

Разработка приложения BooKing'sCross по обмену книгами

Елизавета Добрецова, группа 5040102/40101

Санкт-Петербургский Политехнический Университет

2024-2025

Преподаватель: Иванов Д.Ю.

Разработанное приложение представляет собой удобную платформу для бесплатного обмена книгами между пользователями. Оно позволяет находить и читать интересные и редкие книги, делиться своими книгами и обсуждать прочитанное.

Зачем нужно данное приложение?

- 1 **Доступность книг:** не у всех есть возможность покупать новые книги, в библиотеках не всегда есть в наличии нужные издания, а читать книги в электронном виде после целого дня работы за компьютером может быть тяжело.
- 2 **Экономия ресурсов:** обмены книгами позволяют сэкономить средства на покупке новых книг.
- 3 **Новые знакомства:** приложение объединяет людей с общими интересами, помогая пользователям познакомиться с единомышленниками.
- 4 **Эко-инициатива:** обмены книгами способствуют осознанному потреблению и бережному отношению к природным ресурсам, необходимым для производства бумаги.

1 Пользователи и роли

- *Читатель* - пользователь, берущий книгу другого пользователя на время.
- *Хранитель* - пользователь, предоставляющий свои книги для обмена.
- *Модератор* - пользователь с расширенными правами, проверяющий добавленные книги и отзывы, а также разрешающий спорные ситуации на платформе и следящий за корректностью поведения пользователей.

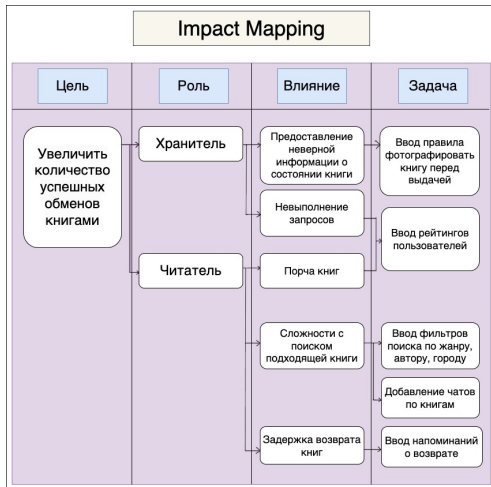
2 Процессы

- *Бронирование* - процесс резервирования книги Читателем у Хранителя.
- *Обмен* - процесс передачи книги от Хранителя Читателю и её последующего возврата.
- *Возврат книги* - процесс передачи книги Читателем Хранителю обратно по завершению срока обмена.
- *Добавление книги* - процесс загрузки Хранителем книги информации о ней в систему.
- *Модерация* - процесс проверки новых книг и новых отзывов Модератором перед публикацией объявления.
- *Оставление отзыва* - процесс написания Читателем отзыва о книге или о Хранителе или Хранителем о Читателе.

3 Данные

- *Книга* - объект обмена, включающий информацию о названии, авторе, состоянии, жанре и других характеристиках.
- *Запрос на бронирование* - заявка Читателя на получение книги у Хранителя.
- *История обменов* - список всех завершённых обменов, выполненных пользователем как в роли Читателя, так и в роли Хранителя.
- *Рейтинг и отзывы* - система оценки пользователей и книг на основе отзывов и рейтингов.
- *Личный чат* - функция для общения между Читателем и Хранителем с целью обсуждения условий обмена.
- *Чат по книге* - обсуждение книги между всеми пользователями, которые ею интересуются или уже брали ее в обмен.

Impact Mapping: стратегическое планирование



На данной диаграмме представлены ключевые пользователи приложения, их влияние на возникновение сложностей при достижении указанной цели и возможные действия, которые можно предпринять при разработке приложения, чтобы это влияние нивелировать.

User story stages: этапы пользовательской истории

| <u>Этапы процесса</u> <ul style="list-style-type: none">Регистрация в системеДобавление книгиПоиск и выбор книгиЗапрос книгиПередача книгиЧтение книгиВозврат книгиОценка и отзыв | Задачи | Работа с книгами | Взаимодействие пользователей друг с другом и с системой | Контроль и уведомления |
|--|---------|---|--|--|
| | Релиз 1 | <ul style="list-style-type: none">добавление книг в системупоиск книг | <ul style="list-style-type: none">регистрация и авторизация пользователейзапрос книги у хранителяорганизация передачи и возврата книги | |
| | Релиз 2 | <ul style="list-style-type: none">отзывы на книги | <ul style="list-style-type: none">чаты по книгамрейтинги пользователей | <ul style="list-style-type: none">напоминания о сроках возвратауведомления о превышении срока |
| | Релиз 3 | <ul style="list-style-type: none">рекомендация книг пользователям на основе отзывов | | <ul style="list-style-type: none">механизм разрешения спорных ситуаций |

