Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

Высшего образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

**Курсовое проектирование**

по дисциплине «Распределенные программные системы»

**«Сайт-платформа для размещения онлайн инди игр»**

Выполнили:

студенты гр. ПРИ-123

О. Р. Стасюк

М. С. Олесиков

Принял:

Тимофеев А. А

Владимир 2025 г.

Цель платформы

Создание веб-платформы для инди-игр, которая сократит среднее время поиска и запуска игры пользователем с 45 до 10 минут за счёт единого каталога с интеллектуальным поиском, мгновенного запуска в браузере и системы рекомендаций.

Задачи платформы

1. Реализовать единый каталог игр с расширенной системой фильтров (по жанру, тегу, рейтингу) и полнотекстовым поиском.
2. Внедрить технологию запуска игр напрямую в браузере для мгновенного доступа.
3. Разработать административный модуль модерации контента для обеспечения качества и безопасности публикуемых материалов.
4. Создать личные кабинеты для разработчиков с инструментами для публикации игр, аналитики и получения обратной связи от игроков.

Список аналогов

1. **Itch.io**
   * Одна из самых известных площадок для размещения инди-игр, как браузерных, так и скачиваемых.
   * Преимущества: гибкий подход к монетизации (разработчик может сам выбирать, какой процент отдавать платформе) простота публикации (незначительные барьеры для входа)
   * Недостатки: большие объемы “шумного” контента и трудно выделяться среди множества проектов; жалобы на то, что игры внезапно теряют видимость в поиске; медленная служба поддержки по жалобам пользователей
   * Особенности: позволяет размещать не только игры, но и комиксы, музыки, ассеты и т. д.
2. **Game Jolt**
   * Является одним из конкурентов Itch.io
   * Преимущества: ориентированное на сообщество, поддержка инди-разработчиков, наличие системы наград / достижений, челленджей.
   * Недостатки / слабые места: меньше известности и меньшая аудитория по сравнению с Itch или крупными платформами.
3. **Steam**
   * Несмотря на то, что это **крупная коммерческая платформа**, многие инди-разработчики публикуют игры там.
   * Преимущества: огромная аудитория, авторитет, инструменты продвижения, видимость (в том числе через рекомендации).
   * Недостатки: высокие барьеры для входа — строгие требования, **конкуренция**, комиссия, требования к качеству.
4. **GOG.com**
   * Также является альтернативной платформой.
   * Преимущества: DRM-free политика (Digital Rights Management — управление цифровыми правами), акцент на качественные проекты.
   * Недостатки: строгий отбор, не все инди-проекты проходят модерацию.
5. **Indie.io**
   * Платформа, ориентированная на поддержку инди-разработчиков, с услугами тестирования, маркетинга, дистрибуции.
   * Преимущества: интегрированные сервисы продвижения и поддержки проекта, возможная помощь в дистрибуции на внешние площадки.
   * Недостатки: возможно, меньшая степень контроля у разработчика или зависимость от услуг платформы.

Команда и роли в команде.

Разработка программной системы ведётся малой командой из двух участников, каждый из которых выполняет полный цикл работ — от проектирования до реализации и тестирования.

Стасюк Олег — подсистема «Платформа для игроков». Цель: обеспечить пользователям (игрокам) доступ к играм, возможностям поиска, оценки и взаимодействия с контентом.

Олесиков Михаил — подсистема «Панель разработчика и администратора». Цель: предоставить инструменты для публикации, модерации и аналитики игр.

Анализ целевой аудитории проекта

**1. Прямая аудитория (основные пользователи)**

Это люди, для которых платформа создаётся в первую очередь:

* **Инди-разработчики** — авторы небольших игр или прототипов. Пример: студент-программист; хобби-разработчик.
* **Игроки (пользователи)** — активные пользователи, которые не только играют, но и пишут отзывы, формируют коллекции, участвуют в обсуждениях.

**2. Кому тоже может быть полезно (косвенная аудитория)**

Эти пользователи не являются основной целью платформы, но могут активно использовать её для своих задач:

* **Независимые издательства и паблишеры** — ищут перспективные проекты для поддержки или коммерческого релиза. Пример: маленькое издательство, мониторящее популярные прототипы.
* **Стримеры и блогеры** — используют платформу как источник свежего контента для видеообзоров и стримов.
* **Журналисты и критики** — ищут материал для статей о тенденциях в инди-разработке.

**3. Непрямой пользователь (контролирующая/обслуживающая аудитория)**

Эти участники не пользуются платформой напрямую для игр, но обеспечивают её стабильность, порядок и развитие:

* **Обслуживающий персонал** — модерация игр, комментариев, соблюдение правил, поддержка серверов, разработка новых функций, устранение багов.
* **Правообладатели и регулирующие органы** — могут вмешиваться при нарушении авторских прав или при размещении запрещённого контента.
* **Инвесторы/партнёры** — наблюдают за развитием платформы, её метриками и перспективами монетизации.

**Потребности целевой аудитории:**

Под проблемами понимаются первоочередные задачи, решения которых являются ключевой фукнциональностью системы:

1. Отсутствие у инди-разработчиков централизованной площадки для демонстрации и распространения своих проектов.
2. Необходимость тестировать систему, исправлять ошибки, обновлять контент и поддерживать актуальность проекта.
3. Не у всех игроков есть возможность скачивать и устанавливать игры. Браузерные игры в таком случае являются альтернативой.
4. Пользователям сложно находить новые проекты среди множества игр без удобного поиска и навигации.
5. Разработчики не имеют прямой аналитики по интересу к их играм и не понимают, что нужно улучшить.
6. Без обратной связи игроки не могут выразить своё мнение, а разработчики — понять реакцию аудитории.

Под ценностями понимаются задачи, решения которых в рамках системы отлично бы эту систему дополнили:

1. Пользователи не могут выражать индивидуальность и управлять своей учётной записью.
2. Пользователи теряются в массиве контента без навигационных подсказок и подборок.
3. Сложность в визуальной презентации игры без дополнительных материалов, что снижает интерес аудитории.
4. Игроки теряются в большом количестве игр и не могут удобно организовать интересные проекты для себя.

Календарный план проекта

В качестве календарного плана приведем диаграмму Ганта на осенний семестр (рис. 1)

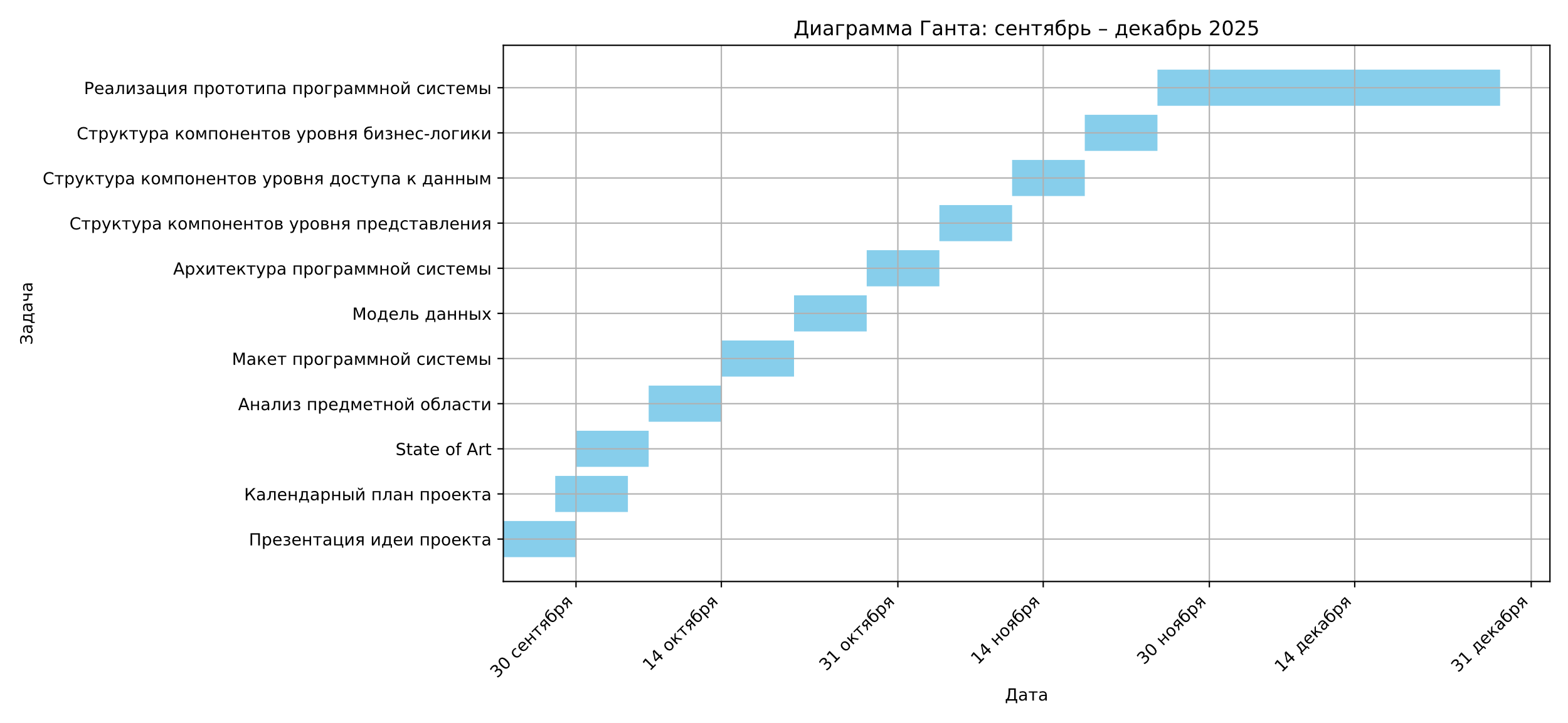


Рисунок 1. Диаграмма Ганта.

State of Art

В настоящее время существует ряд платформ, предназначенных для размещения и распространения инди-игр, среди которых наиболее известны Itch.io, Game Jolt, Steam и GOG.com. Эти сервисы позволяют разработчикам публиковать проекты, получать отзывы игроков и организовывать продажи. Однако большинство подобных систем ориентированы на коммерческое распространение, требуют сложной модерации или ориентированы на скачиваемые версии игр, что создаёт барьер для начинающих авторов.

Современные тенденции в игровой индустрии показывают рост интереса к веб-играм, запускаемым прямо в браузере, и к социальному взаимодействию между разработчиками и игроками. При этом существующие решения не предлагают интегрированной среды для публикации, обратной связи и продвижения небольших инди-проектов, создаваемых студентами и независимыми авторами.

Разрабатываемая система направлена на устранение этих недостатков, обеспечивая доступную, открытую и технически простую платформу для публикации, оценки и обсуждения онлайн инди-игр.

Обзор актуальных технологий

Для создания современной веб-платформы размещения онлайн-игр требуется использование как клиентских (frontend), так и серверных (backend) технологий, обеспечивающих интерактивность, безопасность и масштабируемость.

**Frontend (клиентская часть):**

1. HTML5, CSS3, JavaScript — базовые технологии для построения адаптивного интерфейса и интерактивных элементов.
2. React.js — современная библиотека для построения компонентных интерфейсов; обеспечивает высокую производительность и удобство в сопровождении.
3. Tailwind CSS — утилитарный фреймворк для быстрой стилизации и адаптивной верстки.
4. WebGL / Phaser.js / Three.js — средства для интеграции браузерных (веб-) игр и их предпросмотра прямо на сайте.

**Backend (серверная часть):**

1. Node.js + Express.js — платформа и фреймворк, обеспечивающие создание REST API и обработку запросов в реальном времени.
2. Python (FastAPI / Django) — альтернатива для построения серверной логики с акцентом на простоту и надёжность.
3. База данных: PostgreSQL / MongoDB — используются для хранения данных пользователей, игр, отзывов и статистики.
4. JWT (JSON Web Token) — механизм безопасной аутентификации и разграничения прав пользователей по ролям (гость, пользователь, разработчик, модератор, администратор).

**Сервисы и инфраструктура:**

1. Docker — контейнеризация приложения для облегчения развертывания и тестирования.
2. Nginx — обратный прокси-сервер и средство балансировки нагрузки.
3. GitHub / GitLab — системы контроля версий и совместной разработки.
4. Cloudflare / Firebase / AWS S3 — возможные решения для хостинга, хранения мультимедиа и статических ресурсов.

**Инструменты для аналитики и статистики:**

1. Google Analytics / Matomo — сбор и визуализация статистики просмотров и активности пользователей.
2. Chart.js / Recharts — библиотеки для вывода графиков и метрик в интерфейсе разработчика.

**Выводы о применимости рассматриваемых технологий**

Выбранные технологии отражают современные тенденции в веб-разработке и обеспечивают:

1. гибкость при создании масштабируемых веб-приложений;
2. высокую производительность и отзывчивость интерфейса благодаря React.js;
3. удобную интеграцию игр и мультимедиа за счёт WebGL и JavaScript-фреймворков;
4. надёжную серверную архитектуру на базе Node.js или Python с REST API;
5. безопасность и распределение ролей пользователей при помощи JWT;
6. простоту развёртывания и командной работы с Docker и Git.

Таким образом, указанные технологии полностью применимы для реализации всех компонентов системы — как клиентской, так и серверной части, — и соответствуют требованиям к современным веб-платформам для размещения онлайн-контента. Однако в нашей работе будет использоваться только часть данных технологий строго необходимых для реализации системы.

Анализ аналогов.

**Перечень критериев оценки**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Критерий** | **Вес (значимость)** | **Краткое пояснение** |
| 1 | Низкий порог вхождения для загрузки видеоигр | 10 | Простота публикации без сложных требований и комиссий |
| 2 | Возможность запуска игр онлайн (в браузере) | 10 | Поддержка WebGL/HTML5, возможность играть без установки |
| 3 | Возможность получать обратную связь (комментарии, отзывы) | 8 | Коммуникация между пользователями и разработчиками |
| 4 | Масштабируемость и надёжность системы | 8 | Уровень технической стабильности, производительность |
| 5 | Система управления версиями игр | 6 | Возможность обновления и хранения нескольких версий |
| 6 | Минимальная, но необходимая модерация контента | 4 | Контроль качества без избыточных ограничений |
| 7 | Профили пользователей, коллекции, фильтрация каталога | 2 | Удобство навигации и персонализации |

Таблица 1. Перечень критериев оценки.

**Сравнительная таблица аналогов**

| Платформа | Вход | Онлайн | Обратная связь | Надёжность | Версии | Модерация | Профили | Итог |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Itch.io](https://itch.io/) | 10 | 9 | 8 | 7 | 6 | 3 | 2 | (10×10 + 9×10 + 8×8 + 7×8 + 6×6 + 3×4 + 2×2) / 48 = 8.17 ≈ 8.2 |
| Game Jolt | 9 | 8 | 9 | 7 | 5 | 3 | 2 | (9×10 + 8×10 + 9×8 + 7×8 + 5×6 + 3×4 + 2×2) / 48 = 7.75 ≈ 7.8 |
| Steam | 3 | 2 | 8 | 10 | 10 | 8 | 3 | (3×10 + 2×10 + 8×8 + 10×8 + 10×6 + 8×4 + 3×2) / 48 = 6.58 ≈ 6.6 |
| [GOG.com](https://gog.com/) | 2 | 1 | 6 | 9 | 8 | 9 | 2 | (2×10 + 1×10 + 6×8 + 9×8 + 8×6 + 9×4 + 2×2) / 48 = 5.75 ≈ 5.8 |
| [Indie.io](https://indie.io/) | 6 | 1 | 5 | 6 | 4 | 6 | 1 | (6×10 + 1×10 + 5×8 + 6×8 + 4×6 + 6×4 + 1×2) / 48 = 5.25 ≈ 5.3 |

Таблица 2. Сравнительная таблица аналогов.

Примечание:

* Все оценки по 10-балльной шкале, где 10 — максимально полное соответствие критерию.
* Взвешенная сумма рассчитана через нормализацию по общей сумме весов (48).
* Оценки округлены до десятых для наглядности.

**Выводы по результатам анализа аналогов**

|  |  |  |
| --- | --- | --- |
| **Платформа** | **Итоговый балл** | **Место** |
| **Itch.io** | **8.2** | **1 место** |
| **Game Jolt** | **7.8** | **2 место** |
| **Steam** | **6.6** | **3 место** |
| **GOG.com** | **5.8** | **4 место** |
| **Indie.io** | **5.3** | **5 место** |

Таблица 3. Итоговая таблица с оценками.

1. Itch.io и Game Jolt — наиболее близкие к целям проекта платформы. Они ориентированы на независимых разработчиков, обеспечивают простую публикацию и взаимодействие с аудиторией.
2. Steam и GOG.com показывают высокую техническую надёжность и развитую систему версий, но имеют высокий порог входа и ориентацию на коммерцию, что делает их менее подходящими для начинающих инди-разработчиков.
3. Indie.io выполняет больше функцию агрегатора и каталога, а не полноценной платформы с функционалом обратной связи и публикации.
4. Разрабатываемая система будет сочетать простоту и открытость Itch.io с социальной вовлечённостью Game Jolt, но при этом обеспечит единое ролевое управление, контроль качества контента и запуск игр прямо в браузере.

Описание предметной области

**Состояние до внедрения программной системы**

1. Описание процесса публикации и поиска игр

Процесс публикации новых инди-игр и их поиска игроками является полностью ручным, децентрализованным и неэффективным. Он не поддерживается единой платформой и зависит от множества разрозненных инструментов и сервисов.

2. Измеримые критерии, характеризующие проблемы текущего состояния

2.1. Временныые затраты:

Среднее время публикации игры разработчиком: От завершения разработки до момента, когда игра становится доступна первым игрокам, проходит от 3 до 7 дней. Это время тратится на поиск подходящих площадок, регистрацию, загрузку файлов и настройку страниц.

Среднее время поиска конкретной инди-игры игроком: Поиск игры по жанру/тематике с использованием форумов, социальных сетей и поисковых систем занимает в среднем от 25 до 45 минут.

Время получения первой осмысленной обратной связи разработчиком: После публикации игры разработчик получает первые конструктивные отзывы и баг-репорты в среднем через 5-10 дней из-за низкой концентрации целевой аудитории в выбранных каналах.

2.2. Качественные и количественные метрики охвата и эффективности:

Коэффициент конверсии просмотра в запуск игры: На форумах и в соцсетях только ~1-3% пользователей, увидевших анонс игры, переходят к её скачиванию или запуску из-за множества промежуточных шагов (переход по ссылке, скачивание, распаковка, риск безопасности).

Процент игр, оставшихся незамеченными: Около 40% инди-проектов, опубликованных на персональных ресурсах разработчиков, не получают более 50 уникальных просмотров за первый месяц.

Доля недовольных пользователей из-за технических барьеров: До 15% потенциальных игроков отказываются от попытки запустить игру, столкнувшись с необходимостью установки дополнительного ПО, регистрации на файлообменнике или с распаковкой архива.

2.3. Затраты и операционная неэффективность:

Косвенные финансовые затраты разработчика: Разработчик тратит до 10-15 часов в месяц на нетворкинг и ручное продвижение своего проекта на различных площадках вместо работы над кодом.

Отсутствие структурированной модерации: 100% контента (описания, скриншоты, исполняемые файлы) не проходят никакой стандартизированной проверки, что приводит к появлению низкокачественных, misleading-анонсов и повышает риски безопасности для игроков.

Фрагментация обратной связи: Отзывы и баг-репорты от игроков распределены по в среднем 3-5 разным платформам (комментарии VK, Discord, почта, форум), что увеличивает вероятность потери обратной связи и время на её агрегацию.

3. Сводная проблема

Текущее состояние характеризуется высокими временными затратами на публикацию и поиск, низкой discoverability (возможностью быть обнаруженным) для инди-проектов, отсутствием доверия и безопасности для игроков, а также неэффективным циклом обратной связи. Это создаёт порочный круг, тормозящий развитие как отдельных разработчиков, так и российского инди-сообщества в целом.

**Перечень документов, регулирующих деятельность и имеющих отношение к бизнес-процессу**

Деятельность платформы регулируется следующими типами документов:

* Пользовательское соглашение (Terms of Service): определяет права и обязанности всех сторон (пользователей, разработчиков, администрации платформы).
* Политика конфиденциальности (Privacy Policy): регулирует сбор, хранение и обработку персональных данных пользователей.
* Лицензионное соглашение с конечным пользователем (EULA) для игр: разрабатывается и предоставляется самими разработчиками для их продуктов, размещаемых на платформе.
* Правила модерации контента: внутренний регламент, определяющий критерии допустимого контента (игр, скриншотов, описаний, комментариев) и процедуры его проверки.
* Политика в отношении интеллектуальной собственности (IP Policy): устанавливает процедуры для разрешения споров, связанных с нарушением авторских прав.

**Краткая характеристика бизнес-процесса до автоматизации**

| Этап процесса | Участники | Инструменты и носители информации | Проблемы и недостатки |
| --- | --- | --- | --- |
| **Создание и упаковка игры** | Разработчик | Локальная среда разработки (IDE), системы контроля версий (Git) |  |
| **Публикация и распространение** | Разработчик | Файлообменные сервисы (Google Disk, Яндекс.Диск), форумы, социальные сети | Фрагментация аудитории, риск потери файлов, нестабильные ссылки, отсутствие единого формата представления проекта |
| **Поиск и обнаружение игр** | Игрок | Поисковые системы, рекомендации в социальных сетях, форумы | Высокие временные затраты, сложность верификации качества и безопасности игры, отсутствие централизованного каталога |
| **Обратная связь и поддержка** | Разработчик, Игрок | Комментарии в социальных сетях, личные сообщения, email | Обратная связь неструктурирована, теряется, нет механизма сбора багрепортов, сложно вести диалог с сообществом |
| **Модерация и контроль качества** | Администратор/Модератор (при наличии) | Ручная проверка по жалобам пользователей | Реактивный, а не проактивный подход; высокая нагрузка на модераторов; отсутствие единых стандартов проверки |

Таблица 4. Бизнес-процесс программной системы до автоматизации.

