МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Курсовая работа

Отчёт по курсу «Технологии программирования»

Разработка веб-приложения «ИгроВарка»

Студенты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Богословский А.Д.*

*подпись*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Колупанов А.В.*

*подпись*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Швыденко О.А.*

*подпись*

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Тарасов В.С.*

*подпись*

Зав. Кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Махортов С.Д.*

*подпись*

Воронеж2022

Содержание

[1. Анализ предметной области 4](#_Toc102228937)

[1.1. Постановка задачи 4](#_Toc102228938)

[1.2. Обзор аналогов 5](#_Toc102228939)

[1.2.1. Hot Game – hot-game.info 6](#_Toc102228940)

[1.2.2. Steampay – steampay.com 9](#_Toc102228941)

[1.3. Терминология и сокращения 11](#_Toc102228942)

[1.4. Функциональные требования 13](#_Toc102228943)

[1.5. Технические требования 14](#_Toc102228944)

[1.6. Требования к интерфейсу 14](#_Toc102228945)

[2. Разработка 15](#_Toc102228946)

[2.1. Макеты интерфейса 15](#_Toc102228947)

[2.2. Диаграмма IDEF-0 21](#_Toc102228948)

[2.3. Диаграмма Use-case 22](#_Toc102228949)

[2.4. Диаграмма Workflow 22](#_Toc102228950)

[2.5. Диаграмма состояний 23](#_Toc102228951)

[2.6. Диаграмма активностей 24](#_Toc102228952)

[2.7. Диаграмма DFD 25](#_Toc102228953)

[2.8. Схема приложения 25](#_Toc102228954)

[2.9. Схема Базы данных 26](#_Toc102228955)

[2.10. Технические средства 26](#_Toc102228956)

[2.11. Состав и содержание работ по созданию системы 27](#_Toc102228957)

[3. Сценарии воронки 30](#_Toc102228958)

[3.1. Сценарий 1 30](#_Toc102228959)

[3.2. Сценарий 2 31](#_Toc102228960)

[3.3. Сценарий 3 32](#_Toc102228961)

[3.4. Сценарий 4 33](#_Toc102228962)

[3.5. Сценарий 5 34](#_Toc102228963)

[4. Реализация 36](#_Toc102228964)

[5. Тестирование 36](#_Toc102228965)

[6. Заключение 36](#_Toc102228966)

## 1. Анализ предметной области

В современном мире, возрос спрос на цифровые развлечения. Самый распространённый вид развлечений после кинематографа, идут компьютерные игры. Чтобы получить удовольствие от игры, пользователю мало поиграть игру, нужно ознакомиться с вариантами, выбрать, купить и много последующих действий. Среди современного многообразия игр следует проблема выбора.

Веб-приложение — это программа, которая полностью работает через Интернет и потенциально может быть доступна через несколько сред. Это прикладная программа, которая хранится на удаленном сервере и доставляется по сети, такой как Интернет или интранет, через интерфейс браузера. Сегодня на рынке существует множество веб-приложений. Большинство из них можно доступно бесплатно.

При создании веб-приложения необходимо учитывать несколько моментов. Во-первых, он должен работать в большинстве веб-браузеров. Желательно иметь некоторое представление о том, какой браузер может поддерживать функциональные возможности, необходимые веб-приложению. Еще одна вещь, которую следует учитывать, — это доступность. К веб-приложению должен быть удобен доступ через любую браузерную платформу, и должна быть возможность поддерживать веб-приложения без распространения и установки программного обеспечения на множество разных клиентских компьютеров. Веб-приложения обычно используют комбинацию сценариев на стороне клиента и сценария на стороне сервера для разработки приложения.

## 1.1. Постановка задачи

Целью данной работы является разработка веб-приложения способного упростить для пользователя, задачу поиска и выбора игр. Разработанное веб-приложение должно обладать интуитивно понятным интерфейсом и получать информацию об играх с серверов Steam (Valve). Особенностью веб-приложения является возможность выбора игр из списка по критериям: количество игроков, которые могут одновременно играть и является ли игра платным или бесплатным проектом.

Сайт создается для поиска и просмотра игр, отобранных экспертами сайта, а также подбора рекомендуемых игр с помощью фильтрации всей библиотеки по параметрам, заданным пользователем.

