

Минобрнауки России

Юго-Западный государственный университет

Кафедра программной инженерии

ОТЧЕТ

о преддипломной (производственной) практике

наименование вида и типа практики

на (в) ООО "Предприятие ВТИ-Сервис"

наименование предприятия, организации, учреждения

Студента 4курса, группы ПО-016

курса, группы

Зеленцова Виталия Евгеньевича

фамилия, имя, отчество

Руководитель практики от
предприятия, организации,
учреждения

Оценка

директор

должность, звание, степень

Куркина А. В.

фамилия и. о.

подпись, дата

Руководитель практики от
университета

Оценка

к.т.н. доцент

должность, звание, степень

Чаплыгин А. А.

фамилия и. о.

подпись, дата

Члены комиссии

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

Курск 2024 г.

СОДЕРЖАНИЕ

1	Анализ предметной области	5
1.1	Сервисы для обмена и продажи виртуальных ценностей в MMORPG	5
1.2	Характеристика игры «Stay Out»	6
2	Техническое задание	7
2.1	Основание для разработки	7
2.2	Цель и назначение разработки	7
2.3	Требования пользователя к интерфейсу web-сайта	7
2.4	Функциональные требования к программной системе	8
2.4.1	Вариант использования «Выбор сервера»	9
2.4.2	Вариант использования «Смена сервера»	10
2.4.3	Вариант использования «Поиск»	10
2.4.4	Вариант использования «Просмотр информации о товаре»	11
2.4.5	Вариант использования «Покупка товара»	11
2.4.6	Вариант использования «Регистрация»	11
2.4.7	Вариант использования «Авторизация»	12
2.4.8	Вариант использования «Выставление товара»	13
2.5	Требования к оформлению документации	13
3	Технический проект	14
3.1	Общая характеристика организации решения задачи	14
3.2	Обоснование выбора технологии проектирования	14
3.2.1	Описание используемых технологий и языков программирования	14
3.2.2	Язык программирования JavaScript	15
3.2.2.1	Достоинства языка JavaScript	15
3.2.2.2	Недостатки языка Javascript	16
3.3	Диаграмма компонентов и схема обмена данными между файлами компонента	16
3.4	Диаграмма размещения	18

3.5 Содержание информационных блоков. Основные сущности	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	25

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД – база данных.

ИС – информационная система.

ИТ – информационные технологии.

КТС – комплекс технических средств.

ОМТС – отдел материально-технического снабжения.

ПО – программное обеспечение.

РП – рабочий проект.

СУБД – система управления базами данных.

ТЗ – техническое задание.

ТП – технический проект.

UML (Unified Modelling Language) – язык графического описания для объектного моделирования в области разработки программного обеспечения.

1 Анализ предметной области

1.1 Сервисы для обмена и продажи виртуальных ценностей в MMORPG

В мире массовых многопользовательских онлайн-игр (MMORPG), таких как «Stay Out», игроки могут взаимодействовать в виртуальном мире, выполнять квесты, сражаться с монстрами, развивать своих персонажей и, конечно же, торговать внутриигровыми ценностями. MMORPG представляют собой уникальный игровой жанр, где тысячи игроков со всего мира могут взаимодействовать между собой в одном виртуальном мире, создавая атмосферу постоянного развития и приключений.

Сервисы для обмена и продажи виртуальных ценностей в MMORPG призван упростить процесс торговли между игроками, предоставляя им удобную платформу для взаимодействия. Необходимость в таком сервисе возникает из-за того, что в некоторых играх обмен виртуальными ценностями может быть неудобным и невыгодным для игроков. Это может приводить к созданию различных групп в социальных сетях и мессенджерах, где игроки выкладывают свои товары и договариваются о сделках.

Целью таких сервисов является упрощение процесса торговли, делая его более удобным и безопасным для всех участников. Путем создания специализированной платформы для обмена и продажи виртуальных ценностей, игрокам будет легче находить нужные товары, договариваться о цене и завершать сделки, минимизируя риск обмана и недобросовестных сделок.

Преимущества сервиса для обмена виртуальными ценностями:

- Экономия ресурсов: Сервис не требует значительных ресурсов для использования и позволяет игрокам свободно обмениваться своими предметами.
- Удобство взаимодействия: Пользователи могут легко связываться друг с другом, договариваться о цене и условиях сделки, что делает процесс торговли более удобным и эффективным.

- Минимизация рисков обмана: В отличие от торговли в группах в социальных сетях, где информация о товарах может быть неполной или недостоверной, сервис предоставляет полные характеристики предметов с самого начала сделки, что уменьшает риск обмана.

- Отсутствие вмешательства администрации: В сервисах для обмена виртуальными ценностями нет лишней административной информации и правил, устанавливаемых владельцем группы или сообщества, что делает процесс торговли более прозрачным и справедливым для всех участников.

- Быстрый поиск товаров: Пользователи могут легко найти необходимый товар и просмотреть все предложения на рынке в считанные секунды благодаря удобному интерфейсу и системе поиска.

1.2 Характеристика игры «Stay Out»

«Stay Out» представляет собой онлайн-игру в жанре выживания и RPG, где игроки исследуют постапокалиптический мир, выполняют различные задания и сражаются с мутантами. Для выставления товара на продажу в игре требуется уплата комиссии в размере 4%, а сами предметы могут достигать высоких цен в 1 миллиард, что делает данный процент неоправданно высоким. Кроме того, товары выставляются временно, всего на 2 дня, после чего возвращаются на склад, а комиссия за торговлю исчезает. Поиск товаров в игре осуществляется дословно, что затрудняет процесс поиска нужных предметов.

2 Техническое задание

2.1 Основание для разработки

Основанием для разработки является задание на выпускную квалификационную работу бакалавра «Маркетплейс для реализации цифровых предметов внутри игры "Stay Out"».

2.2 Цель и назначение разработки

Основной задачей выпускной квалификационной работы является разработка и внедрение web-сайта для упрощения торговли внутриигровыми ценностями в «Stay Out».

В процессе разработки web-сайта для упрощения торговли внутриигровыми ценностями в «Stay Out» можно выделить следующие задачи:

- реализация системы аутентификации и безопасности;
- разработка функционала поиска и фильтрации товаров;
- внедрение системы обратной связи;
- разработка системы управления контентом;
- внедрение механизмов модерации.

2.3 Требования пользователя к интерфейсу web-сайта

Сайт должен включать в себя:

- поиск предметов;
- авторизацию;
- регистрацию;
- выбор сервера;
- доступ к профилю.

Композиция шаблона сайта представлена на рисунке 2.1.

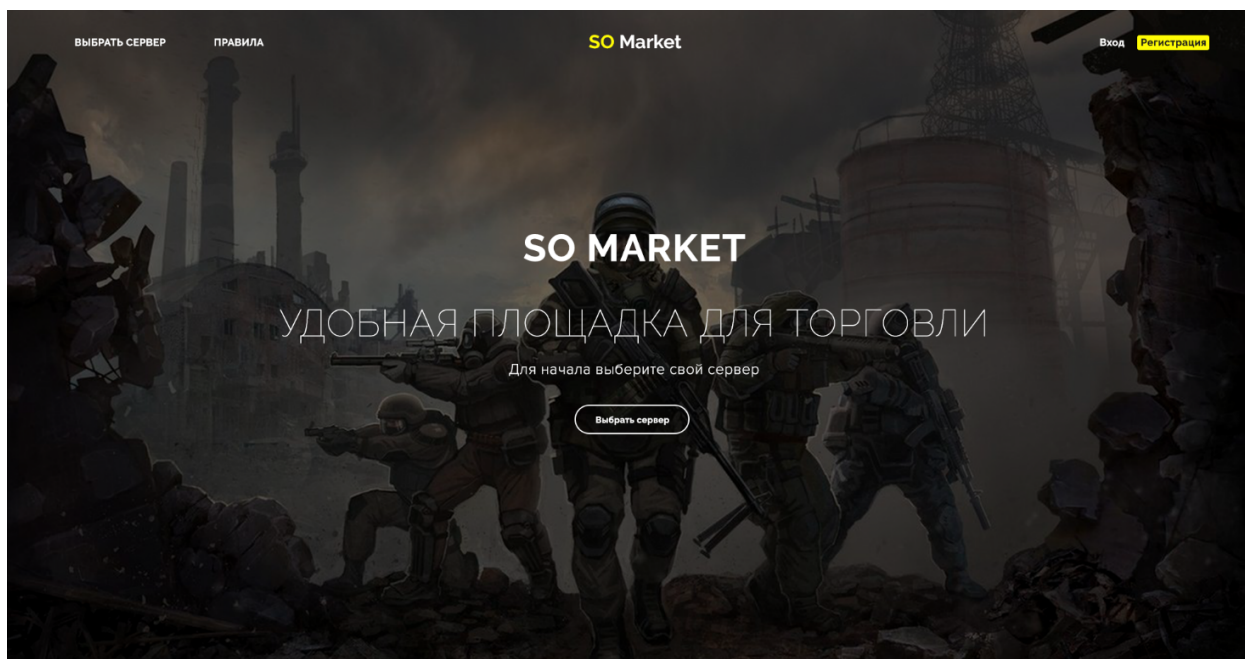


Рисунок 2.1 – Композиция шаблона сайта

2.4 Функциональные требования к программной системе

Для разрабатываемого сайта была реализована модель, которая обеспечивает наглядное представление вариантов использования сайта.

Она помогает в физической разработке и детальном анализе взаимосвязей объектов. При построении диаграммы вариантов использования применяется унифицированный язык визуального моделирования UML.

Диаграмма вариантов описывает функциональное назначение разрабатываемой системы. То есть это то, что система будет непосредственно делать в процессе своего функционирования. Она является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Проектируемая система представляется в виде ряда прецедентов, предоставляемых системой актерам или сущностям, которые взаимодействуют с системой. Актером или действующим лицом является сущность, взаимодействующая с системой извне (например, человек, техническое устройство). Прецедент служит для описания набора действий, которые система предоставляет актеру.

На основании анализа предметной области в разрабатываемой программно-информационной системе должны быть реализованы следующие функции:

1. Регистрация нового пользователя.
2. Вход пользователя в систему.
3. Просмотр списка выставленных предметов.
4. Поиск предметов.
5. Просмотр подробной информации о товаре.
6. Выставление предмета на продажу/покупку.
7. Управление личным профилем.
8. Отправка сообщений другим пользователям.
9. Получение сообщений от других пользователей.

На рисунке 2.2 в виде диаграммы прецедентов представлены функциональные требования к системе для пользователей.

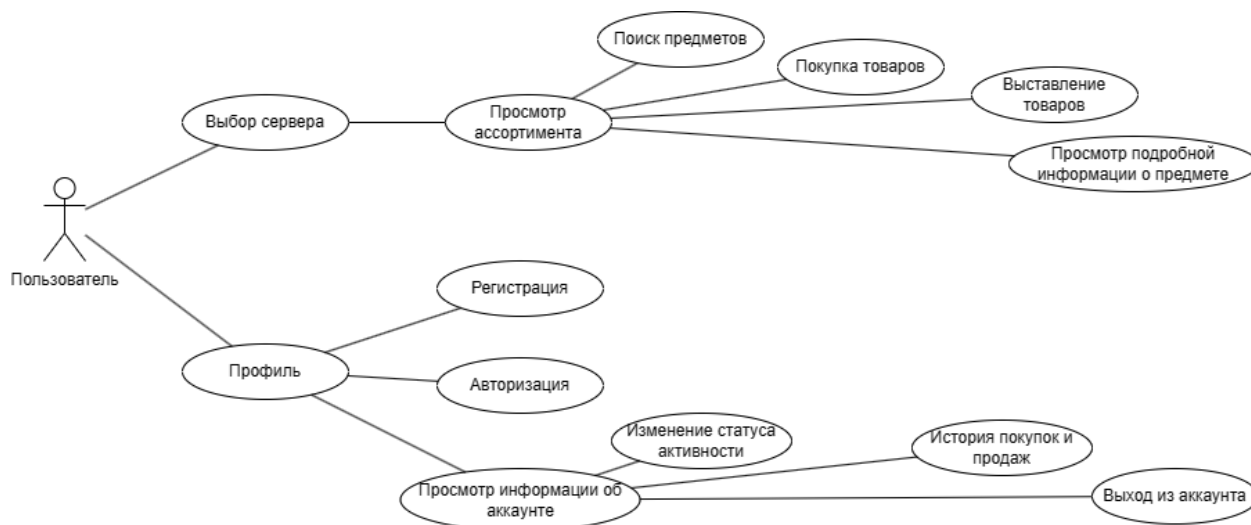


Рисунок 2.2 – Диаграмма прецедентов

2.4.1 Вариант использования «Выбор сервера»

Заинтересованные лица и их требования: пользователь желает начать производить какие-либо действия на сайте, связанные с торговлей. Предусловие: открыта главная страница сайта. Постусловие: открывается нужная

страница в соответствие с выбранным сервером. Основной успешный сценарий:

1. Пользователь нажимает на кнопку выбора сервера на главной странице сайта.
2. На сайте открывается модальное окно с доступными серверами.
3. Пользователь выбирает нужный ему сервер.
4. Приложение открывает страницу, соответствующую выбранному серверу

2.4.2 Вариант использования «Смена сервера»

Заинтересованные лица и их требования: пользователь играет сразу на нескольких серверах и хочет выполнить какие-либо действия, связанные с торговлей, на другом сервере. Предусловие: открыта страница какого-либо сервера. Постусловие: открывается нужная страница в соответствие с выбранным сервером. Основной успешный сценарий:

1. Пользователь нажимает на кнопку «Сменить сервер».
2. На сайте открывается модальное окно с другими серверами.
3. Пользователь выбирает нужный ему сервер.
4. Приложение открывает страницу, соответствующую выбранному серверу

2.4.3 Вариант использования «Поиск»

Заинтересованные лица и их требования: пользователь желает произвести поиск товаров, имеющих в названии одно или несколько ключевых слов. Предусловие: открыта страница какого-либо сервера. Постусловие: лента перезагружается, при наличии товаров, соответствующих поисковому запросу в базе данных, будет отображен их список. Если товары отсутствуют, отображается пустой список. Основной успешный сценарий:

1. Пользователь переходит в режим поиска путем нажатия на элемент интерфейса (значок с изображением лупы) и ввода поискового запроса.
2. Приложение передает поисковой запрос на сервер.

3. Сервер формирует запрос в базу данных и передает результат в приложение.

4. Приложение получает информацию от сервера и производит модификацию списка отображаемых товаров.

2.4.4 Вариант использования «Просмотр информации о товаре»

Заинтересованные лица и их требования: пользователь желает получить больше информации о конкретном товаре. Предусловие: открыта страница какого-либо сервера. Постусловие: появляется модальное окно с полной информацией о товаре. Основной успешный сценарий:

1. Пользователь нажимает на значок с изображением нужного товара.

2. Приложение получает уникальный id предмета и отправляет запрос на сервер.

3. Сервер формирует запрос в базу данных и передает данные в приложение.

4. Приложение получает информацию от сервера и заполняет модальное окно в соответствии с полученной информацией.

2.4.5 Вариант использования «Покупка товара»

Заинтересованные лица и их требования: пользователь увидел интересующий его товар и захотел его приобрести. Предусловие: открыта страница какого-либо сервера. Постусловие: появляется уникальное сообщение, которое нужно скопировать и вставить в игру. Основной успешный сценарий:

1. Пользователь нажимает на кнопку «Купить».

2. Приложение генерирует уникальное сообщение в соответствии с названием товаров и игровым никнеймом пользователя.

2.4.6 Вариант использования «Регистрация»

Заинтересованные лица и их требования: пользователь хочет создать свой аккаунт на сайте. Предусловие: пользователь еще не авторизовался на

сайте. Постусловие: регистрация произведена, появляется значек с изображение профиля. Основной успешный сценарий:

1. Пользователь нажимает на кнопку «Регистрация».
2. Открывается экран регистрации пользователя.
3. Пользователь вводит почту, логин, пароль и повторный пароль.
4. Приложение отправляет запрос на сервер.
5. Сервер формирует запрос в базу данных на создание аккаунта.
6. Если аккаунта с таким же логином или почтой не существует, создается новый аккаунт.
7. Сервер передает информацию в приложение о результате создания аккаунта.
8. Приложение открывает окно пользователю с результатами регистрации.

2.4.7 Вариант использования «Авторизация»

Заинтересованные лица и их требования: пользователь хочет войти в свой аккаунт на сайте. Предусловие: пользователь еще не авторизовался на сайте. Постусловие: авторизация произведена, появляется значок с изображение профиля. Основной успешный сценарий:

1. Пользователь нажимает на кнопку «Авторизация».
2. Открывается экран регистрации пользователя.
3. Пользователь вводит почту, логин, пароль и повторный пароль.
4. Приложение отправляет запрос на сервер.
5. Сервер формирует запрос в базу данных на создание аккаунта.
6. Если аккаунта с таким же логином или почтой не существует, создается новый аккаунт.
7. Сервер передает информацию в приложение о результате создания аккаунта.
8. Приложение открывает окно пользователю с результатами регистрации.

2.4.8 Вариант использования «Выставление товара»

Заинтересованные лица и их требования: пользователь хочет продать какую-либо вещь. Предусловие: пользователь выбрал сервер и авторизовался. Постусловие: предмет добавляется в ленту товаров:

1. Пользователь нажимает на элемент интерфейса (значок с изображением плюса).
2. Приложение открывает модальное окно с полями ввода.
3. Пользователь вводит название предмета полностью или его часть.
4. Приложение выводит все совпадения.
5. Пользователь выбирает нужный ему предмет.
6. Приложение отправляет запрос на сервер.
7. Сервер формирует запрос в базу данных на получения все характеристик предмета.
8. Сервер передает данные в приложение.
9. Приложение заполняет необходимые данные в соответствии с данными.
10. Пользователь вводит недостающие данные и выставляет предмет на продажу.
11. Приложение отправляет запрос на сервер.
12. Сервер вносит данные в базу данных.
13. Сервер передает информацию о результате приложению.
14. Приложение оповещает пользователя об успехе выставления предмета на продажу

2.5 Требования к оформлению документации

Разработка программной документации и программного изделия должна производиться согласно ГОСТ 19.102-77 и ГОСТ 34.601-90. Единая система программной документации.

3 Технический проект

3.1 Общая характеристика организации решения задачи

Необходимо разработать веб-платформу, специально адаптированную для упрощения процесса обмена внутриигровыми ценностями в игре "Stay Out".

Веб-сайт представляет собой комплекс электронных страниц, объединенных в логическую структуру и содержащих разнообразную информацию, включая текст, графику и мультимедийные элементы. Этот онлайн-ресурс доступен через Интернет по определенному доменному имени, например, www.stayoutexchange.com. Каждая страница веб-сайта представляет собой HTML-документ, который обеспечивает визуальное и функциональное взаимодействие с пользователем при помощи языков программирования, таких как HTML, CSS и JavaScript.

Разрабатываемый веб-сайт будет предоставлять пользователям возможность создания учетных записей, выставления и поиска внутриигровых ценностей, а также обмена ими с другими игроками. Он будет обеспечивать удобный интерфейс для взаимодействия между игроками, обеспечивая прозрачность и безопасность сделок.

3.2 Обоснование выбора технологии проектирования

Используемые для создания программно-информационной системы языки и технологии отвечают современным практикам разработки, позволяют достичь высокой производительности и отказоустойчивости программы.

3.2.1 Описание используемых технологий и языков программирования

В процессе разработки web-сайта используются программные средства и языки программирования. Каждое программное средство и каждый язык программирования применяется для круга задач, при решении которых они необходимы.

3.2.2 Язык программирования JavaScript

JavaScript - это интерпретируемый, объектно-ориентированный язык программирования, который широко используется для создания интерактивных и динамических веб-сайтов. Благодаря своей многофункциональности, JavaScript является одним из самых популярных языков программирования в мире веб-разработки.

В отличие от многих других языков программирования, JavaScript выполняется непосредственно в web-браузере пользователя. Это позволяет ему взаимодействовать с элементами HTML страницы, управлять их содержанием, изменять стили, реагировать на пользовательские действия и многое другое, что делает веб-страницы более интерактивными и динамичными.

JavaScript широко используется для разработки различных веб-приложений, таких как интерфейсы пользователя, игры, анимации, формы обратной связи, веб-сервисы и многое другое. Благодаря своей популярности и гибкости, JavaScript также стал широко используемым для разработки серверной стороны приложений с использованием платформы Node.js.

Язык JavaScript имеет динамическую типизацию данных, что означает, что переменные не требуют явного объявления типа данных, и их тип может изменяться в процессе выполнения программы. Это делает JavaScript гибким и удобным для быстрой разработки прототипов и простых скриптов.

Важно отметить, что JavaScript не следует путать с языком Java, несмотря на сходство в названиях. JavaScript и Java - это разные языки программирования с разными синтаксисами, целями и областями применения. JavaScript был создан компанией Netscape в начале 1990-х годов и был изначально разработан для управления взаимодействием с пользователем на веб-страницах.

3.2.2.1 Достоинства языка JavaScript

JavaScript чаще всего используется для написания сценариев работы с веб-страницами в web-браузерах, обеспечивая программирование на сто-

роне клиента. Этот язык предоставляет возможность доступа к элементам разметки web-страницы посредством объектов, что значительно упрощает манипуляции с разметкой. Для решения сложностей с интерпретацией кода различными браузерами применяется библиотека jQuery, которая расширяет возможности JavaScript и унифицирует поведение кода между разными средами.

3.2.2.2 Недостатки языка Javascript

JavaScript обладает некоторыми особенностями, например, непредсказуемость операций сравнения типов данных, что может привести к неожиданным результатам при выполнении математических операций или сравнении значений. Это создает потенциальные сложности для разработчиков, так как различные web-браузеры могут по-разному интерпретировать сценарии JavaScript. Неверно написанный код JavaScript может привести к труднообнаруживаемым ошибкам, и их отладка может быть затруднительной из-за специфики языка.

3.3 Диаграмма компонентов и схема обмена данными между файлами компонента

Диаграмма компонентов является важным инструментом для описания физического представления разрабатываемой системы. Она позволяет разработчикам определить архитектуру системы, выявив взаимосвязи и зависимости между программными компонентами, включая как исходный, так и исполняемый код.

Основными элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними. Компоненты представляют собой логические единицы функциональности системы, которые могут быть как отдельными приложениями, так и более крупными модулями или сервисами.

Интерфейсы определяют способы взаимодействия между компонентами, задавая набор методов, сообщений или протоколов, которыми компоненты обмениваются информацией.

Зависимости указывают на связи между компонентами, определяя порядок и направление потока данных или контроля между ними.

На рисунке 3.1 представлена диаграмма компонентов для проектируемой системы. В центре диаграммы находится сервер с операционной системой, на которой установлена система управления содержимым. Эта система включает в себя базу данных и интерфейс для взаимодействия с клиентскими приложениями.

Помимо этого на диаграмме изображен клиентский компьютер с операционной системой, на которой установлен браузер. Этот компьютер предоставляет интерфейс для взаимодействия пользователя с системой через web-интерфейс.

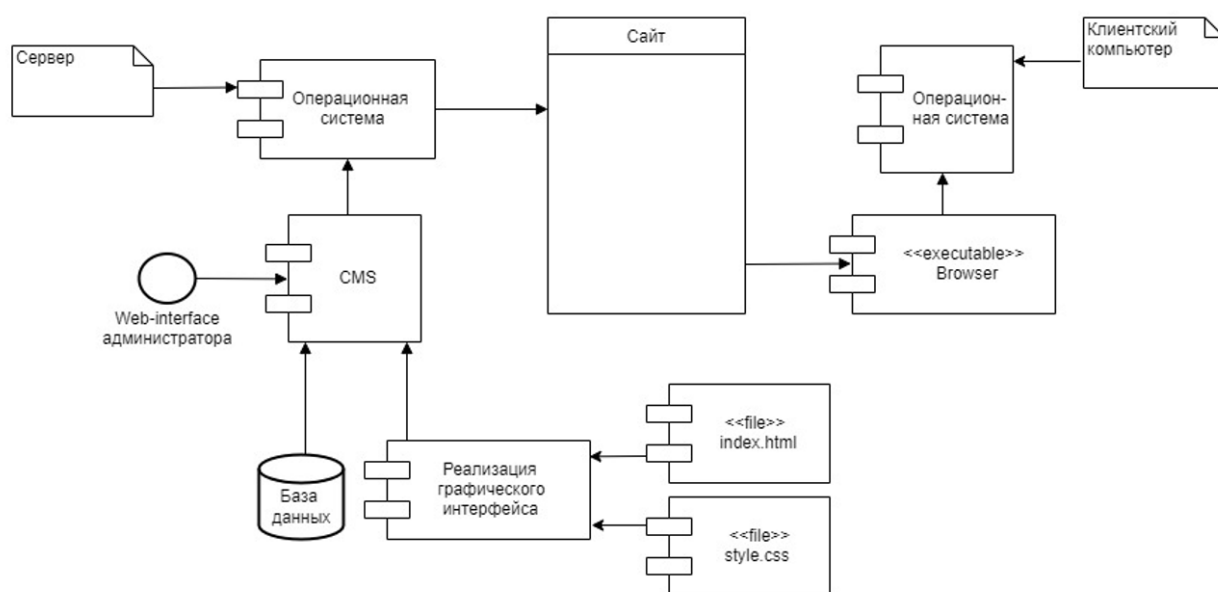


Рисунок 3.1 – Диаграмма компонентов

Любой компонент должен быть вызван в сценарии страницы web-сайта. Web-страница передает данные компоненту в момент вызова последнего.

При вызове компонента в сценарии web-страницы передаются значения параметров компонента, которые далее передаются в сценарий файла `component.php` через массив `$arParams`.

В сценарии файла `component.php` происходит вызов одного из шаблонов компонента с помощью метода `IncludeComponentTemplate` класса `CBitrixComponent`. Id шаблона также определяется в сценарии страницы web-приложения и передается неявно методом `IncludeComponentTemplate`.

После вызова шаблона компонента, в файл `template.php` передается массив `$arParams`, возможно, измененный в сценарии `component.php`, а также массив `$arResult`, сформированный в самом сценарии `component.php`. Оба этих массива доступны также и в файле `result_modifier.php`, который подключается перед `template.php`.

Работа компонента завершается после выполнения сценария файла `component.php`. Однако, если массив `$arResult` будет изменен в сценарии шаблона, изменения не будут переданы обратно в сценарий файла компонента `component.php`.

3.4 Диаграмма размещения

Диаграмма размещения (рис. 3.2) отражает физические взаимосвязи между программными и аппаратными компонентами системы.

Она является хорошим средством для показа маршрутов перемещения объектов и компонентов в распределенной системе.

В таблице 3.1 приведен пример использования пакета `xltabular` с автоматическим расчетом ширины столбца.