**Необходимость реализации программной системы**

Реализация программной системы «Сайт-платформа для размещения онлайн инди игр» необходима для преодоления ключевых проблем, присущих текущему неавтоматизированному процессу:

1. Централизация и структурирование: Создание единой цифровой среды для разработчиков и игроков устраняет фрагментацию, предоставляя стандартизированные инструменты для публикации, поиска и запуска игр.
2. Снижение барьеров для разработчиков: Автоматизация процессов публикации, управления версиями и сбора аналитики позволяет начинающим инди-разработчикам сосредоточиться на творчестве, а не на решении инфраструктурных задач.
3. Систематизация обратной связи: Встроенные механизмы комментариев, рейтингов и коллекций создают структурированный канал коммуникации между игроками и разработчиками, что способствует улучшению качества игр.
4. Обеспечение безопасности и контроля качества: Внедрение процедур предварительной и/или пост-модерации контента, а также единых правил платформы, позволяет защитить пользователей от некачественного или вредоносного контента.
5. Создание сообщества: Платформа формирует ядро целевой аудитории, заинтересованной в инди-играх, способствуя росту и развитию данной отрасли в русскоязычном сегменте интернета.

Словарь предметной области

**Игра** — программный продукт, размещаемый на платформе разработчиком. Может иметь несколько версий, описание, жанр, скриншоты, трейлеры и файлы для запуска (онлайн или для скачивания).

**Версия игры** — конкретная редакция игры, загруженная разработчиком. Содержит дату выхода, список изменений и может заменять предыдущую.

**Жанр игры** — категория, к которой относится игра (например: платформер, шутер, стратегия, RPG и др.). Используется для поиска и фильтрации.

**Каталог игр** — структурированное хранилище опубликованных игр с возможностью сортировки, поиска и фильтрации по жанру, рейтингу, дате добавления и другим параметрам.

**Разработчик** — зарегистрированный пользователь платформы, который размещает игры, управляет их описанием, версиями и отслеживает статистику.

**Игрок (пользователь)** — человек, использующий платформу для поиска, скачивания или запуска игр. Может оставлять отзывы, ставить оценки, добавлять игры в коллекции.

**Профиль пользователя** — совокупность данных об игроке или разработчике, включающая личную информацию, историю активности, опубликованные игры или отзывы.

**Администратор** — пользователь с расширенными правами, отвечающий за модерацию контента (игры, комментарии, описания), управление пользователями и поддержание работы платформы.

**Категория доступа** — роль пользователя на платформе (гость, пользователь, разработчик, администратор), определяющая доступные функции.

**Рейтинг игры** — числовой показатель качества игры, основанный на оценках игроков. Используется для формирования топов и рекомендаций.

**Отзыв** — текстовое сообщение игрока, содержащее мнение об игре. Может сопровождаться оценкой.

**Коллекция игр** — список игр, сформированный пользователем (например: «Избранное», «Пройти позже»).

**Модерация** — процесс проверки и подтверждения соответствия контента (игр, отзывов, описаний) правилам платформы.

**Статистика** — аналитические данные о взаимодействии пользователей с играми (количество скачиваний, запусков, отзывов, средний рейтинг).

**Сущности подлежащие модерации** – это версия игры, отзыв, заявка на разработчика.

**Заявка на разработчика** – это электронная заявка от зарегистрированного пользователя на становление разработчиком ( при одобрении дает возможность пользоваться возможностями роли «Разработчик»).

Пользователи системы и роли

**Характеристика аудитории программной системы**

Аудиторию платформы можно сегментировать на три основные группы, исходя из их целей и взаимодействия с системой:

* **Разработчики (Инди-Студия / Студент-Разработчик / Хобби-Разработчик):**
  + **Цель:** Публикация игровых проектов, получение обратной связи от сообщества, установление связи с игроками и потенциальными издателями.
  + **Техническая грамотность:** Средняя или высокая. Комфортно работают с веб-интерфейсами, системами загрузки файлов, базовой аналитикой.
  + **Мотивация:** Признание, формирование портфолио, монетизация проектов.
  + **Ключевые потребности:** Простой и быстрый процесс публикации, инструменты для управления контентом, доступная аналитика по интересу к их играм.
* **Игроки (Энтузиаст инди-игр / Студент / Casual-игрок):**
  + **Цель:** Поиск новых интересных игр, удобный запуск (особенно браузерных), общение с разработчиками и другими игроками, формирование личной коллекции.
  + **Техническая грамотность:** Различная, от низкой до высокой. Интерфейс должен быть интуитивно понятным даже для неподготовленных пользователей.
  + **Мотивация:** Развлечение, открытие новых проектов, участие в комьюнити.
  + **Ключевые потребности:** Удобный поиск и фильтрация, возможность запускать игры без скачивания, система рейтингов и отзывов, личный кабинет с коллекциями.
* **Административный персонал (Администратор / Модератор):**
  + **Цель:** Обеспечение стабильности, безопасности и соответствия контента правилам платформы.
  + **Техническая грамотность:** Высокая. Являются сотрудниками или привлеченными специалистами, управляющими платформой.
  + **Мотивация:** Поддержание работоспособности и репутации платформы.
  + **Ключевые потребности:** Централизованная панель управления для модерации контента, управления пользователями и просмотра системной статистики.

**Расчетное количество пользователей системы**

На начальном этапе запуска платформы расчетное количество пользователей оценивается следующим образом:

| Роль | Расчетное количество (на старте) | Примечания |
| --- | --- | --- |
| Гости | 500 - 1000 уникальных посетителей/мес | Пользователи, просматривающие каталог без регистрации. |
| Пользователи | 200 - 400 зарегистрированных пользователей | Активные пользователи с личным кабинетом. |
| Разработчики | 20 - 50 зарегистрированных пользователей | Часть из них может быть одновременно и игроками. |
| Модераторы | 2 - 3 пользователя | Внутренние сотрудники или привлеченные администраторы. |
| Администраторы | 1 - 2 пользователя | Технические администраторы системы. |

Таблица 5. Расчетное количество пользователей системы.

**Перечень ролей пользователей**

| Роль | Описание | Ключевые возможности |
| --- | --- | --- |
| Гость | Неаутентифицированный пользователь. | • Просмотр каталога игр, фильтрация и сортировка, а также поиск игр по названию • Просмотр страниц игр и отзывов • Регистрация и вход в систему  • Запуск игры в браузере |
| Пользователь | Зарегистрированный пользователь. | • Все возможности гостя • Оставление отзывов и оценок, а таже просмотр их в профиле • Создание и управление персональными коллекциями игр (удаление, изменение названия, добавление игры) • Подача заявки на получение роли "Разработчик"  • Просмотр и редактирование своего профиля  • Получение и просмотр уведомлений |
| Разработчик | Пользователь с подтвержденными правами на публикацию игр. | • Все возможности пользователя • Публикация, редактирование и удаление собственных игр • Загрузка и управление версиями игр, медиафайлами • Просмотр опубликованных игр, а также статистики по ним (просмотры, скачивания, рейтинги) |
| Модератор | Пользователь, отвечающий за содержание на платформе. | • Все возможности пользователя • Просмотр сущностей, ожидающих модерацию • Модерирование (одобрение или отклонение публикаций, а также выдача роли «Разработчик») • Блокировка/разблокировка пользователей (кроме администраторов)  • Возможность выдать предупреждение |
| Администратор | Пользователь с полным доступом к управлению системой. | • Все возможности модератора • Полный доступ к и глобальной аналитике всех игр • Управление всеми пользователями, включая модераторов (лишение роли)  • Посмотреть пройденные модерации (для отслеживания работы модераторов) |

Таблица 6. Перечень ролей пользователей.

Прецеденты

На основе ранее выделенных ролей и возможностей этих ролей была составлена диаграмма прецедентов. Она представлена на рисунке 2.

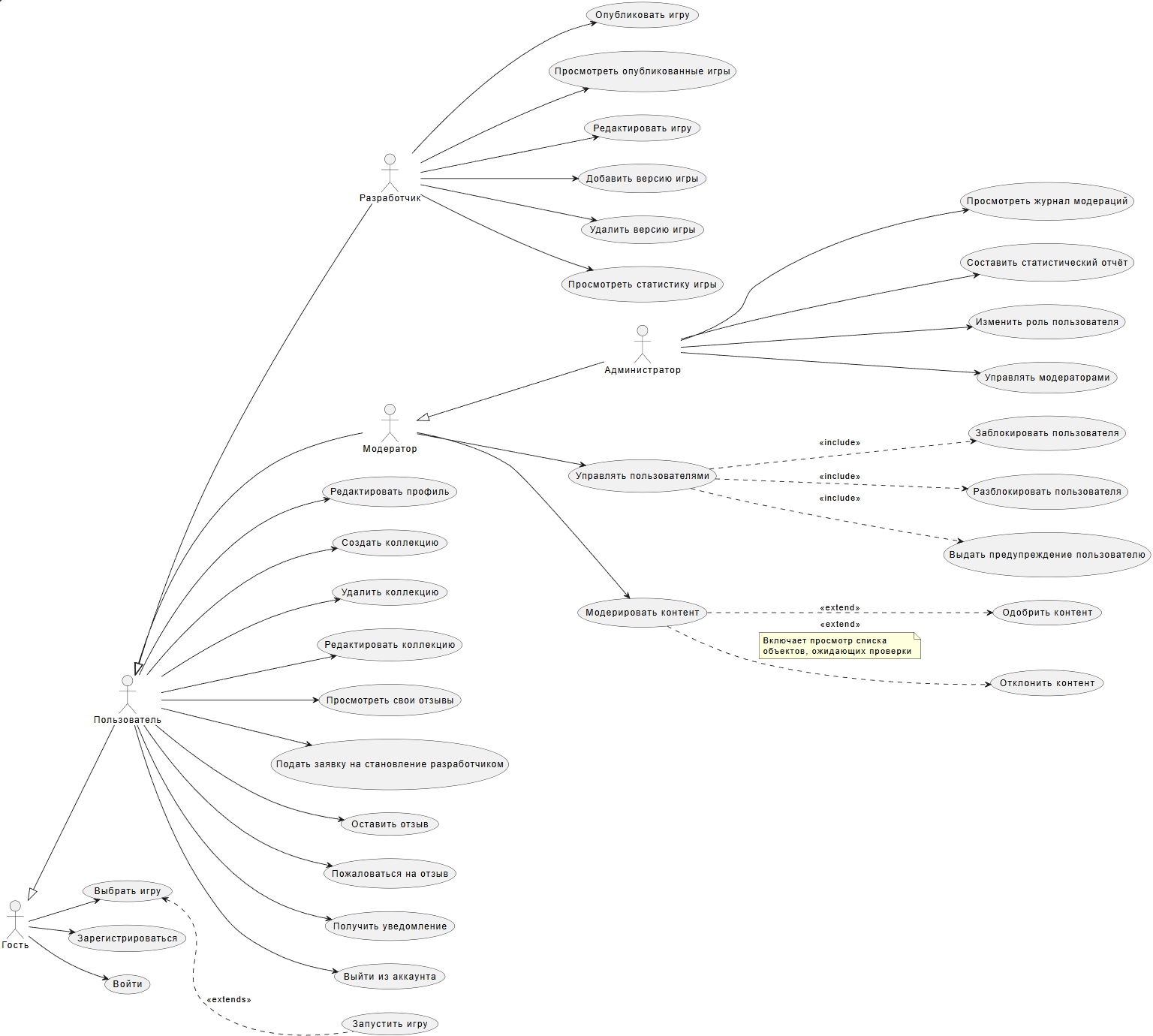


Рисунок 2. Диаграмма прецедентов.

**Описание прецедентов**

Прецедент “Выбрать игру”

**Предусловие**: —

**Действующее лицо**: Гость, Пользователь, Разработчик

**Основной поток**:

1. Гость переходит с главной страницы в каталог;
2. Гость выбирает критерий (предикат) для множества отображаемых игр (поисковая строка, фильтры, сортировка - или любая комбинация трёх);
3. Гость заполняет поисковую строку и/или выбирает фильтры и/или применяет сортировку;
4. Система отображает список игр, удовлетворяющих заданному предикату;
5. Гость просматривает полученный список и выбирает интересующую игру;
6. Система отображает страницу выбранной игры;

**Альтернативный поток**: *Множество игр, удовлетворяющих заданному критерию пустое*

1. Система отображает на странице сообщение о том, что игр, соответствующих критериям поиска нет;
2. Гость изменяет критерий поиска;
3. Переход к шагу 3 основного потока;

**Альтернативный поток**: *Пользователь переходит на страницу игры из профиля*

1. Пользователь заходит в профиль;
2. Переходит к разделу своих коллекций;
3. Выбирает интересующую коллекцию;
4. Нажимает на игру, содержащуюся в коллекции;
5. Система отображает страницу выбранной игры;

**Альтернативный поток**: *Выбранная коллекция пуста*

1. Шаги 1 – 3 альтернативного потока “*Пользователь переходит на страницу игры из профиля*” выполнены;
2. Так как коллекция пуста, чтобы выбрать игру пользователь переходит на главную страницу;
3. Переход к шагу 1 основного потока;

**Альтернативный поток**:

1. Разработчик переходит в профиль
2. Нажимает кнопку «Опубликованные игры»
3. Переходит на страницу опубликованных игр
4. Нажимает на одну из игр
5. Переходит на страницу игры

**Постусловие**: Гость/пользователь/разработчик просматривает страницу выбранной игры

Прецедент “Запустить игру”

**Предусловие**: Выполнен прецедент “Выбрать игру”

**Действующее лицо**: Гость

**Основной поток**:

1. Пользователь выбирает версию игры или оставляет версию по умолчанию (последнюю);
2. Пользователь нажимает кнопку “Играть”;

**Альтернативный поток**: *Версия игры недоступна*

1. Выполнен основной поток.
2. Система выдаёт сообщение об ошибке.
3. Переход к шагу 1 основного потока.

**Постусловие**: Версия игры успешно запущена в браузере гостя

Прецедент “Зарегистрироваться”

**Предусловие**: —

**Действующее лицо**: Гость

**Основной поток**:

1. Гость нажимает на кнопку “Зарегистрироваться”;
2. Гость заполняет форму регистрации;
3. Гость отправляет форму на сервер;
4. Система обрабатывает создаёт новую учетную запись;

**Альтернативный поток**: *Ошибка валидации формы*

1. *Первые 3 шага основного потока выполнены;*
2. *Валидация формы завершается ошибкой (пользователь уже существует, поле повторите пароль не совпадает с полем пароль и т.п.) и система сообщает об этом;*
3. *Переход к шагу 2 основного потока;*

**Постусловие**: Гость создал учетную запись

Прецедент “Войти”

**Предусловие**: —

**Действующее лицо**: Гость

**Основной поток**:

1. Гость нажимает на кнопку “Войти”;
2. Гость заполняет форму входа;
3. Гость отправляет форму на сервер;
4. Система обрабатывает форму и авторизует гостя;

**Альтернативный поток**: *Ошибка валидации формы*

1. *Первые 3 шага основного потока выполнены;*
2. *Валидация формы завершается ошибкой (неверный логин/пароль) и система сообщает об этом;*
3. *Переход к шагу 2 основного потока;*

**Постусловие**: Гость авторизован в системе

Прецедент “Редактировать профиль”

**Предусловие**: Пользователь авторизован в системе

**Действующее лицо**: Пользователь

**Основной поток**:

1. Пользователь переходит в свой профиль;
2. Нажимает кнопку редактировать;
3. Изменяет данные профиля (аватар, статус, имя);
4. Нажимает сохранить изменения;
5. Система обновляет данные в базе;

**Альтернативный поток**: —

**Постусловие**: Пользователь изменил данные профиля

Прецедент “Создать коллекцию”

**Предусловие**: Пользователь авторизован в системе

**Действующее лицо**: Пользователь

**Основной поток**:

1. Пользователь переходит в свой профиль;
2. Просматривает раздел коллекций в профиле;
3. Нажимает кнопку “Создать коллекцию”;
4. Даёт имя коллекции;
5. Нажимает кнопку создать;
6. Система добавляет коллекцию в базу данных;

**Альтернативный поток**: —

**Постусловие**: Новая коллекция создана и добавлена в БД

Прецедент “Удалить коллекцию”

**Предусловие**: Пользователь авторизован в системе

**Действующее лицо**: Пользователь

**Основной поток**:

1. Пользователь переходит в свой профиль;
2. Просматривает раздел коллекций в профиле;
3. Нажимает кнопку “Удалить коллекцию”;
4. Система выдаёт окно подтверждения с вариантами “Подтвердить” и “Отменить”;
5. Пользователь нажимает кнопку “Подтвердить”;
6. Система удаляет коллекцию из базы данных;

**Альтернативный поток**: *Пользователь отменяет удаление коллекции*

1. Первые 4 шага основного потока выполнены;
2. Пользователь нажимает кнопку “Отменить”;
3. Переход к шагу 2 основного потока;

**Постусловие**: Коллекция удалена из БД

Прецедент “Редактировать коллекцию”

**Предусловие**: Пользователь авторизован в системе

**Действующее лицо**: Пользователь

**Основной поток**: *Добавление игры в коллекцию на странице игры*

1. Пользователь открывает страницу игры (см. прецедент “Выбрать игру”);
2. Нажимает кнопку добавить в коллекцию;
3. Выбирает коллекцию из списка своих коллекций;
4. Нажимает ОК;
5. Система обновляет коллекцию в БД;

**Альтернативный поток**: *Удаление игры из коллекции на странице игры*

1. Шаг 1 основного потока выполнен;
2. Пользователь нажимает кнопку удалить из коллекции;
3. Выполняются шаги 3 – 5 основного потока;

**Альтернативный поток**: *Редактирование коллекции в профиле: добавление игры в коллекцию*

1. Пользователь переходит в профиль;
2. Переходит к разделу коллекций в профиле;
3. Нажимает “Редактировать коллекцию”;
4. Пользователь нажимает добавить игру
5. Пользователь выбирает игру (см. прецедент “Выбрать игру”);
6. Нажимает ОК;
7. Система обновляет коллекцию в БД;

**Альтернативный поток** :  *Ошибка при добавлении игры в коллекцию*

1. Шаги 1 – 4 основного потока или шаги 1 – 6 альтернативного потока “*Редактирование коллекции в профиле: добавление игры в коллекцию*” выполнены;
2. Игра уже есть в коллекции;
3. Система отображает сообщение об ошибке;
4. Переход к шагу 3 основного потока или к шагу 5 альтернативного потока;

**Альтернативный поток**: *Редактирование коллекции в профиле: удаление игры из коллекции*

1. Пользователь переходит в профиль;
2. Переходит к разделу коллекций в профиле;
3. Нажимает “Редактировать коллекцию”;
4. Пользователь нажимает удалить игру
5. Пользователь выбирает игру из данной коллекции;
6. Нажимает ОК;
7. Система обновляет коллекцию в БД;

**Альтернативный поток**: *Редактирование коллекции в профиле: изменение названия коллекции*

1. Пользователь переходит в профиль;
2. Переходит к разделу коллекций в профиле;
3. Нажимает “Редактировать коллекцию”;
4. Пользователь нажимает изменить название коллекции;
5. Вводит новое название;
6. Нажимает ОК;
7. Система обновляет коллекцию в БД;

**Постусловие**: Внесенные в коллекцию изменения сохранены в БД

Прецедент “Просмотреть свои отзывы”

**Предусловие**: Пользователь авторизован в системе

**Действующее лицо**: Пользователь

**Основной поток**:

1. Пользователь открывает свой профиль;
2. Переходит во вкладку “Мои отзывы”;
3. Система загружает отзывы на страницу;
4. Пользователь просматривает свои отзывы;

**Альтернативный поток**: *Отзывов нет*

1. Шаги 1 – 2 основного потока выполнены.
2. Система выдаёт сообщение о том, что оставленных отзывов нет;
3. Переход к шагу 1 основного потока;

**Постусловие**: Пользователь просмотрел свои отзывы

Прецедент “Подать заявку на становление разработчиком”

**Предусловие**: Пользователь авторизован в системе; пользователь не является модератором, разработчиком или администратором

**Действующее лицо**: Пользователь

**Основной поток**:

1. Пользователь переходит в профиль;
2. Нажимает “Стать разработчиком”;
3. Заполняет форму разработчика;
4. Нажимает ОК;
5. Система сохраняет заявку в БД;

**Альтернативный поток**: *Пользователь отменяет отправку формы*

1. Первые 3 шага основного потока выполнены;
2. Пользователь нажимает Отмена;
3. Переход к шагу 1 основного потока;

**Постусловие**: Заявка на становление разработчиком сохранена в БД

Прецедент “Оставить отзыв”

**Предусловие**: Пользователь авторизован в системе

**Действующее лицо**: Пользователь

**Основной поток**:

1. Пользователь выбирает игру (см. прецедент “Выбрать игру”);
2. Листает страницу до раздела отзывов;
3. Нажимает “Оставить отзыв”;
4. Заполняет текстовое поле и выставляет оценку игры по пятибалльной шкале;
5. Нажимает “Отправить отзыв”;
6. Система проверяет текст отзыва на запрещенные слова (матерные слова, упоминания политических деятелей и т.п.);
7. Система сохраняет отзыв в БД;

**Альтернативный поток**: *Пользователь отменяет процесс написания отзыва*

1. Шаги 1 – 3 или 1 – 4 основного потока выполнены;
2. Пользователь нажимает “Отмена”;
3. Переход к шагу 2 основного потока;

**Альтернативный поток**: *Отзыв не прошёл валидацию*

1. Шаги 1 – 5 основного потока выполнены;
2. Система выдала сообщение об ошибке валидации;
3. Переход к шагу 4 основного потока;

**Постусловие**: Отзыв пользователя добавлен в БД и помечен как подлежащий модерации

Прецендент «Пожаловаться на отзыв»

**Предусловие**: Пользователь авторизован в системе

**Действующее лицо**: Пользователь

**Основной поток**:

* + - 1. Пользователь выбирает игру (см. прецедент «Выбрать игру»)
      2. Листает страницу до раздела отзывов
      3. Выбирает неприемлемый отзыв
      4. Нажимает кнопку «Пожаловаться»
      5. Система увеличивает количество жалоб на данный отзыв на один

**Альтернативный поток**: –

**Постусловие**: Количество жалоб на данный отзыв увеличено на один

Прецедент “Получить уведомление”

**Предусловие**: Пользователь авторизован в системе

**Действующее лицо**: Пользователь

**Основной поток**:

1. В системе происходит событие (отзыв пользователя прошёл/не прошёл модерацию, игра разработчика прошла/ не прошла модерацию, модератор выдал предупреждение пользователю, администратор опубликовал уведомление о новостях системы и т.п.)
2. Пользователь получает уведомление;
3. Количество непрочитанных уведомлений отображается как цифра рядом с иконкой профиля;

**Альтернативный поток**: —

**Постусловие**: *Пользователь получил уведомление и может его просмотреть в профиле*

Прецедент “Выйти из аккаунта”

**Предусловие**: Пользователь авторизован в системе

**Действующее лицо**: Пользователь

**Основной поток**:

1. Пользователь переходит в профиль;
2. Листает страницу и находит кнопку “Выйти из аккаунта”;
3. Система отображает окно подтверждения;
4. Пользователь нажимает ОК и выходит из аккаунта;
5. Пользователь оказывается на главной странице в роли гостя;

