

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра Программирования и информационных технологий

*Курсовой проект*

*Веб-приложение для публикации и просмотра новостей  
“Школьная новостная доска”*

*09.03.04 Программная инженерия*

Обучающийся \_\_\_\_\_ *Елфимова Е.В., 3 курс*

Обучающийся \_\_\_\_\_ *Стребкова О.В., 3 курс*

Обучающийся \_\_\_\_\_ *Мещеряков И.Г., 3 курс*

Руководитель \_\_\_\_\_ *А.В. Нужных, преподаватель*

Воронеж 2020

## Содержание

Введение	3
Постановка задачи	4
Анализ предметной области	5
Глоссарий	5
Анализ существующих решений	5
Реализация приложения	26
Обоснование архитектуры	26
Приложение.	28
Приложение 1.	28

## ВВЕДЕНИЕ

На сегодняшний день мы привыкли получать любую информацию буквально в несколько кликов. Мы можем узнать все, от прогноза погоды до свежих новостей из дальних стран. Подобным образом обстоит дело и с информацией которую мы получаем от родных, друзей или коллег, мы используем для этого различные мессенджеры и социальные сети.

Но не везде для получения организационной информации удовлетворяющим вариантом является мессенджер или социальная сеть. Например учащимся школ удобно было бы просматривать организационную информацию от учителей онлайн. Но мессенджер не является удобным вариантом для этой цели, а социальная сеть слишком обладает слишком перегруженным функционалом.

В данной курсовой работе рассматривается проблема создания простого, легковесного и не нагруженного излишней функциональностью, но обладающего всеми необходимыми функциями веб-приложения «Школьная новостная доска».

## 2. Постановка задачи

Цель курсовой работы - работая в команде, реализовать веб-приложение для просмотра, публикации и обсуждения школьных новостей, обладающее простым и интуитивно-понятным интерфейсом.

Приложение должно позволять учителю публиковать новости, редактировать их и удалять по истечению их актуальности. Также учитель может участвовать в обсуждении новости в комментариях. Для ученика приложение должно позволять просматривать новости и комментировать их.

Команда состоит из трех человек, среди которых распределены роли: разработчик, тестировщик, проектный менеджер и бизнес-аналитик. Управление проекта ведется по методологии Kanban. Этот метод разработки позволит каждому члену команды иметь представления о реальных проектах в сфере IT-бизнеса.

### 3. Анализ предметной области

#### 3.1. Глоссарий

В рамках данной системы для пользователя определены следующие роли:

- Администратор - авторизованный пользователь, осуществляющий информационную поддержку сайта от имени Заказчика.
- Учитель - авторизованный пользователь в системе, имеющий права на комментирование опубликованных новостей, публикацию и удалению своих новостей, а также удаление комментариев любых пользователей.
- Учащийся - авторизованный пользователь в системе, наделенный правами комментировать новости на доске.
- Новость - текстовая информация, опубликованная учителем.
- Роль - класс пользователей системы, обладающий определенным набором прав доступа.

Соответствующая действительности информации - информация, опубликованная на доске учителем.

#### 3.2. Анализ существующих решений

Рассмотрим существующие решения.

##### 3.2.1. Сайт школы МБОУ СОШ имени Е.А. Болховитинова.

Ссылка: <https://school-38.ru>

Действующий сайт школы №38. Предоставляет полезную информацию для учащихся школы №38, их родителей, а также поступающим в школу..

##### **Достоинства:**

- Разделение информации по категориям.
- Наличие версии “Для слабовидящих”.
- Возможность обратной связи для родителей.

##### **Недостатки:**

- Отсутствие возможности комментирования новостей учащимися.

### 3.2.2. Дневник. ру - цифровая образовательная платформа.

Дневник.ру - единая электронная образовательная среда для учителей, учеников, родителей и органов государственного управления. Она предоставляет информацию об учебном плане, оценках и рейтинге учащихся. (<https://dnevnik.ru>)

#### **Достоинства:**

- Есть возможность просмотра оценок
- Открыт доступ для родителей учащихся

#### **Недостатки:**

- Перегруженная функциональность

#### **Рассмотрим процесс добавления новости:**

Процесс добавления новости на сайте Дневник.ру проходит за 5 кликов. Для этого необходимо войти в систему, выбрать раздел “Моя школа”, написать текст для новости и выбрать

“Отправить” для публикации новости на доске объявлений.

(<https://github.com/Zeleboba99/school-board/blob/master/Документация/bpmn/Добавление%20новости.pdf>)

#### **Рассмотрим процесс создания нового пользователя:**

“Моя школа” и перейти в раздел “Администрирование школы.” Далее учителю необходимо выбрать “Новый человек” для создания нового пользователя, затем выбрать, кого он хочет добавить: нового пользователя или изменить существующего, затем заполнить 35 текстовых полей с личной информацией пользователя (см. приложение 1), кликнуть на кнопку “Далее”, проверить правильность введенных данных и выбрать кнопку “Создать” для создания нового пользователя.

(<https://github.com/Zeleboba99/school-board/blob/master/Документация/bpmn/Добавление%20пользователя.png>)

### 3.3. Документы, на основании которых создается сайт

Устав муниципального бюджетного общеобразовательного учреждения средняя общеобразовательная школа с углубленным изучением отдельных предметов №38 имени Е. А. Болховитинова.

### 3.4. Цели создания сайта

- Упростить процесс распространения информации для учащихся (для добавления новости в нашем приложении требуется 3 клика вместо 5 на сайте Дневник.ру)
- Избежать распространения не соответствующей действительности информации между учащимися (Система предоставляет доступ к публикации новостей только учителям).
- Экономия времени учителей на распространение информации о предстоящих событиях. Так как максимальное число учащихся в классе равно 25 (Приказ от 30.09.2013 г. № 1015 «Об утверждении порядка организации и осуществления образовательной деятельности по основным общеобразовательным программам — образовательным программам начального общего, основного общего и среднего общего образования»), то чтобы обзвонить весь класс понадобится минимум 75 минут, если принять 3 минуты, как время одного телефонного разговора (без учета поиска номера учащегося и затрат времени на ожидание ответа).

### 3.5. Задачи, решаемые при помощи сайта

Система предназначена для публикации и просмотра новостей о проведении школьных событий. Основное назначение системы - предоставление учащимся школы информации о школьных событиях и

возможность для учителей предоставлять информацию о школьных событиях для учащихся.

### 3.6. UML диаграммы

#### 3.6.1. Диаграммы вариантов использования

Рассмотрим диаграмму вариантов использования на рисунке 1.

##### **Актер - пользователь.**

Действия, которые может осуществлять пользователь (отношения ассоциации):

- Войти.

Пользователь может войти в систему как учитель, ученик или администратор. Рассмотрим действия каждого из них.

##### **Актер - администратор.**

Действия, который может осуществлять администратор (отношения ассоциации):

- Добавить учителя в систему;
- Добавить ученика в систему;
- Обновить информацию об ученике;
- Обновить информацию об учителе;
- Удалить ученика из системы;
- Удалить учителя из системы.

##### **Актер - учитель.**

Действия, который может осуществлять учитель (отношения ассоциации):

- Опубликовать новость на доске (отношение включения - создать новую новость);
- Удалить свои опубликованные новости;
- Удалить любые пользовательские комментарии;



- Просмотреть опубликованные новости (отношение включения - просмотреть опубликованные комментарии);
- Прокомментировать опубликованные новости (отношение включения - просмотреть опубликованные комментарии).

**Актер - ученик.**

Действия, которые может осуществлять учитель(отношения ассоциации):

- Просмотреть опубликованные новости (отношение включения - просмотреть опубликованные комментарии);
- Прокомментировать опубликованные новости (отношение включения - просмотреть опубликованные комментарии);
- Удалить свои комментарии.

