и активация учётной записи)

---

### Альтернативный сценарий: Регистрация не подтверждена (шаг №8)

- 8.C.1 Система удерживает учётную запись пользователя в состоянии «неактивна» до истечения установленного срока.
- 8.C.2 По истечении установленного срока система удаляет неактивную учётную запись пользователя.
- 8.C.3 Система информирует пользователя о невозможности использования неактивированной учётной записи при повторной попытке регистрации или авторизации.
- 8.C.4 Альтернативный сценарий завершается без возврата к основному сценарию.

### Исключения (Ошибки, отсутствие данных для отображения)

- E1 Ошибка отправки e-mail.
1. Система фиксирует ошибку при отправке запроса на подтверждение регистрации.
  2. Система уведомляет пользователя о невозможности отправки запроса на подтверждение регистрации.
  3. Система предлагает пользователю повторить отправку запроса на подтверждение регистрации позже.
- E2 Ошибка записи в базу данных.
1. Система фиксирует ошибку записи учётной записи в базу данных.
  2. Система отклоняет регистрацию.
  3. Система выводит пользователю сообщение о технической ошибке и предлагает повторить попытку позднее.
- Постусловия** Учётная запись пользователя создана и активирована ИЛИ Регистрация не завершена, и учётная запись удалена/не создана.
- 

---

## Поиск материалов

---

**Описание** Поиск и просмотр информации о книгах и других материалах в каталоге библиотеки.

**Предусловия**

1. Пользователь имеет доступ к каталогу (гость или авторизованный).
2. Каталог доступен для выполнения поисковых операций.

**Акторы**  
**(Участники)**  
Пользователь, Система

### Основной сценарий

- | <b>№ шага</b> | <b>Действие</b>   |
|---------------|---|
| 1             | Пользователь инициирует поиск материалов в каталоге.  |
| 2             | Пользователь вводит ключевые характеристики искомых материалов (например, название, автора, идентификатор). |
| 3             | Пользователь отправляет запрос на поиск материалов.   |

---

## Поиск материалов

---

- 4 Система выполняет поиск материалов по указанным критериям.
- 5 Система формирует список найденных материалов.
- 6 Система отображает пользователю список найденных материалов.
- 7 Пользователь задаёт дополнительные параметры отбора материалов (фильтрацию, сортировку).
- 8 Пользователь применяет выбранные параметры отбора.
- 9 Система пересчитывает результаты с учётом указанных параметров отбора.
- 10 Система отображает обновлённый список найденных материалов.
- 11 Пользователь запрашивает детальную информацию по выбранному материалу.
- 12 Система отображает пользователю детальную информацию о выбранном материале.

### **Альтернативный сценарий: Поиск без указания критериев (шаг №2)**

- 2.A.1 Пользователь не вводит критерии поиска и отправляет запрос.
- 2.A.2 Система определяет отсутствие критериев поиска.
- 2.A.3 Система информирует пользователя об отсутствии введённых критериев.
- 2.A.4 Система предлагает пользователю список популярных или рекомендованных материалов.
- 2.A.5
- Пользователь выбирает материал из предложенного списка ИЛИ
  - Пользователь возвращается к вводу критериев поиска (возврат к шагу 2 основного сценария).

### **Альтернативный сценарий: Результаты поиска отсутствуют (шаг №6)**

- 6.B.1 Система не находит материалов, соответствующих заданным критериям.
- 6.B.2 Система информирует пользователя об отсутствии результатов поиска.
- 6.B.3 Система предлагает пользователю изменить или уточнить критерии поиска.
- 6.B.4 Пользователь изменяет или уточняет критерии поиска.
- 6.B.5 Пользователь повторно отправляет запрос на поиск.
- 6.B.6 Система возвращается к шагу 4 основного сценария.

### **Альтернативный сценарий: Некорректное сочетание параметров отбора (шаг №9)**

- 9.C.1 Система определяет, что заданные параметры отбора не могут быть корректно применены.
- 9.C.2 Система информирует пользователя о некорректности сочетания параметров отбора.
- 9.C.3 Система предлагает пользователю изменить параметры отбора.
- 9.C.4 Пользователь изменяет параметры отбора.
- 9.C.5 Пользователь повторно применяет параметры отбора.
- 9.C.6 Система возвращается к шагу 9 основного сценария.

### **Исключения (Ошибки, отсутствие данных для отображения)**

- E1 Ошибка связи с базой данных. Поиск временно недоступен.
- E2 Превышено время ожидания. Система предлагает повторить запрос.
- Постусловия** Отображены найденные результаты ИЛИ сообщение об их отсутствии.
-

---

## Оформление брони

---

<b>Описание</b>	Пользователь резервирует материал для последующего получения в библиотеке.
<b>Предусловия</b>	<ol style="list-style-type: none"><li>Пользователь авторизован.</li><li>Материал доступен для бронирования (существуют экземпляры, подлежащие бронированию).</li></ol>
<b>Акторы (Участники)</b>	Пользователь, Система, Библиотекарь
<b>Основной сценарий</b>	
<b>№ шага</b>	<b>Действие</b>
1	Пользователь выбирает в каталоге материал, который требуется забронировать.
2	Пользователь инициирует бронирование выбранного материала.
3	Система проверяет статус экземпляров выбранного материала (доступен, выдан, забронирован).
4	Система проверяет ограничения на бронирование для пользователя (например, лимит активных броней).
5	При отсутствии ограничений система создаёт запись бронирования материала за пользователем.
6	Система формирует и направляет библиотекарю уведомление о новой брони.
7	Система формирует и направляет пользователю уведомление об успешном создании брони.
<b>Альтернативный сценарий: Материал уже забронирован (шаг №3)</b>	
3.A.1	Система определяет, что экземпляры выбранного материала уже забронированы другими пользователями.
3.A.2	Система информирует пользователя о недоступности немедленного бронирования.
3.A.3	Система предлагает пользователю включение в очередь ожидания на данный материал.
3.A.4	Пользователь принимает решение встать в очередь ожидания.
3.A.5	Система создаёт запись о включении пользователя в очередь ожидания.
3.A.6	Альтернативный сценарий завершается без продолжения основного сценария.
<b>Альтернативный сценарий: Превышен лимит бронирований (шаг №4)</b>	
4.B.1	Система определяет, что пользователь превысил допустимый лимит активных бронирований.
4.B.2	Система информирует пользователя о невозможности создания новой брони из-за превышения лимита.
4.B.3	Альтернативный сценарий завершается без продолжения основного сценария.
<b>Альтернативный сценарий: Отмена бронирования до подтверждения библиотекарем (шаг №5)</b>	
5.C.1	Пользователь инициирует отмену ранее созданной брони.
5.C.2	Система удаляет запись о бронировании материалов за пользователем.