**Альтернативный поток**: *Пользователь нажимает “Отмена”*

1. Шаги 1 – 3 основного потока выполнены;
2. Пользователь нажимает Отмена и остаётся в своём профиле;
3. Переход к шагу 2 основного потока;

**Постусловие**: Пользователь вышел из аккаунта и перешёл в роль гостя

Прецедент “Опубликовать игру”

**Предусловие**: Разработчик авторизован в системе

**Действующее лицо**: Разработчик

**Основной поток**:

1. Разработчик переходит в свой профиль;
2. Нажимает “Опубликовать игру”;
3. Заполняет форму создания новой игры (название, которое нельзя сменить, описание, изображения, ссылка на репозиторий GitHub, хэш коммита, версия игры);
4. Нажимает “Опубликовать”;
5. Система создаёт запись о новой версии игры в БД и помечает её как подлежащую модерации;

**Альтернативный поток**: *Пользователь отменяет процесс публикации*

1. Шаги 1 – 3 основного потока выполнены;
2. Пользователь нажимает отмена;
3. Переход к шагу 1 основного потока;

**Альтернативный поток**: *Валидация формы не прошла*

1. Шаги 1 – 4 основного потока выполнены;
2. Система проверяет поля формы;
3. Система выдаёт ошибку валидации формы;
4. Переход к шагу 3 основного потока;

**Постусловие**: Записи новой игры и новой версии игры сохранены в БД. Версия игры помечена как подлежащая модерации.

Прецедент “Просмотреть опубликованные игры”

**Предусловие**: Разработчик авторизован в системе

**Действующее лицо**: Разработчик

**Основной поток**:

1. Разработчик переходит в профиль;
2. Листает страницу до раздела опубликованных игр;
3. Просматривает опубликованные игры;

**Альтернативный поток**: —

**Постусловие**: Разработчик просмотрел опубликованные игры

Прецедент “Редактировать игру”

**Предусловие**: Разработчик авторизован в системе

**Действующее лицо**: Разработчик

**Основной поток**:

1. Пользователь выбирает свою игру (см. прецедент “Выбрать игру”)
2. Нажимает “Редактировать”;
3. Изменяет название и/или описание и/или изображения игры;
4. Нажимает “Сохранить”;
5. Система обновляет данные игры в БД;

**Альтернативный поток**: *Разработчик отменяет изменения*

1. Шаги 1 – 3 основного потока выполнены;
2. Разработчик нажимает “Отмена”;
3. Переход к шагу 1 основного потока;

**Постусловие**: Данные игры отображаемые на странице игры изменены

Прецедент “Добавить версию игры”

**Предусловие**: Разработчик авторизован в системе

**Действующее лицо**: Разработчик

**Основной поток**:

1. Разработчик выбирает свою игру (см. прецедент “Выбрать игру”);
2. Нажимает “Добавить версию”;
3. Заполняет форму добавления новой версии (Ссылка на репозиторий, хэш коммита, список файлов, текст-список изменений);
4. Нажимает ОК;
5. Система загружает файлы с GitHub в БД в виде архива и помечает версию игры как подлежащую модерации;

**Альтернативный поток**: *Разработчик отменяет добавление версии*

1. Шаги 1 – 3 основного потока выполнены;
2. Разработчик нажимает Отмена;
3. Переход к шагу 1 основного потока;

**Постусловие**: Версия игры загружена в БД и помечена как подлежащая модерации

Прецедент “Удалить версию игры”

**Предусловие**: Разработчик авторизован в системе

**Действующее лицо**: Разработчик

**Основной поток**:

1. Разработчик выбирает свою игру (см. прецедент “Выбрать игру”);
2. Нажимает “Удалить версию”;
3. Нажимает ОК;
4. Система удаляет архив файлов версии игры из БД;

**Альтернативный поток**: *Разработчик отменяет удаление версии*

1. Шаги 1 – 2 основного потока выполнены;
2. Разработчик нажимает Отмена;
3. Переход к шагу 1 основного потока;

**Постусловие**: Версия игры удалена из БД

Прецедент “Просмотреть статистику игры”

**Предусловие**: Разработчик авторизован в системе

**Действующее лицо**: Разработчик

**Основной поток**:

1. Разработчик выбирает свою игру (см. прецедент “Выбрать игру”);
2. Листает страницу до раздела статистики;
3. Просматривает статистику

**Альтернативный поток**: —

**Постусловие**: Разработчик просмотрел статистику своей игры

Прецедент “Управлять пользователями”

**Предусловие:** Модератор авторизован в системе.  
**Действующее лицо:** Модератор.  
**Основной поток:**

1. Модератор переходит в раздел "Управление пользователями".
2. Система отображает список пользователей с возможностью фильтрации и поиска.
3. Модератор выбирает целевого пользователя из списка.
4. Система предоставляет модератору набор доступных действий: "Заблокировать", "Разблокировать", "Выдать предупреждение".
5. Модератор выбирает требуемое действие и выполняет соответствующий сценарий (см. вложенные прецеденты).

**Альтернативный поток:** Пользователь не найден.

1. На шаге 2 основного потока система не находит пользователей по заданным критериям.
2. Система отображает сообщение "Пользователи не найдены".
3. Прецедент завершается.

**Альтернативный поток:** Модератор отменяет действие.

1. На любом шаге выбора или заполнения формы модератор нажимает "Отмена".
2. Система возвращает модератора к шагу 2 основного потока (список пользователей).

**Постусловие:** Выбранное действие в отношении пользователя выполнено (или отменено). Состояние пользователя в системе обновлено.

Прецедент “Заблокировать пользователя” (включен в “Управлять пользователями”)

**Предусловие:** Модератор выбрал пользователя для блокировки в рамках прецедента "Управлять пользователями".  
**Действующее лицо:** Модератор.  
**Основной поток:**

1. Модератор нажимает кнопку “Заблокировать” в интерфейсе управления выбранным пользователем.
2. Система отображает форму блокировки (причина, срок).
3. Модератор заполняет форму и нажимает "Подтвердить".
4. Система проверяет, что целевой пользователь не имеет роли Модератор или Администратор.
5. Система помечает пользователя как заблокированного и записывает блокировку в журнал.
6. Система автоматически разлогинивает заблокированного пользователя, если он онлайн.

**Альтернативный поток A:** Попытка блокировки модератора или администратора.

1. На шаге 4 основного потока система обнаруживает, что целевой пользователь имеет роль Модератор или Администратор.
2. Система отображает ошибку: "Блокировка пользователей с повышенными привилегиями доступна только Администратору".
3. Прецедент возвращается в точку выбора действия (шаг 4 основного потока "Управлять пользователями").

**Альтернативный поток B:** Отмена блокировки.

1. На шагах 2-3 основного потока модератор нажимает "Отмена".
2. Прецедент возвращается в точку выбора действия (шаг 4 основного потока "Управлять пользователями").

**Постусловие:** Пользователь заблокирован на указанный срок и не имеет доступа к системе.

Прецедент “Разблокировать пользователя” (включен в “Управлять пользователями”)

**Предусловие:** Модератор выбрал заблокированного пользователя в рамках прецедента "Управлять пользователями".  
**Действующее лицо:** Модератор.  
**Основной поток:**

1. Модератор нажимает кнопку “Разблокировать” в интерфейсе управления выбранным пользователем.
2. Система запрашивает подтверждение с возможностью указания причины.
3. Модератор подтверждает действие.
4. Система снимает метку блокировки с пользователя и фиксирует разблокировку в журнале.

**Альтернативный поток:** Отмена разблокировки.

1. На шаге 2 основного потока модератор нажимает "Отмена".
2. Прецедент возвращается в точку выбора действия (шаг 4 основного потока "Управлять пользователями").

**Постусловие:** Пользователь разблокирован и имеет доступ к системе.

Прецедент “Выдать предупреждение пользователю” (включен в “Управлять пользователями”)

**Предусловие:** Модератор выбрал пользователя в рамках прецедента "Управлять пользователями".  
**Действующее лицо:** Модератор.  
**Основной поток:**

1. Модератор нажимает кнопку “Выдать предупреждение”.
2. Система отображает форму выбора типа/тяжести предупреждения и поля для комментария.
3. Модератор заполняет форму и нажимает "Подтвердить".
4. Система создаёт запись о предупреждении, привязанную к пользователю.
5. Система генерирует уведомление для пользователя (см. прецедент “Получить уведомление”).

**Альтернативный поток:** Отмена выдачи предупреждения.

1. На шагах 2-3 основного потока модератор нажимает "Отмена".
2. Прецедент возвращается в точку выбора действия (шаг 4 основного потока "Управлять пользователями").

**Постусловие:** В системе создана запись о предупреждении, пользователь уведомлен.

Прецедент “Модерировать контент”

**Предусловие:** Модератор авторизован в системе.  
**Действующее лицо:** Модератор.  
**Основной поток:**

1. Модератор переходит в раздел "Модерация контента".
2. Система отображает единый список объектов, ожидающих проверки (версии игр, заявки на разработчика, отмеченные отзывы).
3. Модератор выбирает объект из списка для детального просмотра.
4. Модератор изучает содержимое объекта (код/бинарник игры, текст заявки, текст отзыва).
5. Модератор принимает решение и выполняет сценарий "Одобрить контент" или "Отклонить контент".

**Альтернативный поток:** Нет объектов для модерации.

1. На шаге 2 основного потока система не находит объектов, требующих проверки.
2. Система отображает сообщение "Нет контента для модерации".
3. Прецедент завершается.

**Постусловие:** Выбранный объект проверен, решение модератора зафиксировано в системе.

Прецедент “Одобрить контент” (расширяет “Модерировать контент”)

**Предусловие:** Модератор просматривает объект, требующий модерации, в рамках прецедента "Модерировать контент".  
**Действующее лицо:** Модератор.  
**Основной поток:**

1. Модератор нажимает кнопку "Одобрить".
2. Система запрашивает подтверждение.
3. Модератор подтверждает действие.
4. Система применяет положительный вердикт модерации к объекту:
   * Для версии игры: статус меняется на "Одобрена", игра становится доступна для запуска.
   * Для заявки на разработчика: статус меняется на "Одобрена", пользователю назначается роль "Разработчик".
   * Для отзыва: счётчик жалоб обнуляется, отзыв остаётся видимым.
5. Система уведомляет инициатора объекта (разработчика/пользователя) об одобрении.

**Постусловие:** Объект одобрен, его статус в системе изменён соответствующим образом.

Прецедент “Отклонить контент” (расширяет “Модерировать контент”)

**Предусловие:** Модератор просматривает объект, требующий модерации, в рамках прецедента "Модерировать контент".  
**Действующее лицо:** Модератор.  
**Основной поток:**

1. Модератор нажимает кнопку "Отклонить".
2. Система отображает форму для указания обязательной причины отклонения.
3. Модератор заполняет причину и нажимает "Подтвердить".
4. Система применяет отрицательный вердикт модерации к объекту:
   * Для версии игры: статус меняется на "Отклонена", версия не публикуется.
   * Для заявки на разработчика: статус меняется на "Отклонена".
   * Для отзыва: отзыв скрывается или удаляется.
5. Система уведомляет инициатора объекта (разработчика/пользователя) об отклонении с указанием причины.

**Альтернативный поток:** Отмена отклонения.

1. На шаге 2-3 основного потока модератор нажимает "Отмена".
2. Система возвращает модератора к детальному просмотру объекта (шаг 4 основного потока "Модерировать контент").

**Постусловие:** Объект отклонён, его статус в системе изменён, инициатор уведомлён.

Прецедент “Просмотреть журнал модераций”

**Предусловие:** Администратор авторизован в системе.  
**Действующее лицо:** Администратор.  
**Основной поток:**

1. Администратор переходит в раздел "Администрирование" -> "Журнал модераций".
2. Система отображает таблицу с историей всех действий модерации (кем, когда, какой объект, какое решение, причина).
3. Администратор может применять фильтры (по модератору, типу объекта, дате, решению).

**Постусловие:** Администратор ознакомился с журналом действий модераторов.

Прецедент “Управлять модераторами”

**Предусловие:** Администратор авторизован в системе.  
**Действующее лицо:** Администратор.  
**Основной поток:**

1. Администратор переходит в раздел "Администрирование" -> "Управление модераторами".
2. Система отображает список пользователей с ролью "Модератор", а также кандидатов.
3. Администратор выбирает пользователя из списка.
4. Система предоставляет действия: "Назначить модератором", "Лишить прав модератора", "Просмотреть статистику активности".
5. Администратор выбирает и выполняет необходимое действие.

**Альтернативный поток:** Попытка снять права с единственного администратора.

1. На шаге 4 администратор пытается снять роль модератора/администратора с учётной записи, которая является единственной с ролью Администратор.
2. Система предотвращает это действие и отображает ошибку: "В системе должен остаться как минимум один администратор".

**Постусловие:** Состав и права модераторов обновлены.

Описание документов/сущностей предметной области

Концептуальная диаграмма показывает сущности и их связи, не углубляясь в методы и детали реализации. Поэтому я добавлю только самые важные атрибуты, которые определяют состояние и суть каждой сущности.

Она представлена на рисунке 3.

Также были созданы диаграммы состояний для объектов «Версия игры» и «Заявка на становление разработчиком» (рисунки 4 и 5)

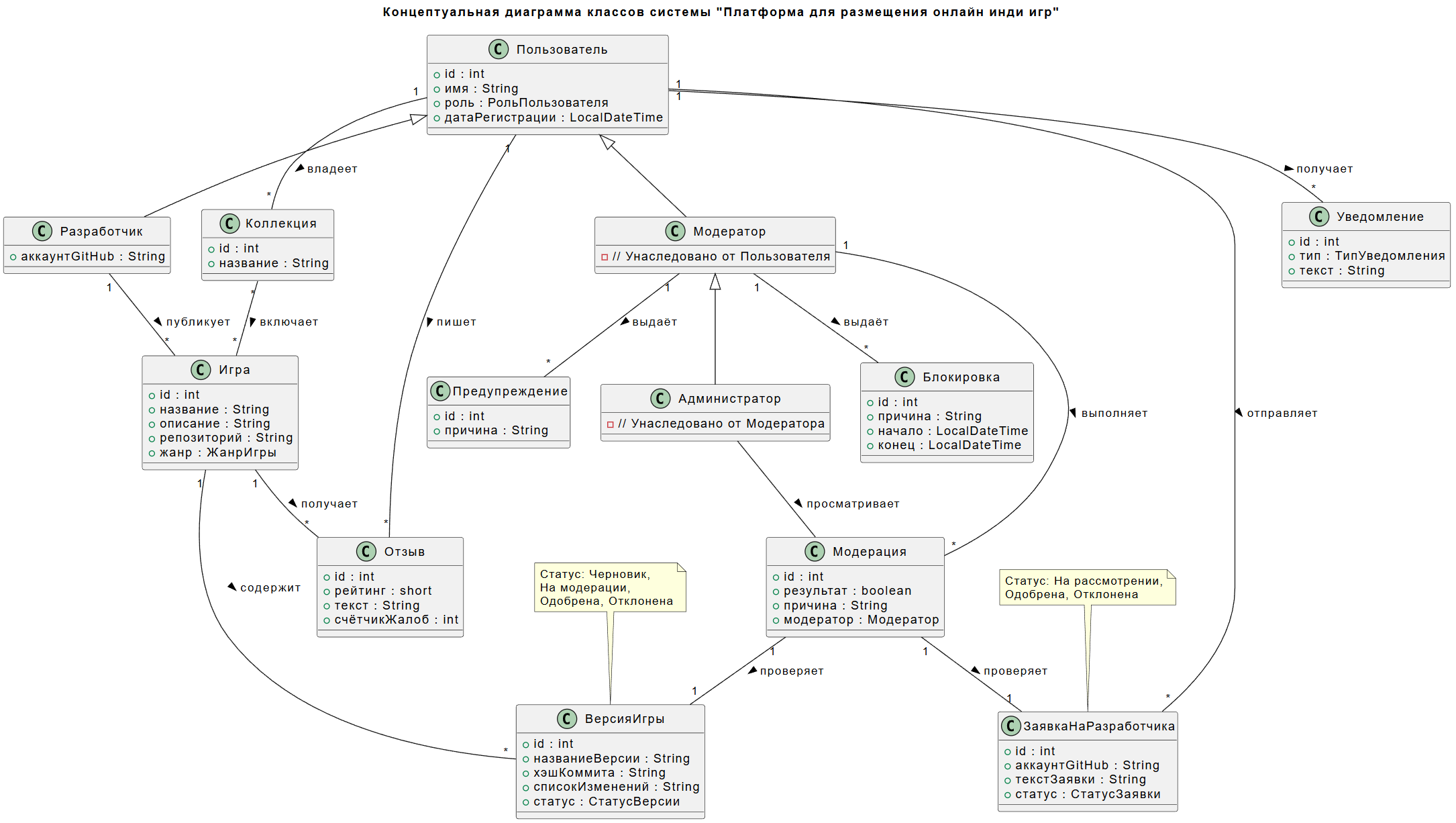


Рисунок 3. Концептуальная диаграмма классов.



Рисунок 4. Диаграмма состояний основных документов **"Заявка на становление разработчиком"**.

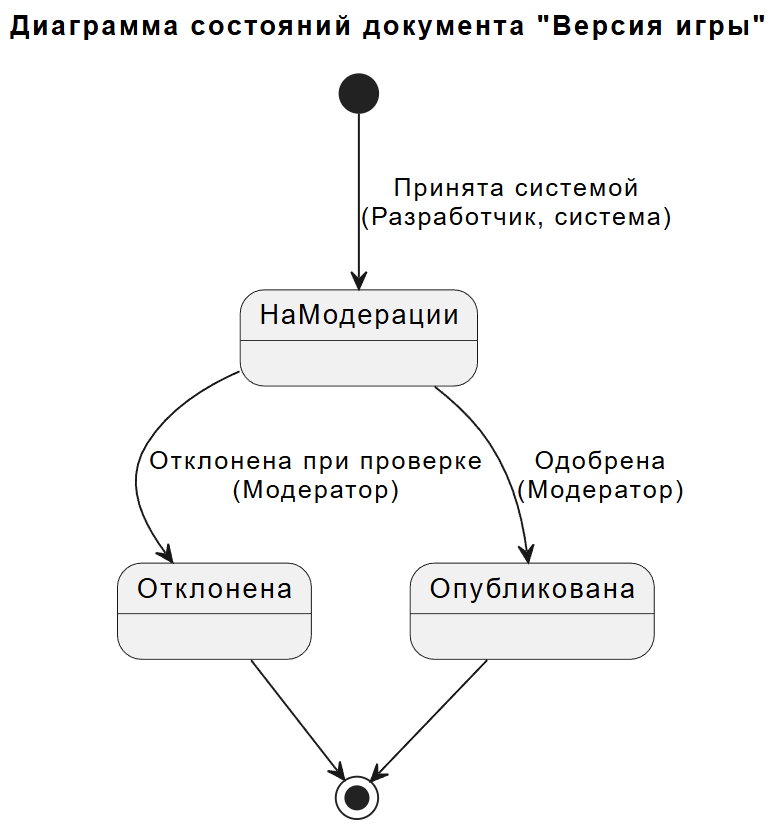


Рисунок 5. Диаграмма состояний основных документов **"Версия игры"**.

Анализ бизнес-процессов

**Основные бизнес-процессы, подлежащие автоматизации**

В рамках платформы для размещения онлайн инди-игр автоматизации подлежат следующие ключевые бизнес-процессы:

1. Модерировать и опубликовать игру.
2. Просмотреть и запустить игру.
3. Управлять обратной связью (отзывы, оценки, комментарии).
4. Управлять пользователями и ролями.
5. Аналитика и отчётность.

**Структурная декомпозиция бизнес-процессов**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Этап | Участники | Входные данные | Выходные данные | Инструменты | Регламентирующие документы | Временные рамки |
| 1. Загрузка игры | Разработчик | Файлы игры (в виде ссылки на Github репозиторий) | Заявление на публикацию | Панель разработчика, веб-интерфейс загрузки | Технические стандарты | До 1 дня |
| 2. Модерация | Модератор | Заявление на публикацию | Решение модерации (Одобрено/Отклонено), уведомление разработчику | Панель модератора | Правила модерации контента | 1 день |
| 3. Публикация | Система | Одобренная игра | Игра в общем каталоге | База данных, файловое хранилище, фронтенд | Правила публикации контента | Автоматически |

Таблица 7. Бизнес-процесс A1. Модерировать и опубликовать игру.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Этап | Участники | Входные данные | Выходные данные | Инструменты | Регламентирующие документы | Временные рамки |
| 1. Поиск и фильтрация | Пользователь/Гость | Запрос, фильтры, жанры | Результаты поиска | Каталог, реализация поиска | Пользовательское соглашение | Мгновенно |
| 2. Просмотр страницы игры | Пользователь/Гость | Выбранная игра | Страница игры (описание, медиа, отзывы) | Веб-интерфейс, фронтенд | Политика конфиденциальности | Мгновенно |
| 3. Запуск игры | Пользователь/Гость | Сборка игры | Запущенная игра в браузере | Игровой плеер | Технические стандарты | Мгновенно |

Таблица 8. Бизнес-процесс A2. Просмотреть и запустить игру.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Этап | Участники | Входные данные | Выходные данные | Инструменты | Регламентирующие документы | Временные рамки |
| 1. Добавление отзыва | Пользователь | Текст, рейтинг, скриншот | Новый отзыв | Интерфейс отзывов | - | Мгновенно |
| 2. Модерация отзывов | Система | Новый отзыв | Одобренный или удалённый отзыв | БД, фронтенд | Правила модерации | Автоматически |
| 3. Отображение отзывов | Система | Проверенные отзывы | Список отзывов на странице игры | БД, API, фронтенд | Правила публикации контента | Автоматически |

Таблица 9. Бизнес-процесс A3. Управлять обратной связью.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Этап | Участники | Входные данные | Выходные данные | Инструменты | Регламентирующие документы | Временные рамки |
| 1. Регистрация и авторизация | Гость/Пользователь | Учетные данные (логин, пароль, email) | Аккаунт пользователя | Форма регистрации\аутентификации | Политика конфиденциальности | Мгновенно |
| 2. Заявка на роль разработчика | Пользователь | Форма заявки | Заявка в очереди | Панель пользователя | Регламент назначения ролей | До 1 дня |
| 3. Управление ролями и блокировками | Администратор, Модератор | Аккаунты пользователей | Обновлённые роли / уведомления | Панель администратора/модератора | Правила модерации/администрирования | По запросу |
| 4. Выдача предупреждений | Модератор | Нарушения правил | Предупреждение пользователю | Панель модератора | Правила модерирования/поведения | По мере необходимости |

Таблица 10. Бизнес-процесс A4. Управлять пользователями и ролями.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Этап | Участники | Входные данные | Выходные данные | Инструменты | Регламентирующие документы | Временные рамки |
| 1. Сбор статистики | Система | Логи посещений, активности пользователей | Сырые данные аналитики | Backend, БД | - | Постоянно |
| 2. Формирование отчётов | Система, Администратор, Разработчик | Статистика из базы данных | Отчёты (графики, таблицы) | Аналитический модуль | - | Периодически/по запросу |
| 3. Анализ показателей | Администратор | Сформированные отчёты | Рекомендации по улучшению | Панель администратора | - | По мере необходимости |

Таблица 11. Бизнес-процесс A5. Аналитика и отчётность.