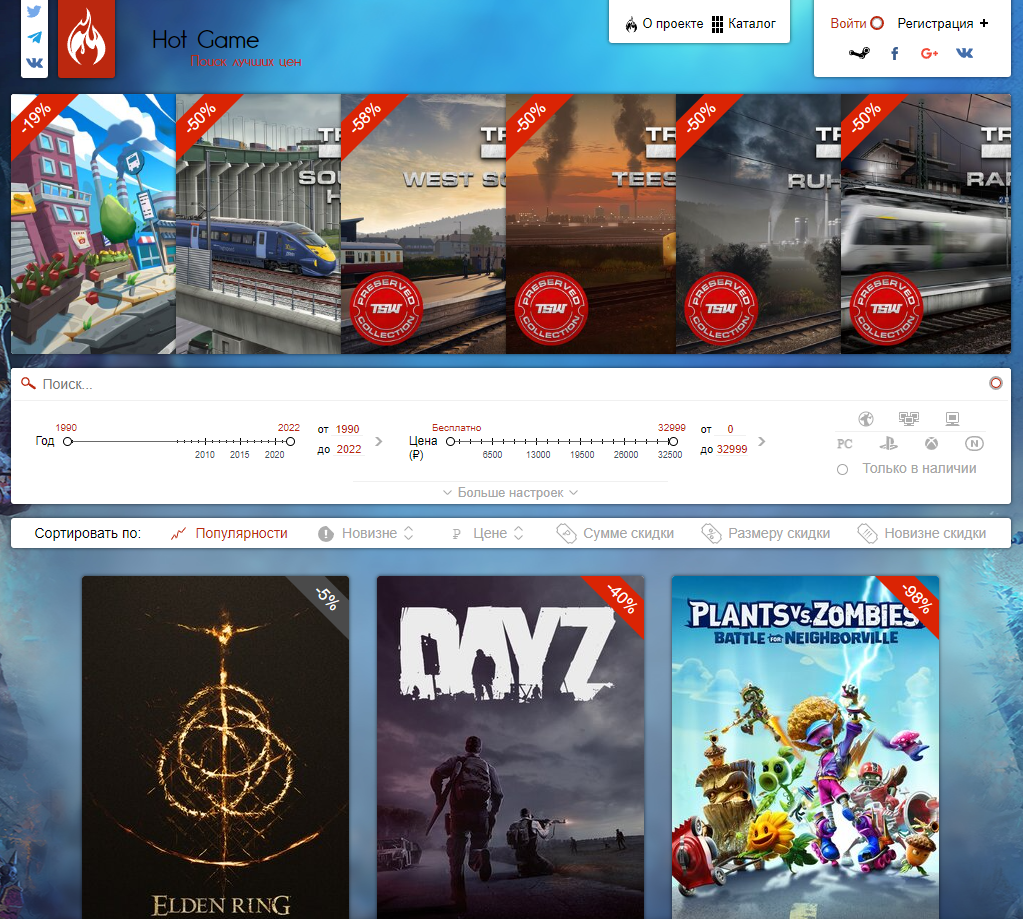
Разрабатываемый проект должен решать следующие задачи:

* Обеспечение доступа к каталогу игр, отобранных экспертами сайта.
* Обеспечение поиска игр по слову в названии.
* Обеспечение возможности просмотра видео-трейлера игры.
* Обеспечение возможности перехода в магазин Steam для дальнейшей покупки.
* Обеспечение добавления игр в понравившиеся и дальнейшего её отображения в профиле.
* Обеспечение возможностей для администраторов сайта (экспертов) модерировать контент на сайте.
* Обеспечение возможности подбора игры.

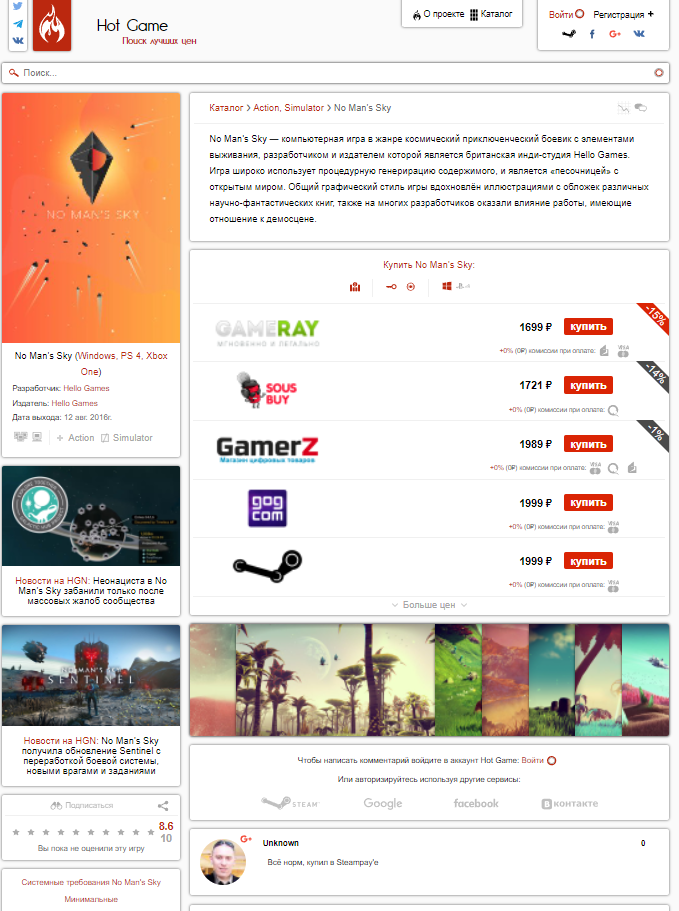
## 1.2. Обзор аналогов

Начиная разработку ПО необходимо произвести анализ рынка и обзор ближайших аналогов. На основе анализа будут сделаны выводы о необходимости реализации различного функционала и интерфейсных элементах.

## 1.2.1. Hot Game – hot-game.info



Главная страница сервиса



Страница игры

Функциональность:

* Поиск игр по названию или фильтрам.
* Сортировка игр на странице.
* Обзор игры, её описание, скриншоты, трейлер, системные требования.
* График цен.
* Агрегация цен из различных онлайн магазинов.
* Система комментариев.
* Аутентификация через различные социальные сети.

