МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

**Объектно-ориентированный подход к проектированию программного обеспечения**

Выполнил студент группы ИВТ-31 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Долженкова М.Л./

Киров 2021

1. Диаграммы моделей взаимодействия

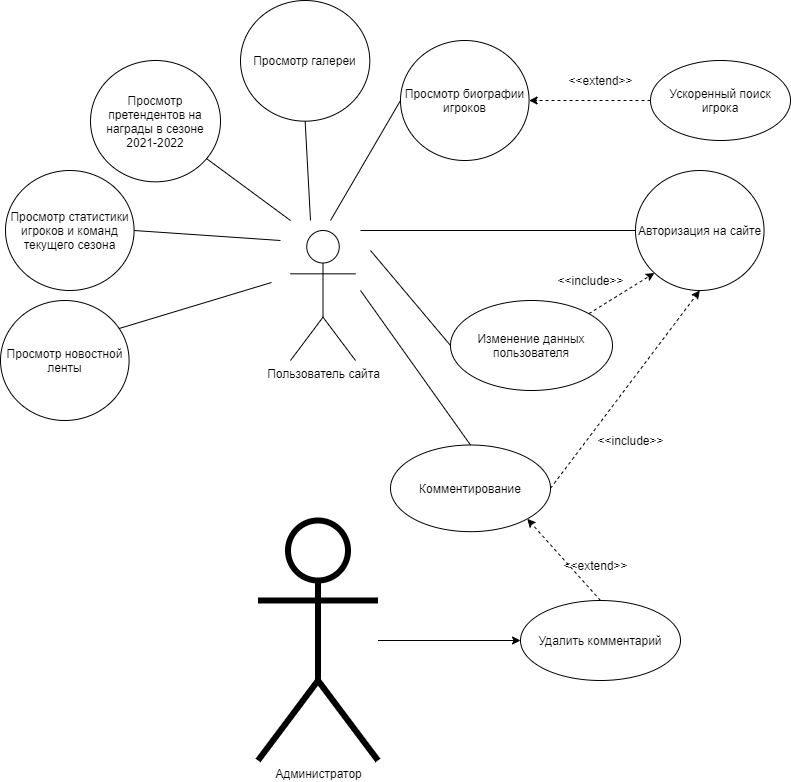


Рисунок 1. Диаграмма «use case»

* **Диаграмма «use case»** отображает отношения между актерами и вариантами использования. В диаграмме для данного проекта отображены возможности, которыми обладает как простой пользователь, так и администратор сайта. Администратор имеет возможность удаления негативных и неприемлемых комментариев. Также показана взаимосвязь таких действий, как комментирование, и его невозможность без авторизации. Благодаря меткам можно увидеть дополнительные возможности определённых страниц сайта и возможность их использования человеком.

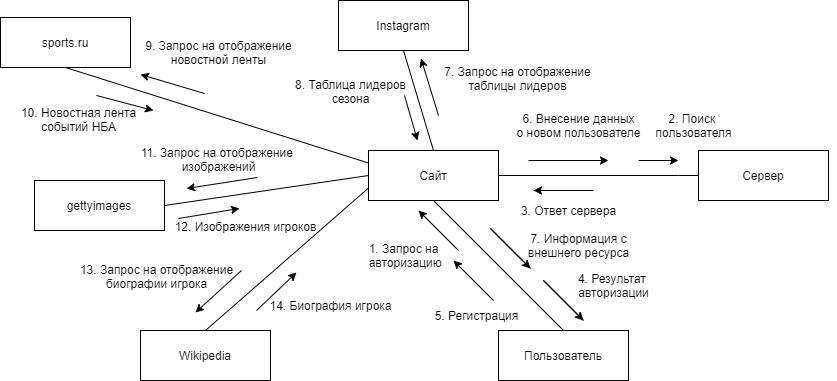


Рисунок 2. Диаграмма кооперации

* **Диаграмма кооперации** указывает обмен сообщениями между объектами при реализации любого варианта использования. При помощи данной диаграммы можно легко отследить, что происходит при выполнении каких-либо действий на сайте. Отслеживается взаимосвязь между пользователем, сервером и всеми внешними ресурсами, задействованными на сайте. Так взаимосвязь сайта и gettyimages показывает, что будет заимствовано с внешнего ресурса для отображения на сайте. При помощи взаимосвязи сайт-сервер можно увидеть, что сайте передаёт серверу и наоборот.