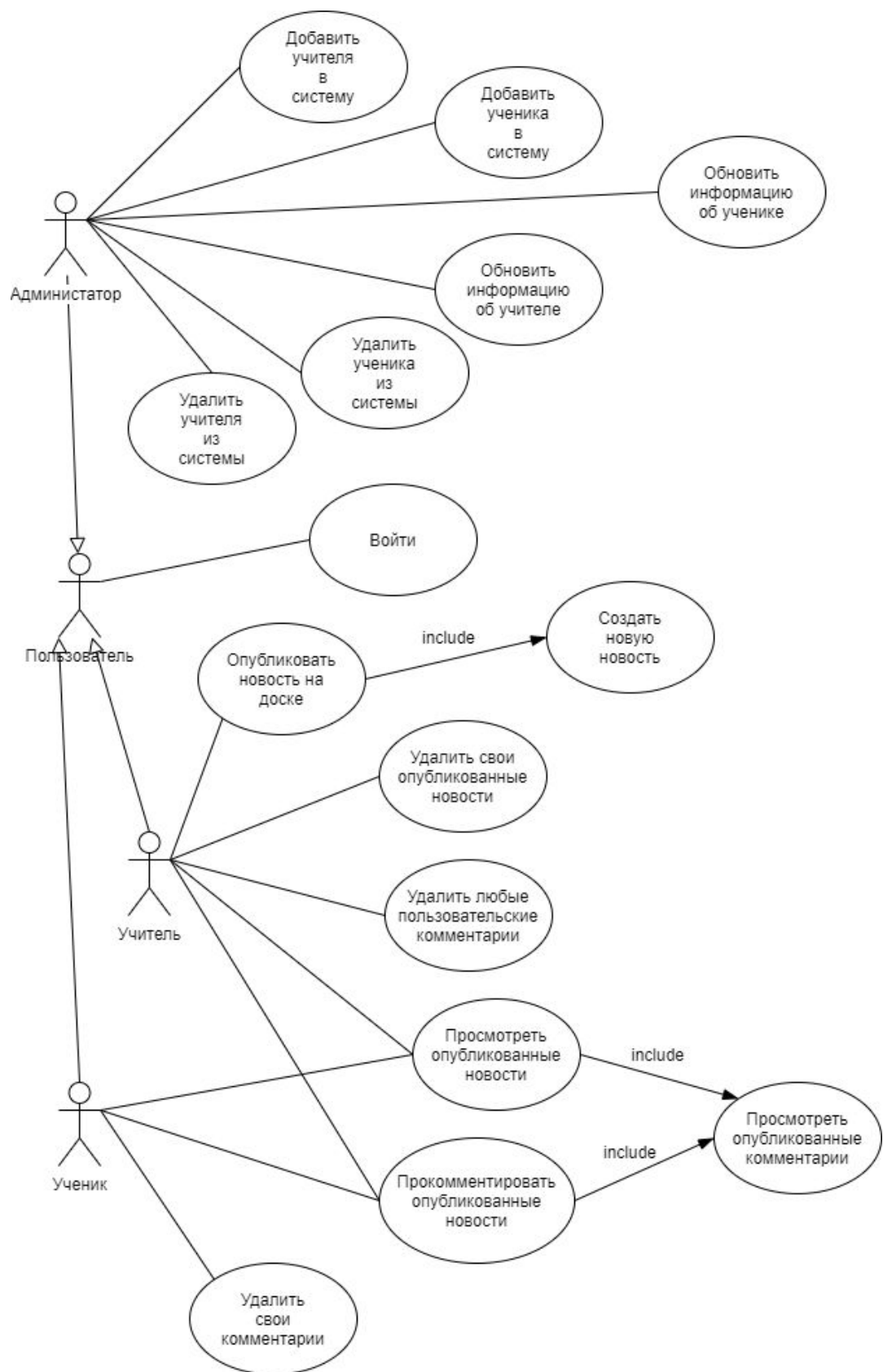


Рисунок1 - Диаграмма вариантов использования

### 3.6.2. Диаграмма последовательности

Диаграмма последовательности для удаления новости - рисунок 2.

Актер - пользователь.

После нажатия на кнопку удаления новости пользователем клиент отображает диалоговое окно для подтверждения удаления. Если пользователя подтверждает удаление, то клиент отправляет запрос об удалении новости на сервер, а сервер, в свою очередь, отправляет запрос в БД об удалении данных. Когда БД возвращает информацию об удалении, сервер возвращает эту информацию клиентской части приложения, которая отображает обновленную страницу новостей.

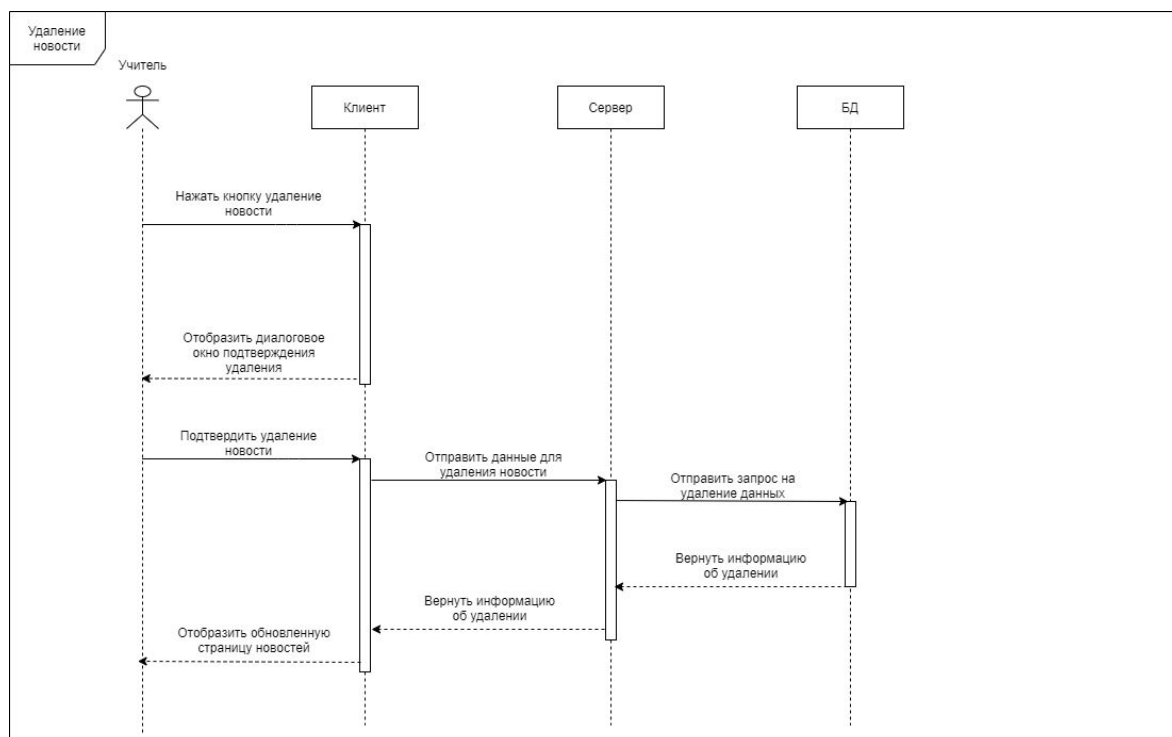


Рисунок 2 - Диаграмма последовательности для удаления новости

Диаграмма последовательности для добавления комментария - рисунок 3.

Актер - учитель или ученик.

После нажатия учителем или учеником кнопки добавления комментария клиент отображает форму добавления комментария. Далее учитель или ученик

вводит комментарий в форму и подтверждает отправку. После этого клиент проверяет введенные данные и отправляет их на сервер, который проверяет права пользователя на добавление комментария, а затем отправляет запрос на добавление данных в БД. Когда БД возвращает информацию о добавлении, сервер возвращает эту информацию на клиентскую часть приложения, которая отображает информацию о добавлении комментария.

