

1. Диаграмма классов

Важнейшим классом модели является **пользователь (User)**, который может быть владельцем библиотеки или читателем. Пользователь создаёт **библиотеку (Library)**, где хранятся его книги.

Каждая книга представлена сущностью **BookFile**, в которой указаны метаданные: название, автор, формат файла, контрольная сумма и дата загрузки.

Через промежуточную сущность **LibraryBook** книга связывается с библиотекой и получает дополнительные пользовательские характеристики — рейтинг, категорию и статус прочтения.

В процессе чтения формируется объект **Reading**, который хранит информацию о последней прочитанной странице, прогрессе и общем времени чтения.

К сущности чтения относятся **закладки (Bookmark)** и **цитаты (Quote)** — это вспомогательные объекты, которые пользователь создаёт для удобства навигации и сохранения интересных фрагментов.

Также присутствует сущность **Genre**, определяющая жанры и теги книг, и **ShareInvitation**, позволяющая пользователям делиться своей библиотекой с другими.

2. Диаграмма вариантов использования

Основные акторы — **владелец библиотеки** и **читатель**.

Для некоторых сценариев также участвует **система хранения**, обеспечивающая работу с файлами книг.

Владелец библиотеки может загружать книги, просматривать их список, организовывать коллекцию и делиться библиотекой с другими пользователями.

Читатель, в свою очередь, может открывать книги для чтения, сохранять прогресс, создавать закладки и цитаты, а также искать нужные книги по названию или автору.

3. Диаграмма деятельности

Процесс начинается с того, что пользователь выбирает книгу из библиотеки. Система загружает необходимые данные (название, обложку, содержимое) и открывает интерфейс для чтения.

Далее действия развиваются параллельно:

- пользователь листает страницы и читает книгу;
- система отслеживает, где именно пользователь остановился, и периодически сохраняет эту информацию.

Если пользователь закрывает книгу, система автоматически сохраняет все данные о прогрессе — последнюю страницу, процент прочтения и время последнего открытия. После этого отображается подтверждение, что прогресс успешно сохранён.

4. Нефункциональные требования

Производительность

- Интерфейс чтения должен работать без заметных задержек.
Время отклика при переходе между страницами — не более 300 миллисекунд.
- Загрузка книги объёмом до 200 МБ должна выполняться не дольше 5 секунд.

Надёжность

- Прогресс чтения сохраняется автоматически каждые несколько секунд или при закрытии книги.
- В случае ошибки загрузки система должна корректно уведомлять пользователя и записывать информацию о сбое в журнал.

Доступность

- Сервис чтения и просмотра книг должен быть доступен не менее чем 99% времени в течение месяца.

Безопасность

- Передача данных между клиентом и сервером осуществляется только по защищённому протоколу HTTPS.