Спецификация требований к программному обеспечению для приложения библиотеки

**1. Введение**

Данный документ представляет собой спецификацию требований к программному обеспечению (ПО) для информационной системы библиотеки. Он содержит описание назначения, функциональных и нефункциональных требований, ограничений и допущений, а также ссылки на сопутствующую документацию. Спецификация предназначена для использования всеми участниками процесса разработки — от проектирования до внедрения и сопровождения системы.

**1.1 Назначение**

Настоящая спецификация охватывает требования к веб-приложению **Library-KNP**, версии 1.0. Цель приложения — автоматизация процессов управления библиотечным фондом, включая учет книг, авторов, жанров, пользователей и истории выдачи. Документ адресован следующим категориям пользователей:

* **Разработчики** — для реализации функциональности в соответствии с требованиями.
* **Тестировщики** — для создания и выполнения сценариев тестирования.
* **Менеджеры проекта** — для контроля объема работ и сроков реализации.
* **Пользователи** (библиотекари, администраторы) — для понимания функций, предоставляемых системой.
* **Составители документации** — для подготовки пользовательских руководств и справочных материалов.

Если проект будет масштабироваться, предполагается обновление спецификации в соответствии с изменениями архитектуры или бизнес-требований.

**1.2 Соглашения, принятые в документах**

В данной спецификации используются следующие соглашения:

* **Жирным** выделены названия компонентов системы и ключевые элементы пользовательского интерфейса.
* *Курсивом* выделены термины, пояснения или обозначения, требующие внимания.
* Все требования нумеруются согласно шаблону REQ-ХХХ, где ХХХ — уникальный идентификатор.
* UML-диаграммы и схемы отображаются с помощью стандартной нотации Unified Modeling Language.

Если иное не указано, в тексте используется стиль, принятый в документации IEEE 830-1998.

**1.3 Границы проекта**

Программное обеспечение **Library-KNP** предназначено для внедрения в библиотеках различного масштаба — от школьных и вузовских до городских и частных. Оно должно обеспечивать:

* Хранение и редактирование информации о книгах, авторах и жанрах.
* Регистрацию пользователей библиотеки и отслеживание истории выдач.
* Поиск и фильтрацию книг по различным параметрам.

Приложение будет разрабатываться с учетом возможности последующего масштабирования и интеграции с внешними системами. В данном документе представлены требования к первой версии системы, реализующей базовую функциональность.

**1.4 Ссылки**

Ниже приведены документы и ресурсы, на которые имеются ссылки в этой спецификации:

* ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»
* IEEE Std 830-1998 «Recommended Practice for Software Requirements Specifications»
* Руководство по стилям пользовательского интерфейса (UI Style Guide)

**2. Общее описание**

В этом разделе представлен обзор программного продукта **Library-KNP**, включая его контекст, целевую аудиторию, рабочую среду, а также ограничения и зависимости, которые следует учитывать при разработке и внедрении системы.

**2.1 Общий взгляд на продукт**

**Library-KNP** — это новое программное решение, разработанное для автоматизации библиотечных процессов. Оно создается как самостоятельный продукт, не являющийся продолжением или модификацией существующей системы. Приложение предназначено для использования в библиотеках различных масштабов и может быть адаптировано под требования конкретных учреждений.

Приложение будет функционировать как клиент-серверная веб-система. Клиентская часть предоставляет интерфейс для пользователей через браузер, серверная часть — отвечает за обработку данных, взаимодействие с базой данных и реализацию бизнес-логики.

Система предполагает возможную интеграцию с внешними сервисами, например:

* сервисами каталогизации (например, ISBN API),
* системой внутренней аутентификации учреждения (LDAP или Active Directory),
* электронными библиотеками.

(Контекстная диаграмма может быть добавлена в будущем приложением к данной спецификации.)