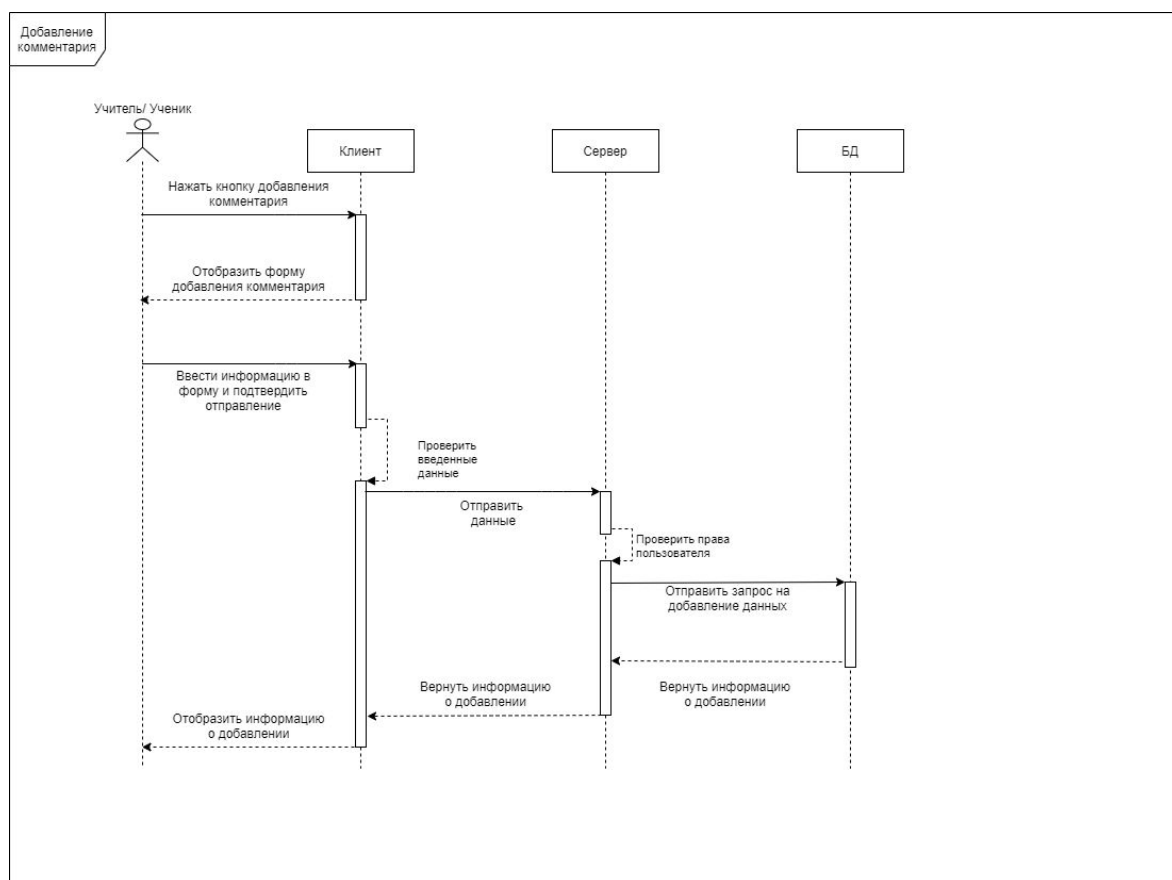


Рисунок 3 - Диаграмма последовательности для добавления комментария

Диаграмма последовательности для добавления новости - рисунок 4.

Актер - учитель.

После нажатия учителем кнопки добавления новости, клиентская часть приложения отображает форму добавления новости. Далее учитель вводит текстовую информацию в форму для добавления новости и подтверждает отправку. Затем клиент проверяет введенные данные и отправляет их на

сервер, который проверяет права пользователя на добавление новости, после чего отправляет запрос на добавление новости в БД. После того, как БД возвращает информацию о добавлении, сервер возвращает эту информацию на клиентскую часть приложения, которая в свою очередь отображает информацию и добавлении.

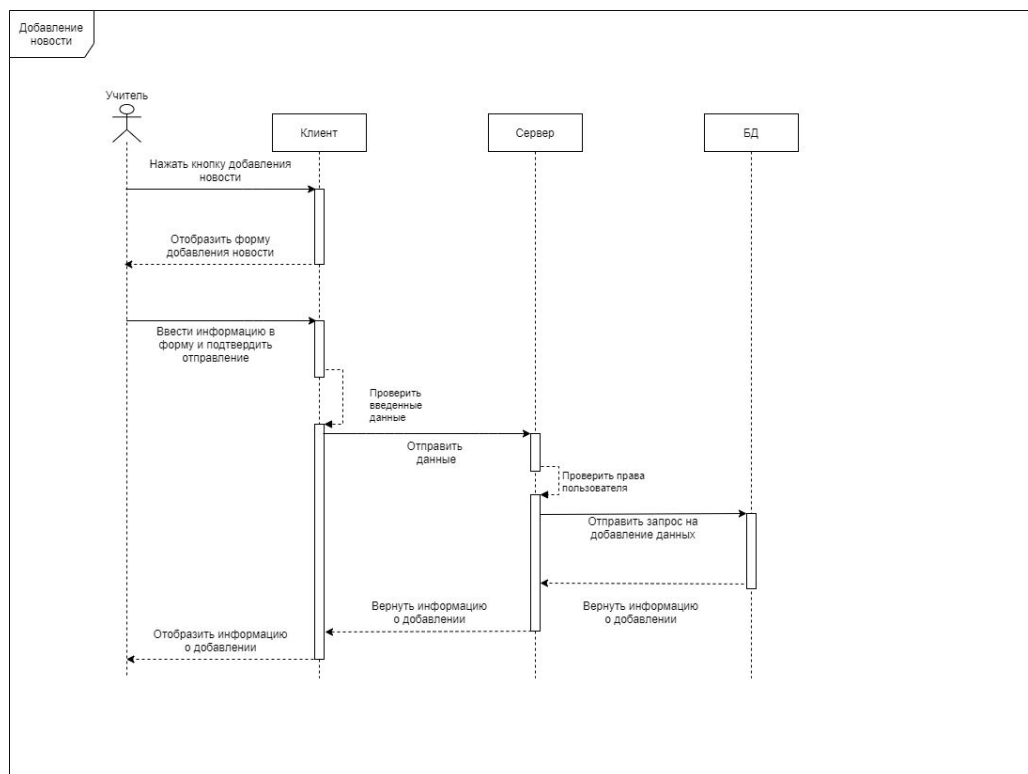


Рисунок 4 - Диаграмма последовательности для добавления новости

Диаграмма последовательности для добавления новости - рисунок 5.

Актер - пользователь.

После ввода пользователем персонального логина и пароля, клиентская часть приложения отправляет данные на сервер, который запрашивает данные о пользователе в БД. Затем БД возвращает данные на сервер, который после создания JWT токена для данного пользователя, возвращает токен и информацию о пользователе на клиентскую часть приложения. Далее клиент запрашивает у сервера данные для главной страницы с JWT токеном в

заголовке, после чего сервер проверяет JWT токен и затем запрашивает данные у БД. Когда БД возвращает эти данные на сервер, он отправляет их на клиентскую часть приложения, которая, в свою очередь, отображает главную страницу приложения.

