**МИНОБРНАУКИ РОССИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «ВГУ»)**

Факультет Компьютерных наук

Кафедра информационных технологий управления

Курсовой проект

Аналог википедии «OnlineCyclopedia»

09.03.02 Информационные системы и технологии

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Воронина Е.С., 3 курс, д/о*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Манукян С.Ш., 3 курс, д/о*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Сторожева А.Р., 3 курс, д/о*

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ старший преподаватель *Тарасов В.С.*

Воронеж 2021

**Содержание**

[1 Введение 3](#_Toc75340669)

[2 Постановка задачи 4](#_Toc75340670)

[3 Анализ предметной области 5](#_Toc75340671)

[3.1 Глоссарий 5](#_Toc75340672)

[3.2 Анализ существующих решений 6](#_Toc75340673)

[3.3 Анализ задачи 9](#_Toc75340674)

[3.3.1. Варианты использования приложения 9](#_Toc75340675)

[3.3.2. Взаимодействие компонентов системы 10](#_Toc75340676)

[3.3.3. Варианты состояния системы 12](#_Toc75340677)

[3.3.4. Варианты действия в системе 13](#_Toc75340678)

[3.3.5. Развёртывание приложения 15](#_Toc75340679)

[3.3.6. IDEF0 17](#_Toc75340680)

[4 Анализ средств реализации 18](#_Toc75340681)

[5 Реализация приложения 19](#_Toc75340682)

[5.1 Сущности 19](#_Toc75340683)

[5.2 Сценарии воронок конверсии 22](#_Toc75340684)

[5.3 Графический интерфейс 22](#_Toc75340685)

[6 Тестирование 23](#_Toc75340686)

[6.1 Дымовое тестирование 23](#_Toc75340687)

[6.2 UI тесты 23](#_Toc75340688)

[6.3 Юзабилити тестирование 23](#_Toc75340689)

[7 Заключение 24](#_Toc75340690)

# Введение

В современном мире у человека обычно мало времени в своём распоряжении, необходимо сделать множество дел, найти множество информации и т.п. В наше время востребованы сервисы, в которых собраны воедино и сгруппированы какие-либо сведения об определённой предметной области. Но бывают и такие сайты, которые включают в себя различные статьи из самых разных предметных областей, явным примером которых является «Википедия».

Желаемый сайт должен облегчить пользователям поиск статей по различным предметным областям. Тем самым необходимости в поиске разных статей на разных сайтах больше не будет, можно будет заходить сразу на проверенный сервис.

Использование веб-сайта, предоставляющего статьи по различным предметным областям, даёт следующие преимущества:

* Возможность поиска статьи, литературы, не выходя из дома
* Возможность загрузить свою статью
* Возможность оценить прочитанную статью другого пользователя
* Возможность проверить автора по рейтинговой системе

Данный курсовой проект посвящен разработке именно такого, простого в освоении, но в то же время выполняющего самые необходимые функции, сайта, способного уменьшить временные затраты каждого человека на поиск необходимых статей из самых различных предметных областей.

# Постановка задачи

Целью данной курсовой работы является создание веб-сайта для просмотра статей и их оценки.

Конечным пользователем сайта будет являться человек, вне зависимости от рода его занятий.

К разрабатываемому продукту предъявляются следующие основные требования:

* стабильная работа на современных веб-браузерах,
* интуитивно понятный пользовательский интерфейс,
* сдержанный внешний вид, выдержанный в едином стиле,
* предоставление Пользователю возможности просмотра Статей
* предоставление Авторизованному пользователю возможности добавление своей статьи, редактирование или удаление
* предоставление возможности оценки статьи, загруженной на сайт
* возможность фильтрации, сортировки и поиска статей, имеющихся на сайте

Для достижения данной цели необходимо проанализировать уже существующие решения для выделения каких-либо недочётов для их учёта при разработке своей системы. Также при наличии удачных вещей попытка внедрения их в свою систему.

# Анализ предметной области

## Глоссарий

* **Статья –** это научный труд небольшого размера, в котором автор исследования по научной тематике излагает полученные результаты.
* **Model (база данных)** - этот компонент отвечает за данные в проекте, а также реагирует на команды контроллера, изменяя свое состояние.
* **Представление данных пользователю, Графический интерфейс (View)** - это компонент, отвечающий за взаимодействие с пользователем. То есть данный компонент определяет внешний вид приложения и способы его использования.
* **Контроллер (Controller)** - это компонент, который управляет запросами пользователя. Его основная функция- вызывать и координировать действие необходимых ресурсов и объектов, нужных для выполнения действий, задаваемых пользователем. Обычно контроллер вызывает соответствующую модель для задачи и выбирает подходящий вид.
* **front-end** - клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.
* **back-end** - программно-аппаратная часть сервиса.
* **REST API** - это стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов.
* **GitHub** - крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.
* **Веб-сервис, интернет-сервис, система, веб-приложение, проект** - идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя.
* **Header** - визуальный элемент, расположенный в верхней части страницы.
* **Footer** - визуальный элемент, расположенный в нижней части страницы.
* **Гость** - неавторизованный на веб-сервисе человек, пользующийся ограниченным функционалом веб-сервиса.
* **Пользователь** - авторизованный на портале человек, пользующийся функционалом веб-сервиса.
* **Администратор** - человек, имеющий доступ к расширенному функционалу веб-сервиса.
* **Личный кабинет** - это раздел сервиса, в котором Пользователь может получить доступ к своим данным
* **Контент** - наполнение сайта.

## Анализ существующих решений

1. ru.wikipedia.org - Википедия

Википедия является свободной энциклопедией, в которой авторами статей для этого всеобъемлющего источника являются сами посетители, за счёт чего происходит быстрое и оперативное обновление информации.

Преимущества:

* Свободная и доступная
* Многоязычная и универсальная (возможность предоставления информации на родном языке)
* Возможность поиска в самых различных сферах
* Способности к самовосстановлению статей

Недостатки:

* Отсутствие процесса экспертной оценки при выкладывании
* Отсутствие персонализации статьи
* Открытый характер модели редактирования

Главный недостаток - субъективизм некоторых статей, нарекания специалистов в достоверности сведений.

1. Krugosvet.ru - Универсальная научно-популярная энциклопедия “Кругосвет”

Содержит более 200 тыс. статей по различным направлениям. Статьи содержат не только классические сведения, но и информацию о современных веяниях и течениях. Стиль опубликованных статей живой, научно-популярный

Преимущества:

* Создатели - профессора и доктора наук (более проверенная информация и экспертная оценка статей при выкладывании)
* Возможность поиска в самых различных сферах
* Отсутствие возможности редактирования чужих статей

Недостатки:

* Присутствует только один язык
* Отсутствие возможности добавления своих статей

1. Encyclopedia.ru - мир энциклопедий

Ресурс содержит не сами статьи, а ссылки на сайты самых разных, универсальных и специализированных, энциклопедий. На странице можно найти ссылки на справочные издания общего типа, ресурсы по различным областям знаний

Преимущества:

* Ссылки на проверенные статьи различных изданий
* Отсутствие возможности порчи статьи

Недостатки:

* Отсутствие возможности добавления своих статей

## Анализ задачи

## Варианты использования приложения



Рисунок 1 - Диаграмма прецедентов

При взаимодействии с сайтом у авторизованного пользователя есть определенный список возможностей, который более наглядно изображен на рисунке 1:

* Посмотреть список статей,
* Найти статьи по фильтрам,
* Изменить персональные данные,
* Добавить, изменить или удалить свою статью,
* Дать оценку статье,
* Получить данные о просмотрах статей в личном кабинете,
* Отправлять свою статью на проверку администратором.

Неавторизованный пользователь имеет возможность:

* Зарегистрироваться,
* Авторизоваться,
* Найти статьи, используя фильтр,
* Просмотреть статьи.

Администратор сервиса может:

* Модерировать статьи пользователей,
* Блокировать статьи или добавлять в список одобренных,
* Выйти из аккаунта.

## Взаимодействие компонентов системы

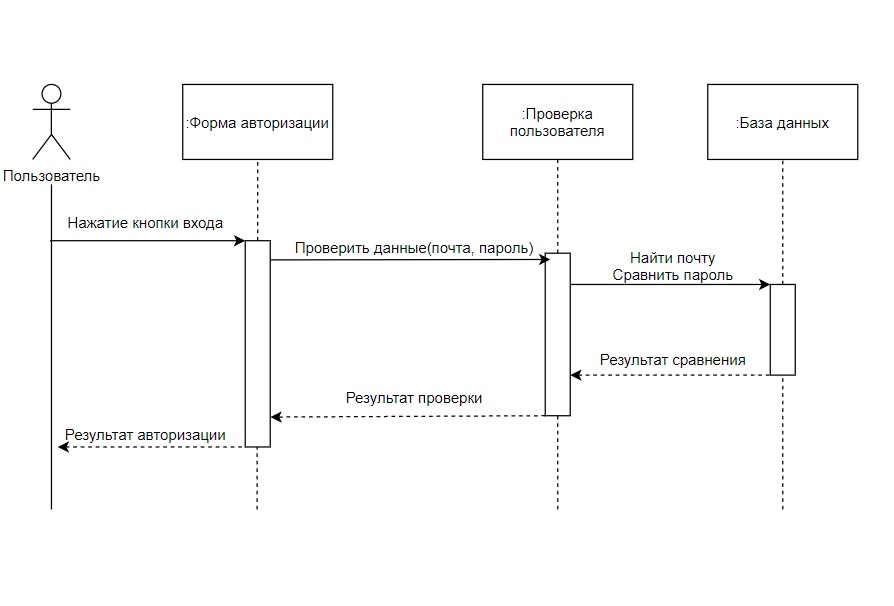


Рисунок 2 - Диаграмма последовательностей

На рисунке 2 показана диаграмма последовательности, на которой изображено упорядоченное во времени взаимодействие объектов при авторизации пользователя.

Для авторизации пользователь обращается к форме авторизации, которая передаёт введённые данные на проверку в модуль проверки пользователя. Тот в свою очередь проверяет существование данного пользователя в базе данных и совпадение введённого пароля с паролем, хранящимся в базе данных. Модуль проверки пользователя посылает статус проверки на форму авторизации, которая выводит пользователю результат авторизации.

## Варианты состояния системы

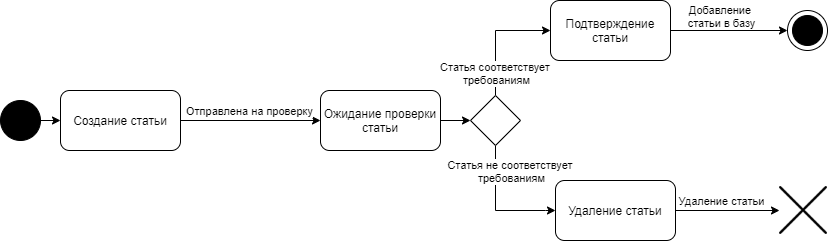


Рисунок 3 - Диаграмма состояний

Диаграмма состояний, изображенная на Рисунке 3, отражает возможные состояния статьи. После создания статьи она отправляется на проверку администратором и переходит в состояние ожидания проверки. Если статья соответствует требованиям (проходит проверку), то она переходит в состояние подтверждения (является подтвержденной) и добавляется в общий список всех статей нужной категории. Если статья не соответствует требованиям (не проходит проверку), то она переход в состояние отклонения (является отклоненной) и удаляется из списка ожидающих проверку статей.

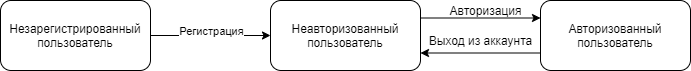


Рисунок 4 - Диаграмма состояний

Диаграмма состояний, изображенная на Рисунке 4, отражает возможные состояния пользователя. Изначально любой пользователь, не прошедший регистрацию, находится в состоянии незарегистрированного пользователя. После прохождения регистрации пользователь переходит в состояние неавторизованного пользователя. После авторизации пользователь переходит в состояние авторизованного пользователя. Если пользователь выходит из аккаунта он возвращается к состоянию неавторизованного пользователя и может авторизоваться снова.

## Варианты действия в системе

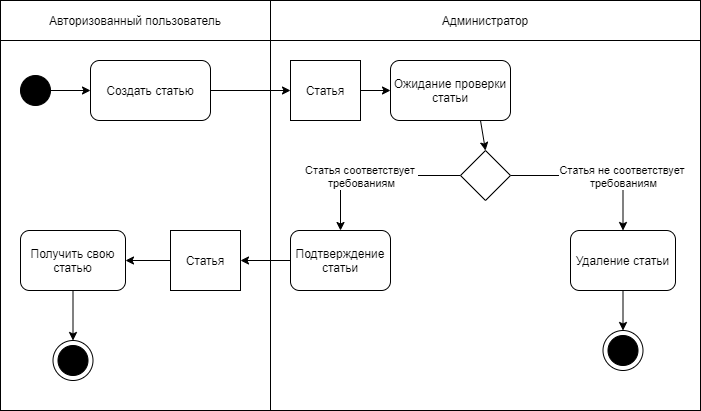


Рисунок 5 - Диаграмма активности

Диаграмма активности, изображенная на Рисунке 5, отражает действия авторизованного пользователя и администратора при создании статьи. После того, как авторизованный пользователь создает статью, она переходит к администратору. При получении статьи администратор должен ее проверить. Если статья не соответствует требованиям, то администратор удаляет ее. Если статья соответствует требованиям, то администратор подтверждает ее, и она (статья) переходит тому авторизованному пользователю, который ее создал. Авторизованный пользователь получает собственную статью, и она в свою очередь закрепляется за ним.

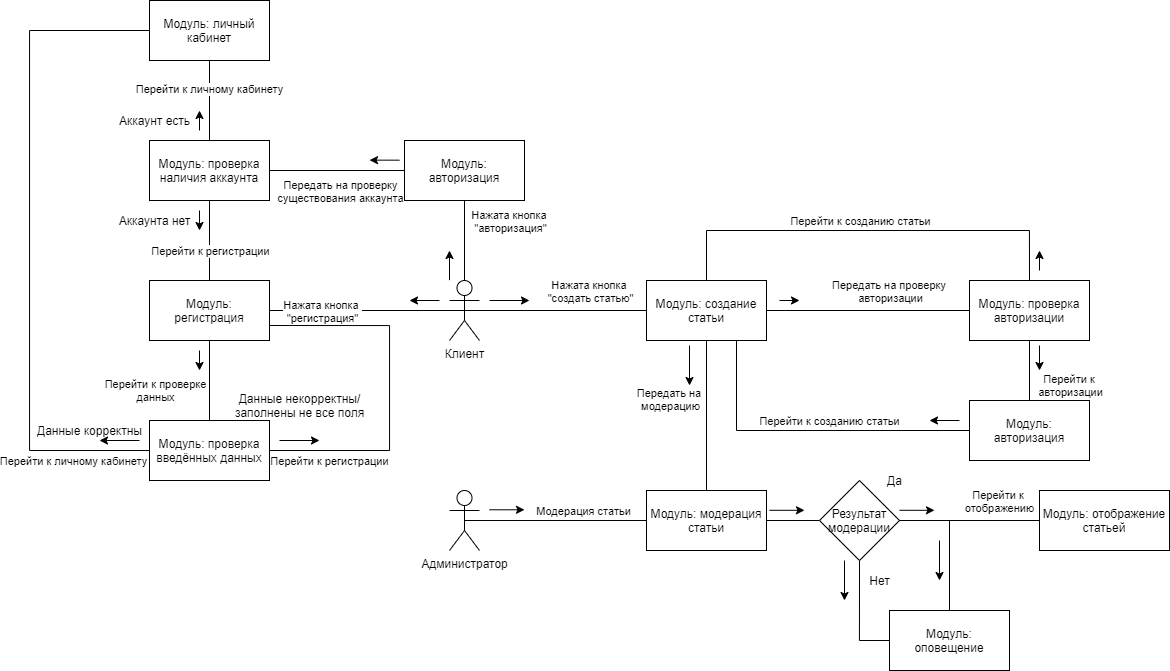


Рисунок 6 - Диаграмма взаимодействий

На Рисунке 6 представлена диаграмма взаимодействий. Она отражает возможные действия пользователя и системы.

Если пользователь захочет выложить статью, система проверит статус авторизации клиента, если он не авторизирован, то система предупредит его об этом и предоставит возможность авторизоваться. Если пользователь авторизован, статья попадает на модерацию, где администратор проверяет корректность предоставленных пользователем данных и выносит вердикт. Если статья одобрена, то клиент получает уведомление об одобрении статьи, и она появляется в списке всех статей, в случае отказа, пользователь получает уведомление об отказе с указанием причины.

Если пользователь захочет пройти процесс регистрации, заполнив необходимые поля, система проверит входящие данные на корректность и в случае ошибки выдаст предупреждение. Если данные корректны, пользователь перейдет в личный кабинет.

При желании пользователя авторизоваться, заполнив необходимые поля, система так же проверит корректность введенных данных. В случае, если пользователя с такими данными нет, система предложит зарегистрироваться.

## Развёртывание приложения

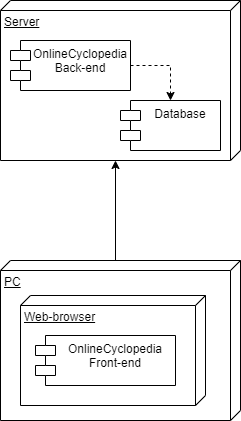


Рисунок 7 - Диаграмма развертывания

На Рисунке 7 представлена диаграмма развертывания, чтобы определить какие аппаратные компоненты («узлы») существуют, какие программные компоненты работают на каждом узле и как различные части этого комплекса соединяются друг с другом. Для разрабатываемого web-приложения узлом устройства является персональный компьютер и сервер, а в качестве узла среды выполнения выступает web-браузер. В браузере развернут front-end приложения, а на серверной части back-end и база данных.