**2.2 Классы и характеристики пользователей**

Система ориентирована на следующие классы пользователей:

* **Библиотекари (администраторы фонда)**  
  Основные функции: добавление, редактирование и удаление информации о книгах, авторах, жанрах, пользователях. Управление выдачей книг и возвратом. Имеют расширенные права доступа.  
  Уровень технической подготовки: средний.
* **Читатели (пользователи библиотеки)**  
  Основные функции: поиск и просмотр доступных книг, подача заявок на бронирование. Уровень доступа ограничен.  
  Уровень технической подготовки: базовый.
* **Системные администраторы**  
  Обеспечивают развертывание, настройку и поддержку серверной части приложения. Управляют правами доступа и интеграциями с внешними системами.  
  Уровень технической подготовки: высокий.

Каждому классу будет доступен различный набор функций, с учетом его полномочий и предполагаемой подготовки.

**2.3 Операционная среда**

Система будет работать в следующей среде:

* **Клиентская часть:**
  + Веб-браузеры: Google Chrome (последняя версия), Mozilla Firefox, Microsoft Edge.
  + Устройства: настольные компьютеры, ноутбуки; опционально — планшеты.
* **Серверная часть:**
  + ОС: Windows Server 2019 или Ubuntu 22.04 LTS.
  + Веб-сервер: IIS или Nginx.
  + СУБД: PostgreSQL 15.
  + Язык серверной логики: C# (ASP.NET Core) или Node.js (в зависимости от окончательного технологического стека).
  + Размещение: внутренняя инфраструктура организации или облачные платформы (например, AWS, Azure).
* **География пользователей**: приложение предназначено для локального или регионального использования, но архитектура допускает расширение на национальный или международный уровень.

**2.4 Ограничения дизайна и реализации**

* Интерфейс должен быть адаптирован для экранов с минимальным разрешением 1280×720.
* Время отклика системы на клиенте не должно превышать 2 секунд при стандартных операциях (поиск, просмотр карточки книги и т.п.).
* Реализация интерфейса должна соответствовать внутреннему гайдлайну по доступности (WCAG 2.1, уровень AA).
* Использование внешних коммерческих библиотек должно быть согласовано с юридическим отделом организации (по лицензиям).
* При проектировании системы необходимо учитывать возможность локализации интерфейса на другие языки.

**2.5 Предположения и зависимости**

**Предположения:**

* Пользователи (библиотекари и читатели) обладают базовыми навыками работы с веб-приложениями.
* Все участники проекта (разработчики, тестировщики, заказчик) используют одну и ту же версию документации и терминологии.
* Система будет разворачиваться в среде с доступом к интернету (для обновлений, работы с внешними API и т.д.).

**Зависимости:**

* Необходимо наличие .NET 8 или Node.js 20 на сервере, в зависимости от выбранного стека.
* Для работы модуля аутентификации требуется доступ к внутреннему LDAP-серверу организации.
* Некоторые функции (например, автоматическое заполнение карточек книг) зависят от доступности внешнего ISBN API.
* Работа системы невозможна без подключения к централизованной СУБД PostgreSQL.

**3. Функции системы**

**3.1 Поиск книг**

**3.1.1 Описание**

Функция позволяет пользователю находить книги по различным критериям: названию, автору, жанру, году издания и другим параметрам. Приоритет: **высокий**.

**3.1.2 Функциональные требования**

* **REQ-301**: Система должна предоставлять строку поиска с возможностью ввода части названия книги.
* **REQ-302**: Система должна поддерживать поиск по имени автора.

**3.2 Фильтрация книг**

**3.2.1 Описание**

Функция позволяет пользователю отфильтровывать книги по жанру, году издания, доступности и другим параметрам. Приоритет: **высокий**.

**3.2.2 Функциональные требования**

* **REQ-310**: Пользователь может выбрать один или несколько жанров для фильтрации.
* **REQ-311**: Пользователь может указать диапазон лет публикации.