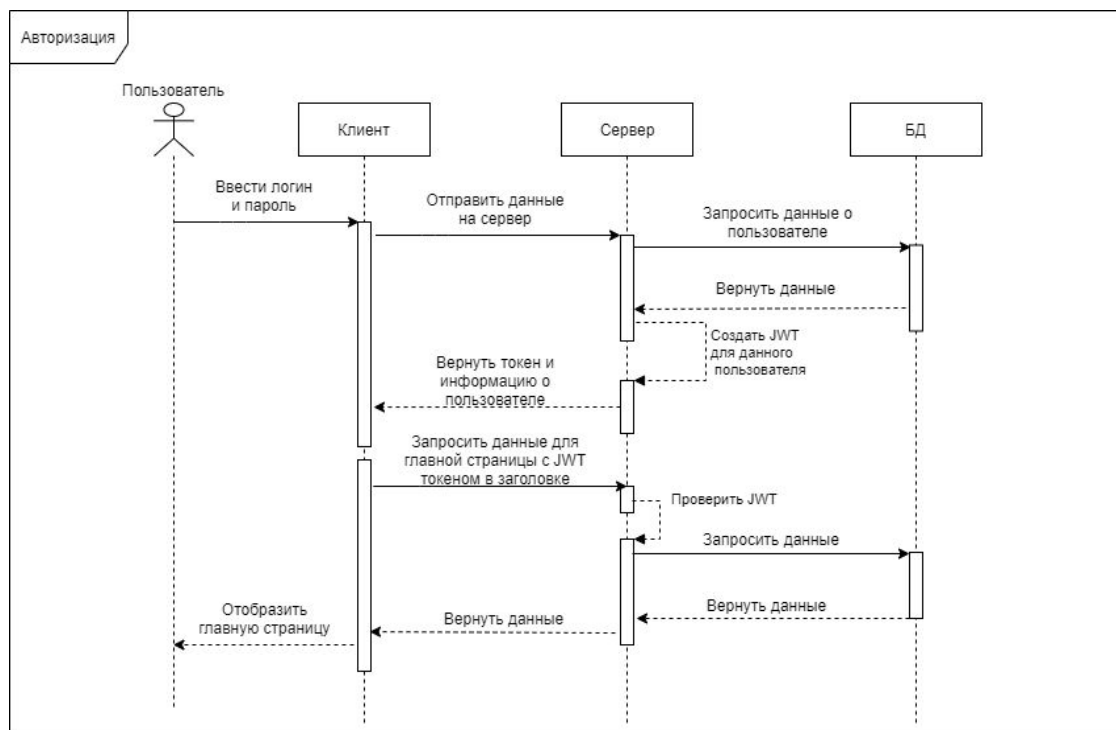


Рисунок 5 - Диаграмма последовательности для авторизации

### 3.6.3. Диаграмма коммуникаций

Диаграмма коммуникаций для авторизации - рисунок 6.

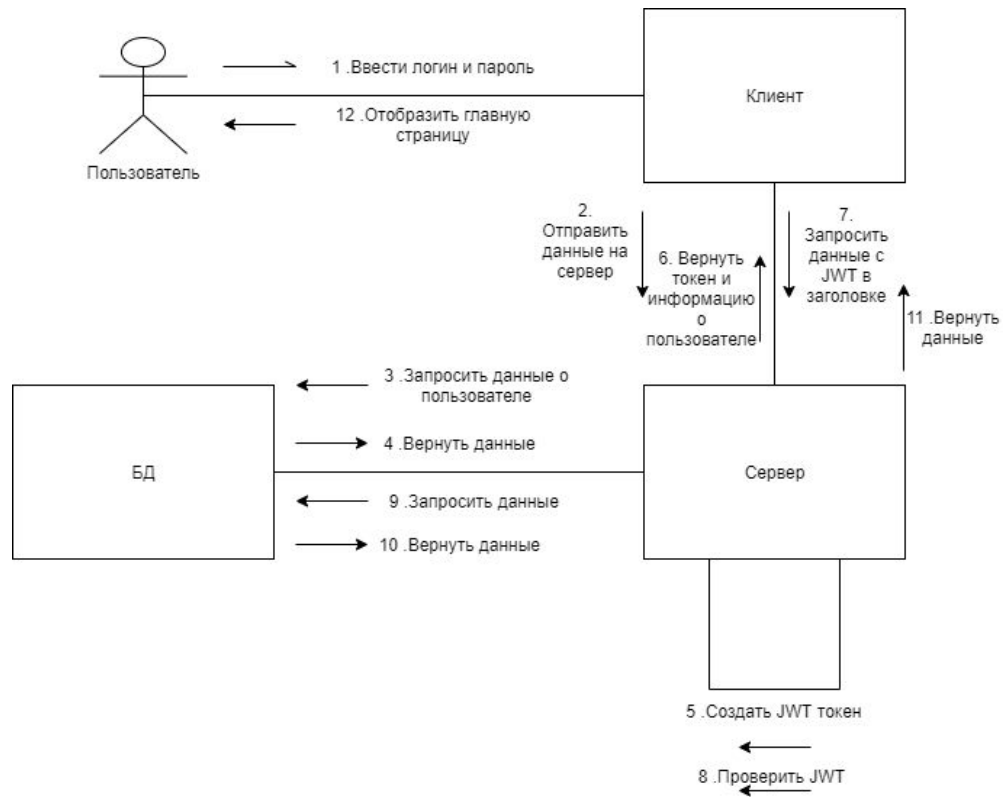


Рисунок 6 - Диаграмма коммуникаций для авторизации

Диаграмма коммуникаций для добавления поста - рисунок 7.

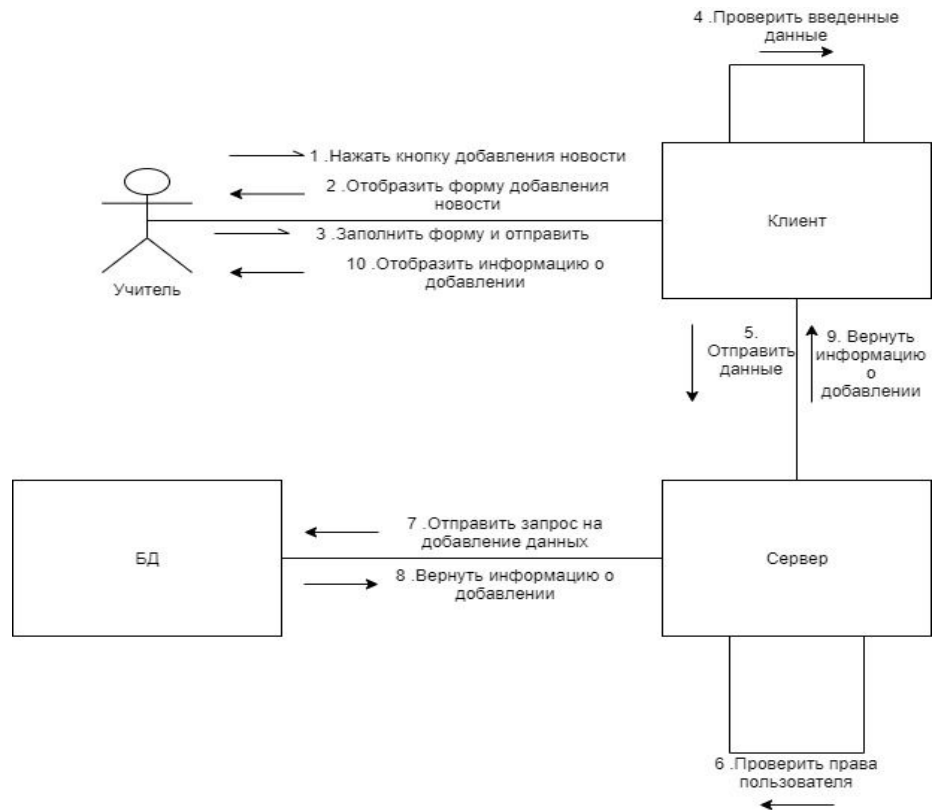


Рисунок7 - Диаграмма коммуникаций для добавления поста



Диаграмма коммуникаций для удаления поста - рисунок 8.