---

## Оформление брони

---

5.С.3	Система формирует и направляет библиотекарю уведомление об отмене брони.
5.С.4	Система формирует и направляет пользователю уведомление об успешной отмене брони.
5.С.5	Альтернативный сценарий завершается без возврата к основному сценарию.
<b>Исключения (Ошибка, отсутствие данных для отображения)</b>	
E1	Ошибка записи в базу данных.
E2	Потеря соединения с сервером при бронировании.
<b>Постусловия</b>	Создана активная бронь или поставлена очередь на материал.

---

## Оформление выдачи материала

---

<b>Описание</b>	Библиотекарь оформляет выдачу материала пользователю.
<b>Предусловия</b>	<ol style="list-style-type: none"><li>Пользователь зарегистрирован в системе.</li><li>У пользователя отсутствуют блокирующие ограничения (например, активные штрафы, критические просрочки).</li><li>Материал доступен для выдачи.</li></ol>
<b>Акторы (Участники)</b>	Пользователь, Система, Библиотекарь
<b>Основной сценарий</b>	
<b>№ шага</b>	<b>Действие</b>
1	Библиотекарь идентифицирует пользователя в системе.
2	Библиотекарь выбирает материал, подлежащий выдаче пользователю.
3	Система проверяет доступность выбранного экземпляра материала.
4	Система проверяет наличие у пользователя ограничений на выдачу (активные штрафы, просроченные материалы и др.).
5	При отсутствии ограничений система регистрирует выдачу материала пользователю с указанием срока возврата и места выдачи.
6	Система формирует и направляет пользователю уведомление о выданном материале и установленном сроке возврата.
<b>Альтернативный сценарий: У пользователя есть просроченные материалы (шаг №4)</b>	
4.А.1	Система обнаруживает у пользователя просроченные материалы.
4.А.2	Система блокирует операцию выдачи нового материала.
4.А.3	Система информирует библиотекаря и пользователя о причине блокировки выдачи.
4.А.4	Альтернативный сценарий завершается без продолжения основного сценария.
<b>Альтернативный сценарий: Материал зарезервирован другим пользователем (шаг №3)</b>	
3.В.1	Система определяет, что выбранный материал зарезервирован другим пользователем.
3.В.2	Система запрещает выдачу материала текущему пользователю.

---

## Оформление выдачи материала

---

3.B.3	Система информирует библиотекаря и пользователя о невозможности выдачи материала.	
3.B.4	Альтернативный сценарий завершается без продолжения основного сценария.	
<b>Альтернативный сценарий: Изменение срока выдачи библиотекарем (шаг №5)</b>		
5.C.1	Библиотекарь принимает решение изменить срок выдачи материала пользователю в соответствии с внутренними правилами библиотеки.	
5.C.2	Библиотекарь задаёт новый срок возврата материала.	
5.C.3	Система обновляет срок возврата материала в записи выдачи.	
5.C.4	Система при необходимости формирует и направляет пользователю актуальное уведомление о сроке возврата.	
5.C.5	Альтернативный сценарий завершается без возврата к основному сценарию.	
<b>Исключения (Ошибка, отсутствие данных для отображения)</b>		
E1	Сбой записи в базу данных при создании выдачи.	Выдача не создана.
E2	Ошибка синхронизации с удалённым филиалом.	Выдача не подтверждена.
<b>Постусловия</b>	Материал числится как выданный пользователю.	

---

## Прецедент 5: Возврат материала

---

### Возврат материала

---

<b>Описание</b>	Пользователь возвращает материал, библиотекарь регистрирует возврат в системе.
<b>Предусловия</b>	1. Материал числится за пользователем как выданный. 2. Пользователь обращается в библиотеку для возврата материала.
<b>Акторы (Участники)</b>	
<b>Основной сценарий</b>	
<b>№ шага</b>	<b>Действие</b>
1	Библиотекарь находит в системе запись о выдаче возвращаемого материала конкретному пользователю.
2	Библиотекарь регистрирует факт возврата материала в системе.
3	Система определяет фактическую дату возврата материала.
4	Система сравнивает фактическую дату возврата с установленным сроком возврата.
5	При отсутствии просрочки система изменяет статус материала на «доступен».
6	При наличии просрочки система рассчитывает размер штрафа в соответствии с действующими тарифами.
7	Система формирует и направляет пользователю уведомление о результатах возврата материала.

---

## Возврат материала

---

8	При наличии рассчитанного штрафа система формирует и направляет пользователю уведомление о начисленном штрафе.
<b>Альтернативный сценарий: Возврат повреждённого материала (шаг №2)</b>	
2.А.1	Библиотекарь при приёме материала обнаруживает повреждение экземпляра.
2.А.2	Библиотекарь фиксирует состояние повреждённого экземпляра в системе.
2.А.3	Библиотекарь инициирует создание акта о повреждении материала.
2.А.4	Система сохраняет сведения об акте повреждения материала.
2.А.5	При необходимости система инициирует процедуру расчёта компенсации или иного вида ответственности в соответствии с правилами библиотеки.
2.А.6	После фиксации повреждения возврат материала продолжает обрабатываться по основному сценарию с шага 3 ИЛИ переходит к отдельной процедуре урегулирования (в зависимости от принятых правил).
<b>Альтернативный сценарий: Досрочный возврат материала (шаг №4)</b>	
4.В.1	Пользователь возвращает материал до наступления срока возврата.
4.В.2	Система фиксирует отсутствие просрочки при выполнении сравнения дат возврата.
4.В.3	Система пропускает расчёт штрафа и изменяет статус материала на «доступен».
4.В.4	Альтернативный сценарий завершается продолжением основного сценария с шага 7.
<b>Альтернативный сценарий: Малая просрочка и отмена штрафа (шаг №6)</b>	
6.С.1	Система определяет, что просрочка возврата материала не превышает установленный порог (например, одни сутки).
6.С.2	Система рассчитывает штраф в соответствии с тарифами.
6.С.3	Библиотекарь принимает решение отменить начисленный штраф при малой просрочке.
6.С.4	Библиотекарь инициирует в системе отмену рассчитанного штрафа.
6.С.5	Система отменяет начисленный штраф и сохраняет соответствующую запись.
6.С.6	Система формирует и направляет пользователю уведомление об отмене штрафа.
6.С.7	Альтернативный сценарий завершается без возврата к основному сценарию.
<b>Исключения (Ошибки, отсутствие данных для отображения)</b>	
E1	Ошибка обновления статуса материала.
E2	Ошибка расчёта штрафа из-за отсутствия данных о тарифе.
Постусловия	Материал возвращён и снова доступен. При необходимости начислен штраф.