1. Диаграммы моделей поведения

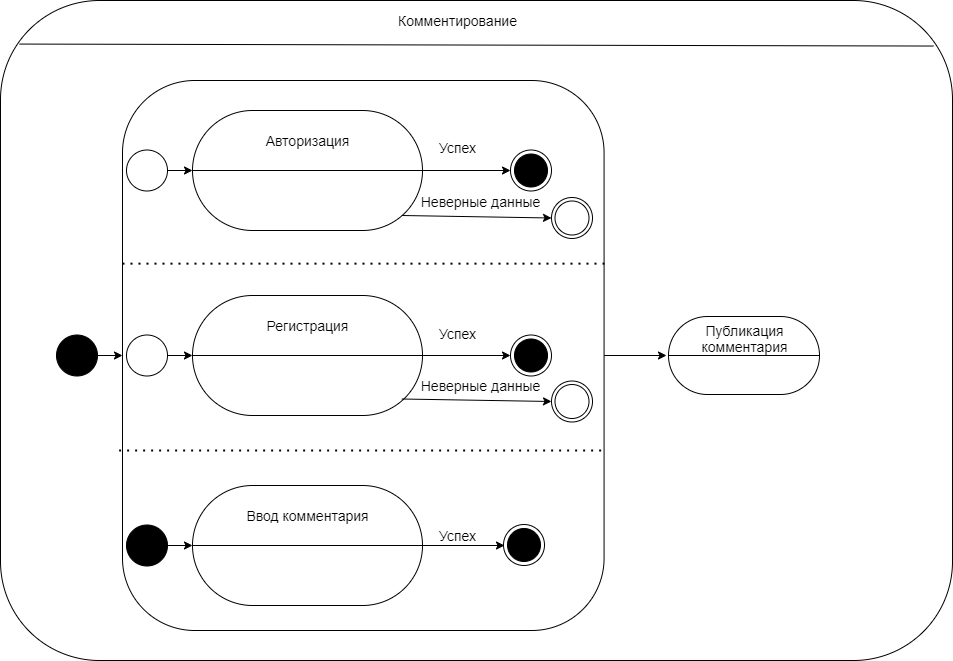


Рисунок 3. Диаграмма состояний входа на сайт

* **Диаграмма состояний** показывает, как объект переходит из одного состояний в другое. Она служит для моделирования сложных динамических аспектов системы. В качестве такого аспекта на диаграмме изображён процесс написания и публикации пользователем комментария. Здесь можно видеть составное состояние, включающее другое состояние, одно из которых содержит также параллельные подсостояния. Так при помощи данной диаграммы можно увидеть, что при неудачной авторизации на сайте у пользователя будет 2 варианты: повторить процесс авторизации или зарегистрироваться по-новому. Комментарий могут оставить только авторизированные на сайте пользователи. Лишь после успешного входа или регистрации у человека появляется возможность оставить комментарий на сайте.

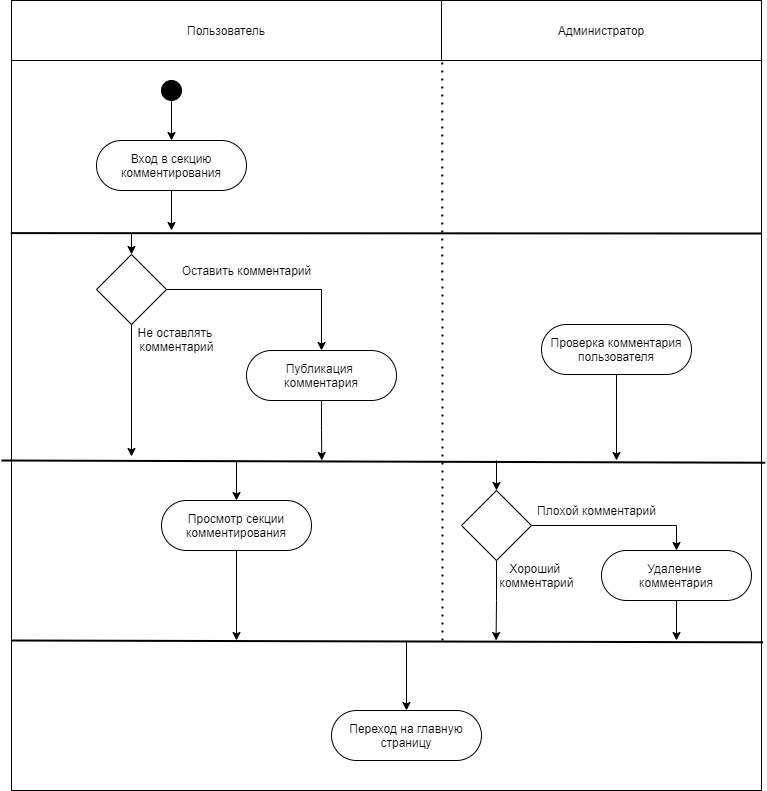


Рисунок 4. Диаграмма деятельности в секции комментирования

* **Диаграмма деятельности** представляет из себя алгоритм каких-либо действий, выполняющихся в системе. В качестве действия было выбрано написание и публикация комментария, а также дальнейшие действия с ними. Благодаря диаграмме можно просмотреть как пользователь и администратор могут взаимодействовать с секцией комментирования. Так для написания комментария необходима авторизация на сайте. Администратор имеет больше возможностей в данной секции чем простой пользователь: он может удалять неприемлемые комментарии или оставлять хорошие.

1. Диаграммы статических моделей

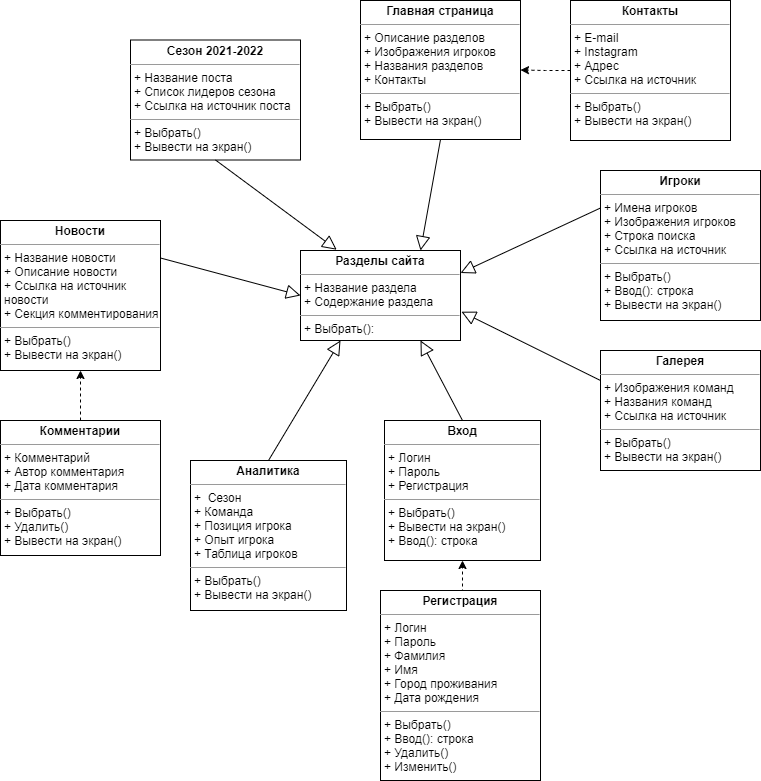


Рисунок 5. Диаграмма классов

* **Диаграмма классов** демонстрирует общую структуру иерархии классов системы, их коопераций, атрибутов, методов, интерфейсов и взаимосвязей между ними. В качестве диаграммы классов было решено описать общую структуру сайта: страницы и их функции. При помощи входных стрелок можно отследить возможности каждой страницы и дополнительные функции, доступные для использования на странице. Так пункт регистрации доступен лишь во вкладке «Вход» или комментирование есть лишь в разделе «Новости».

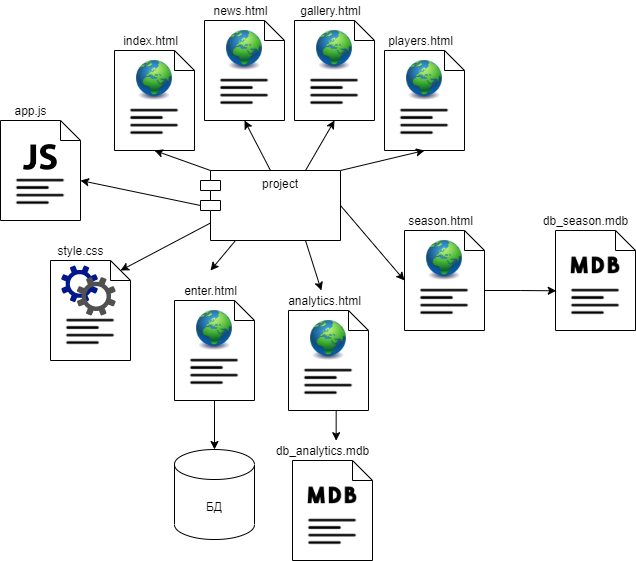


Рисунок 6. Диаграмма реализации

* **Диаграмма реализации** в проекте нужна для визуализации организации компонентов системы и зависимостей между ними. Так в данном проекте project является корневой папкой проекта, в которой располагаются все необходимые файлы. Большую часть сайта занимают html файлы, представляющие страницы главного проекта. База данных пользователей используется при поиске человека на сайте при авторизации. Файлы mdb нужны для отображения и загрузки статических баз данных с игроками и их статистикой. Они доступны во вкладках сезон и аналитика. Также для визуальной составляющей необходимы файлы app.js и style.css, которые задают внешний вид сайта и придают ему динамичность.

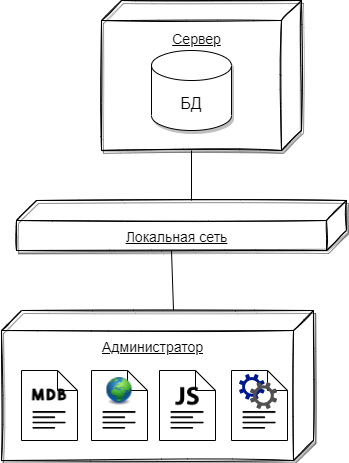


Рисунок 7. Диаграмма размещения

* **Диаграмма размещения** отражает физические взаимосвязи между программными и аппаратными компонентами системы, а также их расположение в системе. В данном проекте в качестве хранилища задействован хост, локальная сеть. В нем хранятся все необходимые для нормального функционирования сайта файлы. В качестве источников данных файлов выступают сервер, который хранит базу данных пользователей, и администратор, который выкладывает в локальную сеть файлы уже самого проекта, описанные в диаграмме компонентов.