Рисунок 8 - Диаграмма коммуникаций для удаления поста

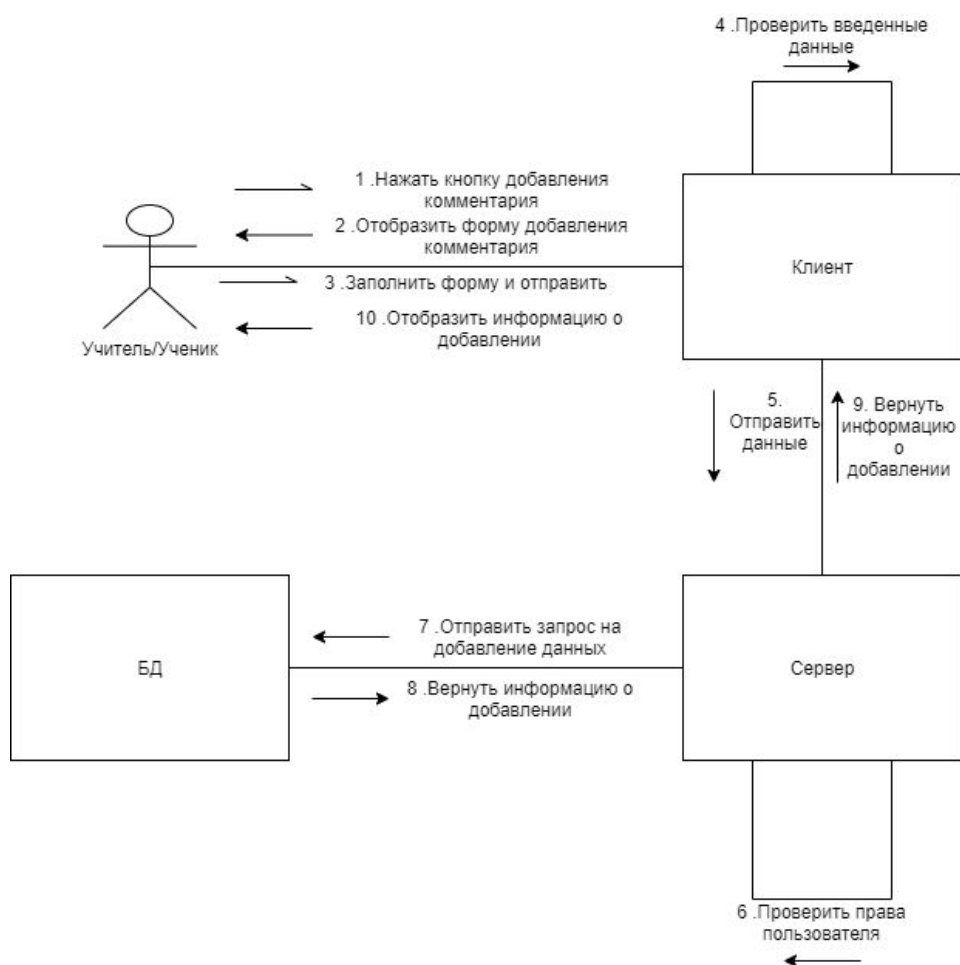


Рисунок 9 - Диаграмма коммуникаций для добавления комментария

Сообщения в диаграмме коммуникаций такие же, что и в диаграмме последовательности, поэтому их повторное описание не приводится.

#### 3.6.4. Диаграмма состояний

Диаграмма состояний для авторизации - рисунок 10.

Диаграмма состояний отражает возможные состояния системы. Когда пользователь зашел в систему, она находится в состоянии ожидания ввода логина и пароля. После ввода пользователем этих данных, система переходит в состояние поиска логина и пароля. В зависимости от исхода этого состояние возможны две основные цепочки состояний.

Если данные, введенные пользователем, являются корректными, то система переходит в состояние отображения главной страницы приложения, в зависимости от роли пользователя.

Если пользователь ввел некорректные данные, система переходит в состояние ожидания ввода логина и пароля.

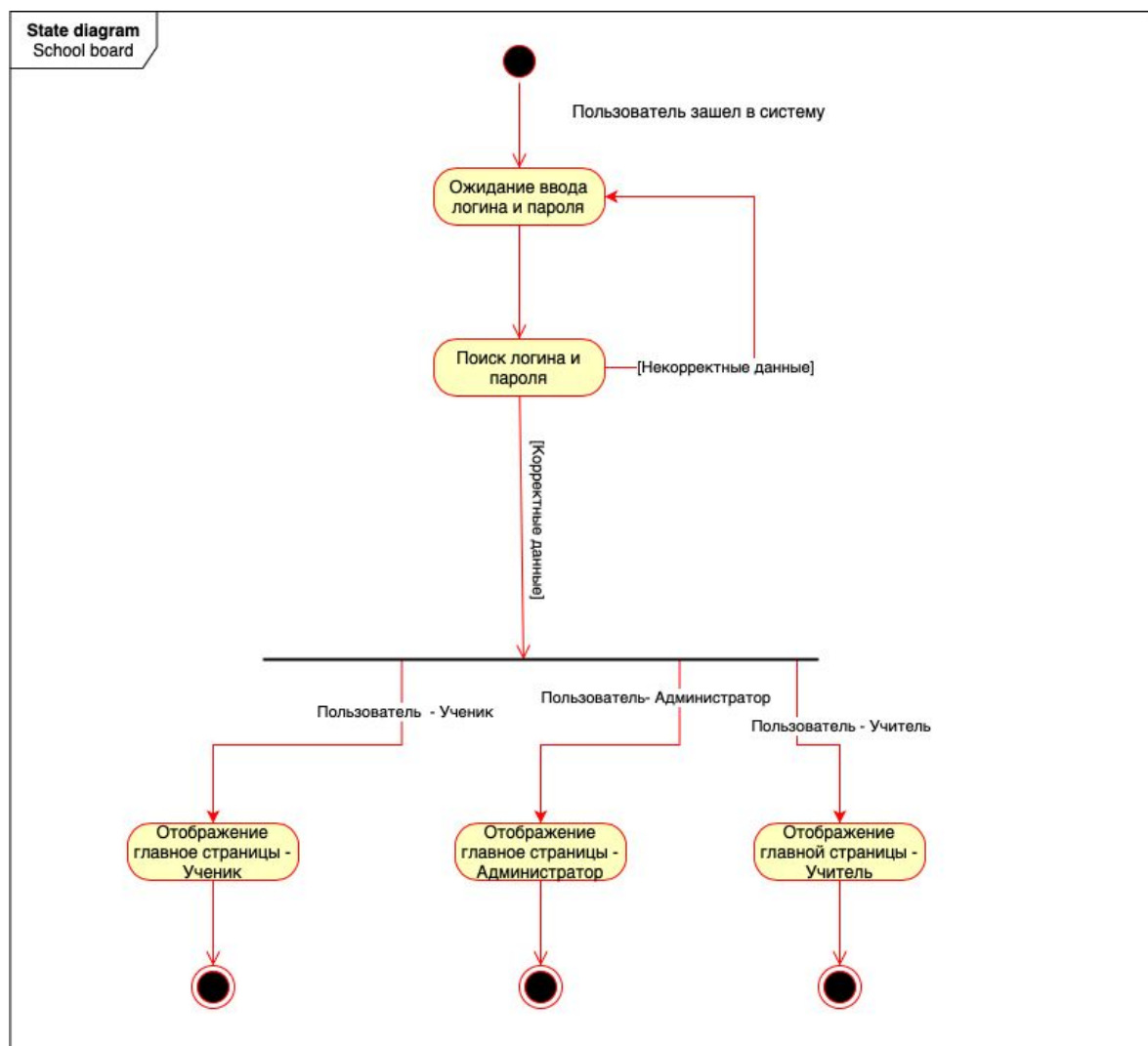


Рисунок 10 - Диаграмма состояний для авторизации пользователя.