---

## Прецедент 6: Погашение штрафа

---

### Погашение штрафа

---

<b>Описание</b>	Пользователь оплачивает начисленный штраф через платёжный сервис.
-----------------	---

---

## Погашение штрафа

---

### Предусловия

1. Пользователь авторизован.
2. В системе существует хотя бы один активный штраф, начисленный пользователю.

### Акторы (Участники)

Пользователь, Система

### Основной сценарий

№ шага	Действие
1	Пользователь запрашивает в системе список своих активных штрафов.
2	Система формирует список активных штрафов пользователя.
3	Система отображает пользователю список активных штрафов.
4	Пользователь выбирает штраф для оплаты.
5	Пользователь инициирует оплату выбранного штрафа.
6	Система передаёт параметры выбранного штрафа в платёжный сервис и передаёт ему управление для проведения операции.
7	Пользователь вводит платёжные реквизиты в платёжном сервисе.
8	Пользователь подтверждает выполнение платежа.
9	Платёжный сервис обрабатывает платёж.
10	Платёжный сервис формирует результат обработки платежа.
11	Платёжный сервис передаёт результат обработки платежа в систему.
12	При успешной оплате система обновляет статус штрафа на «оплачен».
13	Система формирует и направляет пользователю уведомление об успешной оплате штрафа.

### Альтернативный сценарий: Отмена оплаты пользователем (шаг №8)

8.A.1	Пользователь отменяет подтверждение платежа в платёжном сервисе.
8.A.2	Платёжный сервис прекращает обработку операции оплаты.
8.A.3	Платёжный сервис передаёт в систему информацию об отменённой операции.
8.A.4	Система сохраняет исходный статус штрафа как «не оплачен».
8.A.5	Система возвращает пользователя к просмотру списка активных штрафов (аналог состояния после шага 3 основного сценария).
8.A.6	Альтернативный сценарий завершается продолжением основного сценария с шага 3.

### Альтернативный сценарий: Платёж отклонён банком (шаг №9–11)

9.B.1	Платёжный сервис получает от банка отказ в проведении операции оплаты.
9.B.2	Платёжный сервис формирует информацию о неуспешной операции.
9.B.3	Платёжный сервис передаёт в систему информацию об отказе в проведении платежа.
9.B.4	Система сохраняет статус штрафа как «не оплачен».
9.B.5	Система информирует пользователя о том, что платёж отклонён.
9.B.6	Система предлагает пользователю повторить попытку оплаты штрафа.
9.B.7	Альтернативный сценарий завершается, пользователь при желании может повторно иницировать основной сценарий.

### Альтернативный сценарий: Задержка подтверждения от платёжного сервиса (шаг №8)

---

## Погашение штрафа

---

8.C.1	Система фиксирует отсутствие результата от платёжного сервиса в установленный интервал времени.
8.C.2	Система устанавливает для штрафа промежуточный статус (например, «ожидает подтверждения оплаты»).
8.C.3	Система продолжает периодически запрашивать у платёжного сервиса состояние операции.
8.C.4	При получении подтверждения успешной оплаты от платёжного сервиса система обновляет статус штрафа на «оплачен».
8.C.5	Система формирует и направляет пользователю уведомление об успешной оплате штрафа.
8.C.6	При получении информации об отказе система действует по шагам альтернативного сценария В.
8.C.7	Альтернативный сценарий завершается.
<b>Исключения (Ошибки, отсутствие данных для отображения)</b>	
E1	Ошибка соединения с платёжным сервисом. Система уведомляет о невозможности проведения операции.
E2	Сбой записи данных о транзакции. Система не меняет статус штрафа и логирует ошибку.
<b>Постусловия</b>	Штраф оплачен и удалён из списка активных.

---

## Функциональные требования

На основе детализированных прецедентов использования системы сформулированы функциональные требования, которые определяют конкретные возможности системы.

### 1. Регистрация / Авторизация пользователя

- (a) Система должна позволить неавторизованному пользователю инициировать процесс регистрации в системе.
- (b) Система должна позволить неавторизованному пользователю предоставить регистрационные данные: ФИО, электронный адрес, пароль и подтверждение пароля.
- (c) Система должна позволить пользователю запросить повторную отправку письма активации учётной записи.
- (d) Система должна позволить пользователю, имеющему активную учётную запись, иницировать процесс входа в систему.
- (e) Система должна позволить пользователю предоставить свои учётные данные (e-mail и пароль) для авторизации.
- (f) Система должна позволить авторизованному пользователю завершить сеанс работы в системе.
- (g) Система должна проверять корректность введённых пользователем данных регистрации (e-mail, пароль, подтверждение пароля, ФИО).
- (h) Система должна проверять уникальность указанного пользователем e-mail при регистрации.

- (i) Система должна создавать учётную запись пользователя при успешной проверке регистрационных данных.
- (j) Система должна отправлять пользователю письмо с подтверждением регистрации на указанный e-mail.
- (k) Система должна активировать учётную запись пользователя при переходе по ссылке из письма подтверждения.
- (l) Система должна отказывать в регистрации, если указанный пользователем e-mail уже используется другой учётной записью.
- (m) Система должна отображать пользователю сообщение о несоответствии пароля и его подтверждения при их различии.
- (n) Система должна сообщать пользователю об ошибках соединения или сбоях базы данных при выполнении операций регистрации и авторизации.
- (o) Система должна удалять неактивированные учётные записи пользователей по истечении установленного срока.
- (p) Система должна блокировать невалидные сессии и предотвращать доступ при некорректных попытках авторизации.

## 2. Поиск материалов

- (a) Система должна позволить неавторизованному пользователю просматривать каталог материалов без авторизации.
- (b) Система должна позволить пользователю выполнять поиск материалов по каталогу, указав поисковый запрос.
- (c) Система должна позволить пользователю применять фильтры к результатам поиска, указав параметры фильтрации (жанр, год издания, тип материала, наличие).
- (d) Система должна позволить пользователю сортировать результаты поиска, выбрав критерий сортировки (по алфавиту, дате издания, дате добавления).
- (e) Система должна позволить пользователю запрашивать детальную информацию о выбранном материале из результатов поиска.
- (f) Система должна искать материалы по ключевым характеристикам (например, названию, автору, идентификатору, ISBN и др.).
- (g) Система должна отображать пользователю список найденных материалов с основной информацией о каждом материале.
- (h) Система должна поддерживать фильтрацию результатов поиска по заданным пользователем параметрам (жанру, году издания, типу материала, наличию и т. п.).
- (i) Система должна поддерживать сортировку результатов поиска (например, по алфавиту, дате издания, дате добавления).
- (j) Система должна сообщать пользователю о том, что по заданным критериям не найдено ни одного материала.
- (k) Система должна отображать пользователю список популярных или рекомендованных материалов при отсутствии введённого поискового запроса.