**Временные рамки выполнения подпроцессов**

|  |  |
| --- | --- |
| Категория процесса | Средняя продолжительность выполнения |
| Автоматические процессы (загрузка, валидация, публикация) | До 5 минут |
| Действия пользователя (просмотр, отзыв, регистрация) | Мгновенно |
| Модерация контента и ролей | 1–2 рабочих дня |
| Формирование отчётов и аналитика | Периодически (раз в неделю/месяц), либо по запросу |

Таблица 12. Временные рамки выполнения подпроцессов.

**Зоны ответственности пользователей системы (роли)**

|  |  |  |
| --- | --- | --- |
| Роль | Ответственность | Основные функции |
| Пользователь | Пользователь, взаимодействующий с играми (может писать отзывы) | Поиск, фильтрация и запуск игр; оценка и отзыв; создание коллекций |
| Разработчик | Публикация и сопровождение игр | Загрузка версий и медиа; редактирование описания; просмотр статистики |
| Модератор | Контроль качества контента и пользователей | Проверка новых версий игр и отзывов и заявлений; одобрение или отклонение публикаций; блокировки и предупреждения |
| Администратор | Управление системой и пользователями | Назначение ролей (в том числе админов), контроль статистики, поддержание правил и политик |
| Гость | Пользователь, который может только просматривать и играть | Поиск, фильтрация и запуск игр |

Таблица 13. Зоны ответственности пользователей системы (роли).

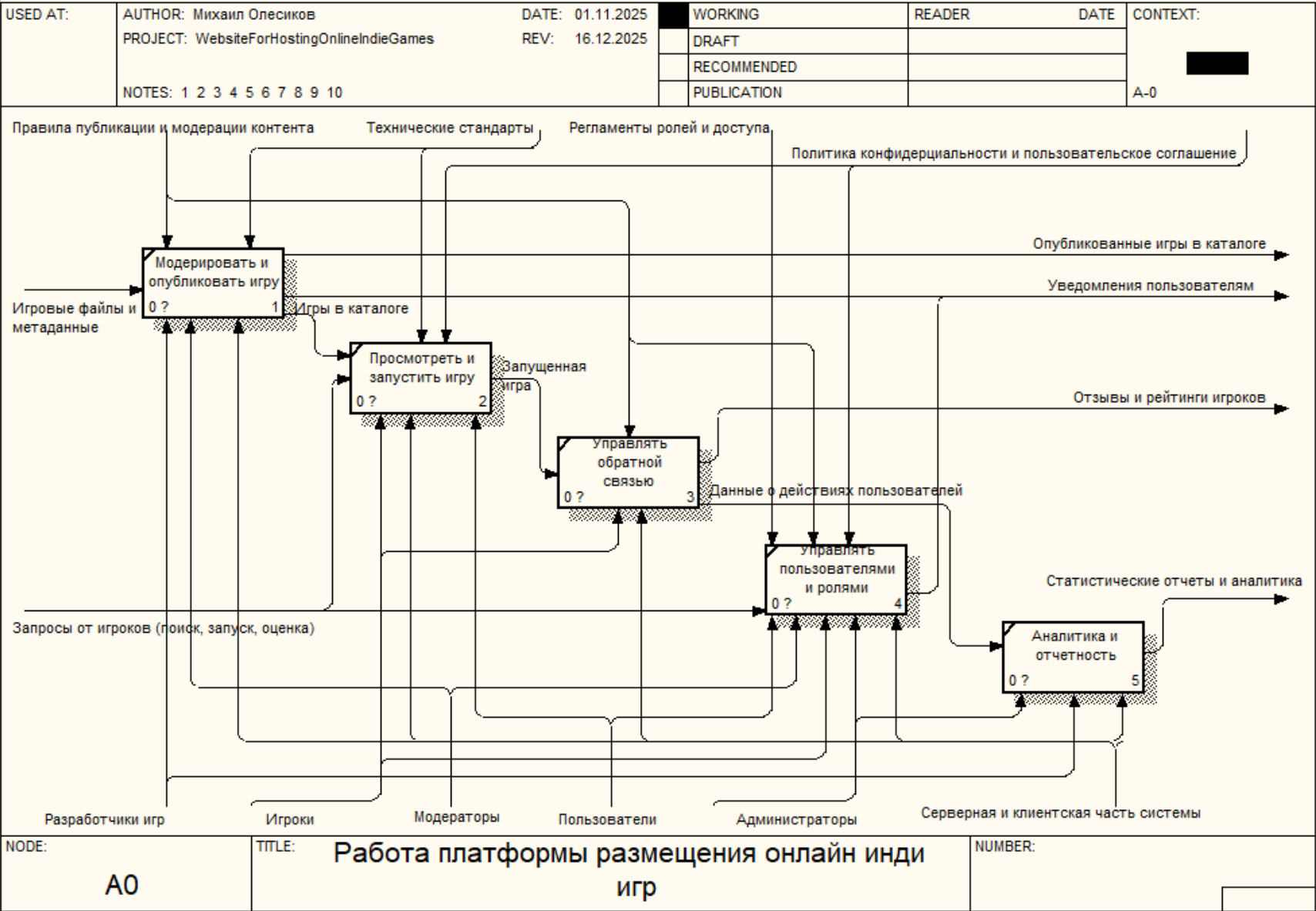


Рисунок 6. Диаграмма автоматизируемого бизнес-процесса IDEF0.

Функциональные требования к системе

**Требования к функциям для разработчиков**

* Система должна обеспечивать возможность загрузки и публикации новой игры, включая:
  + Файлы игры в виде адреса на файлы GitHub.
  + Текстовое описание игры, включающее название, аннотацию и геймплейные особенности.
  + Метаданные: жанр(ы), теги, информация о разработчике.
  + Скриншотов.
  + Видеотрейлеров (в виде ссылок или загруженных файлов).
  + Иконки игры.
* Система должна обеспечивать управление существующими играми разработчика, включая:
  + Просмотр опубликованных игр.
  + Редактирование описания и метаданных игры.
  + Удаление игры (со всеми связанными версиями и медиа).
  + Загрузку новых версий игры с указанием номера версии и списка изменений.
* Система должна обеспечивать предоставление разработчику статистики по его играм, включая:
  + Количество просмотров страницы игры.
  + Количество скачиваний/запусков.
  + Средний рейтинг и динамику его изменения.
  + Список и тексты отзывов.

**Требования к функциям для гостей**

* Система должна обеспечивать просмотр каталога всех опубликованных игр.
* Система должна обеспечивать поиск игр в каталоге по текстовому запросу (по названию и описанию).
* Система должна обеспечивать фильтрацию каталога игр по следующим критериям:
  + Жанр.
  + Тег.
  + Разработчик.
  + Средний рейтинг (выше указанного значения).
* Система должна обеспечивать сортировку списка игр в каталоге по:
  + Дате публикации (новые/старые).
  + Среднему рейтингу (высокий/низкий).
  + Популярности (количеству просмотров).
* Система должна обеспечивать запуск игр непосредственно в браузере пользователя.
* Система должна обеспечивать возможность просмотра отзывов об игре.

**Требования к функциям для пользователей**

* Система должна обеспечивать возможность получения уведомления пользователем.
* Система должна обеспечивать возможность установки пользователем целочисленной оценки игре в диапазоне от 1 до 5.
* Система должна обеспечивать возможность оставления текстового отзыва к игре.
* Система должна обеспечивать создание и управление персональными коллекциями игр ("Избранное", "Играть позже"), включая:
  + Создание новой коллекции с именем.
  + Добавление игры в коллекцию.
  + Удаление игры из коллекции.
  + Удаление коллекции.
* Система должна обеспечивать возможность подачи заявки пользователем на получение роли "Разработчик".
* Система должна обеспечивать возможность взаимодействия пользователя с профилем, включая:
  + Просмотр профиля.
  + Редактирование профиля (изменение имени пользователя, изменение аватара, изменение пароля).
  + Просмотр своих отзывов.
  + Просмотр предупреждений от модераторов.

**Требования к функциям для модераторов**

* Система должна обеспечивать управление учетными записями пользователей, включая:
  + Блокировку и разблокировку учетных записей.
  + Вынос предупреждения о некорректном поведении/контенте.
* Система должна обеспечивать модерацию контента, отправленного на публикацию, включая:
  + Просмотр сущностей, подлежащих модерации.
  + Одобрение сущности (перевод в статус "Одобрено").
  + Отклонение сущности с указанием причины (перевод в статус "Отклонено").

**Требования к функциям для администраторов**

* Система должна обеспечивать возможность контроля над модерацией, включая:
  + Просмотр всех пройденных модераций всех сущностей.
  + Внесение правок в модерацию (уточнение причины отказа, изменение статуса модерации).
* Система должна обеспечивать возможность составления статистического отчета по играм.
* Система должна обеспечивать возможность назначения любой роли для любого пользователя вручную.

**Общесистемные функциональные требования**

* Система должна обеспечивать регистрацию нового пользователя с указанием email, имени и пароля.
* Система должна обеспечивать аутентификацию пользователя по паре email/пароль.
* Система должна обеспечивать разграничение прав доступа к функциям системы на основе ролей (Гость, Пользователь, Разработчик, Модератор, Администратор).
* Система должна обеспечивать надежное хранение и целостность данных:
  + Учетных записей пользователей.
  + Данных об играх, их версиях и медиафайлах.
  + Пользовательских отзывов и оценок.
  + Данных о коллекциях игр.
  + Статистических данных.
* Система должна обеспечивать формирование и отображение на главной странице следующих блоков:
  + Список недавно добавленных игр.
  + Список популярных игр (на основе количества просмотров и оценок).

Нефункциональные требования к системе

**Требования к производительности (быстродействию)**

* Время отклика системы (Time to First Byte) для любой стандартной веб-страницы не должно превышать 1 секунды при пиковой нагрузке до 1000 одновременных пользователей.
* Время полной загрузки интерактивной страницы (например, каталога игр) не должно превышать 3 секунд при ширине канала не менее 10 Мбит/с.
* Система должна обеспечивать обработку не менее 50 транзакций в секунду для операций просмотра и поиска в каталоге игр.
* Время запуска онлайн-игры (от нажатия кнопки "Играть" до начала загрузки игры в браузере) не должно превышать 10 секунд.

**Требования к информационной безопасности**

* Все пароли пользователей должны храниться в базе данных исключительно в виде криптографических хешей с использованием стойкого алгоритма (bcrypt).
* Обмен данными между клиентом и сервером для всех страниц, требующих аутентификации, должен осуществляться по защищенному протоколу HTTPS (TLS 1.2+).
* Доступ к административным и модераторским функциям должен быть разрешен только пользователям с соответствующими ролями.

**Требования к пользовательскому интерфейсу (UX/UI)**

* Интерфейс системы должен быть адаптивным и корректно отображаться на устройствах с шириной экрана 1920px.
* Навигация по основным разделам (Главная, Каталог, Профиль) должна быть доступна не более чем за 3 клика с любой страницы системы.
* Скорость критического сценария (цель проекта): Среднее время успешного поиска конкретной игры через каталог с фильтрами не должно превышать 90 секунд для нового пользователя.

**Требования к надежности и доступности**

* Доступность (Uptime): Система должна иметь доступность на уровне 99.5% в расчете на календарный месяц (максимально допустимый простой ~ 3.5 часа/месяц).

**Требования к масштабируемости**

* Архитектура системы должна позволять горизонтальное масштабирование серверной части для поддержки увеличения количества пользователей до 10 000 без изменения кодовой базы.

**Требования к сопровождению и совместимости**

* Клиентская часть системы должна корректно работать в последних стабильных версиях браузеров Google Chrome, Mozilla Firefox, Microsoft Edge.

Диаграмма классов-сущностей

По концептуальной диаграмме классов была сделана вторая, уточняющая с полноценными java-классами. Она представлена на рисунке 7.

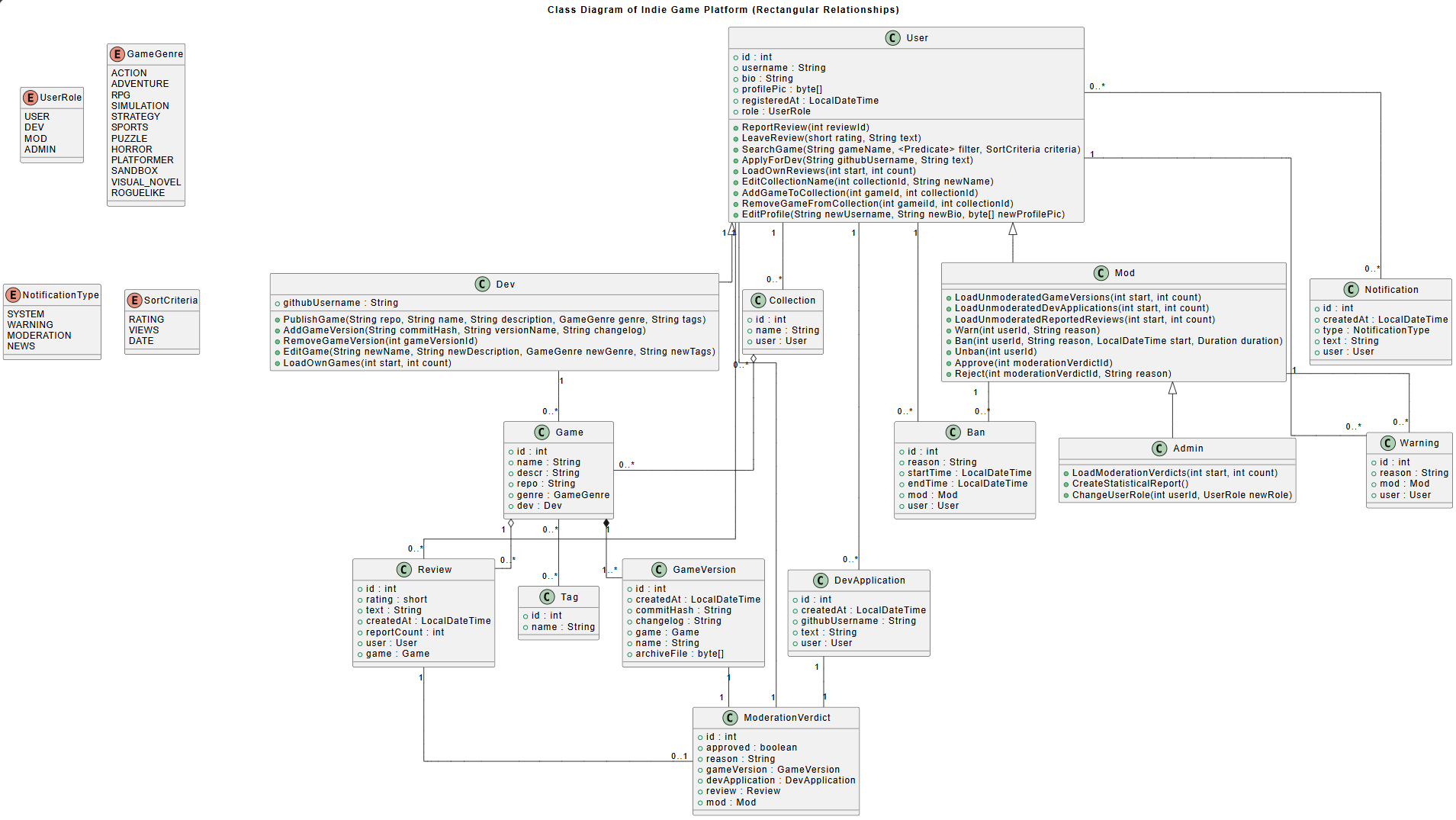


Рисунок 7. Диаграмма классов-сущностей (UML Class Diagram).

**Описание классов-сущностей.**

**User**

Назначение: базовая сущность - зарегистрированный пользователь платформы.

Поля:

* id — уникальный идентификатор.
* username — отображаемое имя пользователя.
* bio — краткая информация о пользователе.
* profilePic — аватар (байтовый массив).
* registeredAt — дата/время регистрации.
* role — роль пользователя (UserRole: USER/DEV/MOD/ADMIN).

Методы (назначение):

* ReportReview(int reviewId) — пометить/сообщить о проблемном отзыве.
* LeaveReview(short rating, String text) — создать отзыв для игры.
* SearchGame(String gameName, Predicate filter, SortCriteria criteria) - поиск игр.
* ApplyForDev(String githubUsername, String text) — подать заявку на роль разработчика.
* LoadOwnReviews(int start, int count) — загрузить список своих отзывов (пагинация).
* Коллекционные методы: EditCollectionName, AddGameToCollection, RemoveGameFromCollection.
* EditProfile(...) - редактирование профиля.

Связи:

* владеет 0..\* Collection;
* оставляет 0..\* Review;
* может иметь 0..\* Notification.

**Dev (наследует User)**

Назначение: пользователь с правом публикации игр (разработчик).

Поля:

* githubUsername — имя/ссылка GitHub (для подтверждения и метаданных).  
  Методы:
* PublishGame(repo, name, description, genre, tags) — создание новой игры/заявки на публикацию.
* AddGameVersion(commitHash, versionName, changelog) — добавить новую версию/релиз игры.
* RemoveGameVersion(gameVersionId) — удалить конкретную версию.
* EditGame(...) — редактировать метаданные игры.
* LoadOwnGames(start, count) — список собственных игр (пагинация)

Связи:

* создатель 1..\* Game (каждая игра имеет одного Dev).

**Mod (наследует User)**

Назначение: модератор — обеспечивает проверку и контроль содержимого.

Методы (назначение):

* загрузка сущностей в очередь модерации: LoadUnmoderatedGameVersions, LoadUnmoderatedDevApplications, LoadUnmoderatedReportedReviews.
* Warn(userId, reason) — выдать предупреждение.
* Ban(userId, reason, start, duration) / Unban(userId) — блокировка/разблокировка пользователей.
* Approve(moderationVerdictId) / Reject(moderationVerdictId, reason) — вынести вердикт по объекту модерации.

Связи:

* создает 0..\* Warning, 0..\* Ban; участвует в ModerationVerdict (каждый вердикт связан с модератором).

**Admin (наследует Mod)**

Назначение: администратор системы, расширенные права управления.

Методы:

* LoadModerationVerdicts(start, count) — управление историей модераций.
* CreateStatisticalReport() — генерация отчётов.
* ChangeUserRole(userId, UserRole newRole) — назначение/снятие ролей.  
  Связи:
* может назначать/отзывать роли модераторов и управлять системой.

**Collection**

Назначение: пользовательская подборка игр (избранное, пройти позже и т.п.).

Поля: id, name, user (владелец).

Методы/поведение: создание/удаление коллекции, модификация списка игр (через методы User/Frontend).

Связи:

* принадлежит одному User (1..1);
* содержит многие игры (many-to-many — через таблицу связи).

**Notification**

Назначение: уведомление пользователя (системное, модераторское и т.п.).

Поля: id, createdAt, type (NotificationType), text, user (получатель).

Поведение: хранится в списке уведомлений пользователя; помечается как прочитанное/непрочитанное.

Связи: принадлежит одному User (many-to-one).

**Game**

Назначение: основная сущность — игровая публикация/проект.

Поля: id, name, descr, repo (ссылка/репозиторий или путь к сборке), genre (GameGenre), dev (владелец-Dev).

Поведение/методы: управление метаданными, получение списка версий, получение рейтинга/статистики.

Связи:

* имеет одного Dev (many-to-one, каждый game → 1 dev);
* содержит 1..\* GameVersion;
* имеет 0..\* Review;
* связано с 0..\* Tag (many-to-many или composition в модели).

**GameVersion**

Назначение: конкретный релиз/версия игры.

Поля: id, createdAt, commitHash, changelog, name (номер/название версии), archiveFile (контент/архив), game (ссылка на Game).

Поведение: хранение бинарных данных релиза, возможность отката/истории версий.

Связи: принадлежит одному Game (many-to-one). Может быть предметом модерации (ModerationVerdict).

**Review**

Назначение: отзыв игрока о конкретной игре.

Поля: id, rating (short 1..5), text, createdAt, reportCount (число флагов), user (автор), game (объект отзыва).

Поведение: публикация, получение флагов/жалоб, модерация (удаление/редактирование).

Связи: many-to-one к User и many-to-one к Game. Может быть связан с ModerationVerdict.

**ModerationVerdict**

Назначение: результат модерации (единое место для вердиктов по разным типам сущностей).

Поля: id, approved (boolean), reason (string), ссылки на объект модерации: gameVersion, devApplication, review, также ссылку на модератора mod.

Поведение: хранит решение модератора по одной из сущностей; создаётся модератором (или админом).

Связи: опционально с GameVersion / DevApplication / Review; связан с Mod как исполнитель вердикта.

**DevApplication**

Назначение: заявка пользователя на получение роли разработчика.

Поля: id, createdAt, githubUsername, text, user (кто подал).

Поведение: попадает в очередь модерации; модератор одобряет/отказывает (через ModerationVerdict).

Связи: many-to-one к User. Может иметь ModerationVerdict.

**Tag**

Назначение: ключевое слово/метка для игр.

Поля: id, name.

Поведение: используется для поиска/фильтрации каталога.

Связи: многие-ко-многим/композиция с Game (игра содержит набор тегов).

**Ban**

Назначение: запись о блокировке пользователя.

Поля: id, reason, startTime, endTime, mod (модератор, наложивший бан), user (заблокированный).

Поведение: при активном бане пользователь лишается ряда прав.

Связи: many-to-one с Mod и many-to-one с User.

**Warning**

Назначение: предупреждение, вынесенное модератором пользователю.

Поля: id, reason, mod, user.

Поведение: накапливаемые предупреждения могут приводить к бану; отображаются в профиле пользователя.

Связи: many-to-one к Mod и User.

**Справочные перечисления (enum)**

* UserRole — роли: USER, DEV, MOD, ADMIN. (Используется в User.role для RBAC.)
* GameGenre — предопределённый список жанров (ACTION, ADVENTURE, RPG, …).
* NotificationType — типы уведомлений: SYSTEM, WARNING, MODERATION, NEWS.
* SortCriteria — критерии сортировки каталога: RATING, VIEWS, DATE.

Схема базы данных

Была составлена физическая ER-диаграмма БД для системы. Она представлена на рисунке 8. Скрипт создания БД представлен в файле db\_create\_script\_v1.2.2.sql.

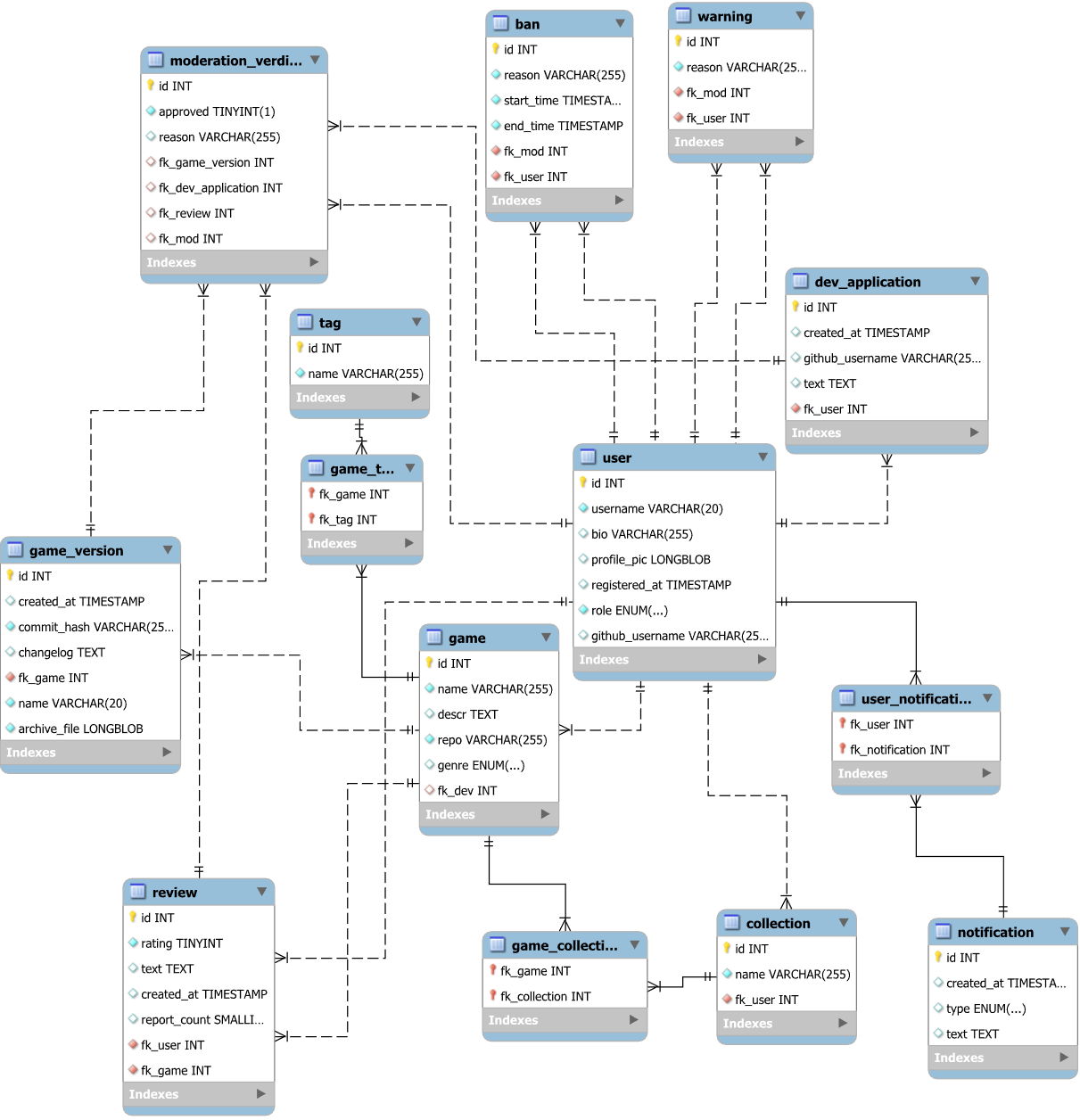


Рисунок 8. Схема БД (ER Diagram).

**Описание реляционных таблиц**

**user**

Назначение: хранит данные о пользователях системы.

Поля:

* id — первичный ключ.
* username — уникальное имя пользователя.
* bio — краткая биография.
* profile\_pic — фото профиля (BLOB).
* registered\_at — дата регистрации (по умолчанию текущее время).
* role — роль пользователя: user, dev, mod, admin.
* github\_username — уникальный GitHub-логин (для разработчиков).

Связи:

* 1 ко многим с ban, warning, collection, review, dev\_application, game (в виде внешних ключей).
* многие ко многим с notification (через user\_notification).

**ban**

Назначение: хранит информацию о банах пользователей.

Поля:

* id — первичный ключ.
* reason — причина блокировки.
* start\_time, end\_time — время начала и окончания.
* fk\_mod — модератор, выдавший бан.
* fk\_user — заблокированный пользователь.

Связи:

* fk\_mod → user.id (модератор, удаляется каскадно).
* fk\_user → user.id (забаненный пользователь, удаляется каскадно).

**warning**

Назначение: предупреждения от модераторов пользователям.

Поля:

* id, reason, fk\_mod, fk\_user.

Связи:

* fk\_mod → user.id
* fk\_user → user.id  
   (Обе связи каскадные при удалении.)

**collection**

Назначение: пользовательские коллекции игр.

Поля:

* id — первичный ключ.
* name — название коллекции (уникально в пределах пользователя).
* fk\_user — владелец коллекции.

Связи:

* fk\_user → user.id
* многие ко многим с game через game\_collection.

**game**

Назначение: хранит информацию об играх.

Поля:

* id, name, descr, repo, genre, fk\_dev.

Описание:

* repo — уникальная ссылка/репозиторий игры.
* genre — жанр (action, rpg, puzzle и др.).
* fk\_dev — разработчик (пользователь, может быть NULL).

Связи:

* fk\_dev → user.id (при удалении — SET NULL).
* многие ко многим с collection через game\_collection.
* многие ко многим с tag через game\_tag.
* 1 ко многим с game\_version и review.

**game\_version**

Назначение: хранит версии игр.

Поля:

* id, created\_at, commit\_hash, changelog, fk\_game, name, archive\_file.

Особенности:

* Комбинация (fk\_game, commit\_hash) уникальна.
* archive\_file — бинарный файл (архив с игрой).

Связи:

* fk\_game → game.id (удаляется каскадно).
* 1 к 1 с moderation\_verdict (через fk\_game\_version).

**tag**

Назначение: хранит уникальные теги игр.

Поля:

* id, name (уникальное).

Связи:

* многие ко многим с game через game\_tag.

**game\_tag**

Назначение: связывает игры и теги (таблица связей many-to-many).

Поля:

* fk\_game, fk\_tag (составной PK).

Связи:

* fk\_game → game.id
* fk\_tag → tag.id  
   (Обе каскадно удаляются.)

**game\_collection**

Назначение: связывает игры с пользовательскими коллекциями.

Поля:

* fk\_game, fk\_collection (составной PK).

Связи:

* fk\_game → game.id
* fk\_collection → collection.id  
   (Обе каскадно удаляются.)

**dev\_application**

Назначение: заявки пользователей на статус разработчика.

Поля:

* id, created\_at, github\_username, text, fk\_user.
* fk\_user уникален — один пользователь может подать одну заявку.

Связи:

* fk\_user → user.id
* 1 к 1 с moderation\_verdict через fk\_dev\_application.

**review**

Назначение: отзывы пользователей об играх.

Поля:

* id, rating, text, created\_at, report\_count, fk\_user, fk\_game.
* Один пользователь может оставить только один отзыв на игру (уникальность по fk\_user, fk\_game).

Связи:

* fk\_user → user.id
* fk\_game → game.id
* 1 к 1 с moderation\_verdict через fk\_review.

**moderation\_verdict**

Назначение: хранит решения модерации (для ревью, версий игр, заявок разработчиков).

Поля:

* id, approved (bool), reason, fk\_game\_version, fk\_dev\_application, fk\_review.

Связи:

* fk\_game\_version → game\_version.id (1 к 1, каскадное удаление).
* fk\_dev\_application → dev\_application.id (1 к 1).
* fk\_review → review.id (1 к 1, каскадное удаление).

**13. notification**

Назначение: уведомления (системные, новостные, модераторские и т. д.).

Поля:

* id, created\_at, type (enum), text.

**Связи:**

* многие ко многим с user через user\_notification.

**user\_notification**

Назначение: связывает пользователей и уведомления (таблица many-to-many).

Поля:

* fk\_user, fk\_notification (составной PK).

Связи:

* fk\_user → user.id
* fk\_notification → notification.id  
   (Каскадное удаление при удалении пользователя или уведомления.)
* **Описание механизмов обеспечения целостности на уровне БД**
* В базе данных **myngdb** механизмы целостности обеспечивают согласованность, непротиворечивость и корректность хранимых данных за счёт использования первичных и внешних ключей, каскадных ограничений, уникальности, ограничений типов данных и проверочных условий (CHECK).
* Каждая таблица имеет первичный ключ, который гарантирует уникальность строк и однозначную идентификацию записей. Например, таблица user использует поле id как основной идентификатор пользователя, а остальные таблицы, такие как ban, warning, collection, review и другие, ссылаются на него через внешние ключи. Это обеспечивает ссылочную целостность: ни одна зависимая запись не может существовать без родительской. При удалении пользователя включены каскадные удаления (ON DELETE CASCADE), что предотвращает появление «висячих» ссылок и автоматически удаляет связанные данные.
* В таблицах, где зависимые записи теряют смысл без родительских, также используется каскадное удаление. Например, при удалении игры удаляются все версии этой игры, её теги, отзывы и записи о принадлежности к коллекциям. В других случаях применяется правило ON DELETE SET NULL, чтобы сохранить зависимую запись, но обнулить связь. Так реализована связь между играми и их разработчиками: если разработчик удалён, игра остаётся, но ссылка на него становится NULL.
* Для обеспечения логической целостности данных используются уникальные ограничения (UNIQUE INDEX). Так, имя пользователя (username) и GitHub-логин (github\_username) уникальны, что исключает дублирование учётных записей. Аналогично, у каждой игры уникальный репозиторий (repo), а пользователь может иметь только одну заявку на получение статуса разработчика — это гарантируется уникальным индексом на поле fk\_user в таблице dev\_application. В таблице review уникальность пары (fk\_user, fk\_game) предотвращает повторное добавление отзыва пользователем на одну и ту же игру.
* Дополнительную роль в обеспечении целостности играют ограничения по типам и допустимым значениям. Поля ENUM ограничивают возможные варианты ролей пользователей и жанров игр заранее заданным списком, что исключает случайные или некорректные значения. Поля с модификатором UNSIGNED (например, рейтинг в отзывах или счётчик жалоб) предотвращают появление отрицательных числовых значений.
* Особое внимание уделено таблице moderation\_verdict, в которой используется ограничение  
   CONSTRAINT chk\_moderation\_one\_target CHECK (((((fk\_game\_version IS NOT NULL) + (fk\_dev\_application IS NOT NULL)) + (fk\_review IS NOT NULL)) = 1)).  
   Оно гарантирует, что каждая запись решения модерации относится **ровно к одному объекту** — либо к версии игры, либо к заявке разработчика, либо к отзыву. Это предотвращает логические ошибки, при которых одно решение могло бы быть одновременно связано с несколькими объектами, или вовсе не имело бы цели модерации.
* Временные поля со значением по умолчанию CURRENT\_TIMESTAMP обеспечивают автоматическую фиксацию времени создания записей, что повышает достоверность временных данных и позволяет корректно отслеживать последовательность событий.
* Система целостности в базе данных myngdb реализована на нескольких уровнях — ссылочном, логическом, типовом и проверочном. Все эти механизмы совместно гарантируют, что данные остаются непротиворечивыми, а структура базы данных надёжно защищена от случайных или логических ошибок при работе с ней.

Макет программной системы

**Описание принципов организации интерфейса**

Интерфейс программной системы спроектирован по принципам логической иерархии, интуитивной навигации и минимизации переходов между страницами. Основная цель - обеспечить пользователю (игроку, разработчику, модератору, администратору) быстрый доступ к функциям своей роли, не перегружая интерфейс избыточной информацией.

Структурная организация интерфейса.

Интерфейс построен по многоуровневой схеме, где основные страницы отражают ключевые сущности системы (каталог, игра, профиль и т.д.). Оконные формы (модальные окна) используются для выполнения локальных действий - редактирования, добавления, подтверждения, - что позволяет избежать полной перезагрузки страницы и сохраняет контекст работы пользователя. Все страницы связаны между собой через единое навигационное меню, обеспечивая непрерывность взаимодействия.

Принципы визуальной и функциональной организации

Главная страница - точка входа, содержит краткие подборки («Популярное», «Новинки») и ссылки на ключевые разделы. Страница каталога организована по принципу карточек игр с возможностью фильтрации и сортировки по тегам, жанрам и рейтингу. Страница игры объединяет информацию об игре, версии, отзывы и форму запуска. Все действия (написать отзыв, добавить версию, редактировать) выполняются в оконных формах, не покидая страницу. Страница профиля служит центром персональных настроек: управление коллекциями, профилем, уведомлениями и публикациями. Каждое действие реализовано отдельным модальным окном.

Страницы модерации и администрирования (просмотр сущностей, список пользователей) реализуют таблицы с фильтрами и действиями, доступными по ролям. Все изменения ролей, блокировки и предупреждения происходят в модальных окнах, чтобы избежать случайных действий.

Универсальные принципы интерфейса

* Единый визуальный стиль: нейтральная цветовая палитра, читаемые шрифты, минимализм.
* Интерактивность: подтверждение действий, всплывающие уведомления, визуальные индикаторы состояния.
* Адаптивность: корректное отображение на экранах.
* Доступность: акценты на основных элементах, контраст, простая структура без перегрузки.
* Ролевая персонализация: отображаются только те элементы, которые соответствуют правам пользователя.

Навигационные принципы

Навигация организована по схеме «главное меню + контекстные переходы». Каждый пользователь всегда имеет доступ к навигации по разделам: «Каталог», «Профиль», «Главная». Для модераторов и администраторов добавлены отдельные пункты меню («Модерация», «Пользователи»). Внутри страниц предусмотрены вспомогательные переходы - например, из профиля к списку своих игр.

**Структура интерфейса, навигация**

**Перечень экранных форм**

В процессе моделирования экранных форм системы было выделено несколько страниц, а также оконных форм (оконная форма – это часть страницы, в которую пользователь переходит, не покидая исходную).

Страницы: Главная страница; Страница регистрации/входа; Страница каталога; Страница игры; Страница профиля; Страница просмотра модераций; Страница сущностей, подлежащих модерации; Страница списка пользователей; Страница запущенной игры.

Оконные формы.

Страница игры: Оконная форма написания отзыва; Оконная форма добавления версии; Оконная форма удаления версии; Оконная форма редактирования игры.

Страница профиля: Оконная форма выхода из аккаунта; Оконная форма редактирования профиля; Оконная форма просмотра своих отзывов; Оконная форма подачи заявки на разработчика; Оконная форма просмотра уведомлений; Оконная форма удаления коллекций; Оконная форма создания коллекций; Оконная форма редактирования коллекций; Оконная форма публикации игры; Оконная форма просмотра опубликованных игр.

Страница списка пользователей: Оконная форма изменения роли пользователя; Оконная форма выдачи предупреждения; Оконная форма разблокировки пользователя; Оконная форма блокировки пользователя.

Страница сущностей, подлежащих модерации: Оконная форма отклонения сущности.

**Диаграмма навигации**

Была составлена диаграмма навигации. Она представлена на рисунке 9.

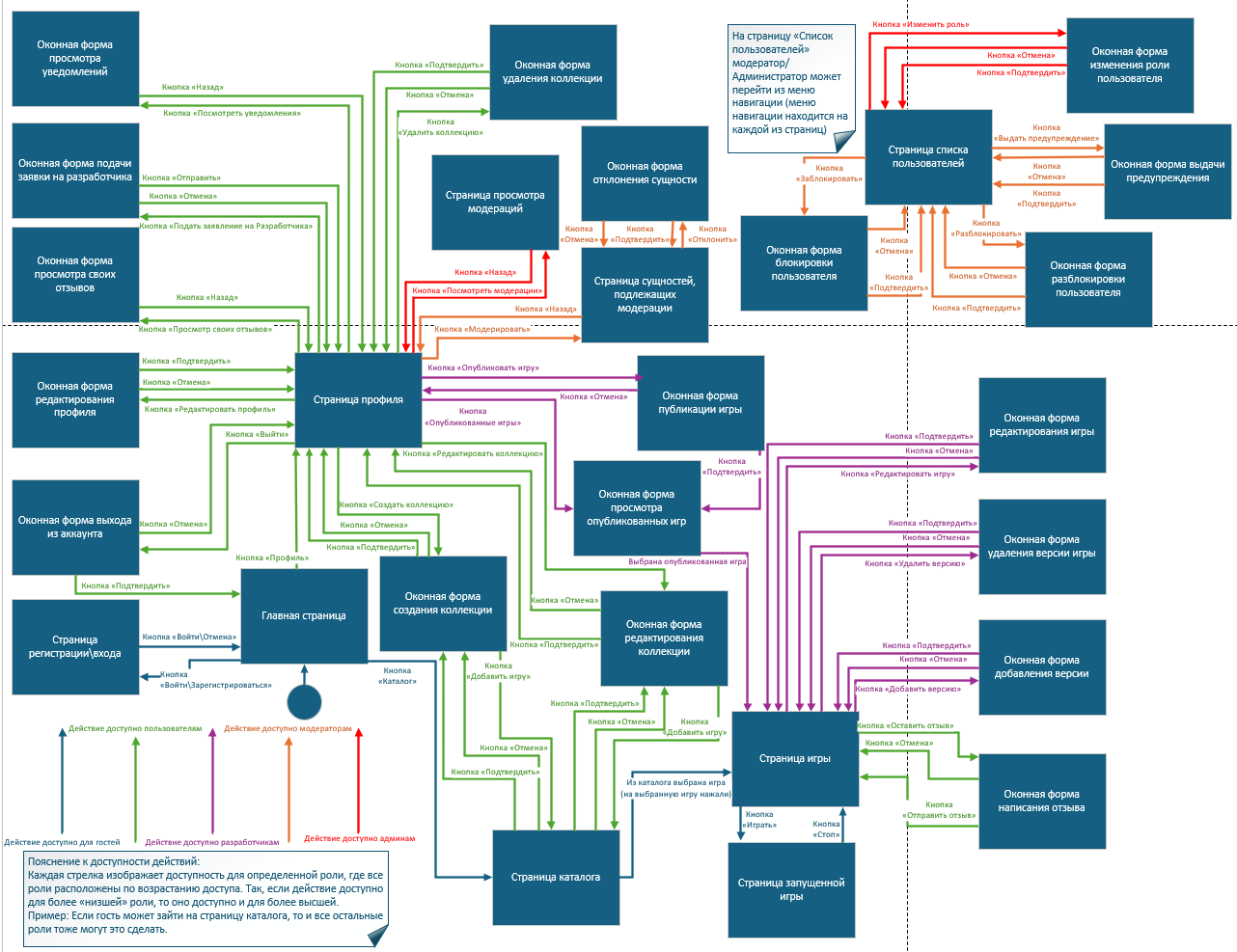


Рисунок 9. Диаграмма навигации.

**Эскизы интерфейса пользователя**

Были разработаны макеты страниц. Данные макеты мы сразу же разметили как html страницы, они представлены на рисунках 10-18 ниже. Для оконных форм также предусмотрены макеты. На данном этапе все оконные формы – это формы по заполнению каких-либо данных. Все сверстанные формы представлены на рисунках 19-37. Рисунки 19-22 – это оконные формы для страницы игры. Рисунки 23-32 – это оконные формы для страницы профиля. Рисунки 33-36 – это оконные формы для страницы списка пользователей. И рисунок 37 – оконная форма для страницы сущностей подлежащих модерации.

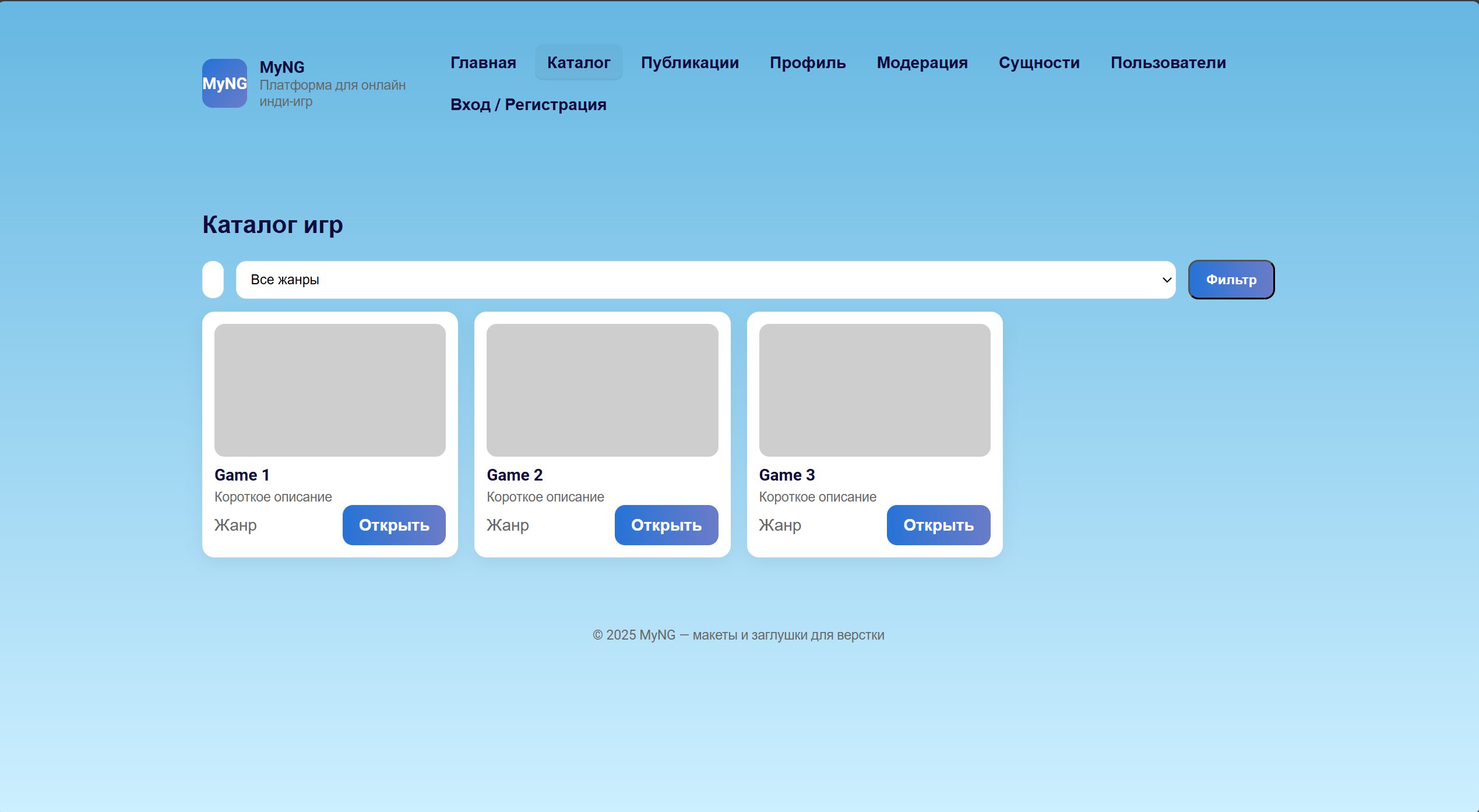


Рисунок 10. Макет главной страницы.