### 3.6.5. Диаграмма развертывания

На рисунке 11 представлена диаграмма развертывания, которая отображает взаимосвязь между аппаратными компонентами приложения и их программными компонентами.

Для описываемого приложения, аппаратными компонентами являются компьютер пользователя и сервер. На узле сервера расположены бекэнд и фронтэнд приложения, а также база данных.

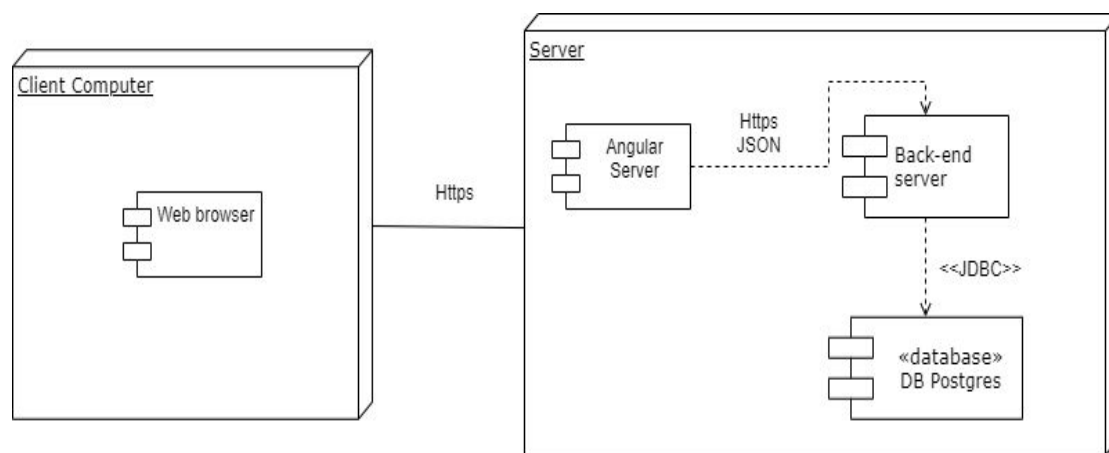


Рисунок 11 - Диаграмма развертывания

### 3.6.6. Диаграмма деятельности

На рисунке 12 представлена диаграмма деятельности для создания нового пользователя администратором.

После того, как страницы входа была открыта, администратор должен ввести свой логин и пароль, после чего система проверяет пароль, если он неверный, то у администратора запрашивается логин и пароль повторно. Если логин и пароль верны, то администратор перенаправляется в личный кабинет.

Далее администратор должен нажать на кнопку для создания нового пользователя. После чего система перенаправляет пользователя на страницу создания нового пользователя, где администратор должен выбрать роль пользователя, ввести имя пользователя, и добавить пароль, который администратор может ввести сам или выбрать создать пароль. Во втором случае система генерирует пароль и возвращает его на страницу создания пользователя. Далее администратор выбирает добавить пользователя, после чего система получает данные нового пользователя, а затем добавляет их. В завершение, система отображает данные нового пользователя в личном кабинете администратора.

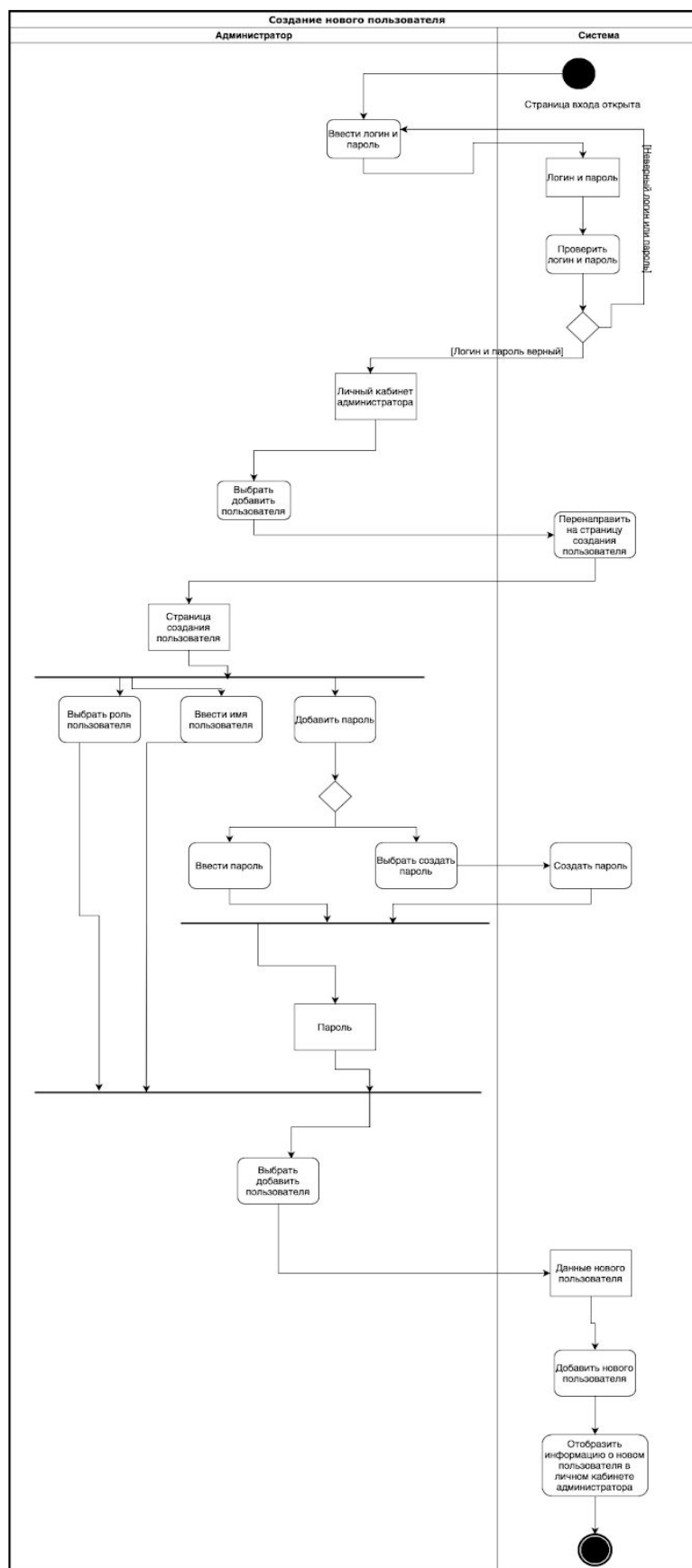


Рисунок 12 - Диаграмма деятельности для добавления нового пользователя.

### 3.6.7. Диаграмма объектов

На рисунке 13 представлена диаграмма объектов.

Пусть в некоторый момент времени в системе существуют следующие объекты:

1. Экземпляры класса “User” :
  - Пользователь “u”;
  - Пользователь “u1”;
  - Пользователь “u2”.
2. Экземпляры класса “Role”:
  - Роль “r1”, имеющая связь с пользователем “u”;
  - Роль “r2” имеющая связь с пользователем “u1” и “u2”.
3. Экземпляр класса “Post”:
  - Новость “p1”, имеющая связь с пользователем “u”.
4. Экземпляры класса “Comment”:
  - Комментарий “c1”, имеющий связь с постом “p1” и пользователем “u2”;
  - Комментарий “c2”, имеющий связь с постом “p1” и пользователем “u1”;
  - Комментарий “c3”, имеющий связь с постом “p1” и пользователем “u2”.