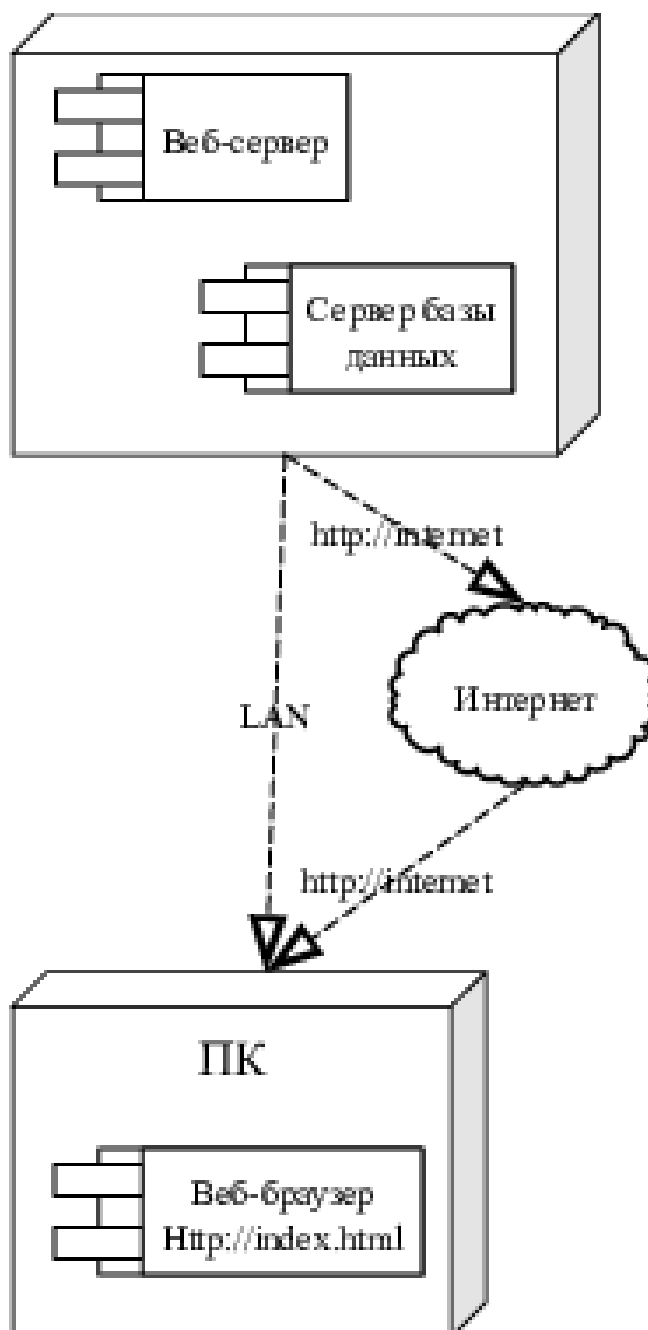


Рисунок 3.2 – Диаграмма размещения. Не помещается на страницу. Очень длинный заголовок

Таблица 3.1 – Сравнение протоколов SSE и WebSocket

	SSE	WebSocket
Направленность	Однонаправленный, полудуплексный: данные посылает только сервер	Двунаправленный, полнодуплексный: и сервер, и клиент могут обмениваться сообщениями
Соединение	HTTP	WS

Продолжение таблицы 3.1

	SSE	WebSocket
Тип данных	Только текст	Бинарные и текстовые данные
Доп. возможности	Встроенный механизм идентификаторов событий и перепоключения	Перепоключение и идентификация события реализуются на стороне приложения

3.5 Содержание информационных блоков. Основные сущности

Проанализировав требования, можно выделить шесть основных сущностей:

- «Пользователь»;
- «Предметы»;
- «Услуги».

В состав сущности «Пользователь» можно включить атрибуты, представленные в таблице 3.3.

Таблица 3.3 – Атрибуты сущности «Пользователь»

Поле	Тип	Обязательное	Описание
1	2	3	4
id	ObjectId	true	Уникальный идентификатор
email	String	true	Почта
login	String	true	Псевдоним
password	String	true	Пароль
role	String	true	Привилегия
status	String	true	Статус

В состав сущности «Пользователь» можно включить атрибуты, представленные в таблице 3.17.

Таблица 3.5 – Атрибуты сущности «Предметы» с использованием различных типов столбцов и многострочным заголовком

Поле	Тип	Обязательное	Описание
1	2	3	4
id	ObjectId	true	Уникальный идентификатор
name	String	true	Название
description	String	false	Описание
img	String	true	Путь до изображения
weight	String	true	Вес
level	String	true	Уровень

В состав сущности «Тип» можно включить атрибуты, представленные в таблице 3.17.

Таблица 3.7 – Атрибуты сущности «Тип» с использованием различных типов столбцов и многострочным заголовком

Поле	Тип	Обязательное	Описание
1	2	3	4
id	ObjectId	true	Уникальный идентификатор
name	String	true	Название
img	String	true	Путь до изображения

В состав сущности «Класс» можно включить атрибуты, представленные в таблице 3.17.

Таблица 3.9 – Атрибуты сущности «Класс» с использованием различных типов столбцов и многострочным заголовком

Поле	Тип	Обязательное	Описание
1	2	3	4
id	ObjectId	true	Уникальный идентификатор
name	String	true	Название
img	String	true	Путь до изображения

В состав сущности «Экипировка» можно включить атрибуты, представленные в таблице 3.17.

Таблица 3.11 – Атрибуты сущности «Экипировка» с использованием различных типов столбцов и многострочным заголовком

Поле	Тип	Обязательное	Описание
1	2	3	4
id	ObjectId	true	Уникальный идентификатор
lifting_capacity	String	true	Грузоподъемность
melee	String	true	Рукопашная
explosion	String	true	Взрыв
electric	String	true	Электрическое
infrared	String	true	Инфракрасное
radiation	String	true	Радиация
biological	String	true	Биологическое
frostbite	String	true	Обморожение
state	String	true	Состояние

В состав сущности «Патроны» можно включить атрибуты, представленные в таблице 3.17.

Таблица 3.13 – Атрибуты сущности «Патроны» с использованием различных типов столбцов и многострочным заголовком

Поле	Тип	Обязательное	Описание
1	2	3	4
id	ObjectId	true	Уникальный идентификатор
type_ammo	String	true	Тип
breaking_throught	String	true	Пробитие
damage	String	true	Урон

В состав сущности «Оружие» можно включить атрибуты, представленные в таблице 3.17.

Таблица 3.15 – Атрибуты сущности «Оружие» с использованием различных типов столбцов и многострочным заголовком

Поле	Тип	Обязательное	Описание
1	2	3	4
id	ObjectId	true	Уникальный идентификатор
moa	String	true	Угловая минута
rate_of_fire	String	true	Темп
breaking_throught	String	true	Пробитие
strength	String	true	Прочность
recoil	String	true	Отдача
handing	String	true	Качание
rocw	String	true	Отказ от состояния
no_pollutin	String	true	Отказ от загрязнения
type_ammo	String	true	Тип патрона

В состав сущности «Гранаты» можно включить атрибуты, представленные в таблице 3.17.

Таблица 3.17 – Атрибуты сущности «Гранаты» с использованием различных типов столбцов и многострочным заголовком

Поле	Тип	Обязательное	Описание
1	2	3	4
id	ObjectId	true	Уникальный идентификатор
number_of_fragments	String	true	Количество осколков
shock_wave_radius	String	true	Радиус взрыва

В системе предусмотрен внутренний механизм связи между разделами и элементами информационных блоков, что обеспечивает эффективную организацию данных без необходимости введения дополнительных идентификаторов при реализации связей между сущностями.

Каждая сущность в системе представлена экземпляром информационного блока, который содержит элементы, отражающие конкретные атрибуты или характеристики данной сущности. Поля и свойства элементов информационных блоков служат для описания атрибутов сущности.

Благодаря этому подходу, данные о сущностях хранятся и управляются структурированно и легко доступны для обработки. Экземпляры сущностей, представленные элементами информационных блоков, обеспечивают гибкость в управлении данными, позволяя эффективно адаптировать информационную структуру системы к различным потребностям и изменениям в бизнес-логике.

Такой подход минимизирует необходимость введения дополнительных идентификаторов или средств для организации связей между сущностями, упрощая процесс разработки и поддержки системы. Кроме того, он способствует повышению производительности и надежности обработки данных,

так как предусматривает оптимальное использование внутренних механизмов управления информацией.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Даффи, К. JavaScript и jQuery: интерактивная веб-разработка / К. Даффи. - Москва : ДМК Пресс, 2017. - 704 с. - ISBN 978-5-97060-826-3. - Текст: непосредственный.
2. Макфарланд, Д. JavaScript. Подробное руководство / Д. Макфарланд. - Санкт-Петербург : Питер, 2018. - 736 с. - ISBN 978-5-496-03829-6. - Текст: непосредственный.
3. Вангельдер, Д. PHP и MySQL для начинающих / Д. Вангельдер. - Киев : Диалектика, 2019. - 416 с. - ISBN 978-617-7597-57-3. - Текст: непосредственный.
4. Лерман, К. PHP и MySQL для начинающих / К. Лерман. - Санкт-Петербург : БХВ-Петербург, 2020. - 416 с. - ISBN 978-5-9775-4974-1. - Текст: непосредственный.
5. Тейлор, Д. Веб-разработка с применением PHP и MySQL / Д. Тейлор. - Москва : ДМК Пресс, 2021. - 368 с. - ISBN 978-5-97060-826-3. - Текст: непосредственный.
6. Мейерс, Э. MySQL для профессионалов / Э. Мейерс. - Москва : ДМК Пресс, 2017. - 736 с. - ISBN 978-5-97060-826-3. - Текст: непосредственный.
7. Бакланов, Г. Основы SQL. Практическое руководство / Г. Бакланов. - Санкт-Петербург : БХВ-Петербург, 2018. - 320 с. - ISBN 978-5-9775-3683-3. - Текст: непосредственный.
8. Диксон, Д. Изучаем SQL. Мастер-класс / Д. Диксон. - Москва : ДМК Пресс, 2019. - 432 с. - ISBN 978-5-97060-826-3. - Текст: непосредственный.
9. Фланаган, Д. JavaScript. Подробное руководство / Д. Фланаган. - Москва : ДМК Пресс, 2017. - 832 с. - ISBN 978-5-97060-826-3. - Текст: непосредственный.
10. Либерти, Л. Введение в MySQL / Л. Либерти. - Санкт-Петербург : Питер, 2018. - 320 с. - ISBN 978-5-4461-1064-1. - Текст: непосредственный.

11. Уилсон, К. Программирование на JavaScript. Самоучитель / К. Уилсон. - Москва : ДМК Пресс, 2019. - 448 с. - ISBN 978-5-97060-826-3. - Текст: непосредственный.

12. Петров, А. Веб-разработка с использованием PHP и MySQL / А. Петров. - Санкт-Петербург : Питер, 2020. - 416 с. - ISBN 978-5-4461-1064-1. - Текст: непосредственный.

13. Рейнхардт, О. JavaScript для начинающих / О. Рейнхардт. - Москва : ДМК Пресс, 2021. - 352 с. - ISBN 978-5-97060-826-3. - Текст: непосредственный.