- (l) Система должна отображать пользователю сообщение о некорректной комбинации фильтров, если заданные параметры отбора не дают результатов.
- (m) Система должна обеспечивать корректную работу поиска по каталогу при высокой нагрузке и большом объёме данных.
- (n) Система должна сообщать пользователю о временной недоступности поиска при сбое соединения или внутренних ошибках.

### **3. Оформление брони**

- (a) Система должна позволить авторизованному пользователю получить доступ к своему личному кабинету после авторизации.
- (b) Система должна позволить авторизованному пользователю просматривать список своих активных бронирований.
- (c) Система должна позволить авторизованному пользователю инициировать бронирование выбранного материала из каталога.
- (d) Система должна позволить пользователю отменять созданную бронь до её подтверждения или обработки библиотекарем.
- (e) Система должна проверять доступность экземпляров материала для бронирования перед созданием записи брони.
- (f) Система должна создавать запись бронирования материала за пользователем при успешной проверке условий.
- (g) Система должна уведомлять библиотекаря о созданной пользователем брони.
- (h) Система должна уведомлять пользователя об успешном создании бронирования материала.
- (i) Система должна предлагать пользователю включиться в очередь ожидания, если все экземпляры выбранного материала уже забронированы.
- (j) Система должна отказывать пользователю в создании новой брони, если превышен установленный лимит активных бронирований.
- (k) Система должна сохранять все операции, связанные с бронированием материалов, в журнале действий.
- (l) Система должна обрабатывать ошибки записи в базу данных при создании или отмене брони и уведомлять пользователя о невозможности выполнения операции.
- (m) Система должна предотвращать дублирование активных броней одного и того же материала для одного пользователя.

### **4. Оформление выдачи материала**

- (a) Система должна позволить библиотекарю получить доступ к функционалу оформления выдачи материала.
- (b) Система должна позволить библиотекарю выполнить поиск пользователя по идентификатору, имени или иному уникальному атрибуту при оформлении выдачи.
- (c) Система должна позволить библиотекарю выбрать материал из каталога для выдачи пользователю.

- (d) Система должна позволить библиотекарю указать срок возврата материала при оформлении выдачи в соответствии с внутренними правилами библиотеки.
- (e) Система должна позволить библиотекарю подтвердить выдачу материала пользователю.
- (f) Система должна позволить авторизованному пользователю просматривать список выданных ему материалов с информацией о дате выдачи и сроке возврата.
- (g) Система должна позволить авторизованному пользователю просматривать историю выданных материалов.
- (h) Система должна проверять отсутствие у пользователя активных блокирующих штрафов или критических просрочек перед выдачей материала.
- (i) Система должна проверять доступность выбранного экземпляра материала для выдачи пользователю.
- (j) Система должна фиксировать факт выдачи материала пользователю с указанием даты выдачи, срока возврата и места (филиала) выдачи.
- (k) Система должна уведомлять пользователя о выданных материалах и установленных сроках возврата.
- (l) Система должна блокировать выдачу новых материалов пользователю при наличии у него просроченных материалов.
- (m) Система должна предотвращать выдачу материала пользователю, если данный экземпляр зарезервирован другим пользователем.
- (n) Система должна обрабатывать ошибки записи данных о выдаче в базе данных и уведомлять библиотекаря о невозможности завершения операции.
- (o) Система должна обеспечивать синхронизацию данных о выдаче материалов между филиалами библиотеки при распределённом хранении данных.
- (p) Система должна хранить журнал всех выданных экземпляров с указанием дат выдачи, возврата и идентификаторов пользователей и материалов.

## 5. Возврат материала

- (a) Система должна позволить библиотекарю получить доступ к функционалу возврата материала.
- (b) Система должна позволить библиотекарю выполнить поиск записи выдачи по идентификатору пользователя, названию материала или идентификатору экземпляра при возврате.
- (c) Система должна позволить библиотекарю зарегистрировать возврат материала в системе.
- (d) Система должна позволить библиотекарю зафиксировать факт повреждения материала при его возврате, указав описание повреждения.
- (e) Система должна позволить библиотекарю вручную отменить начисленный штраф при малой просрочке в рамках установленных правил, подтвердив данное действие.
- (f) Система должна фиксировать факт возврата материала в учётной записи выдачи.

- (g) Система должна проверять фактическую дату возврата материала и сравнивать её с установленным сроком возврата.
- (h) Система должна создавать запись (акт) о повреждении материала с указанием состояния экземпляра и необходимых деталей.
- (i) Система должна начислять пользователю штраф при обнаружении просрочки возврата в соответствии с действующими тарифами.
- (j) Система должна изменять статус возвращённого материала на «доступен» после успешной регистрации возврата.
- (k) Система должна отправлять пользователю уведомление о принятом возврате материала.
- (l) Система должна уведомлять пользователя о начисленном штрафе при наличии просрочки возврата.
- (m) Система должна обрабатывать ошибки при обновлении статуса материала и начислении штрафа, не допуская неконсистентного состояния данных.
- (n) Система должна обрабатывать случаи отсутствия данных о тарифах при расчёте штрафа и уведомлять об этом библиотекаря или администратора.

## 6. Погашение штрафа

- (a) Система должна позволить авторизованному пользователю просматривать актуальный список своих активных штрафов с указанием суммы, причины начисления и срока оплаты.
- (b) Система должна позволить пользователю инициировать оплату выбранного штрафа из списка активных штрафов.
- (c) Система должна перенаправлять пользователя для оплаты штрафа в защищённый платёжный сервис.
- (d) Система должна получать от платёжного сервиса статус выполненной транзакции (успех, отказ, отмена, ожидание подтверждения).
- (e) Система должна обновлять статус штрафа на «оплачен» при получении успешного подтверждения оплаты от платёжного сервиса.
- (f) Система должна отображать пользователю сообщение об успешной оплате штрафа после изменения его статуса.
- (g) Система должна не изменять статус штрафа при отмене пользователем операции оплаты или при отказе в проведении платежа.
- (h) Система должна повторно запрашивать статус операции оплаты у платёжного сервиса при задержке подтверждения результата.
- (i) Система должна обрабатывать ошибки соединения с платёжным сервисом и информировать пользователя о невозможности завершения операции оплаты.
- (j) Система должна обрабатывать ошибки записи данных о транзакции оплаты в базу данных и обеспечивать целостность статусов штрафов.
- (k) Система должна обеспечивать журналирование всех операций, связанных с оплатой штрафов, для последующего аудита.

- (1) Система должна защищать передаваемые платёжные данные пользователя от несанкционированного доступа и утечек.