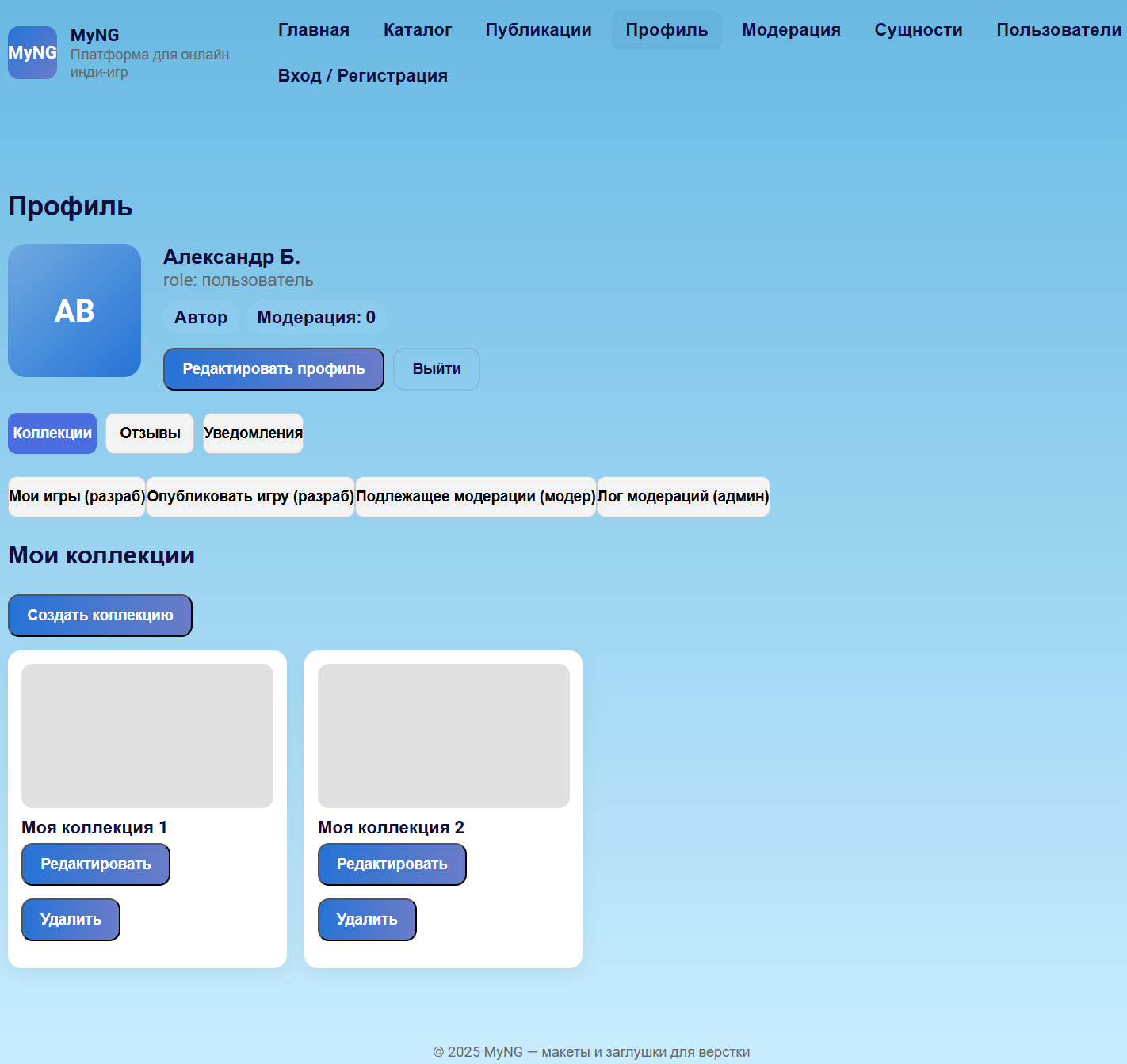


Рисунок 11. Макет страницы профиля.

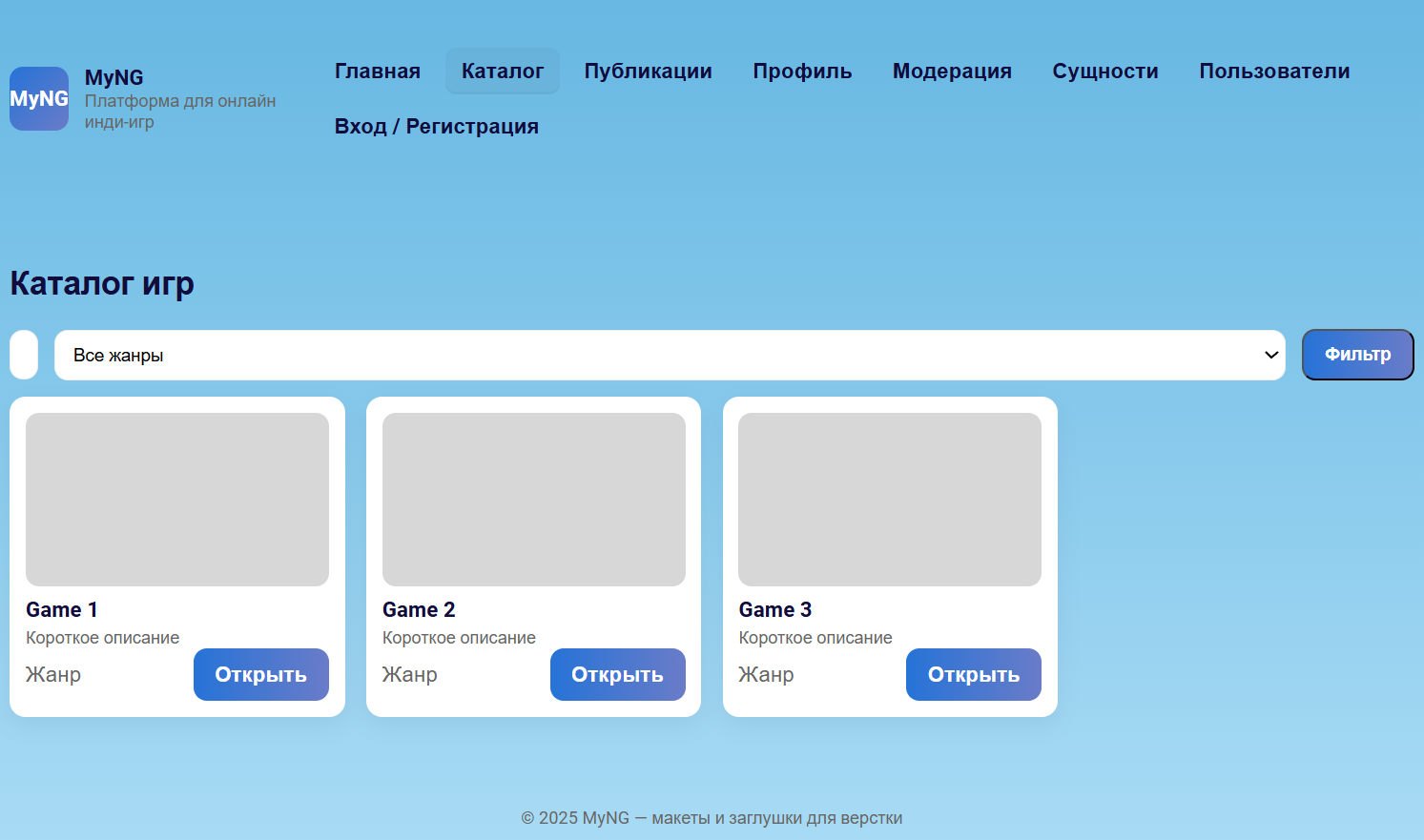


Рисунок 12. Макет страницы каталога.

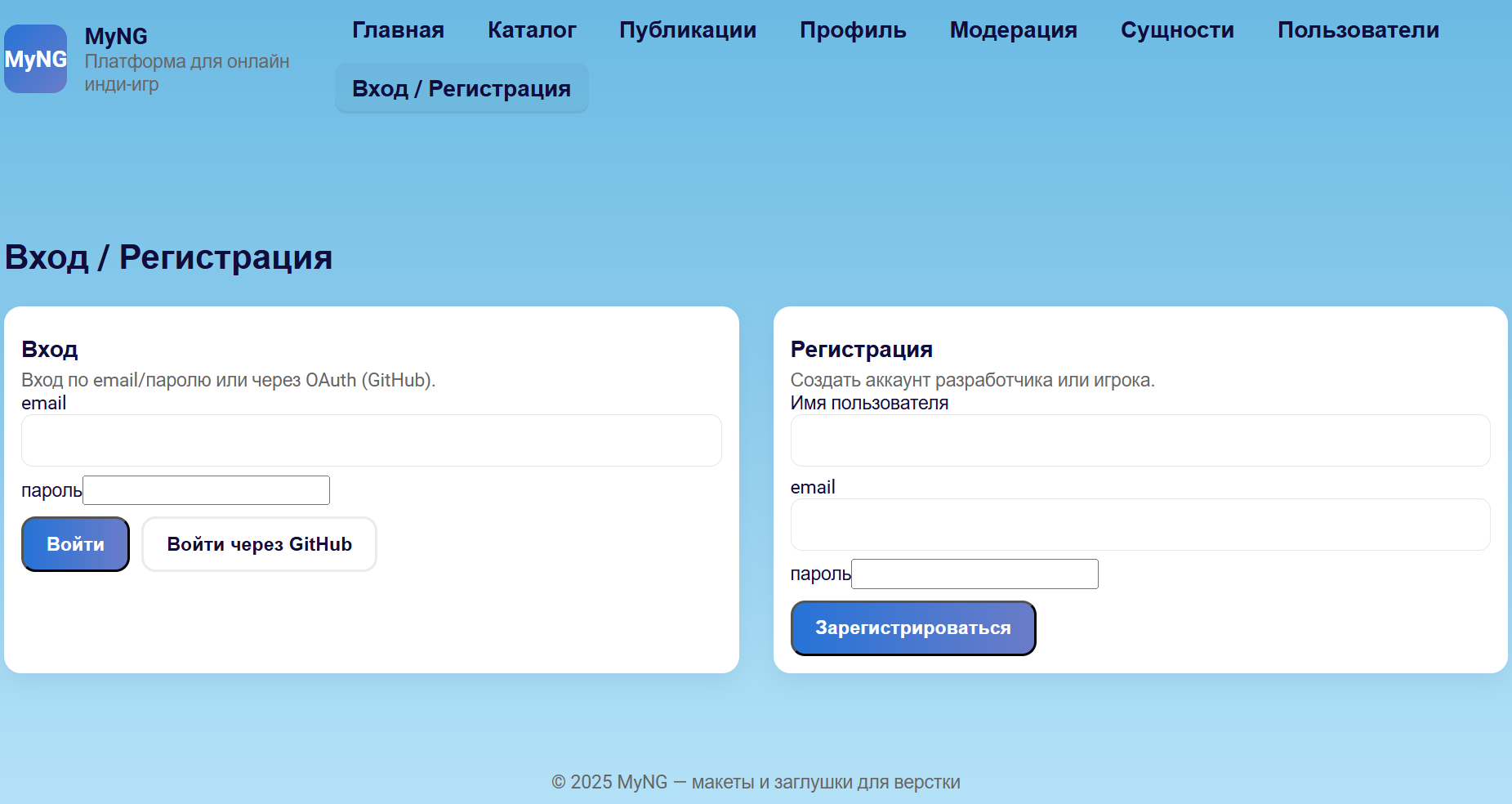


Рисунок 13. Макет страницы входа/регистрации.

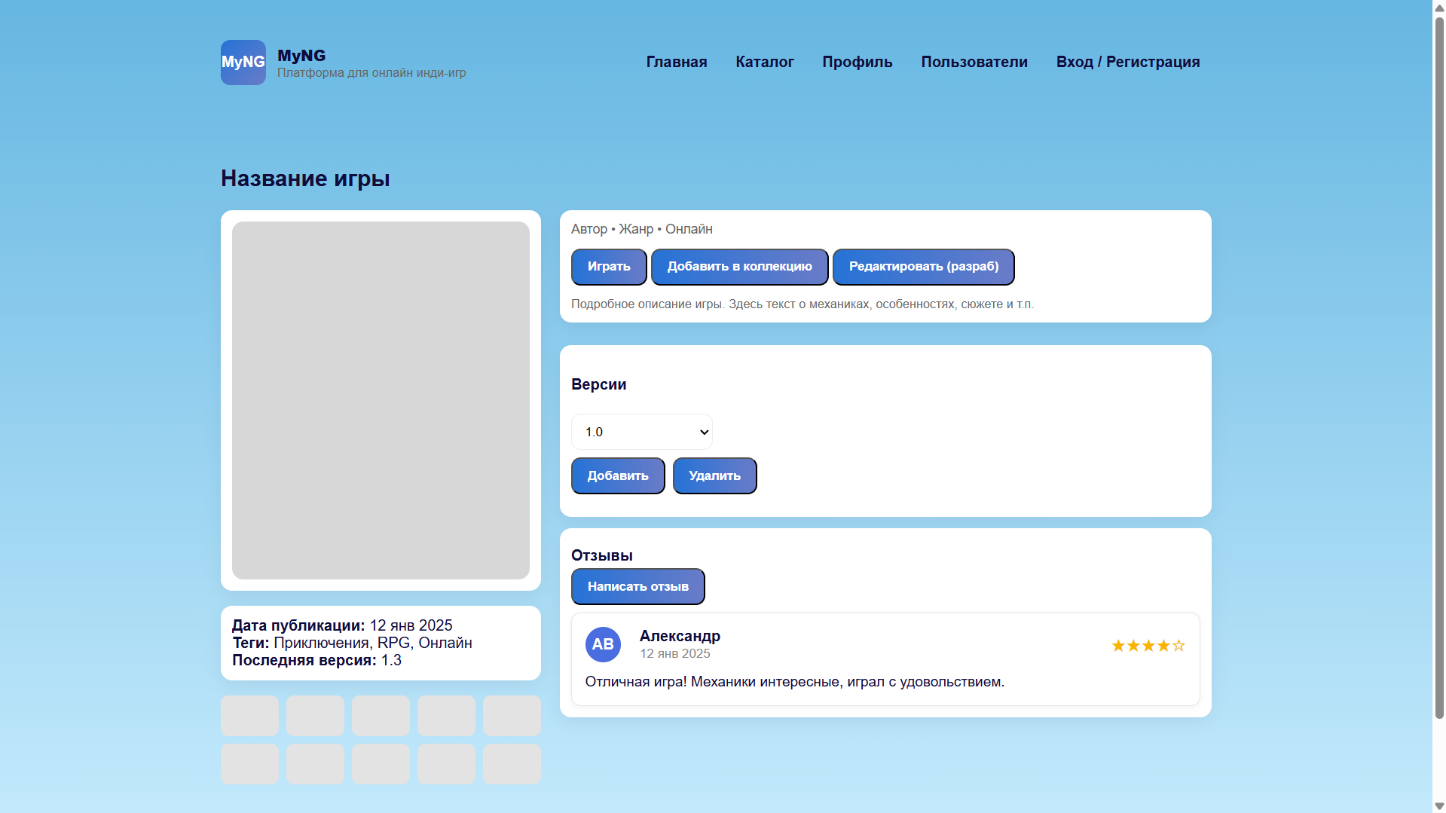


Рисунок 14. Макет страницы игры.

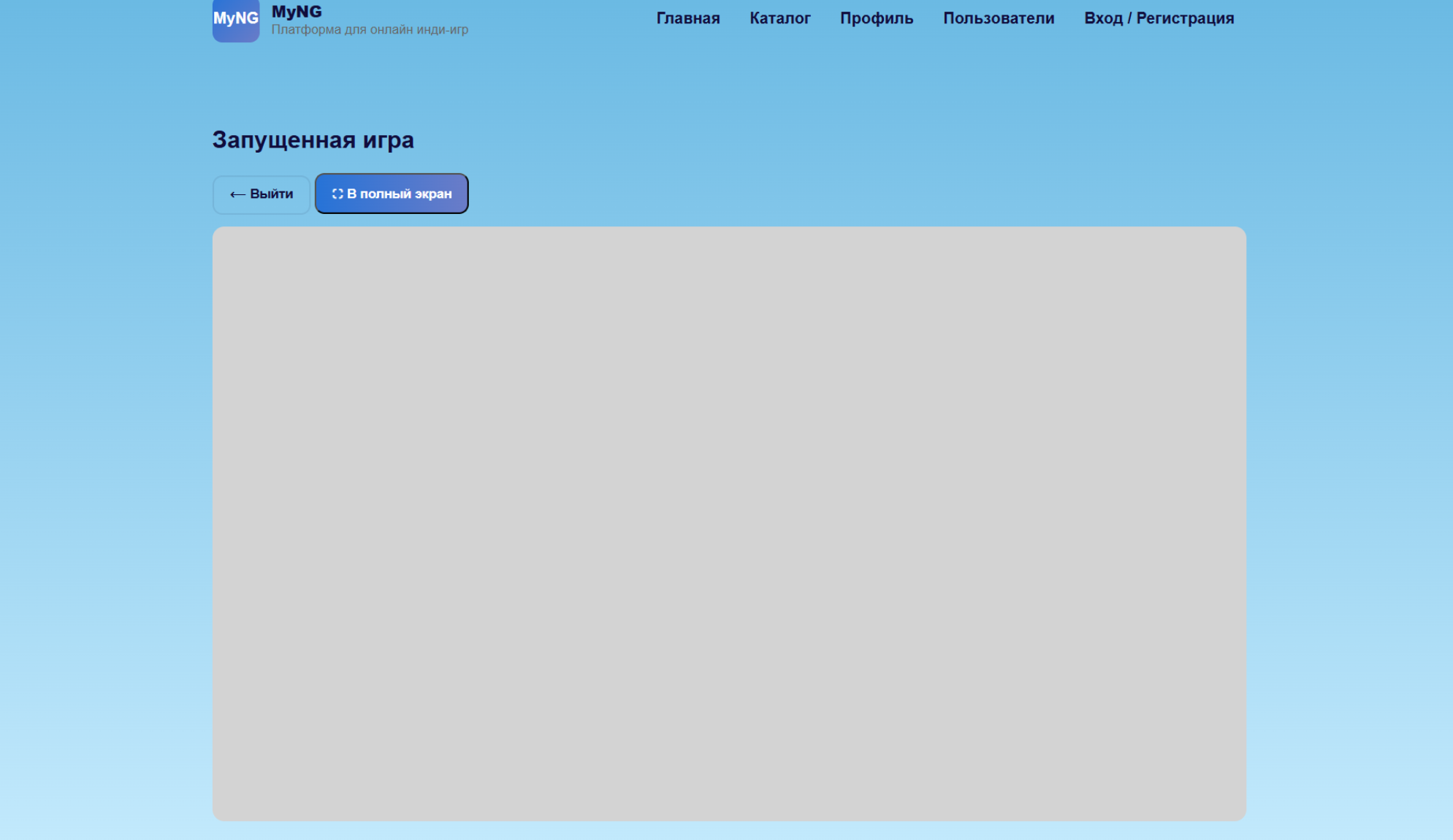


Рисунок 15. Макет страницы запущенной игры.