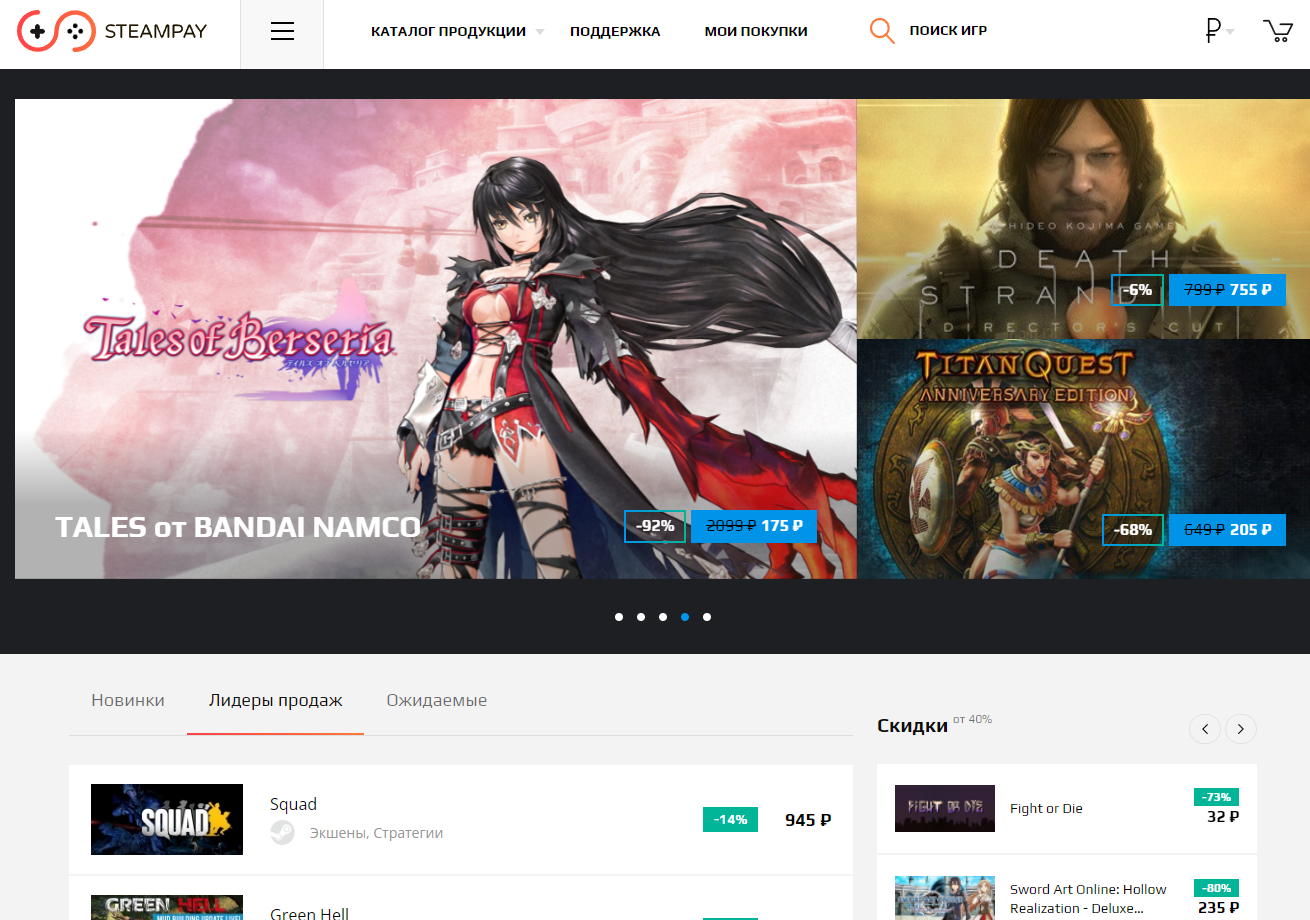
Достоинства:

* Уникальный функционал по агрегации цен (согласно документации разработчиков).
* Собственное расширение для браузера.
* Система учёта комиссий платёжных систем.

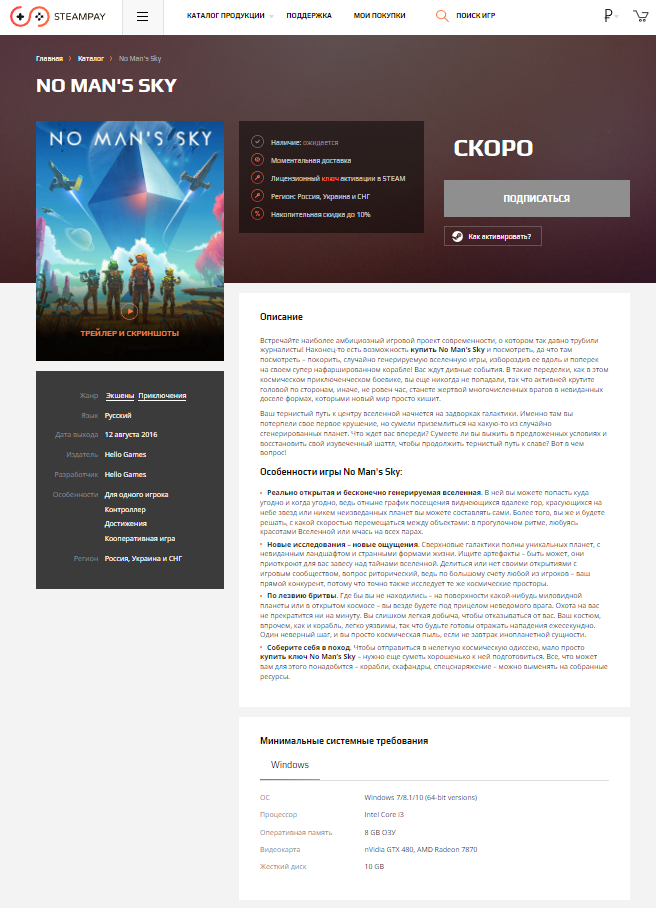
Недостатки:

* Система поиска игр аналогичная Steam.
* Отсутствие подборок.

## 1.2.2. Steampay – steampay.com



Главная страница



Страница игры

Функциональность:

* Поиск игр по названию или фильтрам.
* Сортировка игр на странице.
* Обзор игры, её описание, скриншоты, трейлер, системные требования.
* Система комментариев.