## 7. Управление каталогом и бронированиями (для библиотекаря)

- (a) Система должна позволить библиотекарю просматривать список новых бронирований, требующих обработки.
- (b) Система должна позволить библиотекарю получить доступ к функционалу управления каталогом материалов для добавления, редактирования или удаления записей материалов.

### 3.4 Итерация 4: Диаграммы архитектуры, классов и последовательностей

#### Диаграмма контекста (C1)

Диаграмма контекста определяет границы системы и её взаимодействие с внешними акторами и системами.



Рис. 4: Диаграмма контекста (Context Diagram - C1).

## Диаграмма контейнеров (C2)

Диаграмма контейнеров показывает высокоуровневую архитектуру системы, включая основные приложения, базы данных и внешние системы.

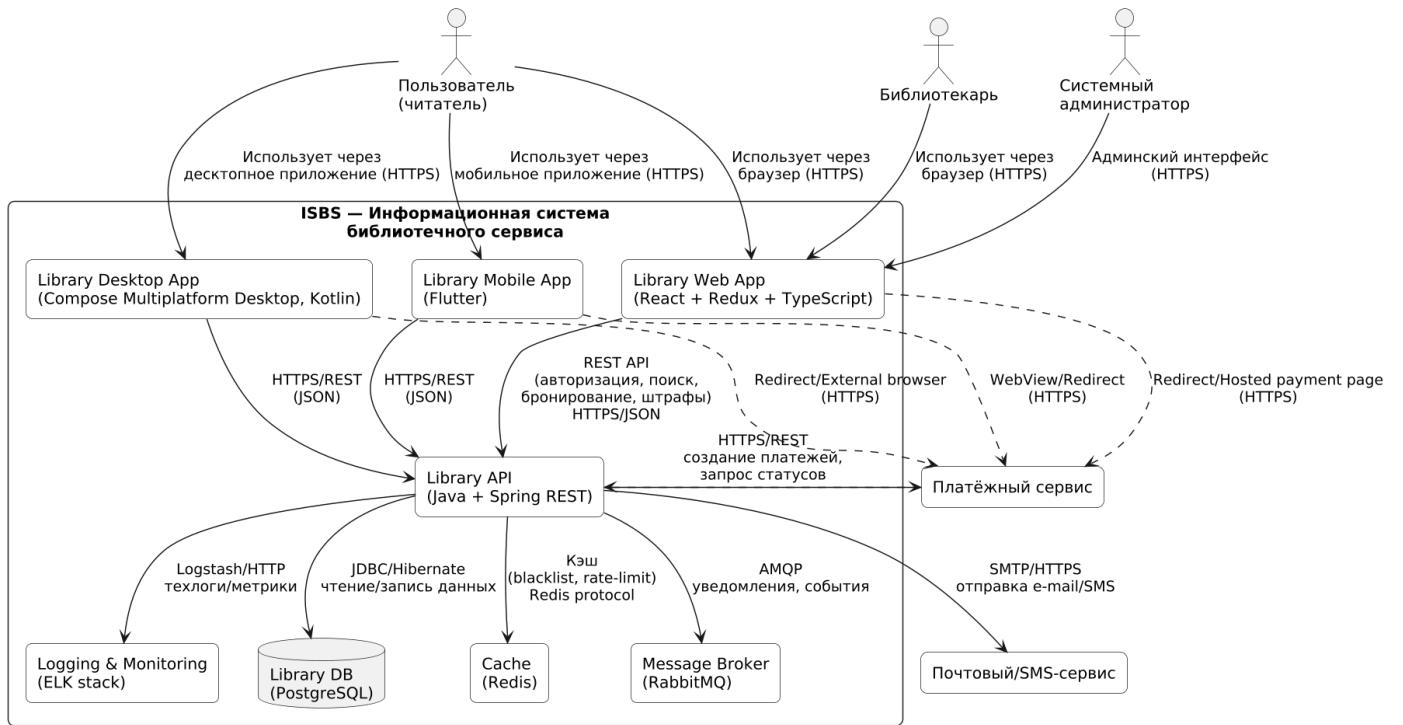


Рис. 5: Диаграмма контейнеров (Container Diagram - C2).

**Клиентские приложения** Система предоставляет три типа клиентских приложений для различных платформ, обеспечивая максимальную доступность и удобство использования для разных категорий пользователей.

- **Library Web App (React, Redux, TypeScript)** — веб-приложение для работы через браузер.

*Технологический стек:*

- **React** — библиотека для построения пользовательских интерфейсов на основе компонентного подхода. Позволяет создавать переиспользуемые UI-компоненты, обеспечивает эффективное обновление интерфейса через виртуальный DOM.
- **Redux** — библиотека управления состоянием приложения. Централизует состояние, упрощает отладку и тестирование, обеспечивает предсказуемое поведение приложения при работе с данными каталога, бронированиями и профилем пользователя.
- **TypeScript** — типизированное надмножество JavaScript. Обеспечивает статическую типизацию, раннее обнаружение ошибок на этапе разработки, улучшает читаемость кода и упрощает рефакторинг больших приложений.

*Особенности:*

- Используется как читателями для поиска и бронирования материалов, так и библиотекарями для управления каталогом и обработки операций выдачи/возврата.
- Коммуникация с backend осуществляется через REST API по протоколу HTTPS с использованием формата JSON.
- Для интеграции с платёжным сервисом используется механизм редиректа во внешний браузер или WebView для безопасной обработки платежей.

- Адаптивный дизайн обеспечивает корректную работу на различных размерах экранов (десктоп, планшет, мобильные устройства).

- **Library Mobile App** (Flutter) — мобильное приложение для iOS и Android.

*Технологический стек:*

- **Flutter** — фреймворк от Google для разработки кроссплатформенных мобильных приложений. Использует язык программирования Dart и собственный движок рендеринга для создания нативно выглядящих интерфейсов.

*Особенности:*

- Единая кодовая база для платформ iOS и Android, что значительно сокращает время разработки и поддержки.
- Высокая производительность благодаря компиляции в нативный код и отсутствию необходимости в мосте между JavaScript и нативными компонентами.
- Богатая экосистема готовых виджетов и пакетов для реализации типичных мобильных паттернов (навигация, формы, списки, уведомления).
- Поддержка push-уведомлений для информирования пользователей о статусе бронирований, напоминаниях о сроках возврата и других важных событиях.
- Коммуникация с backend через HTTPS/REST API в формате JSON, аналогично веб-приложению.

- **Library Desktop App** (Compose Multiplatform Desktop, Kotlin) — нативное десктопное приложение.