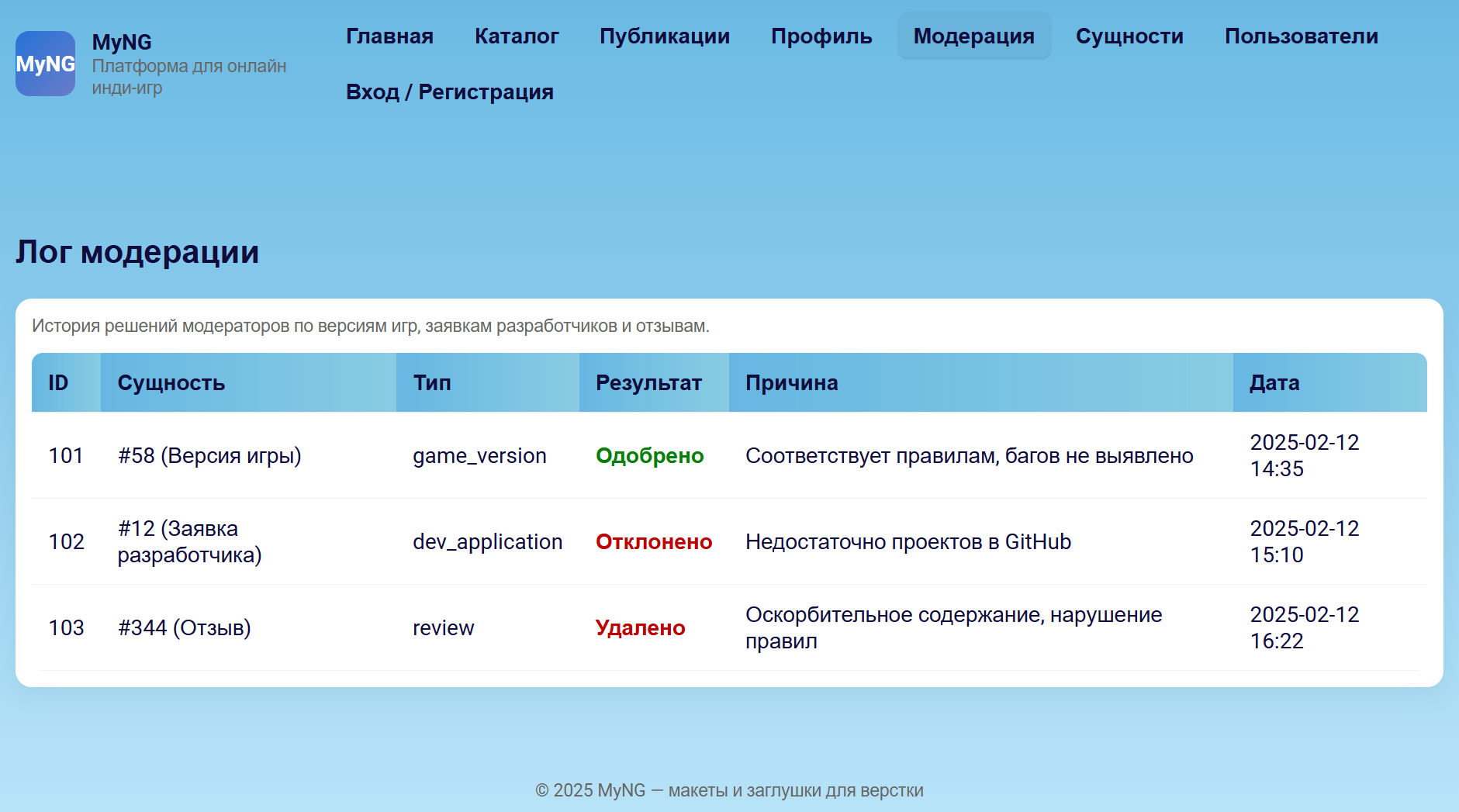


Рисунок 16. Макет страницы просмотра модераций (администратор).

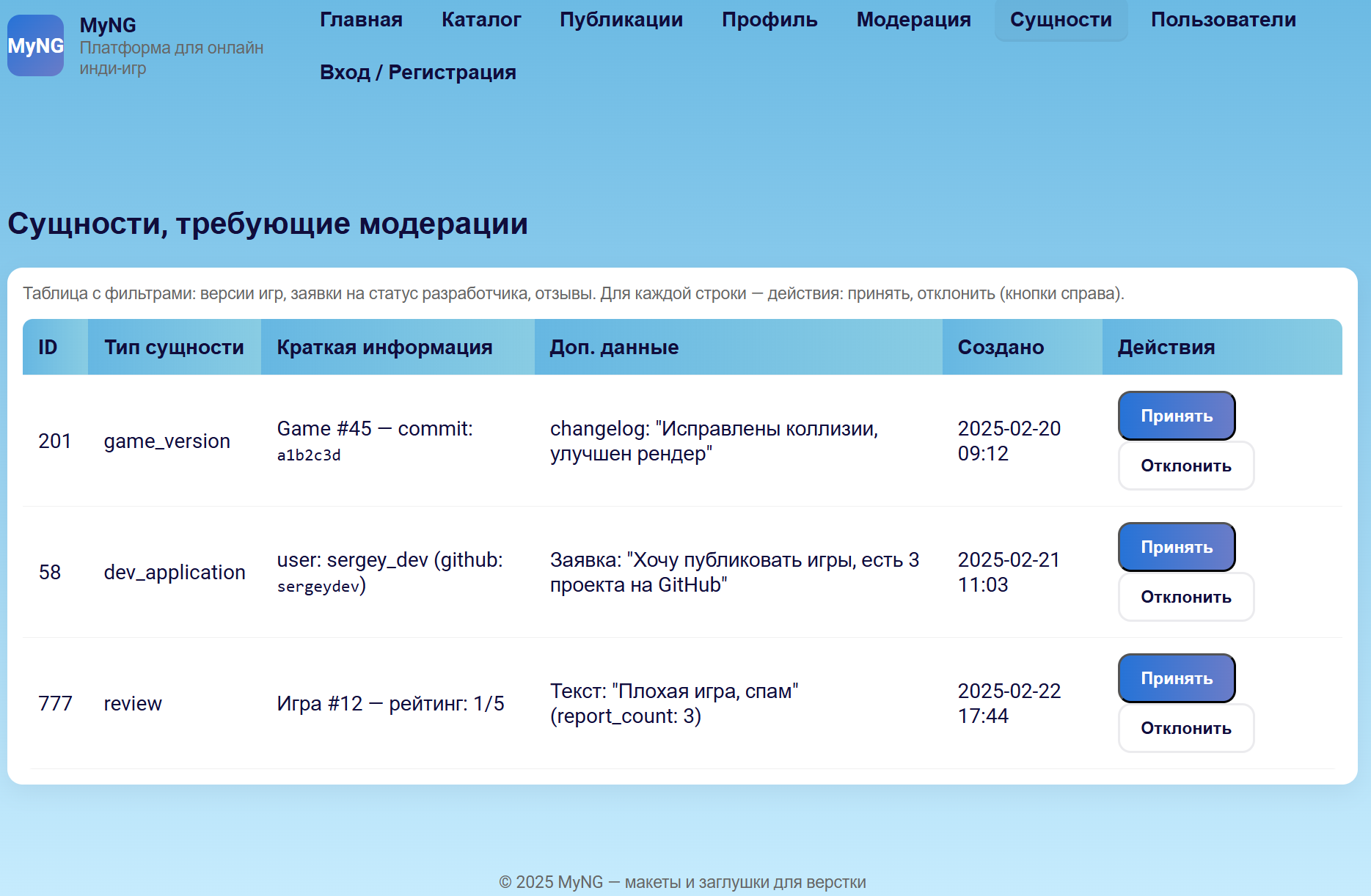


Рисунок 17. Макет страницы сущностей подлежащих модерации (модератор).

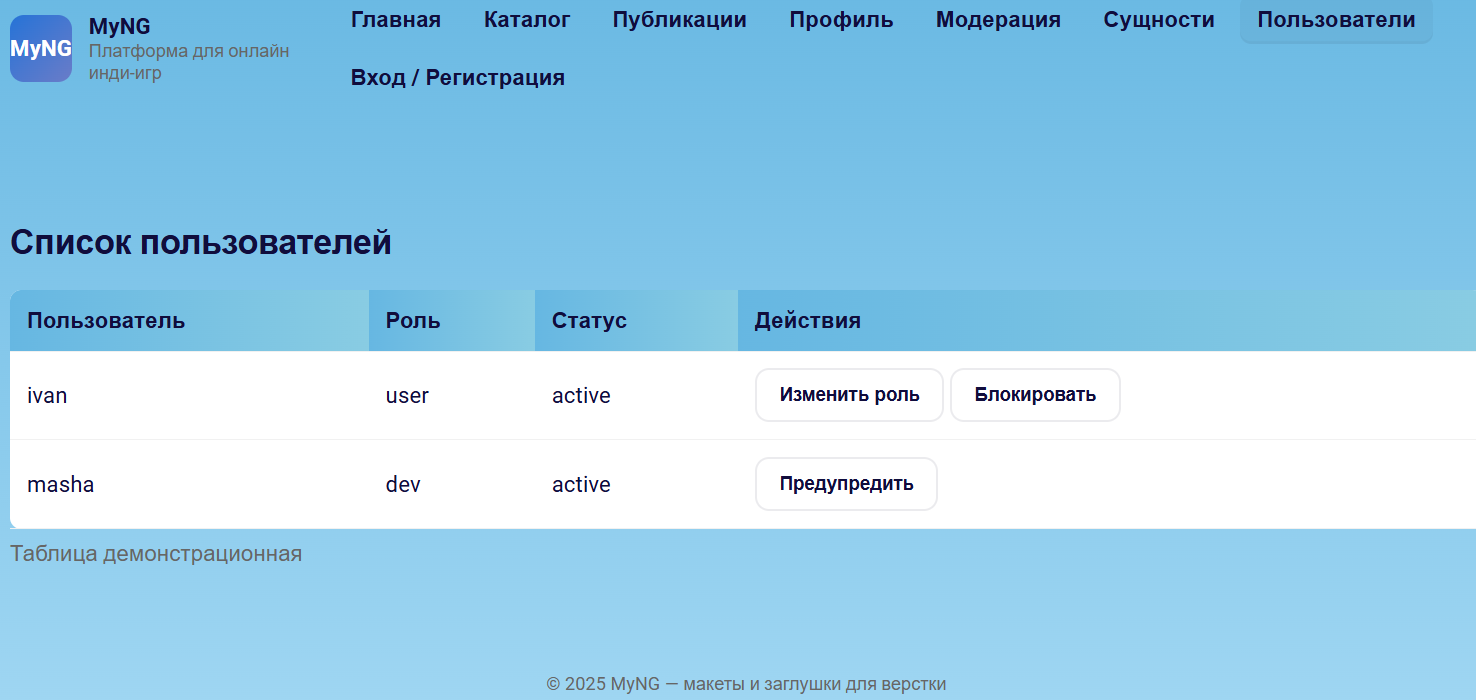


Рисунок 18. Макет «Страница списка пользователей».

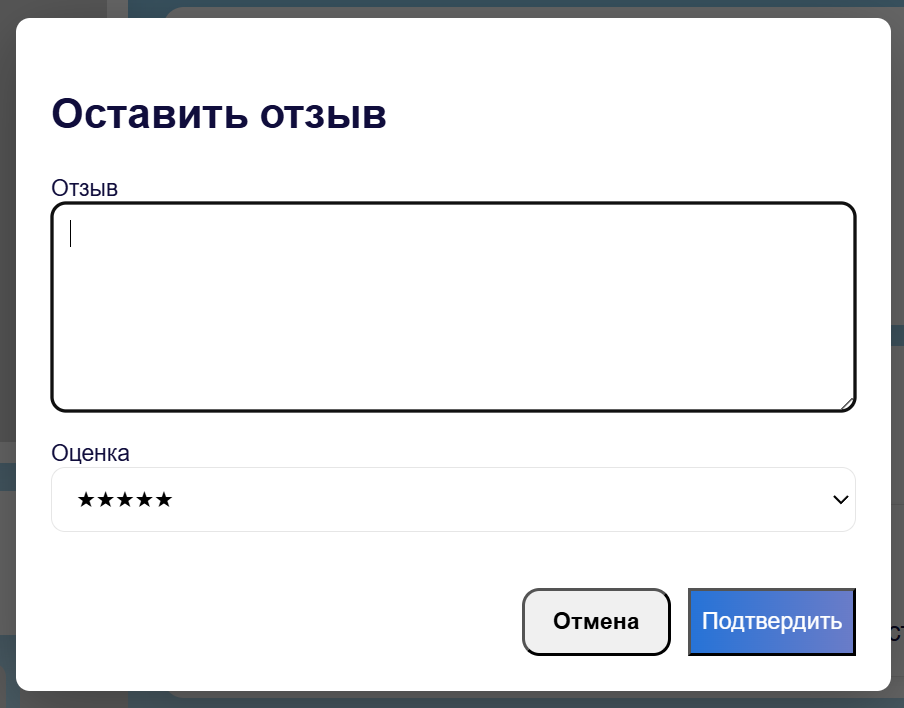
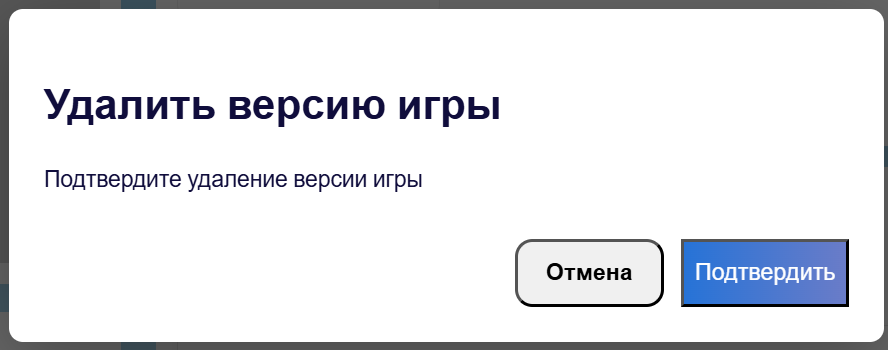


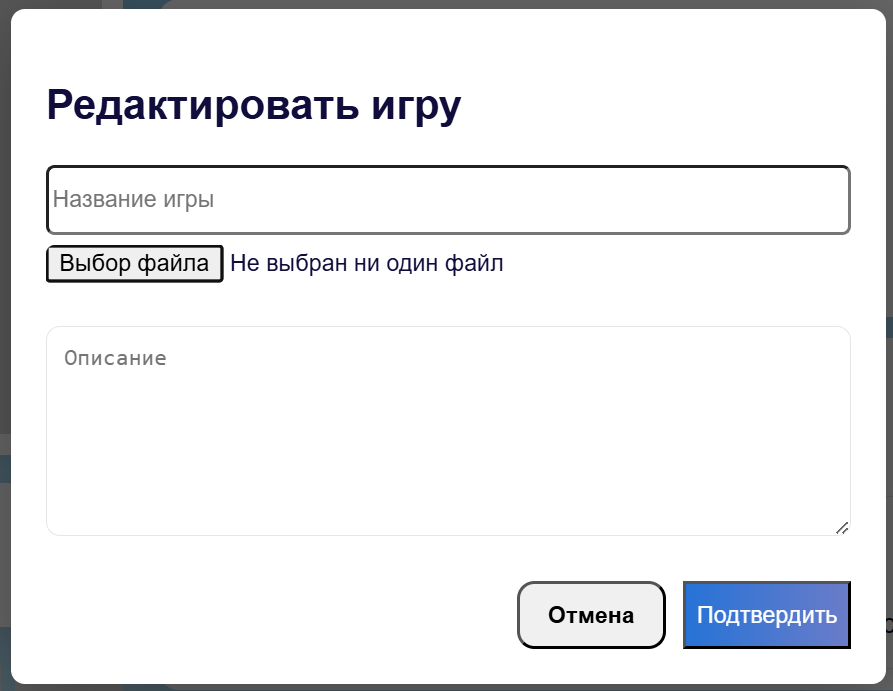
Рисунок 19. Макет «Оконная форма написания отзыва».



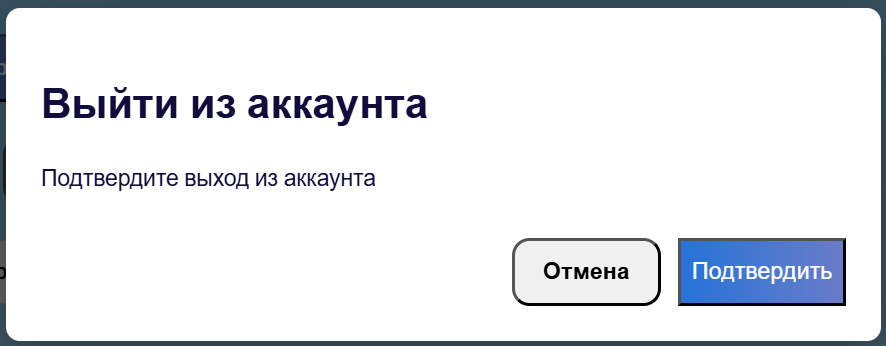
Рисунки 20. Макет «Оконная форма добавления версии».



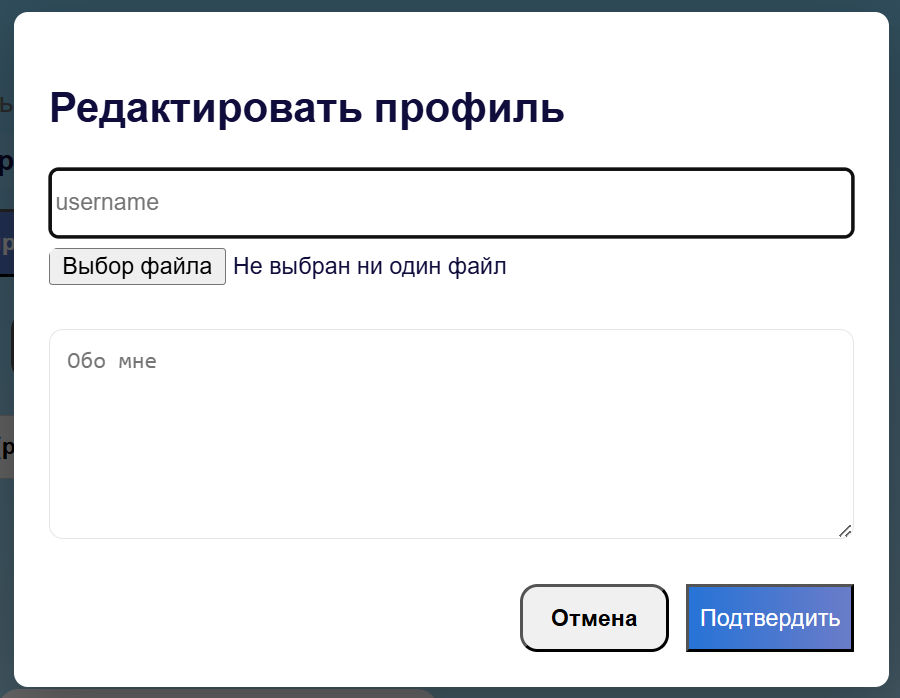
Рисунки 21. Макет «Оконная форма удаления версии».



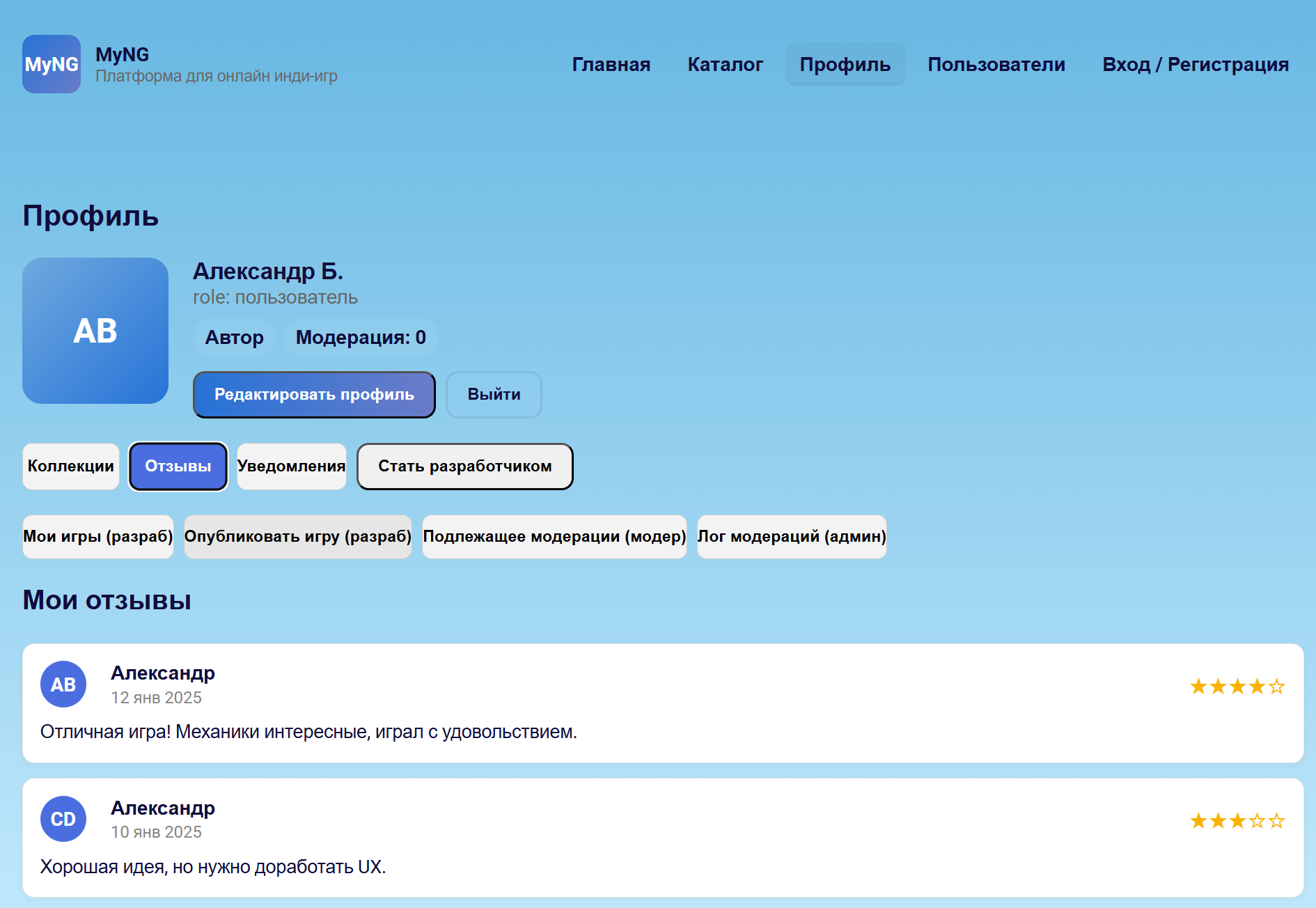
Рисунки 22. Макет «Оконная форма редактирования игры».



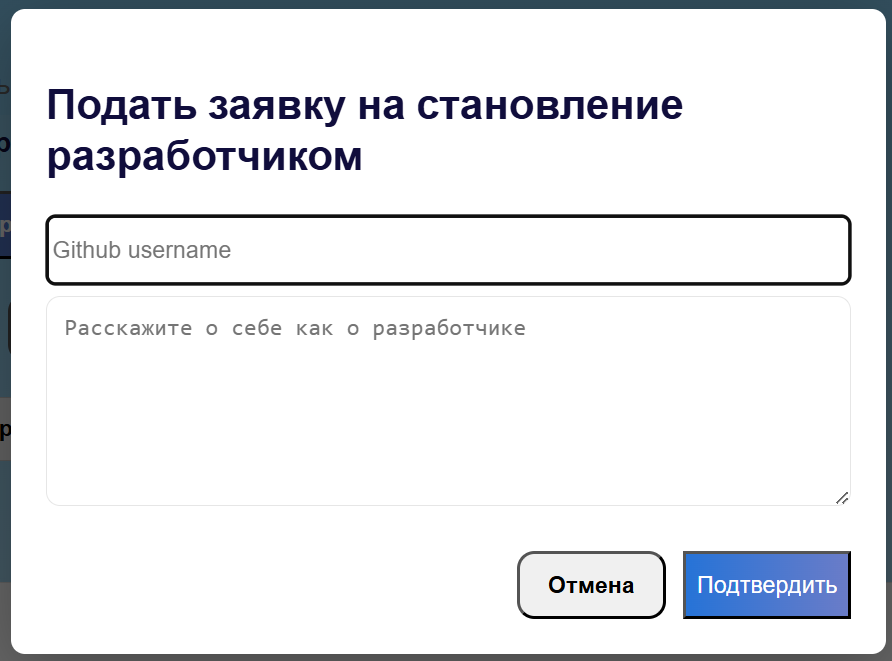
Рисунки 23. Макет «Оконная форма выхода из аккаунта».



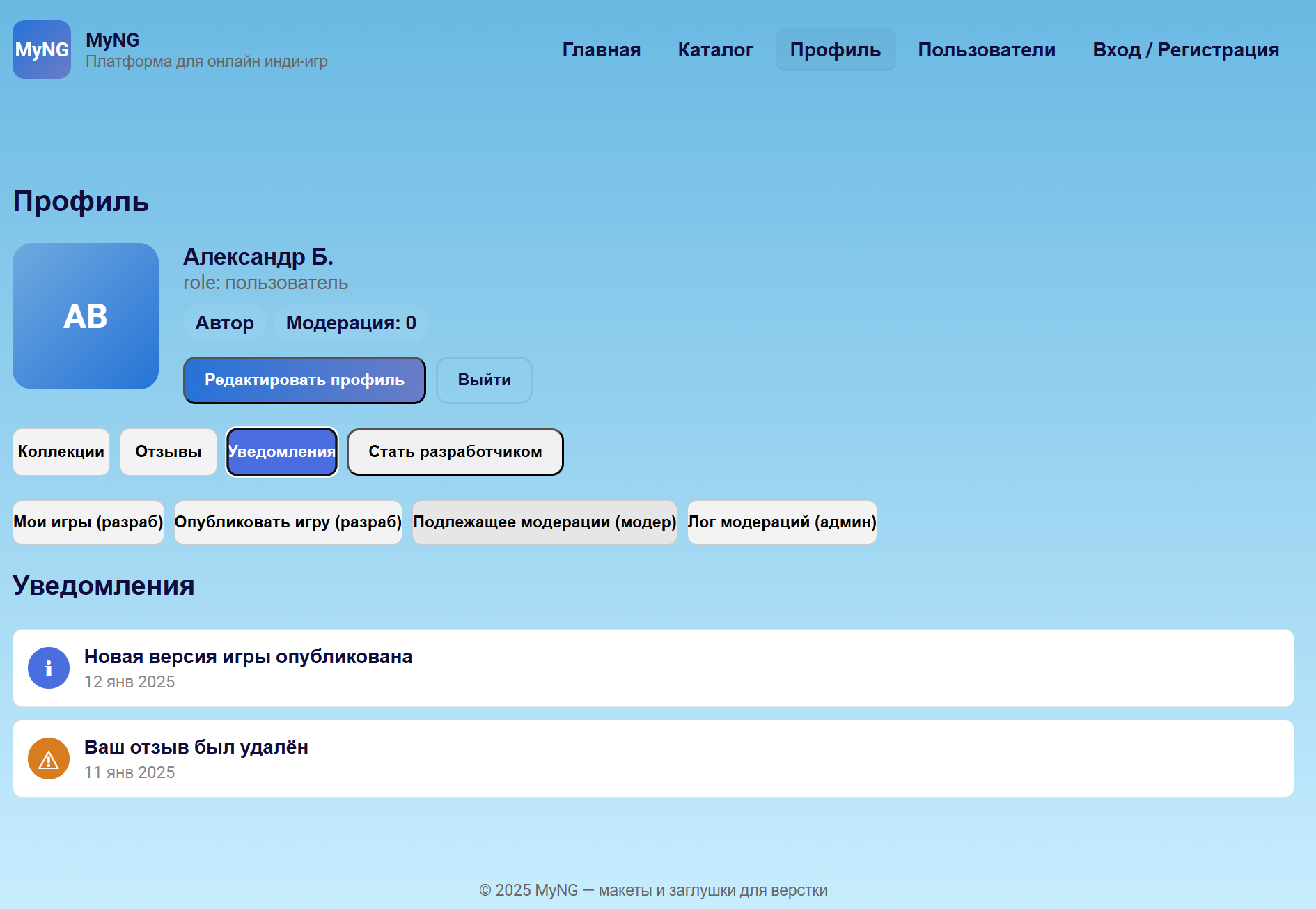
Рисунки 24. Макет «Оконная форма редактирования профиля».



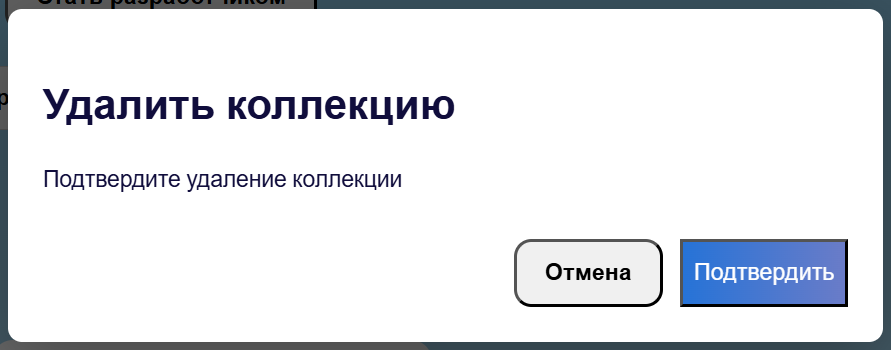
Рисунки 25. Макет «Оконная форма просмотра своих отзывов».



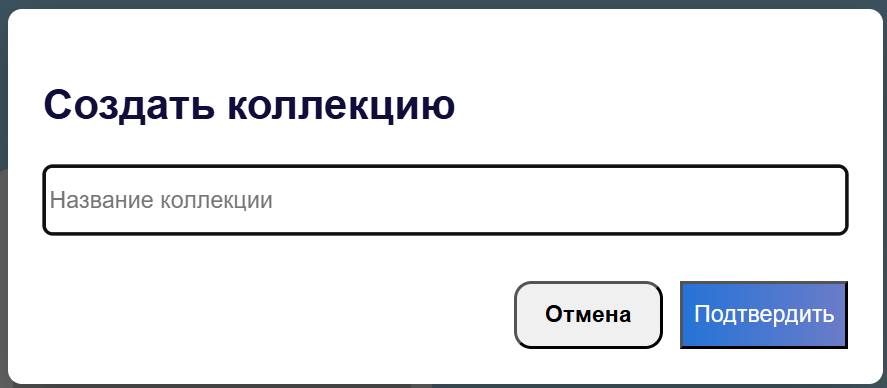
Рисунки 26. Макет «Оконная форма подачи заявки на разработчика».



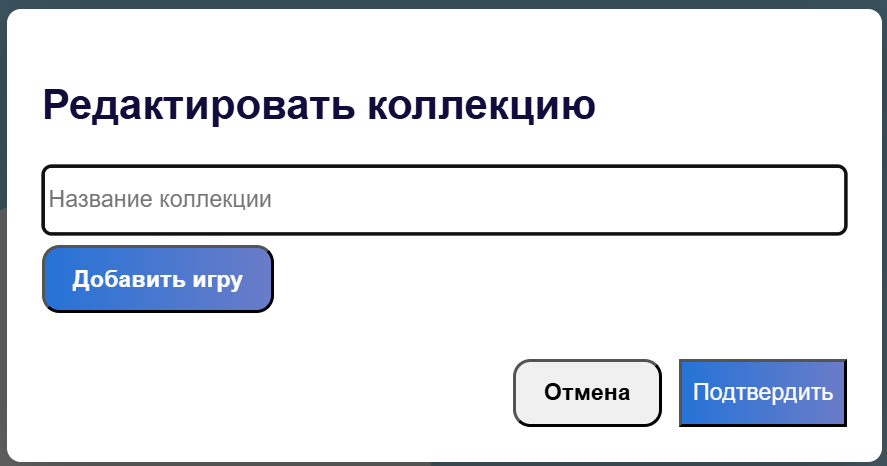
Рисунки 27. Макет «Оконная форма просмотра уведомлений».



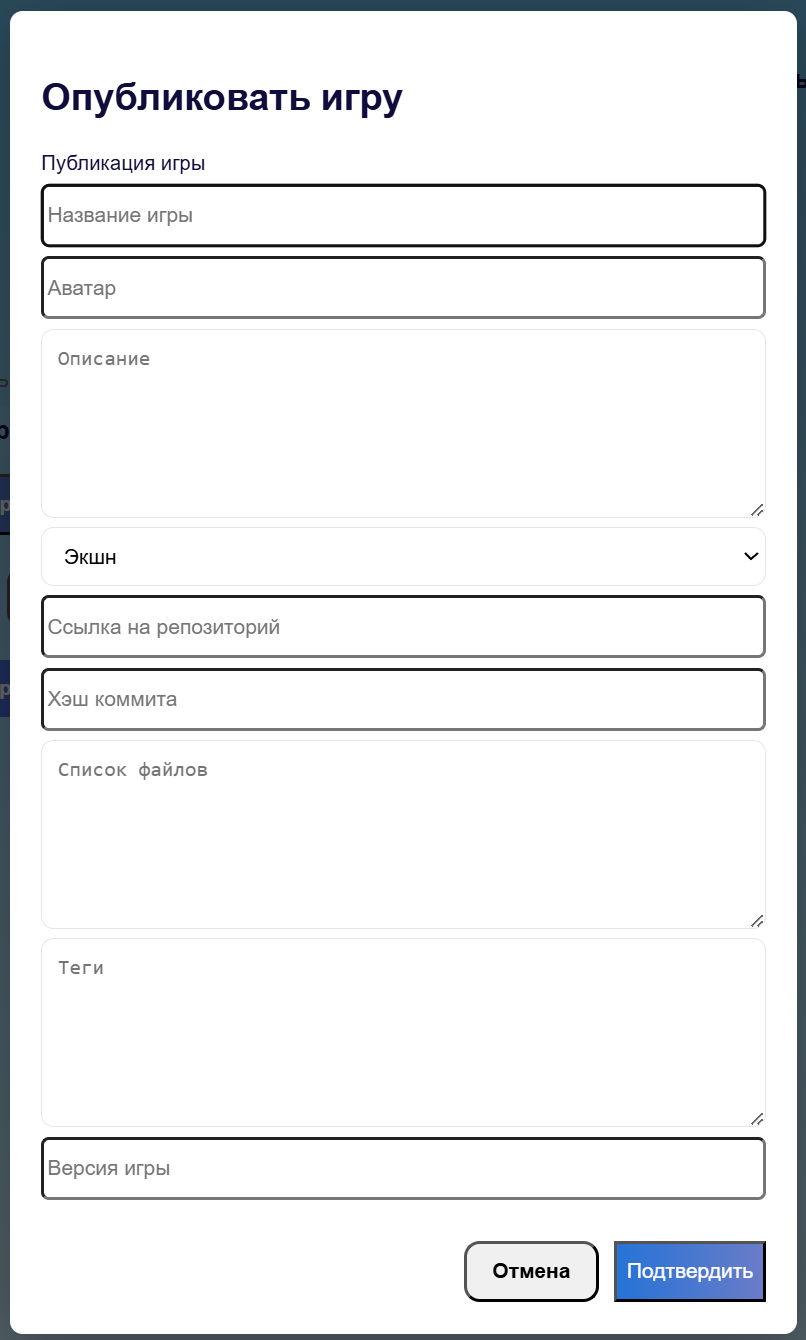
Рисунки 28. Макет «Оконная форма удаления коллекций».



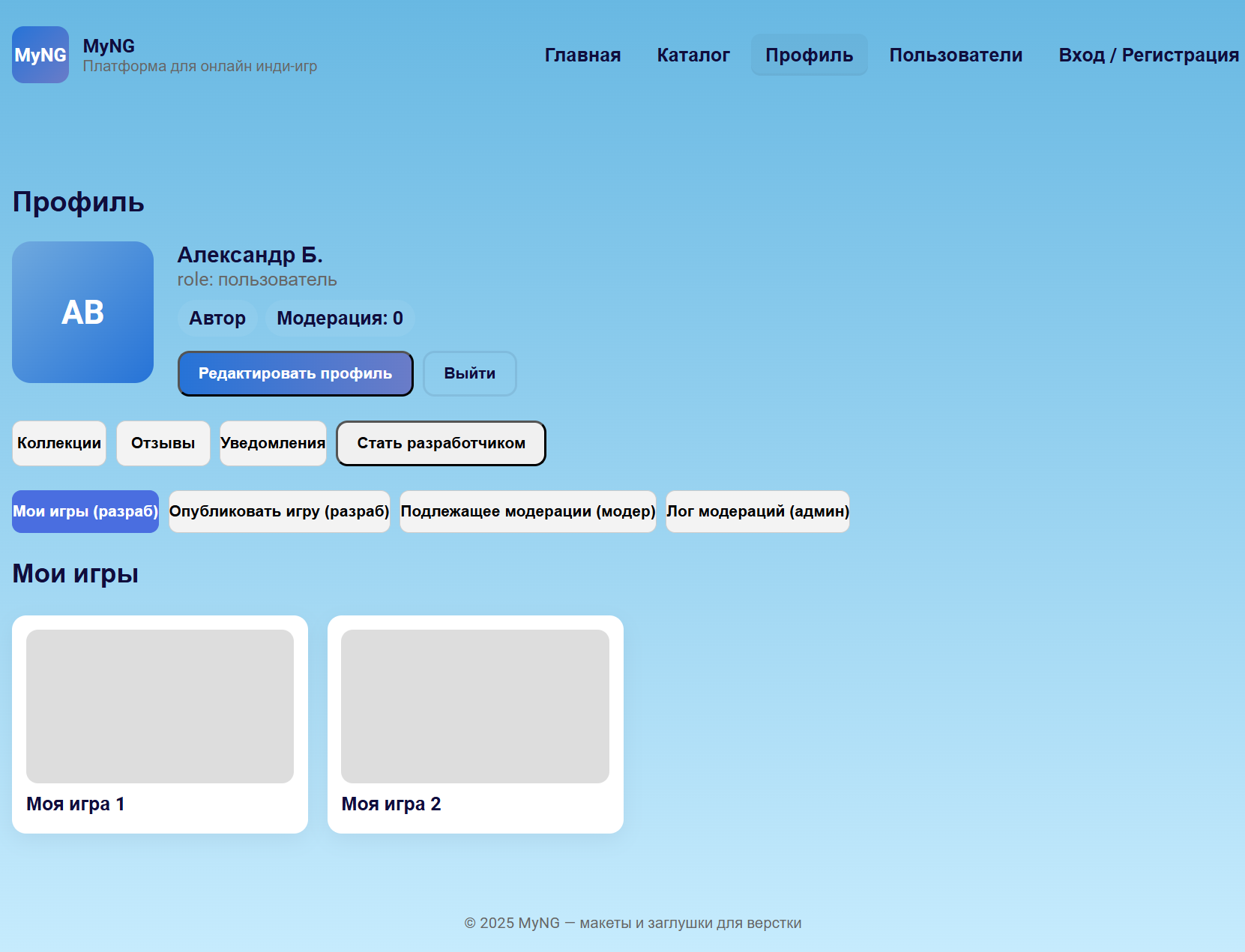
Рисунки 29. Макет «Оконная форма создания коллекций».



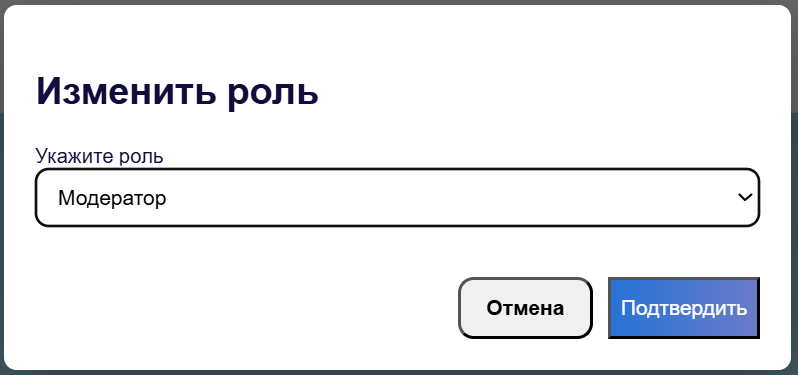
Рисунки 30. Макет «Оконная форма редактирования коллекций».



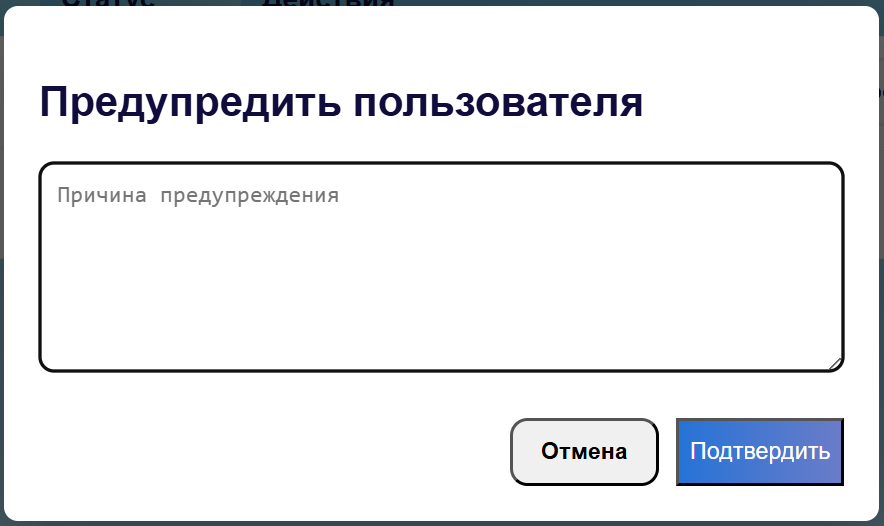
Рисунки 31. Макет «Оконная форма публикации игры».



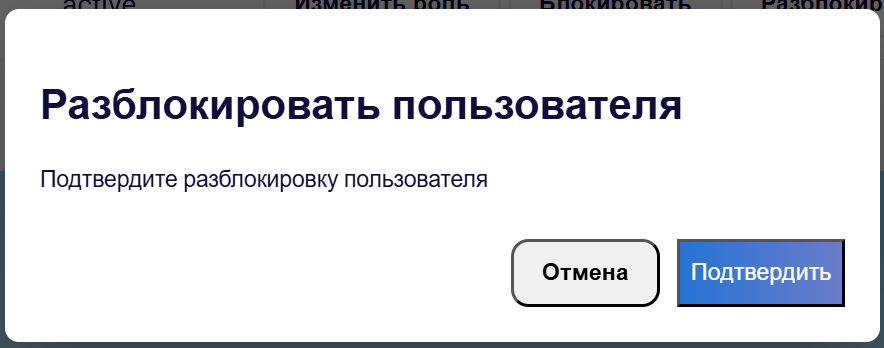
Рисунки 32. Макет «Оконная форма просмотра опубликованных игр».



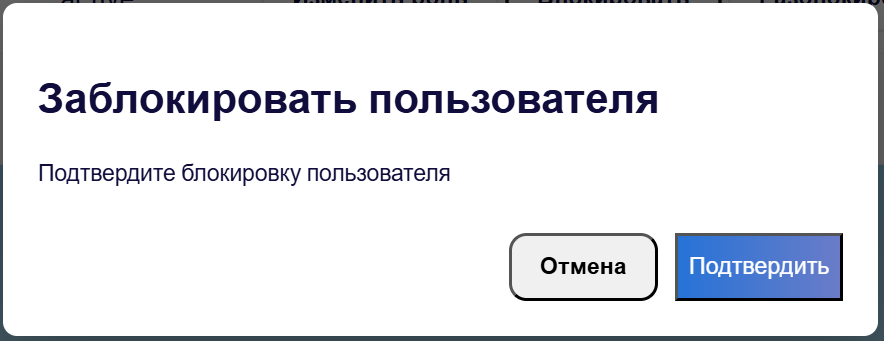
Рисунки 33. Макет «Оконная форма изменения роли пользователя».



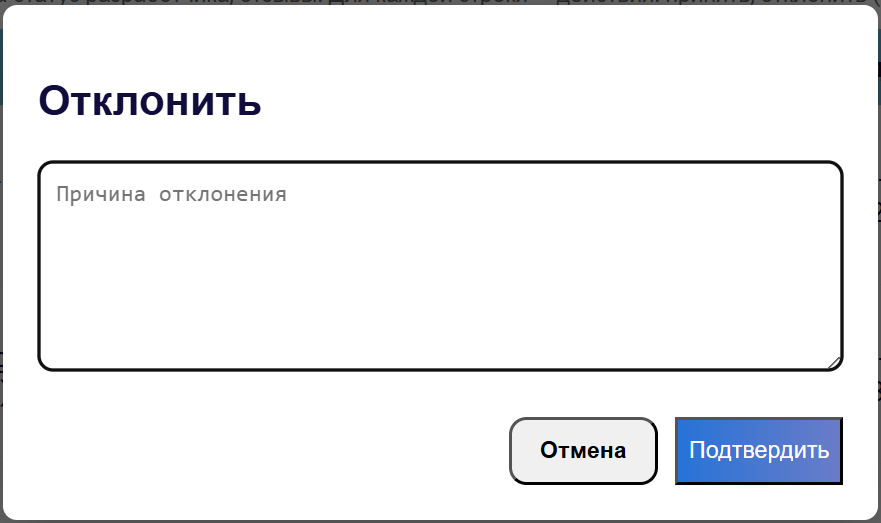
Рисунки 34. Макет «Оконная форма выдачи предупреждения».



Рисунки 35. Макет «Оконная форма разблокировки пользователя».



Рисунки 36. Макет «Оконная форма блокировки пользователя».



Рисунки 37. Макет «Оконная форма отклонения сущности».

**Верстка страниц с заглушками**

Далее по составленным макетам была написана верстка страниц. Представлять скриншоты страниц мы считаем не нужным, так как макеты и страницы одинаковые нет смысла прикреплять верстку еще раз. Единый файл всех страниц представлен в «site.html» - удобный формат, чтобы можно было скачать и открыть 1 страницу с возможностью посмотреть верстку всех, переходя по ссылкам.