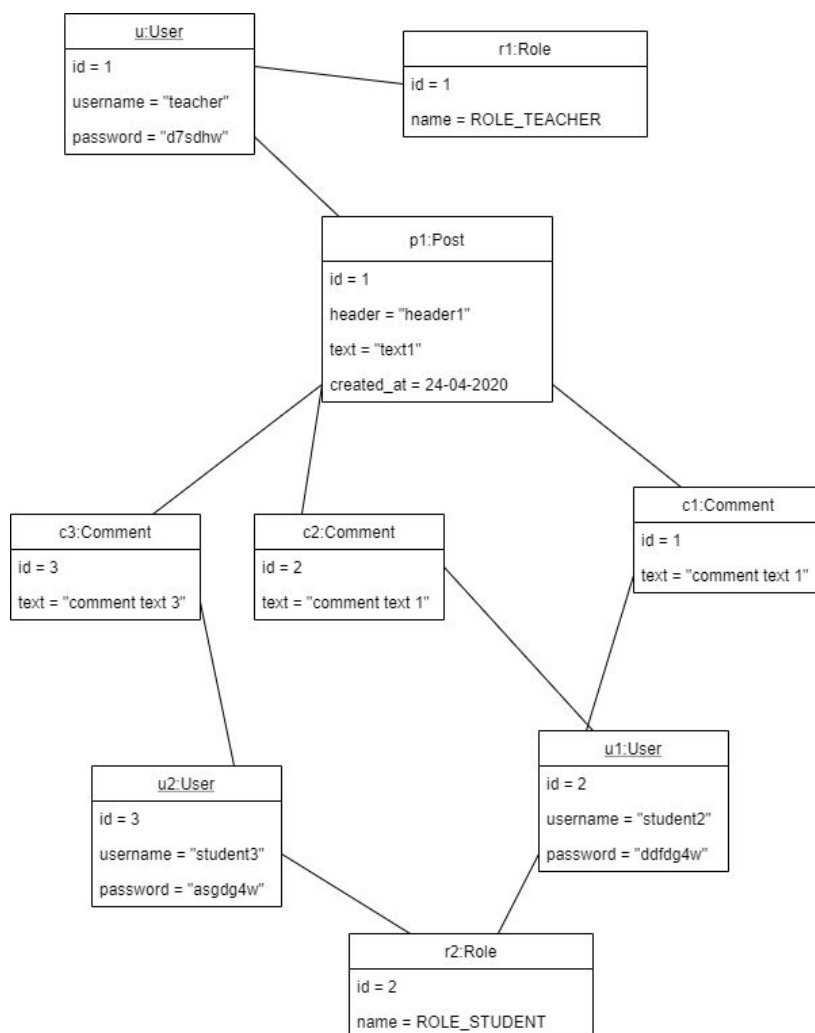


Рисунок 13 - Диаграмма объектов

### 3.7. Схема базы данных

На рисунке 14 представлена схема базы данных.

В базе данных содержится 5 таблиц.

Таблица “users” хранит информацию о пользователях системы, это id, имя пользователя и пароль. Id является первичным ключом. А поля “username” и “password” являются обязательными.

Таблица “roles” хранит информацию о ролях пользователей, это id и название роли. Id - первичный ключ, а поле “name” является обязательным.

Эти две таблицы (users, roles) соединяет таблица “users\_roles”, которая хранит первичный ключ пользователя и первичный ключ роли.



Таблица “posts” хранит информацию о новостях. Это id новости, заголовок, текст новости, дата создания новости и id пользователя, создавшего новость. “Id” является первичным ключом, а поле “user\_id” - внешним.

Таблица “comments” содержит информацию о комментариях пользователей. Это id комментария, текст комментария, id пользователя, оставившего комментарий и id поста, под которым оставили комментарий.

После “id” является первичным ключом, а поле “user\_id” и “post\_id” - внешним.

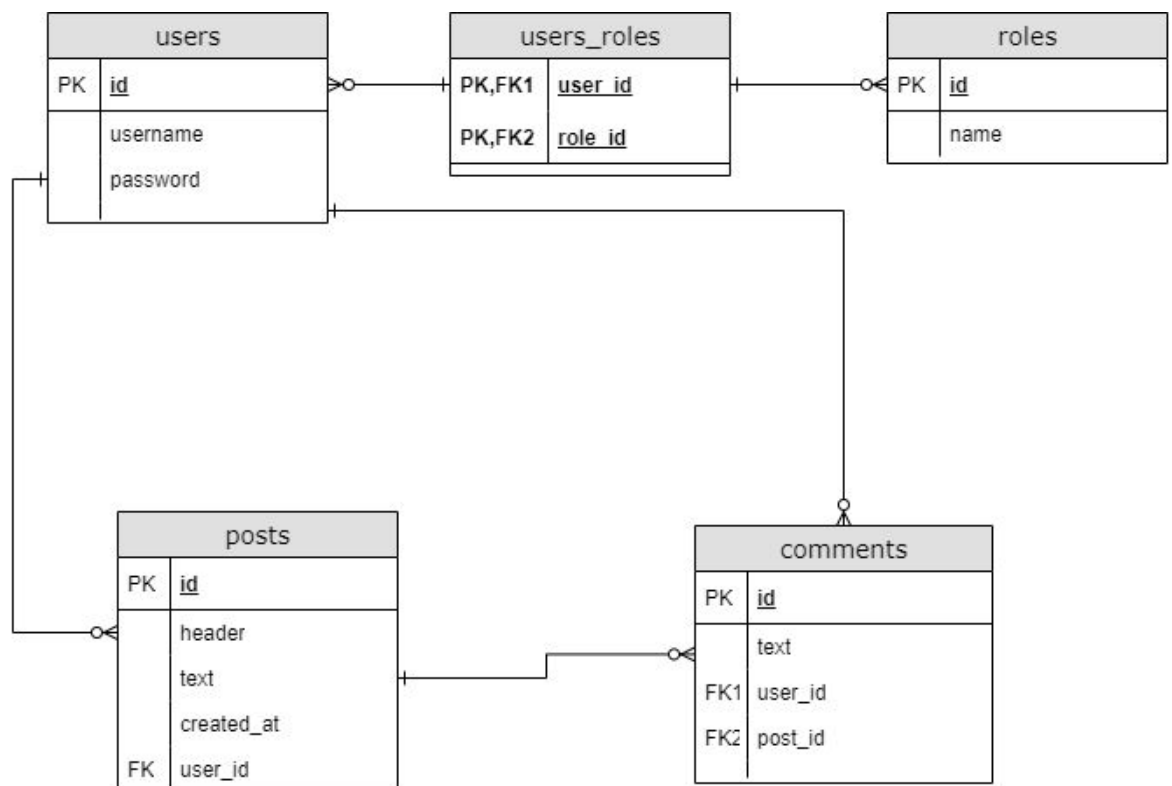


Рисунок 14 - Схема базы данных

### 3.8. Воронки

Были сформированы 3 различных воронки для анализа использования системы:

- Добавление новости для учителя:

Цель создания воронки - определить какое число учителей пользуется системой по назначению.

Воронка состоит из двух шагов:

1. Нажать на кнопку добавления новости;
2. Нажать на кнопку публикации новости.