*Технологический стек:*

- **Compose Multiplatform Desktop** — современный фреймворк для создания десктопных приложений на основе декларативного подхода к построению UI, аналогичного Jetpack Compose для Android.
- **Kotlin** — статически типизированный язык программирования, работающий на JVM. Обеспечивает безопасность типов, лаконичный синтаксис, отличную интеграцию с Java-библиотеками.

*Особенности:*

- Нативный пользовательский интерфейс, соответствующий стандартам операционных систем (Windows, macOS, Linux).
- Высокая производительность за счёт компиляции в нативный код и оптимизации для десктопных платформ.
- Возможность работы в офлайн-режиме с синхронизацией данных при восстановлении соединения.
- Поддержка клавиатурных сокращений и других десктопных паттернов взаимодействия для повышения эффективности работы библиотекарей.
- Прямая интеграция с системными уведомлениями операционной системы.
- Коммуникация с backend через HTTPS/REST API в формате JSON.

Все три клиентских приложения взаимодействуют с единым backend-сервисом **Library API** (Java + Spring REST), что обеспечивает консистентность данных и бизнес-логики независимо от платформы клиента. Каждое приложение оптимизировано под свою целевую платформу, но предоставляет пользователям единообразный функционал системы управления библиотекой.

### Диаграмма компонентов (C3)

Диаграмма компонентов детализирует внутреннюю структуру одного из ключевых контейнеров системы, показывая основные компоненты и их взаимодействие.

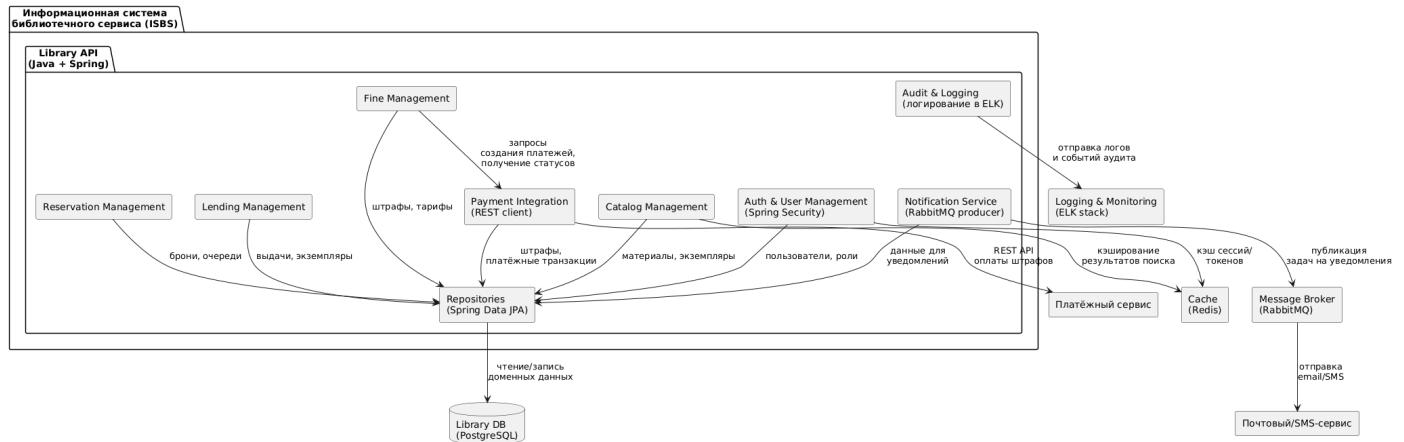


Рис. 6: Диаграмма компонентов (Component Diagram - C3) для одного ключевого контейнера.

### Диаграмма классов (C4)

Диаграмма классов описывает структуру данных системы, основные сущности и их взаимосвязи.

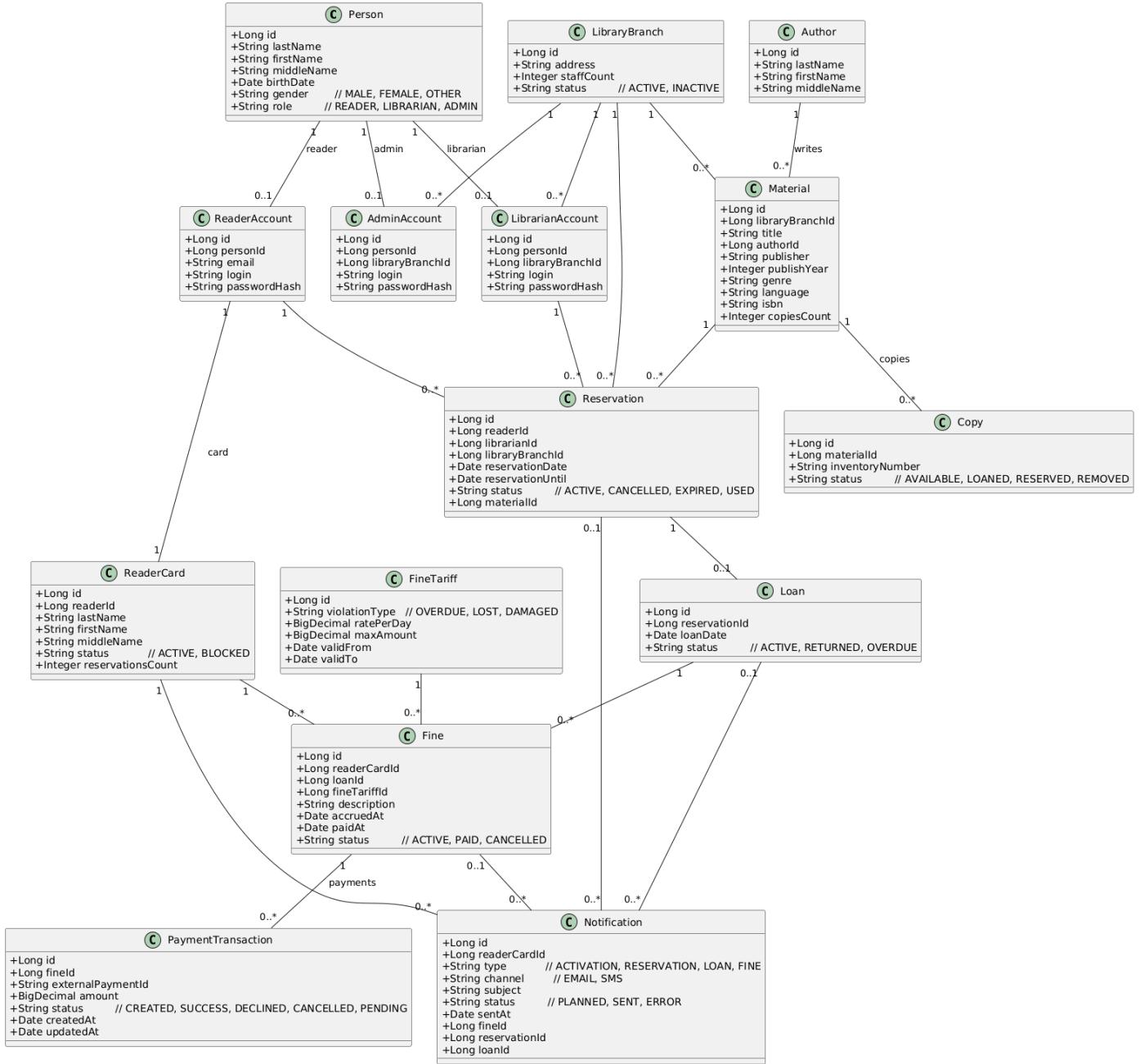


Рис. 7: Диаграмма классов (Class Diagram - C4).