**3.3 Просмотр статистики прочитанных книг**

**3.3.1 Описание**

Функция позволяет пользователю просматривать историю прочитанных книг и статистику по ним. Приоритет: **средний**.

**3.3.2 Функциональные требования**

* **REQ-320**: Пользователь может просматривать список прочитанных книг с датами чтения.
* **REQ-321**: Система отображает общее количество прочитанных книг.
* **REQ-322**: Система показывает диаграмму по жанрам или авторам (опционально).

**3.4 Бронирование книг**

**3.4.1 Описание**

Функция позволяет пользователям бронировать книги для последующего получения. Приоритет: **высокий**.

**3.4.2 Функциональные требования**

* **REQ-330**: Пользователь может подать заявку на бронирование книги, если она недоступна.
* **REQ-333**: Библиотекарь может видеть список бронирований и изменять их статус.

**3.5 Управление фондами (для библиотекаря)**

**3.5.1 Описание**

Функция позволяет библиотекарю добавлять, редактировать и удалять записи о книгах, авторах и жанрах. Приоритет: **высокий**.

**3.5.2 Функциональные требования**

* **REQ-340**: Библиотекарь может добавлять новые книги, указывая все метаданные.
* **REQ-341**: Библиотекарь может редактировать информацию о книгах.
* **REQ-342**: Библиотекарь может удалять книги из фонда.

**3.6 Управление пользователями (для библиотекаря)**

**3.6.1 Описание**

Функция позволяет библиотекарю управлять учетными записями пользователей. Приоритет: **средний**.

**3.6.2 Функциональные требования**

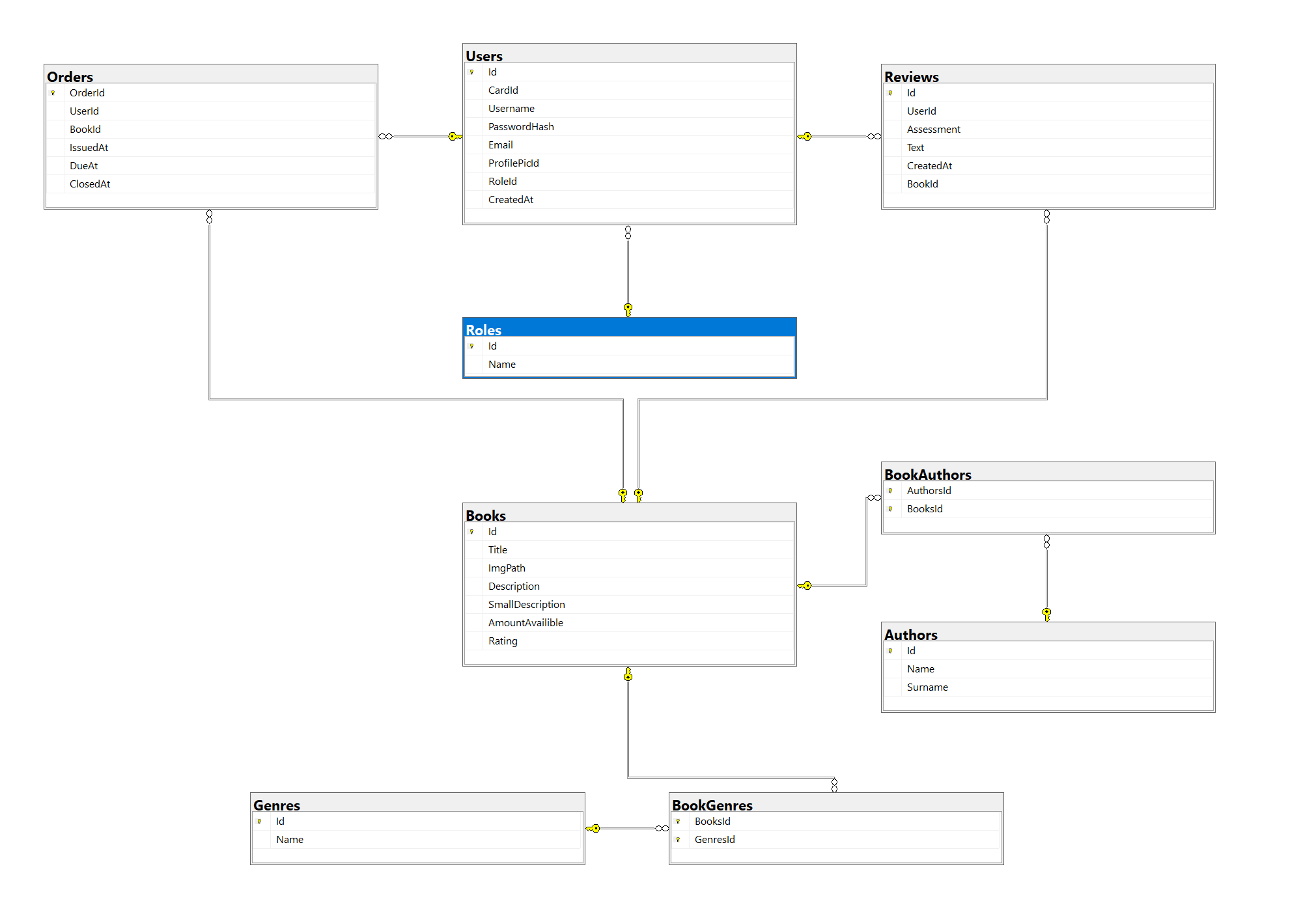
* **REQ-370**: Библиотекарь может создавать и удалять учетные записи.
* **REQ-371**: Библиотекарь может блокировать и разблокировать пользователей.

**4. Требования к данным**

**4.1 Логическая модель данных**

Система работает с такими основными сущностями, как:

* **Книга** (Book) — содержит информацию о названии, авторе, жанре, рейтинге, кол-ве доступных экземпляров.
* **Пользователь** (User) — содержит данные о зарегистрированном пользователе, включая номер карты читателя, логин, e-mail, роль
* **Бронирование** (Order) — связывает пользователя с книгой и содержит дату начала и окончания брони.
* **Жанр**, **Автор, Роль** — справочные сущности.
* **Отзыв**(Review) — связывает пользователя с книгой и содержит оценку, рецензию дату оставления отзыва.



**4.2 Словарь данных**

| **Поле** | **Сущность** | **Тип данных** | **Длина/Формат** | **Описание** |
| --- | --- | --- | --- | --- |
| id | Все сущности | INT |  | Уникальный идентификатор |
| title | Book | STRING | до 255 символов | Название книги |
| Review | Book | double |  | Оценка книги |
| Description | Book | STRING | до 1000 символов | Внешний ключ на таблицу жанров |
| ImgPath | Book | STRING | до 500 символов | Путь к файлу с электронной версией |
| username | User | STRING | до 100 символов | Логин пользователя |
| cardid | User | int | до 100 символов | Логин пользователя |
| hashedPassword | User | STRING | до 100 символов | Результат Хэш функции с введённым пользователем паролем |
| ProfilePicId | User | STRING | до 100 символов | Результат Хэш функции с введённым пользователем паролем |
| email | User | STRING | EMAIL | Адрес электронной почты |
| roleId | User | Int |  | Внешний ключ на таблицу ролей |
| UserId | Order | Int |  | Внешний ключ на таблицу пользователей |
| BookId | Order | Int |  | Внешний ключ на таблицу книг |
| IssuedAt | Order | DATE | ISO 8601 | Дата начала бронирования |
| DueAt | Order | DATE | ISO 8601 | Дата для сдачи |
| ClosedAt | Order | DATE | ISO 8601 | Дата сдачи |