Достоинства:

* Собственная система накопительной скидки.
* Система предзаказов.
* Бонусная программа.
* Календарь игр.
* Отображение цен в различных валютах.

Недостатки:

* Система поиска игр аналогичная Steam.
* Отсутствие подборок.

## 1.3. Терминология и сокращения

* Проект – разрабатываемое командой веб-приложение.
* Клиент, клиентская часть, клиентская сторона – средство (компьютер), принимающее данные от сервера и предоставляющее возможность взаимодействия с системой.
* Сервер – средство (компьютер), принимающее запрос от клиента, производящее вычисления и формирующее веб-страницу для дальнейшей отправки клиенту.
* Python – высокоуровневый объектно-ориентированный язык программирования.
* Front-end – клиентская часть приложения, отвечающая за получение информации от сервера для её дальнейшего отображения на устройстве пользователя.
* Back-end – программно-аппаратная часть приложения, отвечающая за функционирование внутренней части.
* API – специальный протокол для взаимодействия компьютерных программ.
* REST API – архитектурный стиль взаимодействия компонентов распределённого приложения в сети.
* Steam WebAPI – служба Valve, позволяющая получать данные с серверов Steam.
* Веб-сервис – программная система с уникальным веб-адресом (URL), предоставляющая услуги с помощью специальных программ.
* GitHub – веб-сервис для хостинга и совместной разработки IT-проектов.
* Trello – веб-сервис управления проектами и менеджмента задач по методологии канбан.
* Пользователь – человек, имеющий доступ к базовому функционалу веб-приложения.
* Администратор – авторизованный в системе человек, имеющий доступ к расширенному функционалу веб-приложения.

## 1.4. Функциональные требования

Система должна соответствовать следующим функциональным требованиям:

1. Авторизация/Регистрация/Смена пароля

* При первом использовании приложения пользователь имеет возможность зарегистрировать новый аккаунт.
* При повторном использовании приложения пользователь имеет возможность авторизоваться.
* При повторном использовании приложения авторизованный пользователь имеет возможность сменить пароль.

1. Просмотр каталога игр

* Пользователь может просматривать каталог игр.
* Пользователь может искать игры по названию.
* Пользователь может искать игру по тегам.
* Пользователь может отсортировать список игр.

1. Виджет подбора игры (Помощник)

* Только авторизованный пользователь может подобрать игру по тегам, стоимости и количеству игроков, которые в неё будут играть.

1. Просмотр конкретной игры

Неавторизованный пользователь:

* Пользователь может посмотреть обложку игры.
* Пользователь может прочитать описание игры.
* Пользователь может посмотреть теги игры.
* Пользователь может посмотреть трейлер игры.
* Пользователь может воспользоваться виджетом Steam для перехода в магазин (для покупки игры).

Авторизованный пользователь имеет доступ к функциям уровня неавторизованного пользователя, а также:

* Пользователь может добавить игру в список понравившихся игр.
* Пользователь может оставить отзыв об игре в виде цифровой оценки.

## 1.5. Технические требования

Созданное веб-приложение должно иметь архитектуры базового Клиент-Серверного приложения, которому присуще разделение на Front-end и Back-end, взаимодействие которых должно быть реализовано с помощью REST API и получение данных для каталога из Steam WebAPI. Все страницы сайта должны быть реализованы с поддержкой русского языка (RU).

## 1.6. Требования к интерфейсу

Оформление и вёрстка страниц должны соответствовать следующим требованиям:

* Все страницы должны соответствовать единому дизайну.
* Вёрстка должна быть адаптивной и кроссплатформенной (быть единой в разных браузерах).
* Все элементы управления должны быть выделены на фоне основного содержимого страницы.

## 2. Разработка

## 2.1. Макеты интерфейса

При открытии веб приложения пользователь попадает на главную страницу.



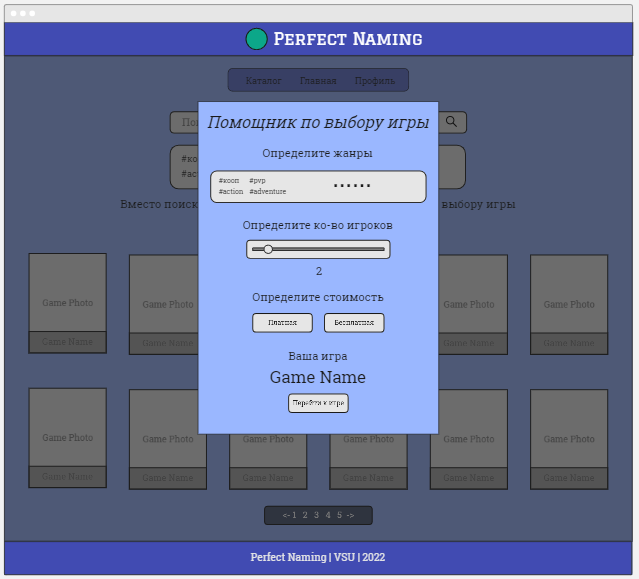
Главная

На странице каталога пользователь, может ознакомиться с играми и начать поиск игры по названию или фильтрам.



Каталог

При помощи помощника пользователь может найти игру по конкретным критериям, сколько игроков в группе и какая стоимость у игры.



Помощник (Подбор игры)

На странице игры пользователь может ознакомиться с информацией об игре, посмотреть трейлер и перейти в Steam, для дальнейшей покупки.