## Диаграмма последовательностей

Диаграмма последовательностей показывает динамическое взаимодействие объектов в рамках одного из ключевых сценариев использования системы.

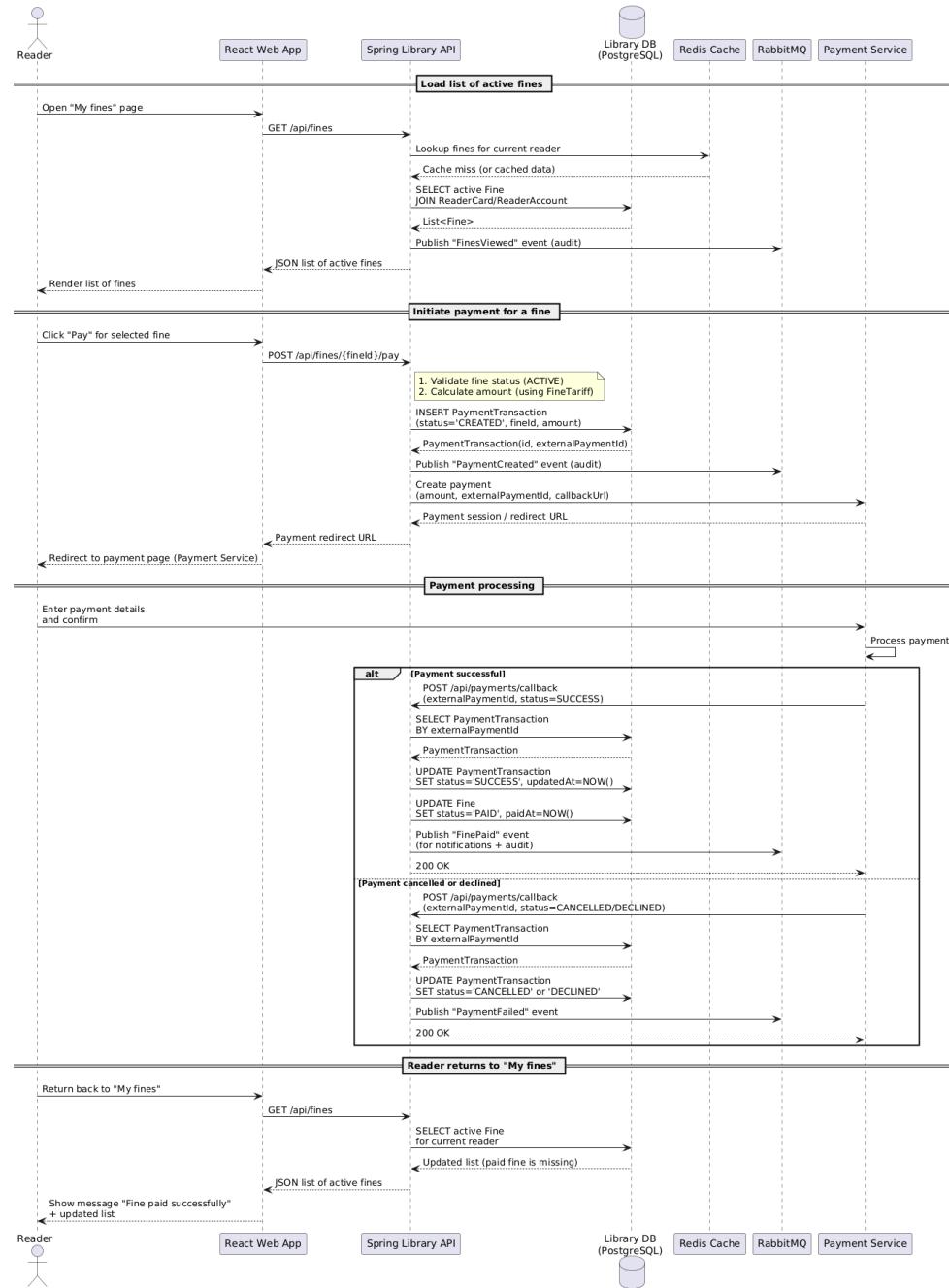


Рис. 8: Диаграмма последовательностей (Sequence Diagram).

## **4 Нефункциональные требования**

Нефункциональные требования определяют качественные характеристики системы, которые должны быть обеспечены при её реализации. Требования сформулированы в измеримом виде для обеспечения возможности их проверки.

### **4.1 Производительность**

1. Система каталога должна обрабатывать не менее 95% поисковых запросов по каталогу при нагрузке до 100 одновременных пользователей за время не более 1.5 секунд в 95% измерений за календарные сутки.
2. Система обработки операций выдачи должна выполнять регистрацию выдачи или возврата одной книги при нагрузке до 100 одновременных библиотекарей за время не более 2 секунд в 95% измерений.
3. Подсистема отчётности должна формировать отчёт о выданных книгах по одной библиотеке объёмом до 100 000 записей за время не более 5 секунд при нормальной нагрузке.
4. Подсистема продления должна регистрировать продление срока одной выдачи пользователем при нагрузке до 200 запросов в минуту за время не более 2 секунд.
5. Подсистема управления каталогом должна сохранять изменения карточки одной книги (редактирование полей или добавление новой книги) при нагрузке до 50 одновременных библиотекарей за время не более 3 секунд.
6. Подсистема поиска с фильтрами должна возвращать пользователю результаты поиска с применёнными фильтрами (автор, жанр, год издания) при объёме каталога до 100 000 записей за время не более 2 секунд.

### **4.2 Надёжность и доступность**

1. Система в целом должна обеспечивать доступность (uptime) не менее 98% в течение календарного месяца при эксплуатации в штатном режиме.
2. Подсистема восстановления должна восстанавливать работоспособность системы после критического сбоя за время не более 2 часов с момента фиксации сбоя в системе мониторинга.
3. Подсистема резервного копирования должна выполнять ежедневное полное резервное копирование базы данных объёмом до 10 ГБ в промежутке с 02:00 до 04:00 по серверному времени без остановки пользовательских операций более чем на 2 минуты.
4. Подсистема восстановления данных должна восстанавливать базу данных из последней успешной резервной копии объёмом до 10 ГБ за время не более 1 часа.

### **4.3 Безопасность**

1. Подсистема аутентификации должна хранить пароли пользователей только в виде хэш-сумм, рассчитанных алгоритмом bcrypt с индивидуальной солью длиной не менее 16 байт во всех пользовательских записях.

- Подсистема авторизации должна разрешать доступ к операциям управления пользователями и каталогом материалов только учетным записям с ролями “Библиотекарь” или “Администратор” при проверке прав в каждой защищаемой операции.
- Транспортный уровень системы должен обеспечивать шифрование всех запросов и ответов между клиентом и сервером с использованием протокола TLS версии не ниже 1.2 во 100% внешних соединений.
- Подсистема аудита должна записывать в журнал аудита все действия библиотекарей и администраторов (создание, изменение, удаление сущностей) с указанием идентификатора пользователя, времени и типа операции для 100% таких действий.
- Подсистема защиты от перебора паролей должна блокировать учётную запись пользователя на 15 минут после 5 подряд неуспешных попыток входа, совершённых в интервале 15 минут.
- Подсистема подтверждения критических операций должна требовать явного подтверждения пользователя и записывать в журнал аудита каждую операцию удаления материалов, начисления штрафов и изменения статуса библиотек в 100% случаев.

#### **4.4 Удобство использования (Usability)**

- Пользовательский веб-интерфейс должен позволять новому пользователю пройти регистрацию и авторизацию при наличии стабильного соединения за время не более 3 минут в 90% сценариев юзабилити-тестирования.
- Пользовательский интерфейс должен позволять читателю оформить первую бронь или выдачу книги при знании названия или автора за время не более 4 минут в 90% тестовых сценариев.
- Интерфейс личного кабинета должен позволять зарегистрированному читателю продлить срок выдачи книги не более чем в два клика мышью (или тапа) без перехода на дополнительные страницы.
- Интерфейс библиотекаря должен позволять добавить карточку новой книги с заполнением обязательных полей за время не более 1 минуты при наличии готовых данных в 95% тестовых сценариев.
- Подсистема поиска должна обеспечивать, чтобы не менее 90% книг, считающихся релевантными экспертами по введённому ключевому слову, отображались на первой странице результатов поиска.
- Подсистема уведомлений должна доставлять электронное уведомление о наступлении просрочки возврата книги пользователю в течение не более 1 часа после наступления даты возврата при доступности внешнего сервиса уведомлений.

#### **4.5 Масштабируемость**

- Система приложений должна поддерживать одновременную работу не менее 200 активных пользователей при среднем количестве 3 запросов в минуту от одного пользователя без увеличения среднего времени ответа основных операций более чем на 20% относительно базового уровня.

2. Подсистема подключения библиотек должна позволять добавить одну новую библиотеку с заполнением конфигурационных данных и активацией в системе за время не более 24 часов без полной остановки сервиса.
3. Подсистема хранения данных должна поддерживать объём каталога до 100 000 записей материалов без роста среднего времени выполнения поискового запроса более чем до 2 секунд.
4. Система обработки операций выдачи и возврата должна выдерживать до 50 одновременных операций выдачи/возврата в секунду без снижения средней скорости ответа поиска по каталогу более чем на 15%.
5. Инфраструктурный слой системы должен автоматически масштабироваться горизонтально (добавление экземпляров приложений) при среднем использовании CPU выше 80% в течение 5 минут, снижая использование до уровня ниже 70% в течение последующих 10 минут.
6. Архитектура приложения должна обеспечивать возможность добавления новых функциональных модулей (например, поддержки новых типов материалов) путём развертывания отдельных сервисов без остановки существующих модулей более чем на 5 минут для одной библиотеки.

## 4.6 Поддерживаемость

1. Подсистема документации должна публиковать обновлённую пользовательскую и административную документацию в онлайн-доступе не позднее 3 рабочих дней после утверждения изменения функциональных требований.
2. Процесс внедрения новой функциональности в промышленное окружение должен занимать не более 2 недель с момента утверждения спецификации в 90% релизов.
3. Подсистема мониторинга должна отображать информацию о критических ошибках и сбоях всех сервисов в админ-панели с задержкой не более 1 минуты от момента возникновения события.
4. Подсистема аудита должна фиксировать любые изменения данных библиотекарем (создание, изменение, удаление материалов и читателей) в журнале аудита в 100% случаев, обеспечивая возможность отката на уровне базы данных.
5. Процесс модификации существующего модуля и его развёртывания в тестовом окружении не должен приводить к полной недоступности тестовой среды более чем на 1 час за один релиз.
6. Система автоматического тестирования должна обеспечивать покрытие автотестами не менее 80% критических пользовательских сценариев (поиск, бронирование, выдача, возврат, оплата штрафов), измеряемое метрикой покрытия по строкам кода и сценариям.

## 5 Заключение

В рамках данного проекта была проведена полная работа по системному анализу и проектированию информационной системы управления библиотекой. В результате выполненной работы были достигнуты следующие результаты:

- Проведён анализ стейкхолдеров системы и разработан план выявления требований, что позволило определить ключевых участников проекта и их интересы.
- Сформулированы бизнес-требования в формате SMART, включающие конкретные измеримые цели по увеличению эффективности работы библиотеки.
- Разработаны BPMN-диаграммы ключевых бизнес-процессов (выдача и возврат книг), которые наглядно демонстрируют логику работы системы.
- Создан Use Case Diagram, описывающий основные сценарии использования системы, и детализированы ключевые прецеденты использования с описанием основных и альтернативных сценариев.
- Разработана архитектура системы, включающая диаграммы контекста (C1), контейнеров (C2), компонентов (C3), классов (C4) и последовательностей, что обеспечивает полное понимание структуры и взаимодействия компонентов системы.
- Определены нефункциональные требования к системе, включающие требования к производительности, надёжности, безопасности, удобству использования, масштабируемости и поддерживаемости.

Разработанная система управления библиотекой решает поставленные задачи по автоматизации процессов выдачи и возврата книг, управления каталогом библиотечных материалов, бронирования и продления сроков выдачи. Архитектура системы обеспечивает масштабируемость, надёжность и безопасность, что позволяет эффективно использовать систему в условиях реальной эксплуатации.

Все требования и архитектурные решения документированы и готовы к использованию на этапе разработки и внедрения системы.