Техническое задание на разработку сервиса «Dino-Log»

Версия 1.3

2025

История изменения документа

|  |  |  |  |
| --- | --- | --- | --- |
| Версия | Дата | Автор | Изменения |
| 1.0 | 23.10.2025 | Мангер В. А. | Создание документа, заполнение общей информацией. |
| 1.1 | 23.10.2025 | Мангер В. А. | Редактирование оформления общей информации. |
| 1.2 | 29.10.2025 | Мангер В. А. | Добавлены новые требования и приложения для макета интерфейса. Доработано оформление документа. |
| 1.3 | 29.10.2225 | Кукаров К.А. | Обновлена информация и добавлена схема в архитектуре системы. |
|  |  |  |  |
|  |  |  |  |

Содержание

[Содержание 3](#_Toc212674017)

[Общие сведения 4](#_Toc212674018)

[Плановые сроки начала и окончания работы по созданию системы 4](#_Toc212674019)

[Назначение и цели создания системы 4](#_Toc212674020)

[Требования к системе 5](#_Toc212674021)

[Функциональные требования 5](#_Toc212674022)

[Нефункциональные требования 7](#_Toc212674023)

[Архитектура системы 8](#_Toc212674024)

[Приложения 9](#_Toc212674025)

Общие сведения

Система – веб-сайт «Dino-Log», специализированная платформа, посвященная фильмам, сериалам и играм о динозаврах. В отличие от существующих решений, Dino-Log предоставляет пользователям централизованный ресурс для поиска, просмотра информации, обсуждения и оценки медиаконтента, посвященного тематике динозавров.

Заказчиком выступает Шнейдер Аркадий Дмитриевич.

Исполнители: Кукаров Кирилл Антонович, Мангер Вероника Андреевна, Латогузов Степан Сергеевич.

Плановые сроки начала и окончания работы по созданию системы

Этапы создания информационной системы с плановыми сроками представлены в таблице.

|  |  |
| --- | --- |
| **Этап** | **Сроки** |
| Планирование и анализ требований | 8 октября – 10 октября |
| Определение требований | 13 октября – 17 октября |
| Проектирование системы | 20 октября – 5 ноября |
| Разработка системы | 7 ноября – 20 декабря |
| Тестирование системы | 21 декабря – 31 декабря |
| Развертывание системы | 2 января – 20 января |

Стоимость разработки системы включает в себя оплату труда двух разработчиков (Backend и Frontend), аренду сервера/хостинга, затраты на приобретение домена и кастомизацию шаблонов дизайна. Общая стоимость разработки MVP (минимальной функциональности) сайта Dino-Log составляет примерно 497 500 рублей. Данная оценка основана на следующих допущениях: длительность MVP - 3 месяца, средняя зарплата разработчика (Backend или Frontend) - 80 000р/мес, минимальная стоимость сервера/хостинга - 2000р/мес, разовая оплата за дизайн 10 000р. Реальная стоимость может отличаться в зависимости от многих факторов.

Назначение и цели создания системы

Система предназначена для предоставления пользователям удобной и централизованной платформы для поиска, просмотра информации, обсуждения и оценки медиаконтента, посвященного тематике динозавров.

Цели создания системы с точки зрения заказчика и пользователей:

* Создать базу данных фильмов, сериалов и игр о динозавров.
* Предоставить пользователям инструменты для поиска, фильтрации и сортировки контента.
* Обеспечить возможность добавления отзывов и оценок контента.
* Создать сообщество пользователей, увлеченных тематикой динозавров.

Требования к системе

Функциональные требования

**Регистрация и авторизация:**

* Любой пользователь имеет возможность зарегистрироваться на сайте, указав логин, пароль и email.
* Любой пользователь имеет возможность авторизоваться на сайте, указав логин и пароль.
* Пользователь может восстановить пароль через email.
* Пользователь может изменить свои данные (email, пароль) в личном кабинете.

**Каталог контента:**

* Система содержит каталог фильмов, сериалов и игр, посвященных тематике динозавров.
* Каждый элемент каталога содержит следующую информацию:
* Название.
* Год выпуска.
* Жанр.
* Описание.
* Рейтинг.
* Трейлер (если доступен).
* Постер.
* Пользователь может просматривать список элементов каталога.
* Пользователь может фильтровать каталог по типу контента (фильмы, сериалы, игры).

**Панель модерации контента:**

* Модератор имеет доступ к отдельной панели для проверки вновь добавленных материалов.
* В панель поступают все новые карточки контента и отзывы, требующие проверки.
* После одобрения модератором материал публикуется в общем каталоге.
* Модератор может редактировать, отклонять или удалять записи.

**Управление пользователями:**

* Администратор имеет доступ к панели управления пользователями.
* В панели отображаются все зарегистрированные аккаунты, с возможностью блокировки, изменения ролей и восстановления доступа.
* Модераторы назначаются вручную администратором.

**Добавление нового контента пользователями:**

* Зарегистрированные пользователи могут предлагать новый контент (фильмы, сериалы, игры).
* При добавлении указываются поля: название, год, описание, жанр, ссылка на трейлер, постер.
* Перед публикацией запись проходит модерацию.

**Автоматическое наполнение каталога (через API):**

* Реализуется интеграция с TMDb API для автоматического импорта фильмов о динозаврах.
* Модератор вручную подтверждает корректность импортированных данных.

**Система уведомлений:**

* Пользователь получает уведомления о новых комментариях под своими отзывами.
* Администратор и модераторы получают уведомления о новых предложениях контента.

**Просмотр карточки контента:**

* Пользователь может перейти на страницу отдельного элемента каталога, где представлена подробная информация об этом элементе.
* Страница содержит:
* Название.
* Год выпуска.
* Жанр.
* Описание.
* Рейтинг.
* Трейлер (если доступен).
* Постер.
* Список отзывов.
* Форму для добавления отзыва.
* Пользователь может посмотреть трейлер прямо на странице (если доступен).

**Отзывы и рейтинги:**

* Зарегистрированный пользователь может оставить отзыв к любому элементу каталога.
* При добавлении отзыва пользователь может выставить оценку по шкале от 1 до 10.
* Отзыв должен содержать текст отзыва и оценку.
* Пользователь может просматривать отзывы других пользователей.
* На основе оценок пользователей вычисляется средний рейтинг элемента каталога.
* Пользователь может просматривать список отзывов, отсортированных по дате добавления или рейтингу.

**Поиск и фильтрация:**

* Пользователь может использовать поисковую строку для поиска контента по названию.
* Пользователь может фильтровать контент по типу (фильмы, сериалы, игры), жанру, и году выпуска.

Нефункциональные требования

**Требования к дизайну**

* Красочный и привлекательный дизайн, ассоциирующийся с тематикой динозавров.
* Удобная навигация и интуитивно понятный интерфейс.
* Использование фирменного стиля, включающего логотип и цветовую схему.

**Требования к адаптивности**

* Сайт должен корректно отображаться на различных устройствах (десктопах, планшетах, смартфонах) с разными разрешениями экрана.
* Должна быть поддержка различных браузеров (Chrome, Yandex).

**Требования к безопасности**

* Защита от SQL-инъекций, XSS-атак и других распространенных веб-угроз.
* Хранение паролей пользователей в зашифрованном виде (с использованием алгоритма хеширования).
* Защита от DDoS-атак.

**Требования к производительности**

* Время загрузки страниц не должно превышать 3 секунды.
* Сайт должен выдерживать большое количество одновременных пользователей.

**Требования к интеграции с внешними API:** Интеграция с TMDb API для получения информации о фильмах.

**Требования к локализации:** Наличие мультиязычности (русский, английский)

**Требования к интерфейсу:**

* Интерфейс должен быть адаптивным, современным, с элементами динамики (React).
* Цветовая палитра оформляется в фирменном стиле проекта (см. Приложение «Цветовая схема»).
* Поддерживаются две темы: *светлая и тёмная.*
* На каждой странице должна быть видна навигационная панель, логотип, поисковая строка и кнопка смены темы.

Архитектура системы

**Основные сведения**

Сервис представляет собой веб-приложение, доступное на любых устройствах благодаря адаптивной верстке. Архитектура системы основана на клиент-серверной модели и разделена на три слоя: базовый, дополнительный и внешний.

**Компоненты архитектуры**

*Базовый слой* включает основные компоненты, обеспечивающие функциональность и взаимодействие системы:

* Бэкенд: реализуется на платформе Spring Boot (Java), обрабатывает бизнес-логику: регистрацию, авторизацию, управление контентом, отзывами, рейтингами, модерацией. Взаимодействует с базой данных через ORM (Hibernate).
* Фронтенд: реализуется на React и Vue.js с использованием HTML, CSS и JavaScript. Обеспечивает интерактивный пользовательский интерфейс, отправляет запросы на сервер через REST API.
* База данных: используется PostgreSQL в качестве основной реляционной СУБД для хранения структурированных данных приложения.

*Дополнительный слой* предназначен для повышения производительности, мониторинга и аналитики:

* Кэширование: реализуется с помощью Redis, используется для ускорения обработки часто запрашиваемых данных и уменьшения нагрузки на основную базу данных (PostgreSQL). Взаимодействие осуществляется через библиотеку Lettuce.
* Сбор метрик: осуществляется при помощи Prometheus. Spring Boot экспортирует метрики (время отклика, количество запросов, ошибки) в Prometheus.
* Визуализация метрик: осуществляется через Grafana, которая подключается к Prometheus и отображает метрики в виде дашбордов. Взаимодействие осуществляется через PromQL.
* Парсеры: Скрипты на Python автоматически собирают данные о медиа-контенте с внешних источников. Передача данных в бэкенд осуществляется через внутренние API или очередь (Pull-механизм).

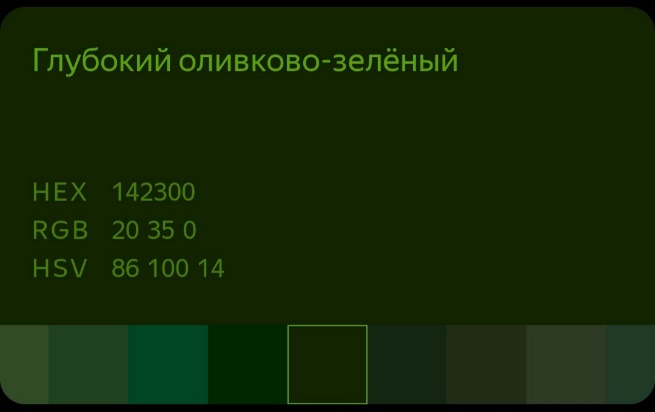
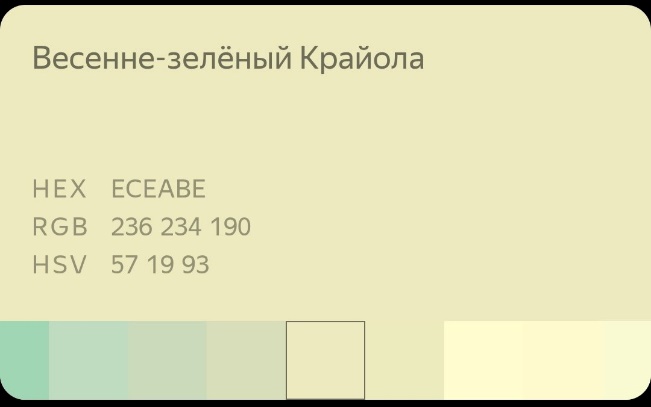
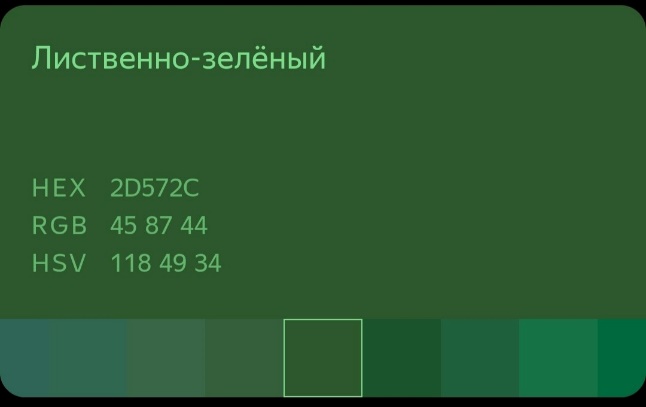
*Внешний слой* содержит интеграции с внешними системами и сервисами:

* Хранилище файлов: используется S3-совместимое хранилище Yandex Cloud и Google Cloud для загрузки и хранения медиафайлов. Взаимодействие с бэкендом осуществляется через REST API.
* Сторонние базы данных медиа: TMDb API используются для автоматического импорта карточек фильмов о динозаврах. Spring Boot отправляет REST-запросы, полученные данные проходят модерацию и сохраняются в PostgreSQL.

Всё это можно увидеть в схеме (см. Приложение «Схема архитектуры»).

Приложения

**Приложение 1. Цветовая схема:**

Основные цвета интерфейса:  
  
Используется плавный градиент перехода, где необходимо.

**Приложение 2. Архитектура системы:**