Данная диаграмма отражает функционал приложения на каждом из трёх этапов внедрения. На первом этапе пользователям доступны только базовые функции (такие как регистрация и авторизация, добавление книг в систему, их поиск), на последующих же этапах будут появляться более продвинутые функции, чтобы приложение было более удобным и функциональным, такие, как, например, рекомендации книг пользователям и разрешение спорных ситуаций.

Logic of Interactions: логика выстраивания взаимодействий

Логика выстраивания взаимодействия

Акторы:

- **Пользователь:** человек, использующий приложение для обмена книгами
- **Хранитель:** пользователь, предоставляющий книгу для обмена
- **Читатель:** пользователь, желающий взять книгу

Объекты системы

- **Интерфейс пользователя:** визуальная часть приложения
- **Сервер:** обрабатывает запросы и управляет данными
- **База данных:** хранит информацию о пользователях, книгах и транзакциях

Последовательность взаимодействия при запросе книги:

1. **Читатель** через **Интерфейс пользователя** отправляет запрос на книгу.
2. **Интерфейс пользователя** передаёт запрос на **Сервер**.
3. **Сервер** обращается к **Базе данных** для проверки доступности книги.
4. **База данных** возвращает информацию о книге на **Сервер**.
5. **Сервер** отправляет подтверждение **Интерфейсу пользователя**.
6. **Интерфейс пользователя** уведомляет **Читателя** о статусе запроса.

Диаграмма демонстрирует, как пользователи и ключевые компоненты системы обмениваются данными в процессе запроса книги. Взаимодействие строится между тремя основными элементами: интерфейсом пользователя, сервером и базой данных. Читатель отправляет запрос на книгу, система обрабатывает его, проверяет доступность книги в базе данных и возвращает пользователю результат.

Use Case: пример использования приложения

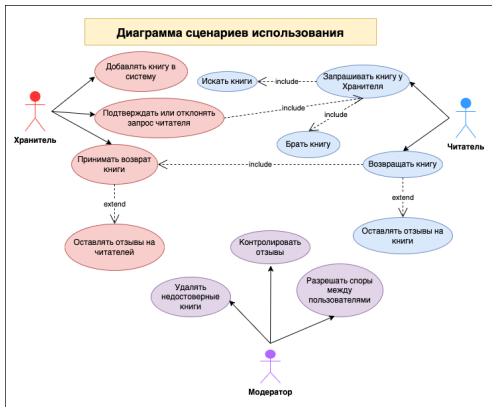
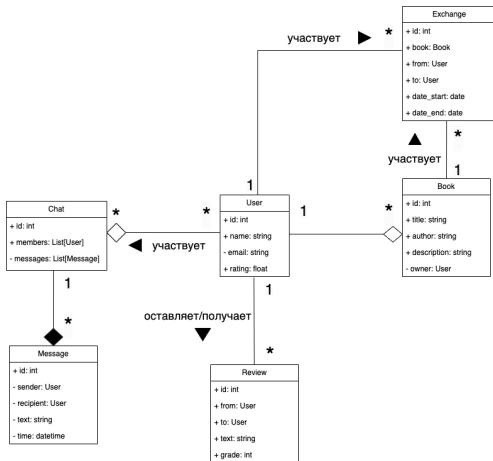


Диаграмма вариантов использования показывает основные сценарии работы с приложением для каждой из трёх ключевых ролей пользователя: Читатель, Хранитель и Модератор. Важно, что пользователь может совмещать роли Читателя и Хранителя, но не может ни одну из этих ролей совмещать с ролью Модератора.

Class Diagram: диаграмма классов



Основу приложения составляют классы *User*, *Book* и *Exchange*. Пользователи могут обмениваться книгами, иницируя обмены (*Exchange*), оставлять отзывы (*Review*), обсуждать прочитанные книги в чатах (*Chat*) и отправлять сообщения (*Message*). Диаграмма отражает связи между данными классами и их взаимодействие в системе.

Диаграммы активности (1/3)

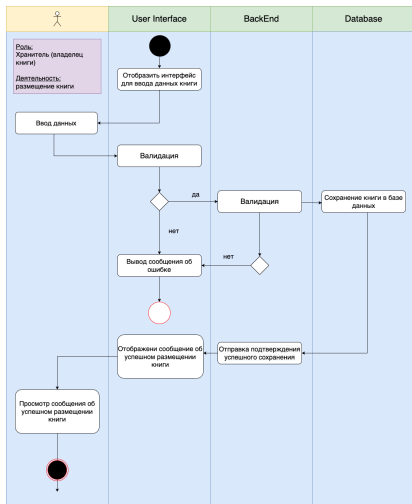


Диаграмма описывает процесс добавления новой книги в систему пользователем в роли Хранителя.

Основные этапы:

- 1 Хранитель открывает интерфейс добавления книги.
- 2 Затем заполняет информацию (название, автор, описание и т. д.).
- 3 Введённые данные отправляются на сервер.
- 4 Сервер проверяет корректность данных и записывает книгу в базу данных.
- 5 После успешного сохранения книги пользователю отображается сообщение об успешном её размещении.

Диаграммы активности (2/3)

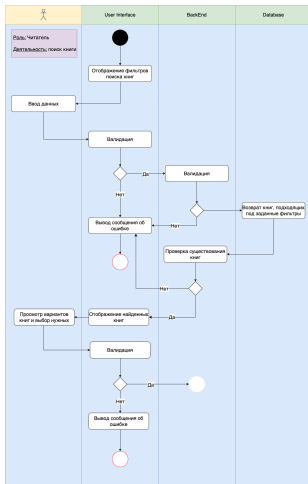


Диаграмма описывает поиск пользователем в роли Читателя книги в системе.

Основные этапы:

- 1 Читатель открывает страницу поиска.
- 2 Вводит ключевые слова (например, название книги или автора).
- 3 Сервер обрабатывает запрос, обращается к базе данных и возвращает список подходящих книг.
- 4 Читатель просматривает результаты и выбирает интересующую его книгу.

Диаграммы активности (3/3)

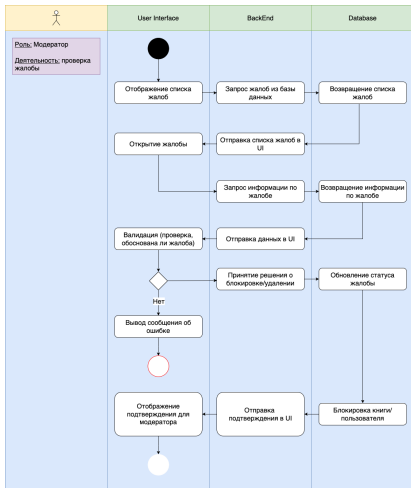


Диаграмма описывает процесс обработки жалоб Модератором.

Основные этапы:

- 1 Модератор получает список жалоб в интерфейсе.
- 2 Затем он выбирает конкретную жалобу для рассмотрения.
- 3 Система загружает информацию о жалобе и связанном контенте.
- 4 Модератор анализирует жалобу и принимает решение:
 - Если жалоба необоснованна, то она отклоняется.
 - Если жалоба обоснована, то объявление или пользователь могут быть заблокированы.
- 5 Система обновляет статус жалобы и уведомляет пользователя о решении.

High Level UI: высокоуровневая диаграмма интерфейса

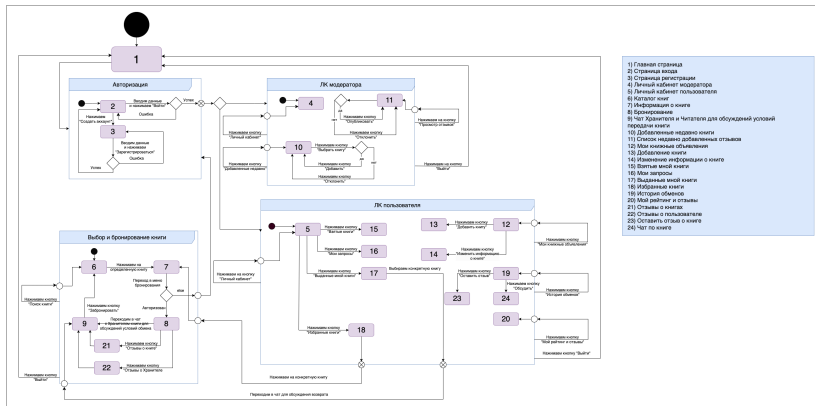
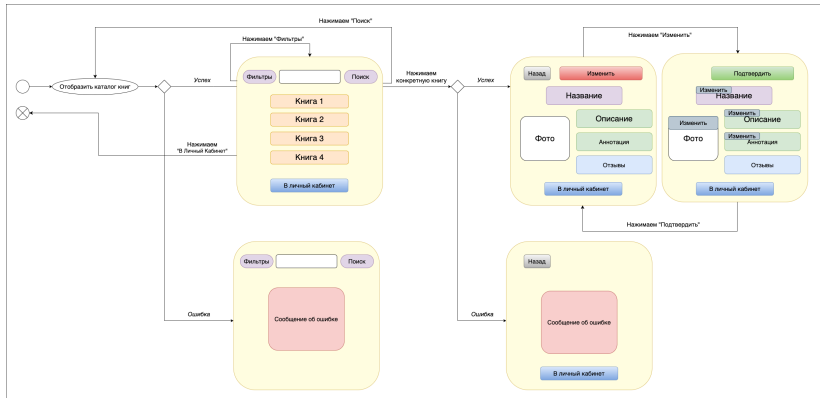


Диаграмма показывает основные экраны приложения и связи между ними.

Low Level UI: Низкоуровневая диаграмма интерфейса



Здесь детализирован процесс поиска и бронирования книги. Пользователь вводит запрос, просматривает результаты, выбирает книгу и отправляет запрос Хранителю. После подтверждения стороны договариваются о передаче через чат.

Component Diagram: диаграмма компонентов

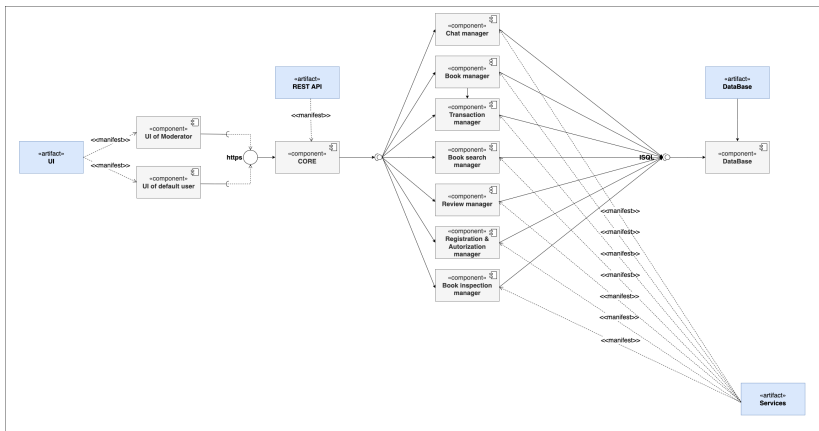


Диаграмма компонентов отображает структуру приложения, разделяя его на модули. Основные части включают интерфейс, API, логику приложения и базу данных. Это помогает понять, как взаимодействуют разные части системы.

Deployment Diagram: диаграмма компонентов

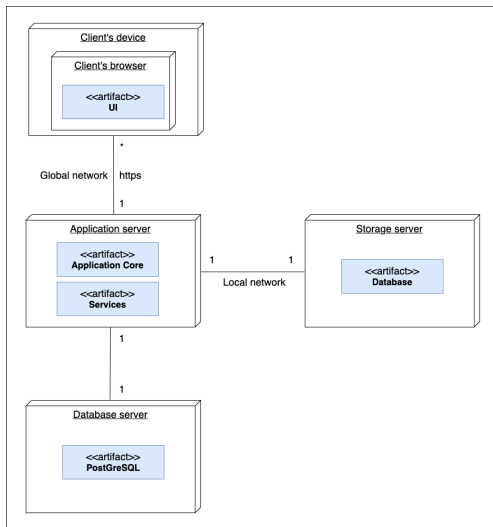


Диаграмма развертывания показывает, как программные компоненты размещены на серверах. Она включает клиентскую часть (веб-интерфейс), серверную часть (бэкенд) и базу данных, обеспечивая стабильную работу приложения.

В ходе работы над приложением по обмену книгами был разработан интерфейс и продумана логика взаимодействия между пользователями и объектами системы, чтобы приложение было удобным и безопасным. UML-диаграммы помогли структурировать архитектуру этого приложения, визуализировать взаимодействия и продумать ключевые процессы, что сделало разработку более чёткой и последовательной.