**Отчет о проделанной работе (первая итерация)**

Оглавление

[1. Общие сведения 2](#_Toc196083223)

[1.1 Наименование проекта 2](#_Toc196083224)

[1.2 Цель и задачи 2](#_Toc196083225)

[2. Проделанная работа 3](#_Toc196083226)

[2.1 Проектирование и реализация базы данных 3](#_Toc196083227)

[2.2 Рецензирование и корректировка структуры базы данных 3](#_Toc196083228)

[2.3 UI/UX-дизайн 3](#_Toc196083229)

[2.4 Реализация серверной части 3](#_Toc196083230)

[2.5 Реализация клиентской части 3](#_Toc196083231)

[2.6 Документация 3](#_Toc196083232)

[3. Используемые инструменты и технологии 5](#_Toc196083233)

[3.1 Языки программирования и библиотеки 5](#_Toc196083234)

[3.2 База данных 5](#_Toc196083235)

[3.3 Система контроля версий 5](#_Toc196083236)

[3.4 Взаимодействие с внешними сервисами 5](#_Toc196083237)

[3.5 Дизайн и прототипирование 6](#_Toc196083238)

[3.6 Средства командной работы 6](#_Toc196083239)

[4. Заключение 7](#_Toc196083240)

[4.1 Результаты первой итерации 7](#_Toc196083241)

[4.2 Объем выполненных задач 7](#_Toc196083242)

[4.3 Предварительная оценка текущего состояния проекта 7](#_Toc196083243)

# Общие сведения

## 1.1 Наименование проекта

Система Управления Библиотекой (СУБ) – это веб-приложение, предназначенное для автоматизации ключевых библиотечных процессов: управление книжным фондом, бронирование, учет операций по выдаче и возврату книг, учет посетителей, формирования отчетности.

## 1.2 Цель и задачи

Цель первой итерации: реализация базовой архитектуры проекта, разработка ключевого пользовательского функционала, связанного с отображением каталога библиотеки и информации о книгах.

Задачи первой итерации:

1. Разработка структуры базы данных;
2. Разработка макетов и первичной стилистики интерфейса;
3. Реализация клиентской части для отображения каталога книг;
4. Подключение клиентской части к серверу и базам данных;
5. Начальное оформление проектной документации.

# Проделанная работа

## Проектирование и реализация базы данных

Ответственный: Павлов Владислав.

* + Спроектирована архитектура базы данных в соответствии с функциональными требованиями системы;
  + Созданы ключевые таблицы: users, roles, books, book\_copies, borrowings, reservations, reading\_room\_visits, notifications;
  + Реализованы связи между сущностями, заданы внешние ключи.

## Рецензирование и корректировка структуры базы данных

Ответственный: Толстолуцкая Полина.

* + Проведён анализ предложенной схемы на предмет полноты и логической целостности;
  + Внесены критически важные правки в структуру, скорректированы таблицы, уточнены атрибуты, удалены лишние поля, а именно:

1. Добавлено разделение книг по жанрам;
2. Удалено лишнее поле с указанием языка книги;
3. Добавлена фиксация исполнителя выдачи книги;
4. Добавлен учет количества мест в читальном зале.
   * Итоговая структура согласована для использования.

## UI/UX-дизайн

Ответственный: Шайсултанова Регина.

* Разработан первичный дизайн пользовательского интерфейса в Figma.

## Реализация серверной части

Ответственный: Павлов Владислав.

* Разработан минимальный сервер на фреймворке Ktor (Kotlin);
* Реализованы REST API-эндпоинты для получения данных о книгах;
* Настроена передача данных в формате JSON;
* Обеспечена интеграция API с базой данных через ORM-библиотеку Exposed.

## Реализация клиентской части

Ответственный: Шайсултанова Регина.

* Сверстаны ключевые страницы веб-приложения: каталог книг, страница с подробной информацией о книге;
* Обеспечена интеграция с базой данных: данные о книгах загружаются и отображаются на фронте.

## Документация

Ответственный: Толстолуцкая Полина.

* Организована структура проекта в репозитории (/frontend, /backend, /docs);
* Подготовлен стартовый файл README.md;
* Начата работа по организации GitHub Wiki, создан план разделов и подготовлены заглушки под содержание;
* Подготовлен первый отчет о проделанной работе.

# Используемые инструменты и технологии

## Языки программирования и библиотеки

1. Kotlin: Язык программирования, используется для разработки серверной части приложения. Имеет полную совместимость с экосистемой Java.

* Ktor: Фреймворк на Kotlin для создания асинхронных серверных приложений. Используется в качестве основы для бэкенда. Позволяет быстро поднимать REST API, легко конфигурируется, а также поддерживает расширяемость;
* Exposed: Kotlin SQL-DSL и ORM-библиотека, используемая для взаимодействия с базой данных PostgreSQL. Позволяет писать SQL-запросы в Kotlin-стиле, обеспечивая безопасность типов и удобочитаемость кода, а также упрощает миграции и работу с транзакциями.

1. JavaScript: Язык программирования, используется для разработки интерактивной части клиентского интерфейса. Применяется для управления логикой пользовательского взаимодействия, отправки запросов к серверу и динамического обновления интерфейса.

* Axios: JavaScript-библиотека для выполнения HTTP-запросов. Используется для отправки запросов к серверу, получения и обработки данных из API;
* HTML, CSS: Стандартные языки разметки и стилизации. Применяются для создания структуры и оформления веб-интерфейса. Использовались при верстке основных страниц: каталога книг, карточки книги и других компонентов интерфейса.

## База данных

PostgreSQL: Реляционная СУБД.

В рамках текущего проекта база данных содержит логически связные таблицы для хранения информации о пользователях, ролях, книгах, экземплярах, бронированиях, визитах и уведомлениях.

## Система контроля версий

Используется GitHub, как платформа для совместной работы, хранения репозитория и ведения документации.

## Взаимодействие с внешними сервисами

Imgur: Используется как хранилище изображений обложек книг. Позволяет загружать изображения и использовать прямые ссылки (URL), которые подставляются в карточки книг для отображения визуального контента без хранения изображений в собственной базе данных.

## 3.5 Дизайн и прототипирование

Figma: Приложение для создания макетов интерфейса и UI/UX-дизайна. В проекте использовалось для разработки макетов страниц и согласования визуального стиля.

## 3.6 Средства командной работы

Telegram: Используется как основной канал для общения команды, оперативного обсуждения возникающих задач, синхронизации действий участников проекта, проведения групповых звонков.

Yandex Tracker: Система управления задачами и отслеживания прогресса. Применяется для распределения обязанностей, контроля сроков и фиксации этапов разработки в рамках итерационного подхода.

# Заключение

## 4.1 Результаты первой итерации

В ходе первой итерации сформирован фундамент проекта:

* Разработана и утверждена структура базы данных;
* Выполнена верстка и начата интеграция клиентской части;
* Оформлена базовая документация, налажены процессы взаимодействия в команде.

## 4.2 Объем выполненных задач

Командой выполнены все запланированные задачи итерации. Реализована ключевая функциональность: отображение каталога и карточек книг с данными из базы, а также структура, позволяющая развивать проект в последующих итерациях.

## 4.3 Предварительная оценка текущего состояния проекта

Проект находится на стадии активной разработки. Заложен основной архитектурный фундамент, создана основа дизайна. Есть задел для расширения функциональности и перехода к более сложным модулям (авторизации, бронированию, выдаче книг и т.д.).