**4.3 Получение, целостность, хранение и утилизация данных**

* **Получение данных**:
  + Изначально база заполняется библиотекарем вручную или через импорт CSV.
  + Пользовательские данные вносятся при регистрации.
  + История и статистика формируются автоматически на основе действий пользователей.
* **Целостность данных**:
  + Все поля с внешними ключами валидируются при сохранении.
  + Для полей типа email и year предусмотрены маски ввода и серверная валидация.
  + Используются транзакции при обновлении связанных записей.
* **Хранение данных**:
  + Все данные хранятся в реляционной базе данных.
  + Резервное копирование БД — ежедневно, с сохранением копий за 7 дней.
* **Утилизация данных**:
  + Метаданные (например, лог входа) хранятся 1 год, затем архивируются.

**5. Требования к внешним интерфейсам**

**5.1 Пользовательские интерфейсы**

**Library-KNP** предоставляет пользователю доступ к функциям приложения через графический пользовательский интерфейс (GUI), доступный в веб-браузере и десктопной версии. Интерфейс должен быть интуитивно понятным, адаптированным к различным устройствам, в том числе ноутбукам и мониторам с разрешением от 1280×720 и выше. Основные требования:

* **Стандарты**: интерфейс следует рекомендациям [Material Design] и корпоративным стилевым гайдлайнам.
* **Шрифты**: по умолчанию используется шрифт Segoe UI или Roboto, размер шрифта от 12 до 16 pt.
* **Цветовая схема**: тёмная тема (по умолчанию) с возможностью переключения на светлую.
* **Кнопки и ссылки**: единый стиль кнопок: прямоугольные с закруглёнными краями, цвет кнопок соответствует значению действия (например, зелёный — подтвердить, красный — отменить).
* **Сочетания клавиш**:
  + Ctrl+F — поиск книги.
  + Ctrl+S — сохранить изменения.
  + Esc — отмена текущего действия или закрытие окна.
* **Проверка данных**:
  + Email: проверяется формат.
  + Год издания: целое число в диапазоне 1000–текущий год.
  + Поля, обязательные к заполнению, помечаются \*.
* **Адаптация под локализацию**: все тексты вынесены в отдельный словарь; поддерживаются минимум два языка — русский и английский.
* **Доступность**:
  + Поддержка навигации с клавиатуры.
  + Контрастность интерфейса не менее 4.5:1.
  + Атрибуты aria-label и role для элементов интерфейса.

**5.2 Интерфейсы ПО**

**Library-KNP** взаимодействует со следующими внешними компонентами:

* **База данных**: PostgreSQL 14+.
  + Обмен данными в формате SQL и JSON.
  + Поддержка транзакций, внешних ключей, процедур и представлений.
* **Авторизационный сервис**: OAuth 2.0 / OpenID Connect.
  + Используется для входа пользователей и получения JWT-токенов.
* **Сторонний сервис загрузки книг** (например, National Library API).
  + REST API, формат запросов/ответов: JSON.
  + Используются методы: GET /books, POST /books.
* **Файловое хранилище** (локальное или S3-совместимое).
  + Книги в формате PDF, EPUB, DOCX.
  + Ограничение по размеру файла — 50 МБ.
* **Почтовый сервер**:
  + SMTP-протокол.
  + Поддержка TLS.
  + Формат уведомлений: HTML + plaintext.

**5.3 Интерфейсы оборудования**

**Library-KNP** в минимальной конфигурации не требует специфического оборудования, за исключением:

* **Рабочая станция**: x64 процессор, минимум 4 ГБ ОЗУ, экран с разрешением 1280×720.
* **Сканер штрихкодов** (опционально):
  + Подключение по USB.
  + Эмуляция ввода с клавиатуры (keyboard wedge).
  + Используется в библиотеке для быстрой регистрации книг.
* **Серверная часть**: Linux-сервер или Windows Server 2019+, от 2 CPU, 4+ ГБ ОЗУ, 50+ ГБ дискового пространства.

**5.4 Коммуникационные интерфейсы**

Для обеспечения сетевого взаимодействия приложение использует:

* **HTTPS**: для всех внешних и внутренних API-запросов, поддерживается TLS 1.2 и выше.
* **WebSocket**: используется для отображения уведомлений в реальном времени (например, о скором сроке сдачи книги).
* **SMTP**: для отправки писем о бронированиях, напоминаний и отчётов библиотекарям.
  + Ограничения: вложения не более 10 МБ, допустимы только .pdf, .xlsx.
* **REST API**:
  + Префикс маршрутов: /api/v1/
  + Ответы в формате JSON.
  + Авторизация через Bearer Token.

**6. Атрибуты качества**

**6.1 Удобство использования**

* Интерфейс приложения должен позволять новому пользователю освоить основные функции (поиск, бронирование, чтение книг) за не более чем 15 минут.
* Среднее время выполнения стандартной операции (например, бронирование книги или скачивание файла) не должно превышать 3 кликов.
* Для предотвращения ошибок все опасные действия (удаление, отклонение бронирования) должны сопровождаться подтверждением.
* Интерфейс должен включать всплывающие подсказки и системную справку.
* Поддержка специальных возможностей (WCAG 2.1, уровень AA):
  + Контрастность текста — не менее 4.5:1.
  + Навигация с клавиатуры.
  + Атрибуты aria-\* и поддержка скринридеров.

**Приоритет**: высокий.

**6.2 Производительность**

* Загрузка домашней страницы пользователя — не более **1,5 секунд** при стандартной скорости подключения 10 Мбит/с.
* Время выполнения поискового запроса (по ключевым словам и фильтрам) — не более **1 секунды** для локальной базы объемом до 50 тыс. записей.
* Массовая генерация отчета (до 1000 записей) — не более **10 секунд**.
* Система должна поддерживать одновременную работу не менее **200 пользователей** без деградации производительности.

**Приоритет**: высокий.

**6.3 Безопасность**

* Все данные пользователя (в том числе бронирования, история чтения) должны храниться в зашифрованном виде (например, AES-256).
* Аутентификация — через JWT с истечением срока действия в 1 час; возможность обновления токена.
* Авторизация — по ролям: администратор, библиотекарь, пользователь.
* Защита от XSS, CSRF, SQL-инъекций — обязательна.
* Журналирование входов и подозрительных действий, включая попытки доступа к запрещенным разделам.
* Соответствие требованиям **GDPR** для защиты персональных данных.

**Приоритет**: критически высокий.

**6.4 Техника безопасности**

* Все изменения и удаление данных требуют подтверждения или наличия прав администратора.
* Система должна сохранять резервную копию базы данных не реже одного раза в сутки.
* Восстановление системы из резервной копии должно занимать не более 1 часа.
* При потере соединения с сервером пользователь должен быть уведомлен, и его данные сохранены в локальном кеше (если возможно).

**Приоритет**: средний.

**6.5 Надежность и устойчивость**

* Система должна быть доступна не менее 99.5% времени в течение календарного месяца.
* В случае сбоя веб-сервера должна быть реализована автоматическая перезагрузка сервиса.
* Данные пользователя не должны теряться при обновлении системы или её аварийной остановке.

**Приоритет**: высокий.

**6.6 Масштабируемость**

* Приложение должно поддерживать горизонтальное масштабирование серверной части без изменений в архитектуре.
* Добавление новых модулей (например, рекомендательная система) не должно требовать значительной переработки кода основной системы.

**Приоритет**: средний.

**6.7 Поддерживаемость**

* Код должен быть документирован в соответствии с внутренним стандартом команды.
* Должна быть доступна внутренняя административная панель для просмотра логов, состояния базы данных и пользовательской активности.
* Все конфигурационные параметры должны храниться в отдельном конфигурационном файле (например, .env или appsettings.json).

**Приоритет**: средний.

**7. Требования по интернационализации и локализации**

**7.1 Поддержка языков**

* Система должна поддерживать переключение интерфейса между как минимум двумя языками: **русским** и **английским** (американский вариант).
* Все текстовые элементы интерфейса, уведомления, сообщения об ошибках и справочная информация должны храниться в отдельном файле ресурсов, поддерживающем расширение.
* В дальнейшем должна быть возможна локализация под другие языки без изменений в логике приложения.

**Приоритет**: высокий.

**7.2 Форматирование данных**

* Даты должны отображаться в локальном формате пользователя:
  + ДД.ММ.ГГГГ для русской локали.
  + MM/DD/YYYY для английской локали (США).
* Валюты (если будут использоваться, например, при учете штрафов) должны отображаться с учетом локали:
  + ₽ для русского языка.
  + $ для английского.
* Числа должны форматироваться в соответствии с локалью:
  + Разделитель разрядов — пробел (русский) или запятая (английский).
  + Десятичный разделитель — запятая (русский) или точка (английский).

**7.3 Форматы личных данных**

* Формы ввода имен должны быть независимыми от порядка "Имя-Фамилия" или "Фамилия-Имя" — пользовательский ввод не ограничивается конкретным порядком.
* Поля адресов должны быть адаптированы под международные стандарты (свободный ввод, без жёсткой валидации структуры).
* Номера телефонов должны поддерживать международный формат +<код страны> <номер>.

**7.4 Часовые пояса и время**

* Вся информация, связанная с датами и временем (бронирования, загрузки, отчёты), должна сохраняться в формате UTC и отображаться пользователю с учётом его локального часового пояса.
* Пользователь должен иметь возможность указать свой часовой пояс в настройках.

**7.5 Культурные и нормативные различия**

* В интерфейсе не должно быть символов, цветов, метафор и изображений, которые могут быть восприняты как культурно чувствительные или неуместные в разных странах.
* Приложение должно соответствовать требованиям законодательства о защите персональных данных в соответствующей юрисдикции (например, **GDPR** для пользователей из ЕС, **ФЗ-152** для РФ).

**8. Прочие требования**

**8.1 Лицензирование и юридические требования**

* Программное обеспечение должно использовать только **легальные и лицензированные библиотеки**, компоненты и фреймворки с открытым исходным кодом, лицензии которых допускают коммерческое использование и модификацию (например, MIT, Apache 2.0).
* Все сторонние компоненты и библиотеки должны быть задокументированы с указанием версии и лицензии.
* Система должна соответствовать законам о защите персональных данных:
  + В РФ — **ФЗ-152 "О персональных данных"**.
  + В ЕС — **GDPR**, если доступ к системе будет международным.

**8.2 Стандарты и соответствие**

* Приложение должно быть разработано в соответствии с принципами **SOLID**, паттерном **MVVM** (в части WPF-клиента) и с использованием RESTful-архитектуры при взаимодействии с сервером.
* Интерфейс должен соответствовать рекомендациям по UX/UI и быть интуитивно понятным для массовой аудитории (библиотекари, учащиеся, читатели).
* Для веб-части (если предусмотрена) — соответствие WCAG 2.1 уровня AA по доступности.

**8.3 Условия внедрения и поддержки**

* Развёртывание возможно как в **локальной инфраструктуре библиотеки**, так и на внешнем сервере (по решению заказчика).
* Предусматривается подготовка **пользовательского руководства** для библиотекарей и администраторов системы.
* Разработчик обязан предоставить **инструкции по установке**, **техническую документацию** и **описание API**, если предусмотрен доступ сторонних систем.
* Планируется поддержка системы не менее **1 года** с даты внедрения, включая исправление критических ошибок и консультации.

**8.4 Ограничения ответственности**

* Разработчик не несёт ответственности за данные, утерянные в результате **несвоевременного создания резервных копий** администраторами.
* Также не гарантируется корректная работа в случае изменения конфигурации среды, не предусмотренной требованиями (например, замена ОС, переход на неподдерживаемые версии библиотек).