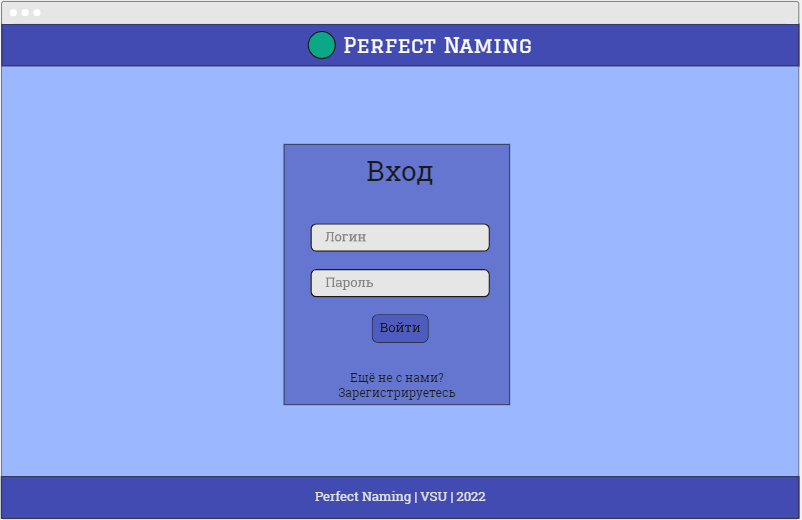
Страница игры

На странице Профиль пользователь может ознакомиться со своими данными и посмотреть игры, которые он добавил в список понравившихся.



Профиль

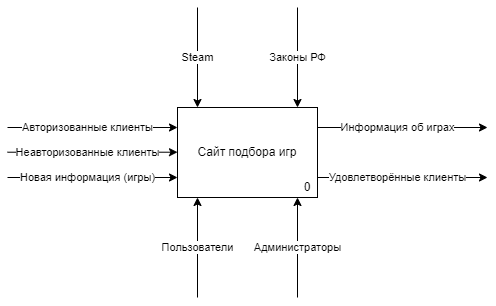
На странице Авторизация пользователь может войти в свой профиль или перейти к регистрации.



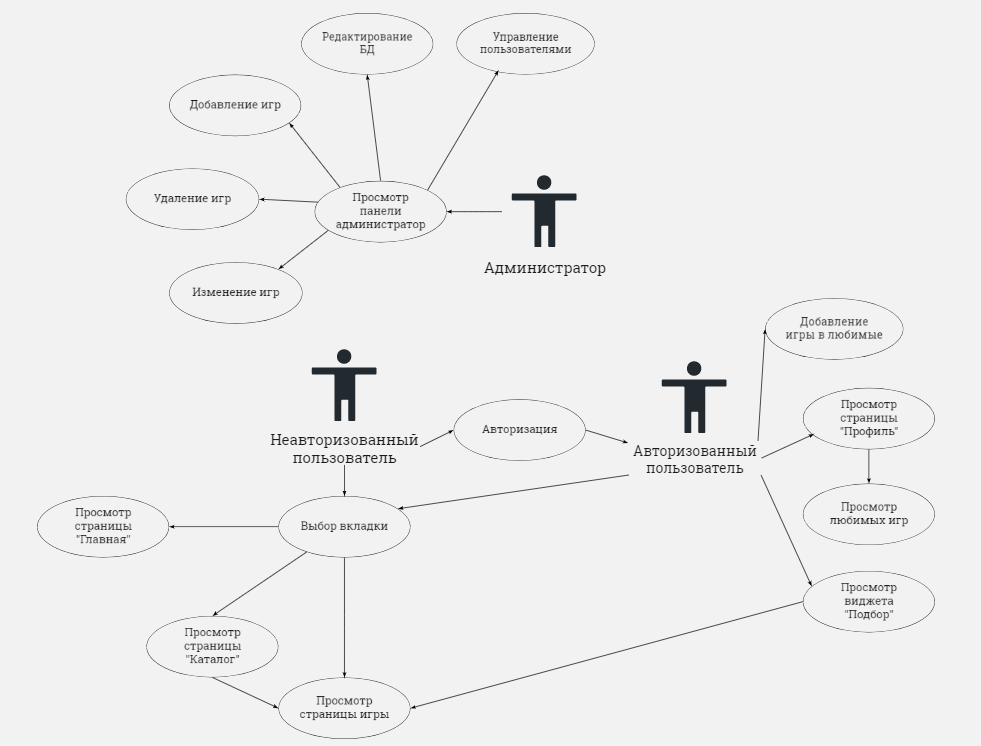
Авторизация

Примечание: Страница Регистрация аналогична странице Авторизация.

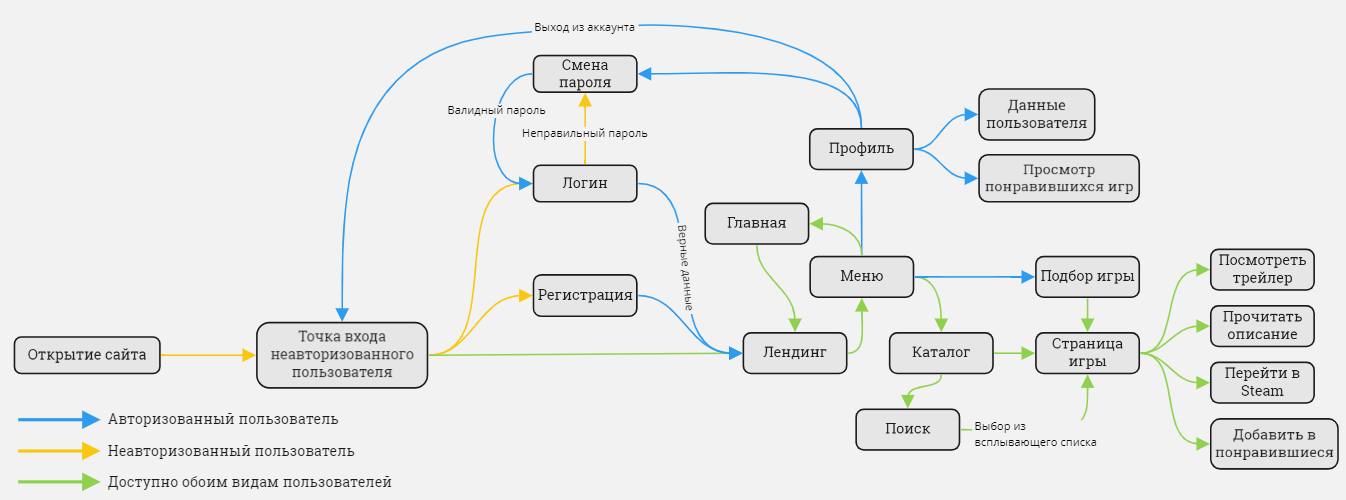
## 2.2. Диаграмма IDEF-0



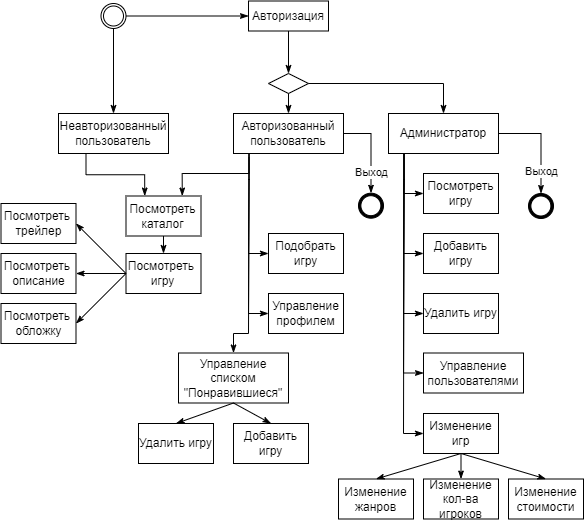
## 2.3. Диаграмма Use-case



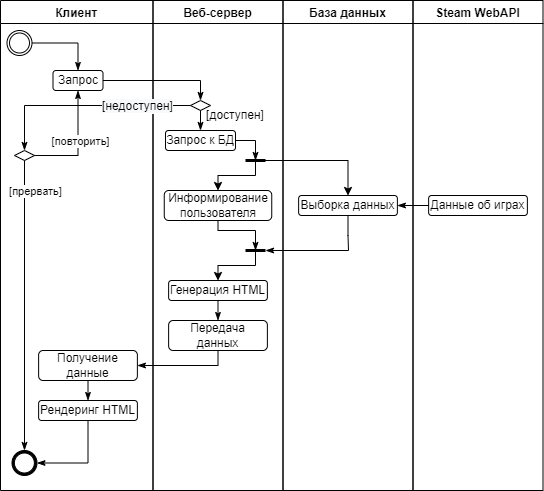
## 2.4. Диаграмма Workflow



## 2.5. Диаграмма состояний



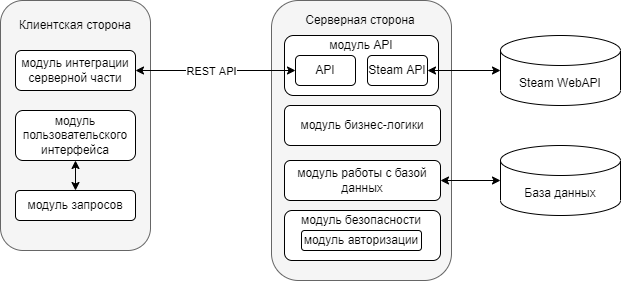
## 2.6. Диаграмма активностей



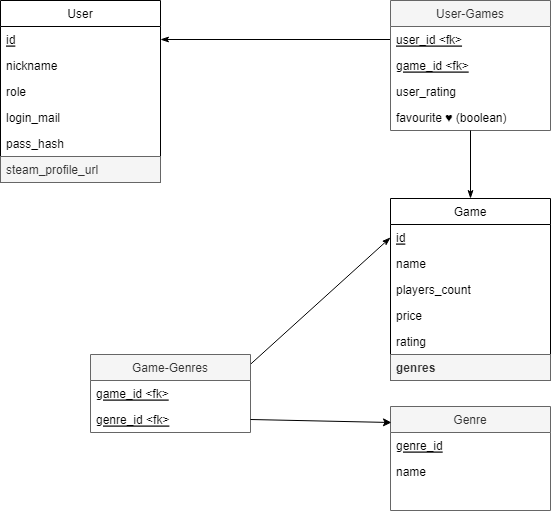
## 2.7. Диаграмма DFD



## 2.8. Схема приложения



## 2.9. Схема Базы данных



## 2.10. Технические средства

* Язык программирования Python
* Язык разметки HTML
* Язык описания внешнего вида CSS + Язык JavaScript
* Steam WebAPI
* СУБД SQLite
* IDE PyCharm
* Swagger
* Flask
* Система контроля версий Git (Github)
* Таск-менеджер Trello
* Хостинг проекта Heroku
* Хостинг и средство разработки интерфейса TeleportHQ