**Редактировать цель**

Название:

☐ Ретаргетинг

Тип условия:

☐ Количество просмотров ☐ Посещение страниц ☐ JavaScript-событие ☒ Составная цель

Задайте до пяти шагов, которые должны быть последовательно выполнены во время одного визита. Шагами могут быть просмотры страниц или JavaScript-события. [Подробнее.](#) [Диагностика проблем.](#)

**Шаг 1**

Условие:

Код цели для сайта:

**Шаг 2**

Условие:

Код цели для сайта:

Рисунок 15 - Этапы для воронки “добавление новости для учителя”

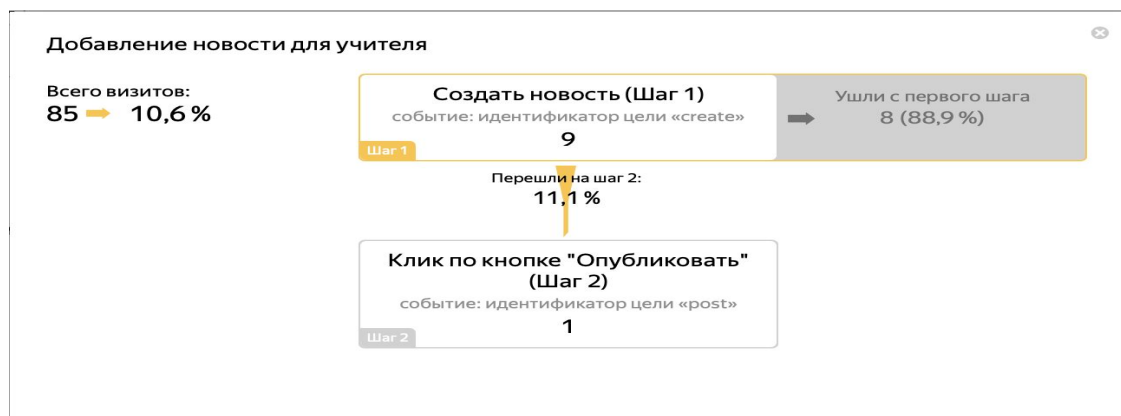
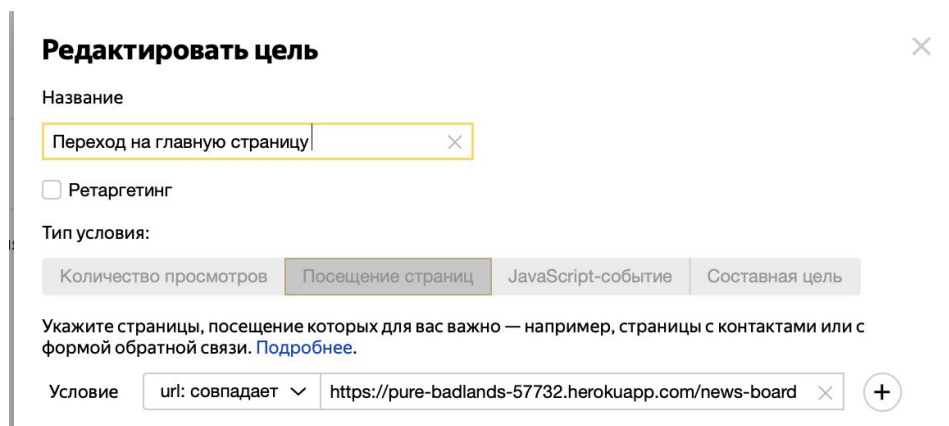


Рисунок 16 - воронка для добавления новости для учителя

- Переход на главную страницу:

Цель создания воронки - определить сколько пользователей заходит в систему.



**Редактировать цель** [X]

Название

☐ Ретаргетинг

Тип условия:

Количество просмотров | **Посещение страниц** | JavaScript-событие | Составная цель

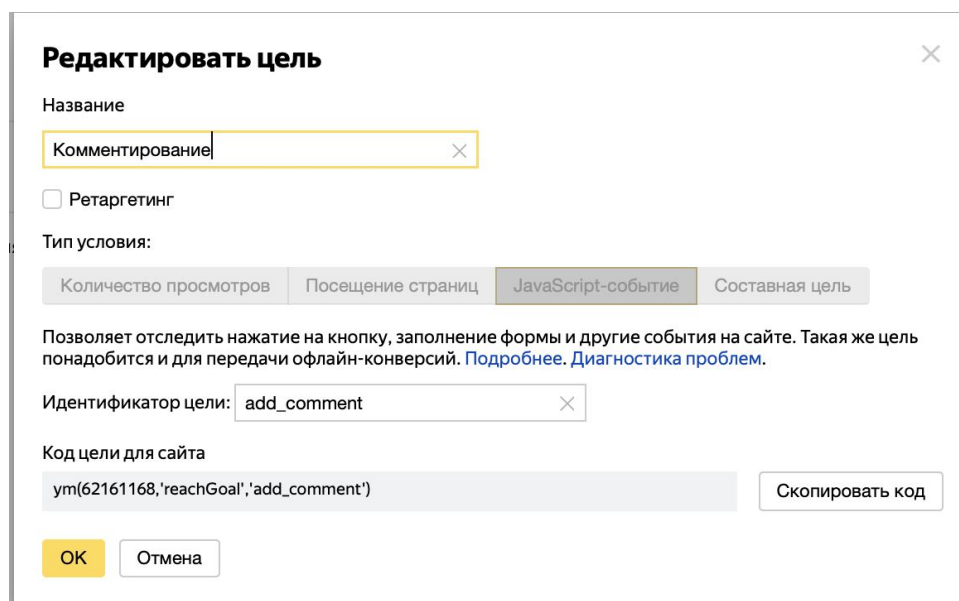
Укажите страницы, посещение которых для вас важно — например, страницы с контактами или с формой обратной связи. [Подробнее.](#)

Условие   [X] [ + ]

Рисунок 16 - Воронка для перехода на главную страницу

- Комментирование новостей:

Цель создания - определить, является ли комментирование новостей актуальным для данной системы.



**Редактировать цель** [X]

Название

☐ Ретаргетинг

Тип условия:

Количество просмотров | Посещение страниц | **JavaScript-событие** | Составная цель

Позволяет отследить нажатие на кнопку, заполнение формы и другие события на сайте. Такая же цель понадобится и для передачи офлайн-конверсий. [Подробнее.](#) [Диагностика проблем.](#)

Идентификатор цели:

Код цели для сайта

Рисунок 17 - Воронка для добавления новости для учителя

## 4. Реализация приложения

### 4.1. Обоснование архитектуры

В качестве паттерна проектирования серверного и клиентского приложения был выбран MVC (Model-View-Controller). Данный паттерн был выбран по ряду причин:

- Разделение логики работы приложения;
- Возможность параллельной разработки;
- Облегчается поддержка и тестирование кода;
- Соответствие выбранному фреймворку.

Среди аналогов можно выделить паттерны MVP (Model-View-Presenter) и MVVM (Model-View-View-Model). Основная причина по которой выбор не был сделан в их пользу это несоответствие выбранного фреймворка этим паттернам. Также в MVP архитектуре требуется реализация view интерфейсов и классов presenter для каждой модели, что является неоправданным усложнением в сравнении с MVC. MVVM имеет еще более витиеватую, в сравнении с MVP, архитектуру, что является также неоправданным для приложения.



## Приложение.

### Приложение 1.

Администрирование » Люди

#### Создание новой персоны

1. Способ добавления персоны → 2. **Добавление персоны** → 3. Проверка данных

Роль в школе:

- ☐ Сотрудник
- ☒ Ученик
- ☐ Родитель

Внимательно выберите тип новой персоны!

Указанные даты прибытия будут внесены в систему как даты зачисления ученика в соответствующий класс и дату поступления.

При необходимости вы можете указать дату, тип и номер приказа на зачисление.

Внимательно заполните все поля!

Код страны согласно ГОСТ 7.87-2003 можно посмотреть [здесь](#).

Личный данные

Фамилия:

Имя:

Отчество:

Дата рождения:

Пол: ☒ Мужской ☐ Женский

Место рождения:

Гражданство:

СНИЛС:

Виза:

Свидетельство о рождении

Серия:

Номер:

Как выдан:

Дата выдачи:

Место выдачи:

Номер вклейки:

Документ, удостоверяющий личность

Тип документа:

Серия:

Номер:

Как выдан:

Место выдачи:

Дата выдачи:

Контактные данные

Адрес постоянной регистрации:

Адрес временной регистрации:

Дата окончания временной регистрации:

Одновременно проживающий в этом помещении: ☐ Соответствует с адресом постоянной регистрации

Подтверждение проживания на территории, заселенной за ОО: ☐ Да ☒ Нет

Email:

Рабочий телефон:

Мобильный телефон:

Домашний телефон: