**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| **КП.**09.02.071.24.221.16ПЗ |  |

**ВЕБ-ПРИЛОЖЕНИЕ «**ИГРОВОЙ СПРАВОЧНИК»

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Председатель ВЦК: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Н.Р. Огородникова) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Е.С. Кубата) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Д.М. Рединов) |

Иркутск

2025

**СОДЕРЖАНИЕ**

**Элементы оглавления не найдены.**

# ВВЕДЕНИЕ

В современном мире компьютерные игры стали неотъемлемой частью досуга миллионов людей, предлагая не только развлечение, но и возможность для общения, соревнований и совместного времяпрепровождения. С ростом популярности многопользовательских проектов, таких как игра «Honkai: Star Rail», увеличивается потребность в удобных инструментах, которые помогают игрокам лучше ориентироваться в игровом процессе, изучать механики и планировать развитие своих персонажей.

Игровые справочники и базы знаний позволяют систематизировать большой объём информации, делая её доступной в любое время. Пользователи могут быстро находить данные о персонажах, снаряжении, навыках, ресурсах и стратегиях, что значительно упрощает освоение игры и повышает эффективность игрового процесса.

Веб-приложение «Игровой справочник» призвано стать удобным инструментом для игроков, которые хотят оптимизировать свой прогресс в игре «Honkai: Star Rail». Оно будет предоставлять структурированную информацию, включающую описания игровых элементов, рекомендации по сборкам персонажей и советы по прохождению. Такой сервис позволит пользователям экономить время на поиске информации в разрозненных источниках и сосредоточиться на самом игровом процессе.

Особое внимание в разработке уделяется удобству интерфейса и доступности информации. Веб-приложение будет поддерживать поиск и фильтрацию по ключевым параметрам, что позволит игрокам быстро находить интересующие данные. Кроме того, использование современных веб-технологий обеспечит кроссплатформенность, благодаря чему справочник будет одинаково удобен как на компьютерах, так и на мобильных устройствах.

Таким образом, создаваемое веб-приложение «Игровой справочник» не только расширит возможности игроков Honkai: Star Rail, но и станет незаменимым помощником для тех, кто стремится к более глубокому пониманию механик игры и эффективному использованию её ресурсов.

1. Предпроектное исследование

1.1 Описание предметной области веб приложения

Современные видеоигры отличаются высокой сложностью, большим количеством внутриигровых механик и постоянно расширяющимся контентом. Игроки сталкиваются с необходимостью запоминать и анализировать значительные объёмы информации: характеристики персонажей, их навыки и уровни, виды экипировки, материалы для улучшений и ресурсы, необходимые для развития. В условиях регулярных обновлений и добавления нового контента становится всё труднее удерживать в памяти все эти данные, что приводит к ошибкам при планировании игрового процесса и замедляет прогресс.

Веб-приложение «Игровой справочник» направлено на решение этой проблемы, предлагая пользователям удобный инструмент для систематизации и быстрого поиска информации, связанной с Honkai: Star Rail. Оно создаётся как единый ресурс, который позволит игрокам получать доступ к актуальным данным по персонажам, артефактам, оружию и другим элементам игры в удобной и структурированной форме.

Проблема дезорганизации информации особенно актуальна для игроков, которые активно участвуют в событиях, развивают несколько персонажей одновременно или стремятся максимально эффективно использовать ограниченные игровые ресурсы. Поиск сведений в разрозненных источниках: форумах, видеороликах или социальных сетях требует значительных затрат времени и не всегда гарантирует достоверность. Веб-приложение «Игровой справочник» создаётся для того, чтобы решить эти задачи, предоставляя проверенную и систематизированную базу знаний.

Приложение ориентировано на разные категории пользователей:

* новичков, которым сложно разобраться в большом объёме информации и игровых механиках;
* опытных игроков, которые хотят быстро находить данные и строить оптимальные стратегии развития;
* любителей анализа и планирования, стремящихся сравнивать персонажей, подбирать наиболее эффективные сборки и маршруты прогресса;
* участников игровых сообществ, которым важно иметь под рукой удобный источник информации для обсуждений и совместного прохождения.
* Веб-приложение будет предоставлять пользователям ряд функций, направленных на упрощение взаимодействия с игрой:
* каталог персонажей с описанием характеристик, навыков и рекомендуемых сборок;
* база данных артефактов, оружия и ресурсов с возможностью поиска и фильтрации;
* советы и рекомендации по развитию персонажей и прохождению игровых событий;
* личный кабинет с возможностью составления собственных заметок и подборок информации;
* система обновлений, которая позволит всегда быть в курсе актуального игрового контента.

Потенциальные сценарии использования включают:

* Быстрый поиск характеристик персонажей и предметов во время игрового процесса.
* Подготовку к новым событиям и планирование развития аккаунта с учётом ограниченных ресурсов.
* Создание собственных подборок и стратегий для оптимизации игрового времени.
* Использование справочника в рамках игровых сообществ для обмена информацией и совместного планирования.

Таким образом, веб-приложение «Игровой справочник» становится незаменимым инструментом для игроков Honkai: Star Rail, позволяя им систематизировать информацию, повышать эффективность игрового процесса и получать больше удовольствия от игры.

1.2 Обзор инструментов, используемых в разработке программного продукта

Инструменты разработки программного продукта определяют его функциональность, удобство использования и возможности дальнейшего расширения. Для реализации веб-приложения «Игровой справочник» необходимо использовать инструменты, обеспечивающие простоту разработки, гибкость, кроссплатформенность и лёгкость поддержки.

Приложение будет состоять из двух частей – клиентской и серверной. Для проектирования архитектуры и интерфейса будут применяться современные средства визуализации и прототипирования, а для реализации логики – язык PHP и вспомогательные технологии.

Клиентская часть отвечает за отображение данных и взаимодействие с пользователем. Она будет разрабатываться с использованием HTML, CSS и JavaScript.

1. HTML – язык разметки, отвечающий за структуру страниц.
2. CSS – каскадные таблицы стилей, позволяющие задавать оформление и адаптивность интерфейса.
3. JavaScript – язык программирования, обеспечивающий динамику и интерактивность (например, фильтрацию, анимацию, валидацию форм).

Серверная часть отвечает за хранение, обработку и предоставление данных. Для её реализации используется:  
 4. PHP – язык сценариев, выполняющийся на сервере, подходящий для обработки запросов пользователей, взаимодействия с базой данных и генерации динамического содержимого.  
 5. MySQL – система управления базами данных, которая будет хранить информацию о персонажах, оружии, артефактах и других игровых элементах.

Инструменты проектирования и разработки:  
6. Draw.io – сервис для построения диаграмм, который будет использоваться для проектирования архитектуры приложения и базы данных.  
7. Figma – онлайн-платформа для создания прототипов интерфейсов. Используется для визуализации интерфейса и его последующей реализации.  
8. Visual Studio Code – интегрированная среда разработки (IDE), поддерживающая PHP, JavaScript, HTML и CSS. Обеспечивает удобство написания кода, отладки и интеграцию с системой контроля версий Git.

Таблица 1 – Сравнение языков программирования для реализации серверной части

|  |  |  |  |
| --- | --- | --- | --- |
| Язык программирования | PHP | Python | JavaScript (Node.js) |
| Простота освоения | + | + | + |
| Поддержка веб-фреймворков | + | + | + |
| Скорость разработки | + | - | + |
| Богатое сообщество и библиотеки | + | + | + |
| Интеграция с MySQL | + | + | + |

Таблица 2 – Сравнение вариантов СУБД для хранения данных

|  |  |  |  |
| --- | --- | --- | --- |
| СУБД | MySQL | PostgreSQL | SQLite |
| Поддержка больших проектов | + | + | - |
| Простота использования | + | - | + |
| Масштабируемость | + | + | - |
| Поддержка сложных запросов | + | + | - |
| Подходит для локальной разработки | + | + | + |

Таблица 3 – Сравнение IDE для разработки веб-приложения

|  |  |  |  |
| --- | --- | --- | --- |
| IDE | Visual Studio Code | PhpStorm | Sublime Text |
| Поддержка PHP, JS, HTML, CSS | + | + | + |
| Удобство работы с UI | + | + | - |
| Инструменты отладки | + | + | - |
| Интеграция с Git | + | + | - |
| Простота использования | + | - | + |
| Расширяемость плагинами | + | + | + |
| Стоимость | Бесплатно | Платно | Условно бесплатно |

Таким образом, для разработки веб-приложения «Игровой справочник» используются современные инструменты, обеспечивающие удобство проектирования, простоту разработки и поддержку дальнейшего расширения проекта.

## **2 Проектирование игрового справочника**

## **2.1 Структурная схема игрового справочника**

Рисунок 1 демонстрирует диаграмму вариантов использования Use Case View интернет-магазина. Диаграмма прецедентов описывает высокоуровневые функциональные возможности системы с точки зрения пользователей.

Изображение выглядит как текст, диаграмма, рисунок, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1 – Диаграмма прецедентов

На диаграмме представлены два основных прецедента:

* Пользователь: Может просматривать список персонажей, их конфигурации, статьи по сюжеты, комментировать их и оценивать.
* Администратор: Обладает полными правами на управление системой, включая назначение новых администраторов, добавление статей, персонажей и их конфигураций.

Диаграмма деятельности показывает поток выполнения операций в системе, в данном случае – процесс добавления персонажа.

Изображение выглядит как текст, снимок экрана, диаграмма, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2 – Диаграмма деятельности

Процесс начинается с авторизации для проверки прав, и при наличии прав администратора, осуществляется переадресация на личный кабинет администратора, где находится панель для администрирования сайтом. Затем, администратор, нажимая на кнопку добавления персонажа переходит на соответствующую страницу, заполняет форму отправляет данные в БД. Затем система после обработки, проверки и занесения данных в БД, переносит на страницу созданного персонажа для проверки отображения элементов и данных.

Диаграмма детализирует процесс игрового справочника – добавление персонажа и сборки. Она демонстрирует последовательность шагов и ключевые операции, такие как заполнение формы и отправки данных в БД, что соответствует реализованному функционалу

## **2.2 Функциональная схема интернет-магазина**

Контекстная диаграмма – это высокоуровневое визуальное представление, которое показывает взаимодействия между разрабатываемой системой и её внешними объектами.

Изображение выглядит как текст, диаграмма, линия, Параллельный

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 3 – Контекстная диаграмма

Диаграмма показывает, что система получает информацию о жалобах пользователей на комментарии, а администратор проверяет жалобу в соответствии с политикой сайта.

Диаграмма декомпозиций (А1) предназначена для детализации работы, то есть декомпозицией называется разделение бизнес-процессов на более мелкие составляющие.

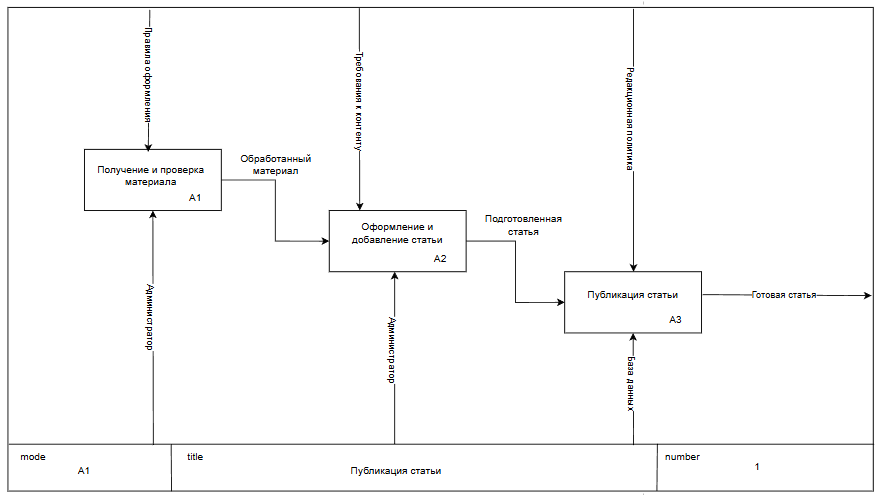


Рисунок 4 – Диаграмма декомпозиций А1

Процесс публикации лор-статьи использует информацию из источника "Мнимый уголок", обрабатывает входные данные на трех этапах и выдает "Готовую статью". На первом этапе (A1) администратор получает и проверяет материалы, создавая "Обработанный материал". На втором этапе (A2) происходит оформление и добавление статьи в систему с созданием "Подготовленной статьи". На третьем этапе (A3) осуществляется публикация статьи на сайте. Управление процессом осуществляется через "Правила оформления контента", "Требования к контенту" и "Редакционную политику", а механизмами выступают "Администратор", "База данных" и "Система уведомлений".

Диаграмма декомпозиции A2 (IDEF0) – это диаграмма, которая описывает детализацию функций системы на втором уровне в нотации IDEF0

Изображение выглядит как диаграмма, текст, План, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 5 – Диаграмма декомпозиции A3

Диаграмма A2 демонстрирует детальную последовательность операций при оформлении и добавлении лор-статьи. Четко прослеживается разделение на логические этапы: форматирование текста, добавление медиа-контента, заполнение метаданных, сохранение в базу данных. Такая декомпозиция позволяет эффективно распределить ответственность между модулями системы и обеспечивает прозрачность бизнес-процесса публикации контента от источника.

Изображение выглядит как текст, диаграмма, чек, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 6 – Диаграмма классов

Диаграмма классов представляет статическую структуру данных системы:

* Users – модель пользователя с распределением ролей через атрибут role (user, admin). Администраторы имеют эксклюзивные права на управление контентом (добавление персонажей и публикация статей), обычные пользователи взаимодействуют с контентом через комментарии, лайки и избранное
* Characters – сущность персонажа игры Honkai Star Rail с полным набором характеристик: путь (path), тип урона, навыки, артефакты, световые конусы, рекомендуемые статы и материалы для прокачки. Персонажи добавляются исключительно администраторами
* Articles – лор-статья, содержащая контент от лороведов. Атрибут author\_id хранит идентификатор администратора, который добавил статью в систему. Публикация статей доступна только администраторам
* Comments – комментарий пользователя, связанный с персонажем или статьей, с поддержкой статусов модерации. Любой зарегистрированный пользователь может оставлять комментарии
* Likes– лайк, фиксирующий оценку пользователем статьи или комментария через универсальную схему (target\_type, target\_id). Доступен всем пользователям
* Favorites– избранное пользователя, позволяющее сохранять персонажей или статьи для быстрого доступа. Каждый пользователь формирует личную коллекцию избранного

Связи отображают бизнес-логику с четким разграничением прав доступа администраторы (определяются через role в Users) создают и публикуют статьи, добавляют персонажей с их сборками. Обычные пользователи взаимодействуют с опубликованным контентом: просматривают персонажей и статьи, оставляют комментарии, ставят лайки и формируют список избранного. Атрибут author\_id в таблице Articles связывает публикацию с конкретным администратором, что в будущем позволит реализовать журнал действий администраторов для отслеживания, кто именно добавил или модерировал контент.

Диаграмма классов демонстрирует хорошо продуманную объектную модель с разделением ролей и полномочий. Четкое разграничение между администраторами (создание контента) и пользователями (потребление и реакции на контент), наличие необходимых атрибутов и логичные связи между классами обеспечивают основу для устойчивой и расширяемой архитектуры справочной системы. Модель полностью поддерживает ключевые сценарии использования: административное управление персонажами и статьями, пользовательский просмотр контента, комментирование, систему лайков и избранного, а также модерацию комментариев администраторами.

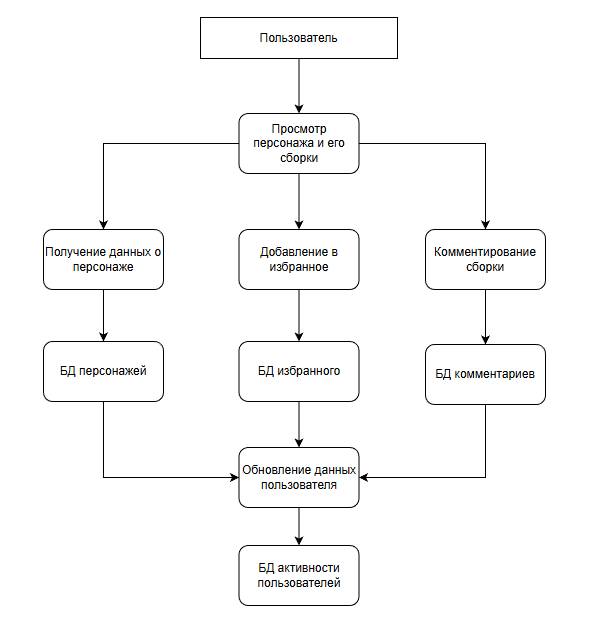
****

Рисунок 7 – Диаграмма потоков данных DFD

Диаграмма потоков данных (DFD) показывает движение информации в системе:

* Внешние сущности: Пользователь, Администратор
* Процессы: Справочная система hsrlib (основной обработчик)
* Потоки данных: запросы на просмотр персонажей и их сборок, запросы на чтение лор-статей, операции комментирования и лайков, добавление в избранное, создание и публикация нового контента администраторами, аутентификация и регистрация пользователей
* Хранилища данных: БД пользователей, БД персонажей, БД статей, БД комментариев, БД лайков, БД избранного

Диаграмма иллюстрирует, как данные поступают от пользователей и администраторов, обрабатываются системой, сохраняются в базе данных и возвращаются в виде отображения персонажей со сборками, лор-статей, комментариев и пользовательских коллекций избранного.

Разработанный комплект диаграмм полностью охватывает функциональную схему справочной информационной системы hsrlib:

* IDEF0 диаграммы (A0, A1, A2) эффективно отображают иерархию бизнес-процессов, начиная от контекстного представления всей системы и заканчивая детализированными операциями публикации лор-статей с форматированием текста, добавлением медиа-контента, заполнением метаданных и сохранением в базу данных.
* Диаграмма классов четко определяет структуру данных системы, показывая сущности (Users, Characters, Articles, Comments, Likes, Favorites), их атрибуты и взаимосвязи с четким разграничением ролей администраторов и пользователей, что соответствует реализованной в проекте объектной модели.
* DFD диаграмма наглядно демонстрирует потоки информации между внешними сущностями (пользователи, администраторы) и системой, отражая ключевые процессы справочной системы: регистрацию и авторизацию, просмотр игровых персонажей и сборок, чтение лор-статей, управление контентом, взаимодействие через комментарии и лайки, работу с избранным.

Все диаграммы взаимосвязаны и непротиворечивы: сущности из диаграммы классов соответствуют хранилищам данных в DFD, а бизнес-процессы IDEF0 отражают функциональность, реализованную через процессы обработки данных и методы взаимодействия с базами данных. Такой комплексный подход к проектированию обеспечивает полное понимание архитектуры справочной системы hsrlib для игры Honkai Star Rail и облегчает дальнейшую разработку и сопровождение проекта.

## **2.3 Функциональная схема интернет-магазина**

Перед началом разработки программного обеспечения критически важно провести тщательное проектирование базы данных. Этот этап включает в себя определение типов данных, с которыми будут взаимодействовать пользователи системы, а также выявление взаимосвязей между этими данными. Именно в этом и заключается суть процесса проектирования.

Инфологическое моделирование играет ключевую роль в этом процессе. Его главная задача, как показано на рисунке 8, заключается в разработке наиболее интуитивных и удобных для человека методов сбора и представления информации, которая будет храниться в создаваемой базе данных.

Изображение выглядит как рисунок, шаблон, диаграмма, зарисовка

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 8 – Инфологическая модель базы данных

Инфологическая модель представляет собой семантическое описание предметной области в терминах сущностей и отношений:

* Пользователи – сущность «Пользователь» с персональными данными (имя пользователя, пароль, почта, яндекс id), ролью (user, admin) и датой регистрации. Является центральной сущностью для разграничения прав доступа: администраторы управляют контентом, пользователи взаимодействуют с ним.
* Персонажи – сущность «Персонаж игры Honkai Star Rail», включающая полный набор игровых характеристик: название, уникальный идентификатор (slug), редкость (stars), путь (path), тип урона (damage\_type), основные и дополнительные характеристики (main\_stats, sub\_stats), навыки (skills), рекомендуемые световые конусы (cones), артефакты и реликвии (artifacts, relics), материалы для прокачки (materials), описание, лор-информацию и изображение персонажа (artwork).
* Статьи – сущность «Лор-статья», содержащая контент от лороведов: заголовок (title), уникальный slug, текстовый контент (content), связь с администратором-публикатором (id пользователя), даты создания и обновления. Представляет образовательный и развлекательный контент о сюжете игры.
* Комментарии – сущность «Комментарий пользователя», связывающая пользователя с конкретной статьей или персонажем, содержащая текст комментария (content), статус модерации (status) и дату публикации. Обеспечивает интерактивность и обратную связь от сообщества.
* Лайки – сущность «Оценка контента», фиксирующая положительную реакцию пользователя на статью или комментарий через универсальную структуру (id пользователя, тип цели, id цели). Позволяет измерять популярность контента.
* Избранное – сущность «Коллекция избранного», представляющая персональную подборку персонажей или статей пользователя для быстрого доступа. Содержит связь пользователя с сохраненным объектом (id пользователя, id персонажа/статьи) и дату добавления.

Связи между сущностями:

* Пользователь может создавать множество статей (администраторы) и комментариев (1:M)
* Пользователь может иметь множество лайков и элементов избранного (1:M)
* Статья принадлежит одному пользователю-администратору, но может иметь множество комментариев и лайков (M:1 и 1:M)
* Персонаж может иметь множество комментариев и быть добавлен в избранное многими пользователями (1:M)
* Комментарий принадлежит одному пользователю и относится к одной статье или персонажу (M:1)
* Лайк связывает пользователя с конкретным контентом через универсальную схему (M:1)
* Избранное реализует связь многие-ко-многим между пользователями и персонажами/статьями через промежуточную таблицу

Инфологическая модель демонстрирует логичную структуру предметной области справочной системы. Четкое разделение между контентом (персонажи, статьи) и пользовательским взаимодействием (комментарии, лайки, избранное), а также разграничение ролей администраторов и пользователей обеспечивают гибкость и масштабируемость системы hsrlib для игрового сообщества Honkai Star Rail.

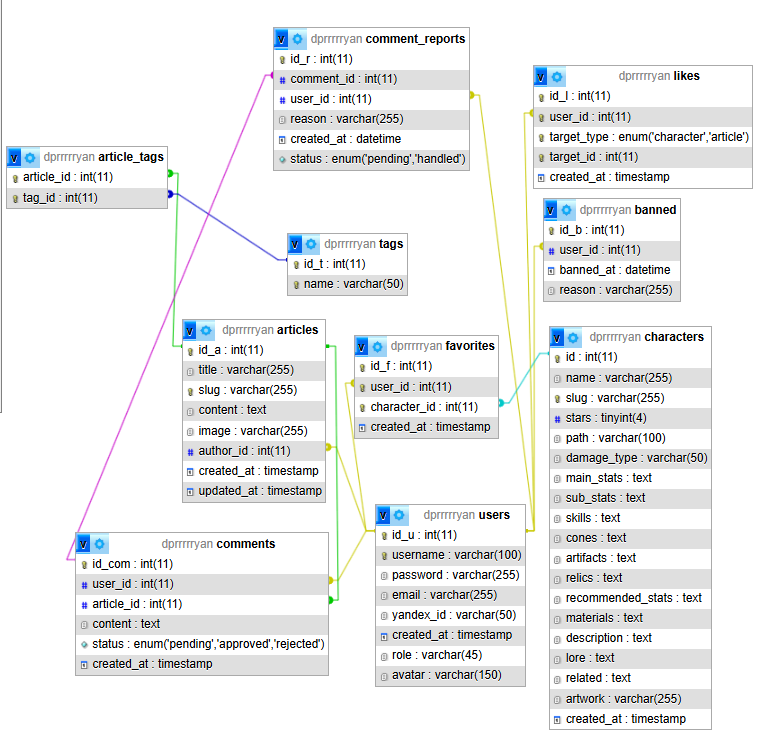


Рисунок 9 – ER диаграмма

Представленная ER‑модель соответствует третьей нормальной форме (3NF):

* Первая нормальная форма (1NF): Все атрибуты атомарны, нет повторяющихся групп. Каждое поле содержит единичное значение (username, email, title, content и т. д.).
* Вторая нормальная форма (2NF): Нет частичных зависимостей от составного ключа — все сущности имеют простой первичный ключ (id\_u, id\_c, id\_a, id\_com, id\_l, id\_f).
* Третья нормальная форма (3NF): Нет транзитивных зависимостей — все неключевые атрибуты зависят только от первичного ключа своей таблицы. Например, в таблице Articles атрибуты title, slug, content зависят только от id\_a, а информация об авторе хранится через внешний ключ author\_id, указывающий на таблицу Users.

Обоснование выбора 3NF:

* Устраняет избыточность данных — информация о пользователях хранится только в таблице Users, а не дублируется в статьях и комментариях.
* Обеспечивает целостность данных — связи через внешние ключи гарантируют ссылочную целостность.
* Упрощает поддержку и модификацию БД — изменение данных пользователя требует обновления только одной записи.
* Оптимальна для OLTP‑систем — справочная система hsrlib работает с онлайн‑транзакциями (создание комментариев, лайков, добавление в избранное).
* Гибкая система лайков через универсальную схему (target\_type, target\_id) позволяет оценивать разные типы контента без создания отдельных таблиц.

База данных справочной системы hsrlib создана и реализована в СУБД MySQL и состоит из 10 таблиц: Users (пользователи с ролями и аватарами), Characters (персонажи игры с полными характеристиками, навыками, артефактами и историей), Articles (лор статьи с авторами и изображениями), Comments (комментарии пользователей), Comment\_Reports (жалобы на комментарии), Likes (система оценок для статей и персонажей), Favorites (избранное пользователей), Banned (блокировки пользователей), Tags (теги для статей), Article\_Tags (связь статей с тегами). Все таблицы связаны через внешние ключи, обеспечивая целостность и согласованность данных предметной области.

Таблица 4 – Таблица Users

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| user\_id | INT | Уникальный идентификатор пользователя |
| username | VARCHAR(100) | Имя пользователя |
| email | VARCHAR(255) | Электронная почта |
| password | VARCHAR(255) | Хешированный пароль |
| yandex\_id | VARCHAR(150) | Идентификатор Yandex для авторизации |
| role | VARCHAR(45) | Роль пользователя в системе |
| avatar | VARCHAR(150) | Ссылка на аватар пользователя |
| created\_at | TIMESTAMP | Дата регистрации |

Таблица 5 – Таблица Articles

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id\_a | INT | Уникальный идентификатор статьи |
| title | VARCHAR(255) | Заголовок статьи |
| slug | VARCHAR(255) | Уникальный URL-идентификатор |
| content | LONGTEXT | Полный текст статьи |
| excerpt | TEXT | Краткое описание статьи |
| image | VARCHAR(255) | Ссылка на изображение статьи |
| author\_id | INT | Идентификатор автора статьи |
| created\_at | TIMESTAMP | Дата создания статьи |
| updated\_at | TIMESTAMP | Дата последнего обновления статьи |

Таблица 6 – Таблица Comments

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id\_com | INT | Уникальный идентификатор комментария |
| user\_id | INT | Идентификатор пользователя-автора |
| article\_id | INT | Идентификатор статьи |
| content | TEXT | Текст комментария |
| created\_at | TIMESTAMP | Дата создания комментария |

Таблица 7 – Таблица Banned

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id\_b | INT | Уникальный идентификатор записи о блокировке |
| user\_id | INT | Идентификатор заблокированного пользователя |
| banned\_until | DATETIME | Дата окончания блокировки |
| reason | VARCHAR(255) | Причина блокировки |

Таблица 8 – Таблица Tags

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id\_t | INT | Уникальный идентификатор тега |
| name | VARCHAR(50) | Название тега |

Таблица 9 – Таблица Article\_Tags

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| article\_id | INT | Идентификатор статьи |
| tag\_id | INT | Идентификатор тега |

Таблица 10 – Таблица Likes

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id\_l | INT | Уникальный идентификатор лайка |
| user\_id | INT | Идентификатор пользователя |
| target\_type | VARCHAR(50) | Тип объекта (character/article) |
| target\_id | INT | Идентификатор объекта |
| created\_at | TIMESTAMP | Дата создания лайка |

Таблица 11 – Таблица Favorites

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id\_f | INT | Уникальный идентификатор записи избранного |
| user\_id | INT | Идентификатор пользователя |
| character\_id | INT | Идентификатор персонажа |
| created\_at | TIMESTAMP | Дата добавления в избранное |

Таблица 12 – Таблица Comment\_Reports

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id\_r | INT | Уникальный идентификатор жалобы |
| comment\_id | INT | Идентификатор комментария |
| user\_id | INT | Идентификатор пользователя, подавшего жалобу |
| reason | VARCHAR(255) | Причина жалобы |
| created\_at | DATETIME | Дата создания жалобы |
| status | ENUM('pending','handled') | Статус обработки жалобы |

Таблица 13 – Таблица Characters

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | INT | Уникальный идентификатор персонажа |
| name | VARCHAR(255) | Имя персонажа |
| slug | VARCHAR(255) | Уникальный URL-идентификатор |
| class | TINYINT(4) | Класс персонажа |
| path | VARCHAR(100) | Путь персонажа |
| damage\_type | VARCHAR(50) | Тип урона |
| rarity | INT | Редкость персонажа |
| sub\_stats | TEXT | Дополнительные характеристики |
| skills | TEXT | Навыки персонажа |
| cones | TEXT | Световые конусы |
| artifacts | TEXT | Артефакты |
| relics | TEXT | Реликвии |
| recommended\_stats | TEXT | Рекомендуемые характеристики |
| materials | TEXT | Материалы для прокачки |
| description | TEXT | Описание персонажа |
| lore | TEXT | История персонажа |
| related | TEXT | Связанные персонажи |
| artwork | VARCHAR(255) | Ссылка на изображение |
| created\_at | TIMESTAMP | Дата создания записи |

Спроектированная база данных полностью соответствует требованиям предметной области справочной системы hsrlib для игры Honkai Star Rail. ER-модель, приведённая к третьей нормальной форме, обеспечивает оптимальную структуру для хранения информации о пользователях, персонажах игры с их сборками, лор-статьях от контент-создателей, комментариях пользователей, системе лайков и избранного, минимизируя избыточность и обеспечивая логическую целостность данных.

Чётко определённые связи между сущностями позволяют эффективно реализовать все бизнес-процессы системы: от просмотра каталога персонажей и чтения лор-статей до добавления комментариев, проставления лайков и формирования персональных коллекций избранного. Разграничение ролей через атрибут role в таблице Users обеспечивает безопасное управление контентом: администраторы добавляют и публикуют персонажей со сборками и лор-статьи, а обычные пользователи взаимодействуют с контентом через комментарии, лайки и избранное.

Универсальная схема хранения лайков (target\_type, target\_id) и избранного (entity\_type, entity\_id) обеспечивает гибкость системы оценок без создания избыточных таблиц. Выбранные типы данных, ограничения целостности и внешние ключи гарантируют корректное хранение сведений и предотвращают возникновение противоречивых состояний в базе данных, обеспечивая надёжность и расширяемость разработанной справочной информационной системы для игрового сообщества Honkai Star Rail.

## **2.4 Проектирование интерфейса**

Для проектирования пользовательского интерфейса игрового справочника «hsrlib» использовался профессиональный онлайн-инструмент [Draw.io](https://Draw.io). Данный инструмент был выбран за его кроссплатформенность, простоту использования, широкий набор элементов для построения схем и удобство совместной работы, что позволило быстро и наглядно создать прототипы всех основных экранов системы.

Изображение выглядит как текст, снимок экрана, диаграмма, Параллельный

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 10 – Страница главная

Прототип главной страницы справочной системы hsrlib. Страница разделена на три основные секции. В левой части расположен блок "Персонаж дня" с крупным изображением рекомендуемого персонажа. Центральная секция содержит сетку популярных персонажей (6 карточек) с изображениями и названиями, позволяющую пользователю быстро ознакомиться с наиболее востребованными героями игры. В правой части представлен блок "Последние статьи" с тремя превью лор-статей, включающими заголовок, краткое описание и дату публикации. Верхняя навигационная панель обеспечивает доступ к основным разделам: Главная, Персонажи и сборки, Лор, Форум, а также содержит аватар пользователя для быстрого доступа к профилю.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 11 – Страница входа

Прототип страницы авторизации справочной системы hsrlib. Страница имеет центрированную форму входа с минималистичным дизайном. Форма содержит два основных поля ввода: "Имя пользователя" и "Пароль", обеспечивающие базовую аутентификацию пользователя. Для защиты от автоматизированных атак интегрирована CAPTCHA с чекбоксом "Я не робот". Основная кнопка "Войти" позволяет выполнить аутентификацию с использованием учетных данных системы. Дополнительно реализована возможность быстрой авторизации через внешний сервис "Войти через Яндекс ID", что упрощает процесс входа для пользователей, уже имеющих учетную запись Яндекса. В нижней части формы размещена ссылка "Нет аккаунта? Зарегистрируйтесь", направляющая новых пользователей на страницу регистрации.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 12 – Страница регистрации

Прототип страницы регистрации новых пользователей справочной системы hsrlib. Страница имеет центрированную форму регистрации с минималистичным дизайном. Форма содержит четыре основных поля ввода: "Имя пользователя", "Email", "Пароль" и "Подтвердить пароль", обеспечивающие полный набор данных для создания новой учетной записи. Для защиты от автоматизированных регистраций интегрирована CAPTCHA с чекбоксом "Я не робот" от Yandex Cloud. Основная кнопка "Зарегистрироваться" позволяет завершить процесс создания аккаунта и автоматически авторизоваться в системе. В нижней части формы размещена ссылка "Уже есть аккаунт? Войти!", направляющая существующих пользователей на страницу авторизации.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 13 – Страница личного кабинета администратора

Прототип панели управления контентом справочной системы hsrlib для пользователей с ролью "Администратор". Страница имеет шапку с идентификацией пользователя (аватар и название панели) и навигационными элементами "Выход" и "Главная" для быстрого доступа к основным функциям. Центральная часть содержит персонализированное приветствие с указанием роли пользователя. Основной функционал представлен пятью кнопками административных действий: "Добавить персонажа", "Добавить сборку" и "Добавить статью" для управления контентом системы, "Права пользователей" для управления доступами пользователей, и "Создать бэкап БД" для резервного копирования базы данных. Кнопки сгруппированы логически: управление контентом в верхнем ряду, служебные функции в нижнем ряду.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 14 – Страница личного кабинета пользователя

Прототип страницы личного кабинета обычного пользователя справочной системы hsrlib. Страница содержит шапку с логотипом системы и навигационными ссылками "Главная" и "Выход". Верхний блок профиля включает круглый аватар пользователя размером 120x120 пикселей с кнопкой "Изменить аватар", а также базовую информацию: имя пользователя, роль (Пользователь), дату регистрации и email. Основной контент организован в четыре карточки: "Настройки аккаунта" с отображением имени пользователя, email и пароля (скрыт точками) и кнопкой "Изменить данные"; "Статистика активности" с двумя счетчиками — количество комментариев и элементов в избранном; "Мои коллекции" с двумя крупными кнопками быстрого доступа к разделам "Избранное" и "Мои комментарии"; "Уведомления" с тремя чекбоксами для настройки email-рассылки новых статей, уведомлений об ответах на комментарии и уведомлений о новых персонажах.

Изображение выглядит как текст, снимок экрана, диаграмма, Параллельный

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 15 – Страница персонажи

Прототип страницы просмотра всех персонажей игры Honkai Star Rail в справочной системе hsrlib. Страница имеет стандартную шапку с логотипом, навигационным меню и аватаром пользователя. Основной контент разделен на две колонки: левая боковая панель содержит блок фильтров с тремя группами — фильтрация по редкости (4 или 5 звёзд), типу урона (Физический, Огонь, Лёд, Молния, Ветер, Квант, Мнимое) и пути персонажа (Разрушение, Охота, Эрудиция, Гармония, Небытие, Сохранение); правая основная область включает заголовок страницы "Персонажи", поле поиска и адаптивную сетку карточек персонажей. Каждая карточка содержит вертикальное изображение персонажа с иконкой избранного в правом верхнем углу, имя персонажа, количество звёзд (редкость) и тип урона в виде бейджа. В нижней части реализована навигация по страницам с кнопками "Назад", номерами страниц и "Вперёд", позволяющая просматривать полный каталог персонажей.

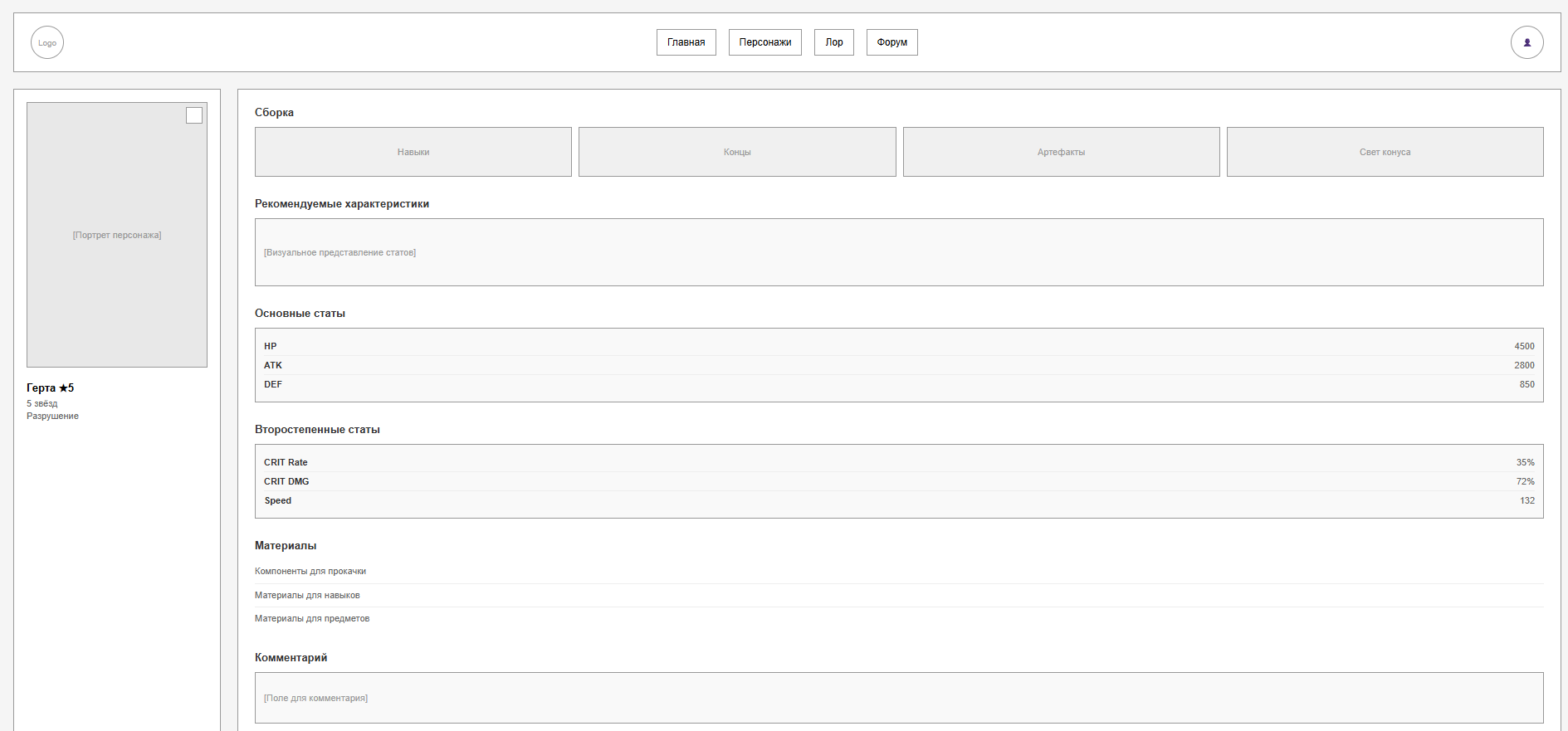


Рисунок 16 – Страница сборки

Прототип страницы детализированного просмотра и управления сборкой персонажа в справочной системе HSRLib. Страница имеет стандартную шапку с логотипом, навигационным меню (Главная, Персонажи, Лор, Форум) и аватаром пользователя. Основной контент разделен на две колонки: левая боковая панель содержит полноразмерное портретное изображение персонажа с кнопкой добавления в избранное в правом верхнем углу, имя персонажа с его рейтингом редкости (количество звёзд) и информацией о пути персонажа (одно из восьми путей: Разрушение, Охота, Эрудиция, Гармония, Небытие, Сохранение и др.); правая основная область включает четыре основных раздела управления сборкой: блок Сборка с четырьмя компонентами (Навыки, Консты, Артефакты, Свет конуса), раздел Рекомендуемые характеристики с визуализацией приоритетных статов через прогресс-индикаторы, раздел Основные статы с текущими значениями базовых характеристик персонажа (Атака, Защита, HP, Скорость) и раздел Второстепенные статы с дополнительными параметрами. В нижней части располагаются информация о необходимых материалах для развития персонажа, кнопки действий ("Сохранить сборку", "Обсудить") и ссылка на полное описание персонажа, позволяющие пользователю полностью управлять и оптимизировать развитие персонажа.

Изображение выглядит как снимок экрана, текст, диаграмма, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 17 – Страница статей

Прототип страницы каталога аналитических статей и исследований по сюжету игры Honkai Star Rail в справочной системе HSRLib. Страница имеет стандартную шапку с логотипом, навигационным меню (Главная, Персонажи, Лор, Форум) и аватаром пользователя. Основной контент разделен на две колонки: левая боковая панель содержит три группы фильтров — фильтрация по категориям статей (Персонажи, События, Миры, Теория, История), типам контента (Разбор, Анализ, Инсайты, Гайд) и актуальным патчам (3.0, 2.7, 2.6, Архив); правая основная область включает заголовок страницы "Лор", поле поиска по названию статей и адаптивную сетку карточек аналитических материалов.

Каждая карточка статьи содержит горизонтальное изображение-превью статьи (иллюстрация, скриншот из игры или обложка) с иконкой избранного в правом верхнем углу, заголовок статьи (с переносом текста для длинных названий) и категорию-бейдж, указывающую на тип контента. Статьи включают разборы скрытых деталей в диалогах и книгах (например, "Сирены Амфореуса, Гисиленса и Фагуса", "Владычица Безмолвия Полька Какамонд"), аналитику текущих событий патча ("Разбор событий патча 3.0"), теоретические исследования ("Титаны и их роль в судьбе миров", "Эффект бабочки в сюжете") и хронологические справки. В нижней части реализована пагинация с кнопками "Назад", номерами страниц и "Вперёд", позволяющая просматривать полный архив статей по лору, и система поиска для быстрого доступа к интересующему материалу.



Рисунок 18 – Страница просмотра статьи

Прототип страницы детального чтения и изучения аналитической статьи по лору игры Honkai Star Rail в справочной системе HSRLib. Страница имеет стандартную шапку с логотипом, навигационным меню (Главная, Персонажи, Лор, Форум) и аватаром пользователя. Под шапкой располагается хлебная крошка навигации для быстрого возврата к каталогу лора и категориям.

Основной контент разделен на две колонки: левая основная область содержит полный текст статьи с заголовком, метаинформацией (автор, дата публикации и обновления, время чтения), полноразмерным изображением-иллюстрацией, основным телом текста со структурированными разделами (подзаголовки, абзацы, цитаты), и подвалом с кнопками действий ("В избранное", "Поделиться", "Обсудить") и рейтингом статьи. Правая боковая панель включает блок информации о статье (категория, тип, сложность, наличие спойлеров), систему тегов для навигации по связанным темам, раздел "Связанные персонажи" с ссылками на профили упомянутых в статье героев, и блок "Рекомендуемые статьи" с предложениями похожих материалов. Такая структура обеспечивает глубокое погружение в контент и навигацию по связанной информации без необходимости возврата к главной странице лора.

5 Разработка веб-приложение

# 5.1 Разработка базы данных веб-приложение

В проекте для создания и управления базой данных hsrlib использовался язык структурированных запросов SQL. С его помощью была определена структура всех таблиц — Users, Characters, Articles, Comments, Likes, Favorites, Banned, Tags, Article\_Tags, Comment\_Reports — указаны типы данных полей, первичные и внешние ключи, а также ограничения целостности данных. Проектирование базы начиналось с составления схемы таблиц и их взаимосвязей, после чего с помощью SQL-команд создавались таблицы, устанавливались связи между ними и задавались индексы для обеспечения корректной работы системы и повышения производительности запросов.

Ниже представлен SQL код для создания таблиц:

CREATE TABLE users (

id\_u INT(11) PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(100) NOT NULL,

password VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL,

yandex\_id VARCHAR(150),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

role VARCHAR(45) DEFAULT 'user',

avatar VARCHAR(150),

UNIQUE KEY unique\_username (username),

UNIQUE KEY unique\_email (email)

);

CREATE TABLE banned (

id\_b INT(11) PRIMARY KEY AUTO\_INCREMENT,

user\_id INT(11) NOT NULL,

reason VARCHAR(255),

banned\_until DATETIME,

FOREIGN KEY (user\_id) REFERENCES users(id\_u) ON DELETE CASCADE

);

CREATE TABLE articles (

id\_s INT(11) PRIMARY KEY AUTO\_INCREMENT,

title VARCHAR(255) NOT NULL,

slug VARCHAR(255) NOT NULL,

meta\_text TEXT,

image VARCHAR(255),

author\_id INT(11),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

UNIQUE KEY unique\_slug (slug),

FOREIGN KEY (author\_id) REFERENCES users(id\_u) ON DELETE SET NULL

);

CREATE TABLE tags (

id\_t INT(11) PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50) NOT NULL,

UNIQUE KEY unique\_name (name)

);

CREATE TABLE article\_tags (

article\_id INT(11) NOT NULL,

tag\_id INT(11) NOT NULL,

PRIMARY KEY (article\_id, tag\_id),

FOREIGN KEY (article\_id) REFERENCES articles(id\_s) ON DELETE CASCADE,

FOREIGN KEY (tag\_id) REFERENCES tags(id\_t) ON DELETE CASCADE

);

CREATE TABLE comments (

id\_com INT(11) PRIMARY KEY AUTO\_INCREMENT,

user\_id INT(11) NOT NULL,

article\_id INT(11) NOT NULL,

content TEXT NOT NULL,

status ENUM('pending', 'approved', 'rejected') DEFAULT 'pending',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(id\_u) ON DELETE CASCADE,

FOREIGN KEY (article\_id) REFERENCES articles(id\_s) ON DELETE CASCADE

);

CREATE TABLE comment\_reports (

id\_r INT(11) PRIMARY KEY AUTO\_INCREMENT,

comment\_id INT(11) NOT NULL,

user\_id INT(11) NOT NULL,

reason VARCHAR(255),

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

status ENUM('pending', 'handled') DEFAULT 'pending',

FOREIGN KEY (comment\_id) REFERENCES comments(id\_com) ON DELETE CASCADE,

FOREIGN KEY (user\_id) REFERENCES users(id\_u) ON DELETE CASCADE

);

CREATE TABLE characters (

id INT(11) PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255) NOT NULL,

slug VARCHAR(255) NOT NULL,

stars TINYINT(4),

path VARCHAR(100),

damage\_type VARCHAR(50),

element VARCHAR(50),

sub\_stats TEXT,

skills TEXT,

cones TEXT,

artifacts TEXT,

relics TEXT,

recommended\_stats TEXT,

materials TEXT,

description TEXT,

lore TEXT,

related TEXT,

artwork VARCHAR(255),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE KEY unique\_slug (slug)

);

CREATE TABLE favorites (

id\_f INT(11) PRIMARY KEY AUTO\_INCREMENT,

user\_id INT(11) NOT NULL,

character\_id INT(11) NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE KEY unique\_favorite (user\_id, character\_id),

FOREIGN KEY (user\_id) REFERENCES users(id\_u) ON DELETE CASCADE,

FOREIGN KEY (character\_id) REFERENCES characters(id) ON DELETE CASCADE

);

CREATE TABLE likes (

id INT(11) PRIMARY KEY AUTO\_INCREMENT,

user\_id INT(11) NOT NULL,

target\_type ENUM('character', 'article') NOT NULL,

target\_id INT(11) NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE KEY unique\_like (user\_id, target\_type, target\_id),

FOREIGN KEY (user\_id) REFERENCES users(id\_u) ON DELETE CASCADE

);

В ходе работы была разработана структура базы данных, обеспечивающая хранение необходимых данных и поддерживающая целостность информации. Схема отражает основные объекты и их связи, что позволяет эффективно управлять данными и обеспечивать быстрый доступ для дальнейшей обработки и использования в веб-приложении.

# 5.2 Разработка кода клиентской части веб-приложения

Для разработки интерфейса веб-приложения использовались HTML, CSS и PHP. В клиентской части определялась структура данных, с которыми работает интерфейс: набор необходимых полей, их типы и методы обработки. Такая организация обеспечивает корректный обмен информацией между клиентом и сервером, позволяет обрабатывать ввод пользователя, выполнять валидацию данных и подготовку их к хранению в базе данных. На рисунке 19 представлен пример кода регистрации пользователя в веб-приложении.

Изображение выглядит как текст, документ, Шрифт, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 19 – Код регистрации в веб-приложении

Для взаимодействия клиентской части с серверной логикой, реализованной на PHP, используется отдельный слой обработки данных. В нём определяются методы отправки данных через формы, получение POST-запросов, обработка введённой информации и её преобразование в формат, удобный для работы интерфейса. Такой подход позволяет изолировать серверную логику от клиентской части и обеспечивает безопасное и корректное взаимодействие между пользователем и системой. На рисунке 20 представлен пример кода регистрации пользователя в веб-приложении.

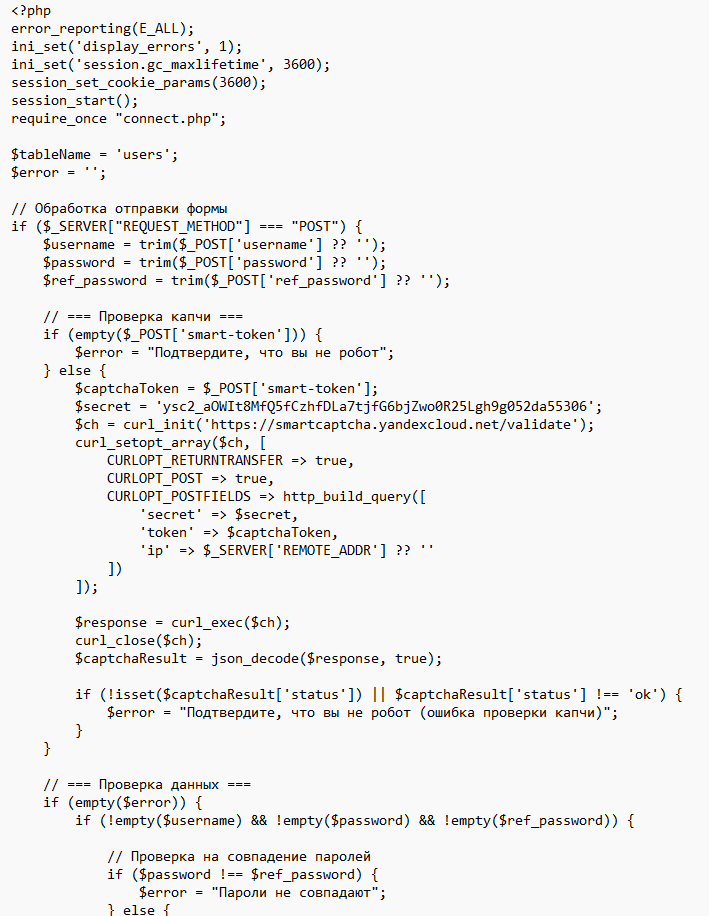


Рисунок 20 – Код регистрации в веб приложение

На рисунке 21 представлена итоговая страница регистрации в веб-приложение.

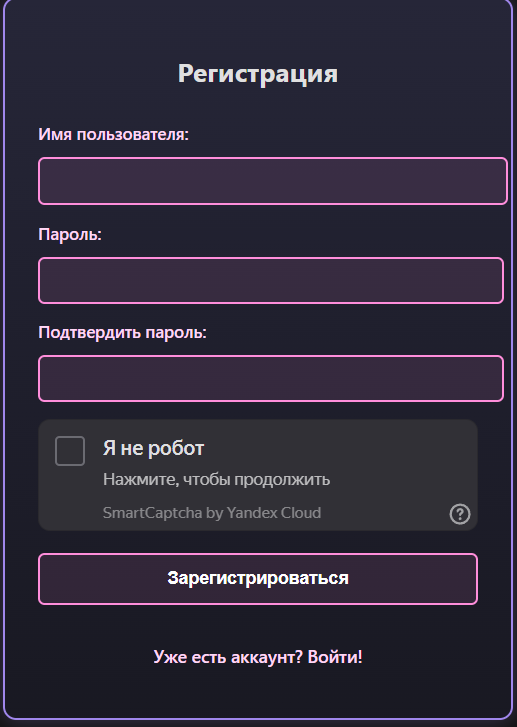


Рисунок 21 – Страница регистрации в веб приложение

В результате работы реализован интерфейс, где каждая часть веб-приложения выполняет строго определённую функцию. Это позволяет создавать расширяемую и легко поддерживаемое веб-приложение.

# 5.3 Разработка кода серверной части веб-приложение

В веб-приложении для реализации функционала входа пользователя использовался PHP совместно с HTML и CSS. Серверная часть обрабатывает POST-запросы от формы входа, проверяет наличие всех обязательных полей и выполняет валидацию данных. Дополнительно реализована проверка через SmartCaptcha Яндекс для защиты от ботов. После успешной проверки данных выполняется поиск пользователя в базе данных MySQL, проверка хеша пароля и блокировок в таблице banned. При необходимости хеш пароля обновляется для поддержания актуального уровня безопасности. При успешной аутентификации создаются сессии для хранения имени пользователя и роли, после чего выполняется перенаправление в личный кабинет в зависимости от роли (Admin или User). В клиентской части предусмотрено отображение ошибок и динамическое снятие подсветки полей при корректном вводе данных, обеспечивая удобство и безопасность взаимодействия с системой.

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 22 – Код авторизации

В процессе разработки серверной части была создана архитектура, которая надёжно обрабатывает запросы и обеспечивает безопасность работы с данными. Такая структура облегчает дальнейшее развитие проекта и гарантирует стабильную работу сервера.

6 Тестирование веб-приложения

6.1 Тестовые сценарии

Первым этапом функционального тестирования веб-приложения проводится разработка тестовых сценариев, направленных на проверку реализации требований и корректности работы ключевых функций системы. Функциональное тестирование позволяет убедиться, что каждая функция выполняется в соответствии с заданной логикой и обеспечивает ожидаемый результат при различных вариантах поведения пользователей. В данном разделе представлены позитивные и негативные сценарии, охватывающие основную функциональность системы. Позитивные сценарии демонстрируют корректную работу функций при использовании системы по назначению, тогда как негативные позволяют проверить устойчивость системы к ошибочным действиям и некорректным данным. Первый функциональный сценарий (таблица 14) направлен на поиск персонажа через строку поиска.

Таблица 14 – Сценарий тестирования успешного поиска персонажа

|  |  |
| --- | --- |
| Параметр | Описание |
| ID теста | POS–01 |
| Название теста | Поиск персонажа по имени |
| Предусловия | Пользователь находится на странице персонажей веб-приложения |
| Шаги | 1. Ввести в строку поиска имя персонажа, например «Герта»  2. Нажать кнопку «Поиск»  3. Дождаться загрузки результатов |
| Тестовые данные | «Герта» |
| Ожидаемый результат | Веб-приложение отображает персонажа с указанием класса, редкости и кнопкой для перехода на страницу персонажа |

Следующий позитивный сценарий (таблица 15) направлен на добавление персонажа в избранное

Таблица 15 – Сценарий тестирования успешного поиска персонажа

|  |  |
| --- | --- |
| Параметр | Описание |
| ID теста | POS–**02** |
| Название теста | Добавление персонажа в избранное |
| Предусловия | Пользователь нашел персонажа на странице |
| Шаги | 1. В карточке персонажа нажать на «сердечко»  2. Перейти в «Избранное»  3. Дождаться загрузки результатов |
| Тестовые данные | «Герта» |
| Ожидаемый результат | Страница отображает персонажа |

Следующий позитивный сценарий (таблица 16) направлена на создание комментария.

Таблица 16 – Сценарий тестирования успешного создания комментария

|  |  |
| --- | --- |
| Параметр | Описание |
| ID теста | POS–03 |
| Название теста | Создание комментария |
| Предусловия | Пользователь должен быть авторизирован |
| Шаги | 1. На странице просмотра статей выбрать любую и нажать на нее  2. Внизу написать комментарий  3. Дождаться загрузки страницы |
| Тестовые данные | «Герта» |
| Ожидаемый результат | Отображение созданного комментария |

Негативные тестовые сценарии проверяют поведение системы при некорректных действиях пользователя или попытках нарушения правил работы системы. Первый негативный сценарий (таблица 17).

Таблица 17 – Сценарий тестирования валидация данных пользователя при регистрации

|  |  |
| --- | --- |
| Параметр | Описание |
| ID теста | NEG–04 |
| Название теста | Валидация учетных данных |
| Предусловия | Пользователь не зарегистрирован |
| Шаги | 1. На странице Регистрации в поле «Логин» указать любой логин  2. В поле «Пароль» ввести 1  3. Нажать «Зарегистрироваться» |
| Тестовые данные | Пароль: 1 |
| Ожидаемый результат | Отображение ошибки: «Длина пароля должна быть не менее 8 символов» |

Второй негативный сценарий (таблица 18) направлен на проверку устойчивости системы к попыткам выполнения SQL– инъекций при авторизации. Данный тестовый сценарий позволяет убедиться, что веб-приложение корректно обрабатывает вредоносные строки во входных данных, не допускает обхода механизма аутентификации и предотвращает выполнение потенциально опасных SQL– конструкций.

Таблица 18 – Сценарий тестирования попытка выполнении SQL – инъекции

|  |  |
| --- | --- |
| Параметр | Описание |
| ID теста | FUNC–NEG–05 |
| Название теста | Попытка выполнения SQL– инъекции в поле логина при авторизации |
| Предусловия | 1. Система доступна по домену 2. Пользователь не авторизован.   Отключены любые механизмы, автоматически выполняющие авторизацию. |
| Шаги | |  | | --- | |  |  |  | | --- | | 1. Ввести в поле «Имя пользователя» вредоносную SQL-строку 2. Ввести произвольный пароль 3. Нажать кнопку «Войти» | |
| Тестовые данные | 1. email: ' OR '1'='1, 2. email: '; DROP TABLE users;– – , 3. email: admin'– – , 4. email: 1' OR '1'='1' /\*,   password: test |
| Ожидаемый результат | Пользователь не проходит аутентификацию, сервер корректно обрабатывает вводимые данные, SQL-ошибки отсутствуют |

Третий негативный сценарий (таблица 29) направлен на проверку устойчивости системы к попыткам входа на аккаунт администратора. Данный тестовый сценарий позволяет убедиться, что пользователь не сможет зайти на учетную запись администратора.

Таблица 18 – Сценарий тестирования попытка выполнении SQL – инъекции

|  |  |
| --- | --- |
| Параметр | Описание |
| ID теста | ROLE–NEG–06 |
| Название теста | Проверка прав доступа пользователя к админ–панели |
| Предусловия | Пользователь не авторизирован |
| Шаги | |  | | --- | |  |  * 1. Перейти на страницу «Авторизоваться»   2. Ввести данные   3 Нажать «Войти» |
| Тестовые данные | admin  123456 |
| Ожидаемый результат | Доступ запрещен. Пользователь будет оповещен что введены неверные данные |

Разработанные функциональные тестовые сценарии подтвердили корректность работы основных функций веб-приложения и соответствие заданным требованиям. Позитивные сценарии показали, что веб-приложение правильно обрабатывает запросы авторизованных пользователей и возвращает необходимые данные в зависимости от их роли. Негативные сценарии продемонстрировали устойчивость приложения к ошибочным действиям, отсутствию авторизации, попыткам SQL-инъекций и перебор входа в аккаунт. Проведённое тестирование подтвердило надёжность, безопасность и корректность ключевых механизмов системы.

7 Документирование веб-приложения

7.1 Руководство по установке веб-приложения

**Техническое задание**

**Введение**

Настоящее техническое задание разработано для создания программного продукта HSRLib — игрового справочника по игре Honkai: Star Rail. Техническое задание определяет цели, требования и основные характеристики разрабатываемого программного обеспечения и предназначено для последующей реализации и тестирования системы.

**Общие сведения**

Наименование программного продукта: HSRLib  
 Разработчик: Рединов Д. М.  
 Тип программного продукта: информационно-справочная система  
 Платформа: веб-приложение (HTML, PHP, JS)  
 Средства разработки: PHP, MySQL/PostgreSQL, HTML, CSS, JavaScript, дополнительные библиотеки для анимаций и интерактивного интерфейса

Программный продукт разрабатывается для пользователей, интересующихся игровым контентом Honkai: Star Rail, желающих быстро получать структурированную информацию о персонажах, оружии, событиях и других игровых элементах.

**Назначение и цели создания программного продукта**

Назначение:  
Создание удобного и наглядного справочника, который позволяет пользователю:

* получать полные сведения о персонажах (характеристики, способности, навыки);
* просматривать информацию о доступном оружии, его параметрах и редкости;
* отслеживать события игры и актуальные обновления;
* использовать поисковые и фильтровые функции для быстрого нахождения информации.

**Цели создания:**

* Систематизация информации по игре Honkai: Star Rail в единой базе данных.
* Обеспечение удобного пользовательского интерфейса с возможностью быстрого поиска.
* Повышение информативности и удобства доступа к игровому контенту.
* Создание основы для возможного расширения функционала (например, рейтинги, избранное).

**Требования к системе в целом**

**Требования к структуре и функционированию программного продукта**

* Программный продукт должен предоставлять интерфейс для поиска и фильтрации информации о персонажах, оружии и событиях.
* Данные должны храниться в базе данных с возможностью добавления, редактирования и удаления записей через административный интерфейс.
* Интерфейс должен поддерживать адаптивное отображение на различных устройствах (ПК, планшеты, смартфоны).
* Программный продукт должен обеспечивать быстрый отклик на действия пользователя (поиск, фильтры, навигация).
* Реализация анимаций и интерактивных элементов должна быть безопасной и не снижать производительность системы.

**Требования к надежности**

* Система должна сохранять данные пользователей и справочную информацию без потерь.
* При ошибках или сбоях система должна корректно уведомлять пользователя об их возникновении.
* Предусмотреть резервное копирование базы данных.
* Обеспечить защиту от несанкционированного доступа к административной части системы.

**Требования к документированию**

* Разработчик должен подготовить документацию, включающую:
* описание структуры базы данных;
* описание пользовательского интерфейса и функций;
* инструкции по установке и запуску продукта;
* руководство пользователя и администратора.

**Состав и содержание работ по созданию программного продукта**

* Разработка структуры базы данных.
* Реализация пользовательского интерфейса с поддержкой поиска, фильтров и адаптивного дизайна.
* Создание административного интерфейса для управления контентом.
* Интеграция интерактивных элементов (анимации, визуализация данных).
* Тестирование программного продукта на соответствие требованиям.
* Подготовка документации и инструкций для пользователя и администратора.