## 2.11. Состав и содержание работ по созданию системы

Плановый срок начала работ – Март 2022 года

Плановый срок окончания работ – Июнь 2022 года

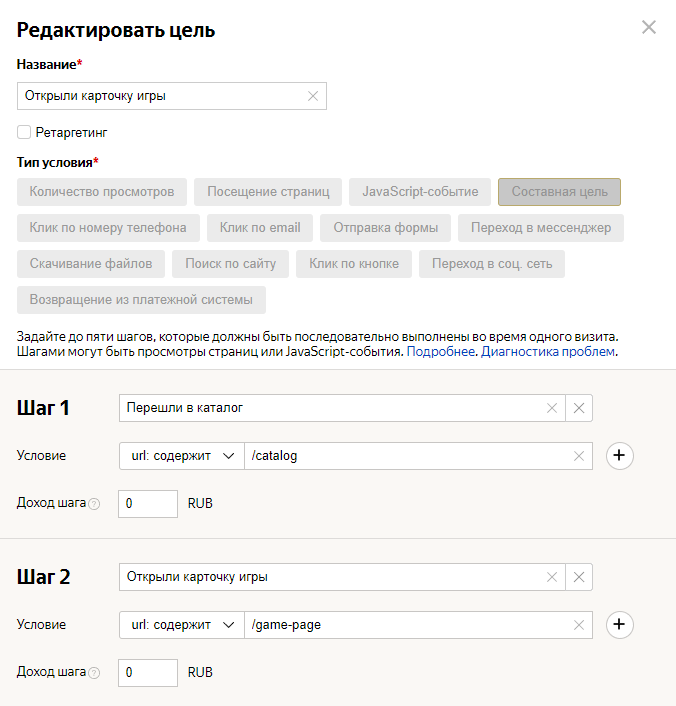
Основные этапы по разработке системы, их содержание и примерные сроки выполнения приведены в Таблице 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Этап | Содержание работ | Порядок приёмки документы | Сроки | Ответственный |
| 1. Техническое задание | Разработка требований к системе | Утверждение технического задания | До 16.03.22 | Разработка – Исполнитель  Согласование – Заказчик |
| 2. Техническое проектирование | Разработка сценариев работы системы | Доска и ссылка на miro.com | До 16.03.22 | Исполнитель |
| Разработка дизайн-макета | Представление изображений дизайн-макета | До 16.03.22 | Исполнитель |
| 3. Разработка системы | Разработка серверного модуля, модуля базы данных | Приём результатов работы осуществляется в процессе тестирования и испытаний | В течение 2 календарных месяцев с момента утверждения технического задания | Исполнитель |
| Разработка динамической части веб-сайта |
| Разработка статической части веб-сайта |
| 4. Предварительные автономные испытания | Проверка соответствия функциональным требованиям | Согласно техническому заданию | В течение недели (7 дней) с момента завершения разработки | Исполнитель |
| Проверка документации |
| Доработка и повторные испытания до устранения обнаруженных недостатков |
| 5. Разработка курсового проекта и сопровождающей документации | Разработка курсового проекта, содержащего аналитическую информацию о проекте на основе технического задания | В течение всего времени работы над проектом с момента утверждения технического задания | До 25.05.22 | Исполнитель |
| 6. Опытная эксплуатация | Предоставление доступа к текущей версии проекта, ограниченному количеству участников с целью тестирования на соответствие функциональным требованиям | Создание и ведение внутреннего документа | До 01.06.22 | Исполнитель |
| Доработка и повторные испытания до устранения обнаруженных недостатков |

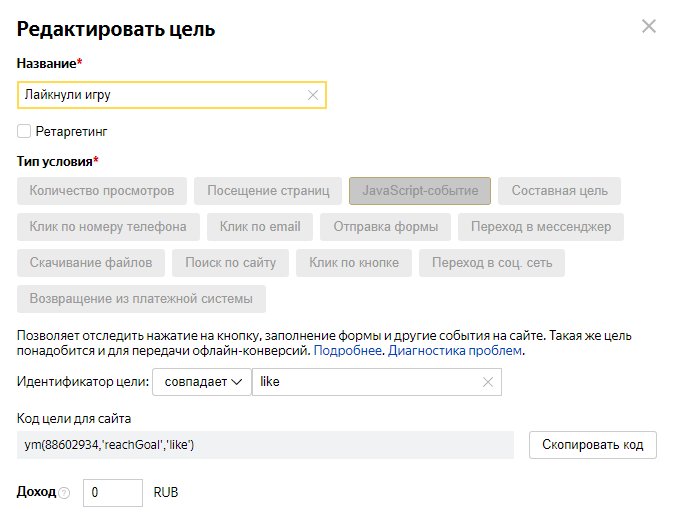
Таблица 1 - Основные этапы разработки системы

## 3. Сценарии воронки

## 3.1. Сценарий 1



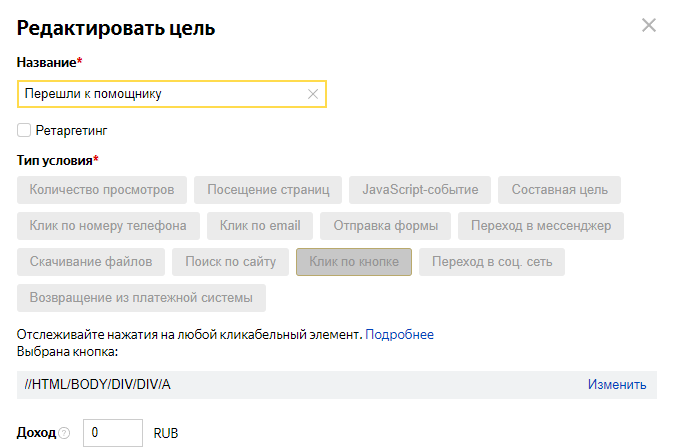
## 3.2. Сценарий 2



На странице игры авторизованный пользователь добавил игру в свой список, понравившихся игр, путём нажатия кнопки.



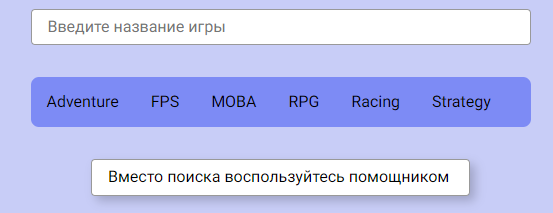
## 3.3. Сценарий 3



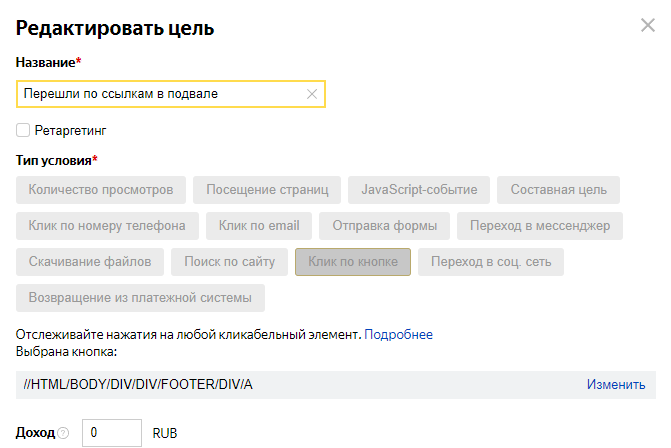
Авторизованные пользователи перешли к помощнику по подбору игр либо переходом из меню



либо нажатием кнопки на странице каталога



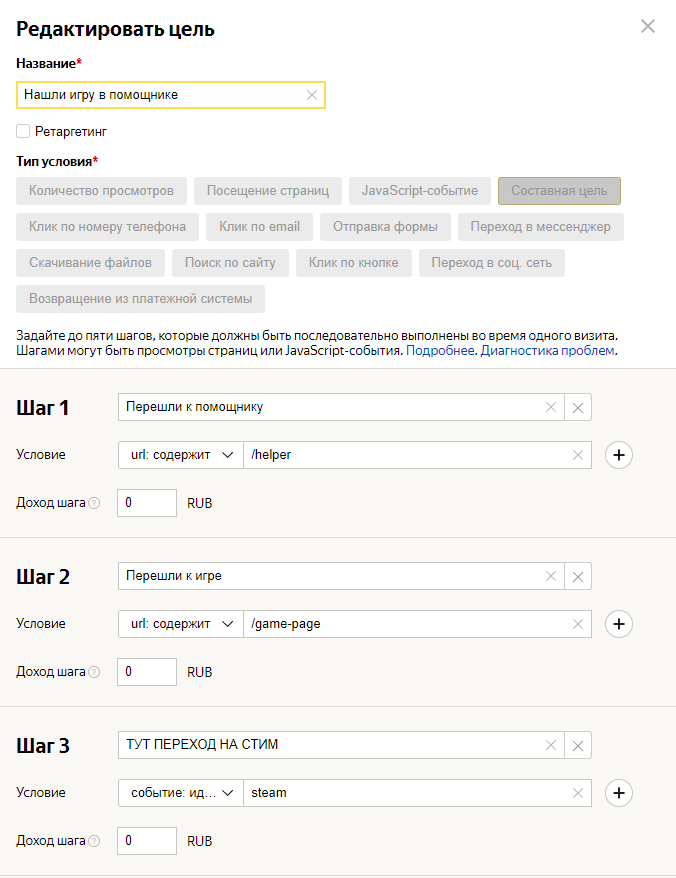
## 3.4. Сценарий 4



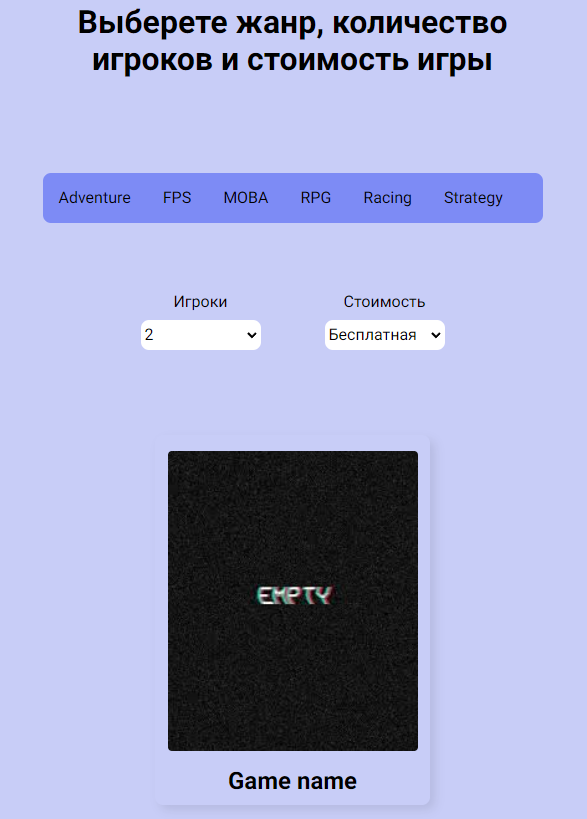
Пользователи перешли по ссылкам в подвале страниц.

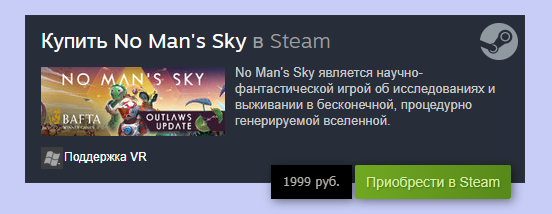


## 3.5. Сценарий 5



Авторизованные пользователи перешли к помощнику, воспользовались им и перешли на игру, которую помощник нашёл и со страницы данной игры перешли в магазин Steam.





## 4. Реализация

## 5. Тестирование

## 6. Заключение