## IDEF0

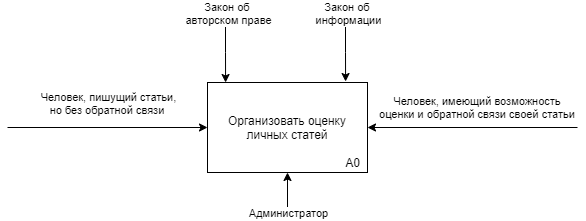


Рисунок 8 – IDEF0

На Рисунке 8 изображена IDEF0 диаграмма. Работу сервиса регулируют «Закон об авторском праве» и «Закон об информации». Работу системы обеспечивает Администратор. На вход в Систему поступает человек, пишущий статьи. На выходе Система выдаёт человека, имеющего возможность выложить свой труд в сеть и получить обратную связь.

# Анализ средств реализации

В качестве средств реализации приложения были выбраны следующие технологии:

* HTML, CSS, JS и сторонние библиотеки JS – web

ориентированные языки HTML и CSS необходимы для разработки front-end части. JS и сторонние библиотеки дают доступ к огромному количеству инструментов, заточенных под разные задачи и упрощающие процесс разработки,

* в качестве СУБД была выбрана MySQL. Она является хорошо масштабируемой, в равной степени легко может быть использована для работы, как с малыми, так и с большими объемами данных. А за счет упрощения некоторых используемых в ней стандартов система имеет высокую производительность,
* в качестве языка разработки в back-end использовался язык PHP.

# Реализация приложения

## Сущности

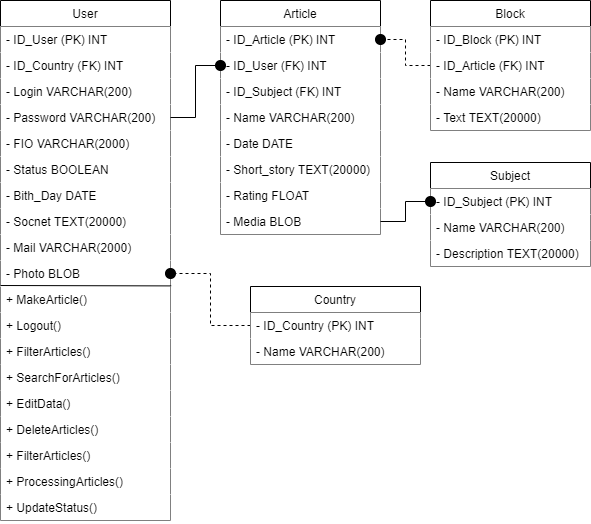


Рисунок 9 - Диаграмма классов

На Рисунке 9 изображена диаграмма классов, отражающая их отношения.

1. Класс «Country» – представляет собой отражение сущности страна. Класс имеет следующие свойства:

* «ID\_Country» – уникальный идентификатор,
* «Name» – название страны.

1. Класс «Subject» – представляет собой отражение сущности предметная область. Класс имеет следующие свойства:

* «ID\_Country» – уникальный идентификатор,
* «Name» – название предметной области,
* «Description» - описание предметной области.

1. Класс «Article» – представляет собой отражение сущности статья. Класс имеет следующие свойства:

* «ID\_Article» – уникальный идентификатор,
* «ID\_Subject» - предпочитаемая предметная область (FK),
* «ID\_User» - пользователь, создавший статью (FK),
* «Name» – название статьи,
* «Date» – дата создания статьи,
* «Short\_story» - краткое содержание статьи,
* «Rating» - рейтинг статьи,
* «Media» - медиа данные статьи.

1. Класс «Block» – представляет собой отражение сущности блок статьи. Класс имеет следующие свойства:

* «ID\_Block» – уникальный идентификатор,
* «ID\_Article» – идентификатор принадлежности блока текста к определённой статье (FK),
* «Name» – заголовок блока,
* «Text» - текст блока.

1. Класс «User» – представляет собой отражение сущности пользователь. Класс имеет следующие свойства:

* «ID\_User» – уникальный идентификатор,
* «ID\_Country» – выбранная страна пользователя (FK),
* «Login» – логин пользователя,
* «Password» – пароль пользователя,
* «FIO» – ФИО пользователя (FK),
* «Status» – статус пользователя (модератор или нет),
* «Bith\_Day» – день рождения пользователя,
* «Socnet» – ссылки на социальные сети пользователя.
* «Mail» – почта пользователя,
* «Photo» – фото пользователя.

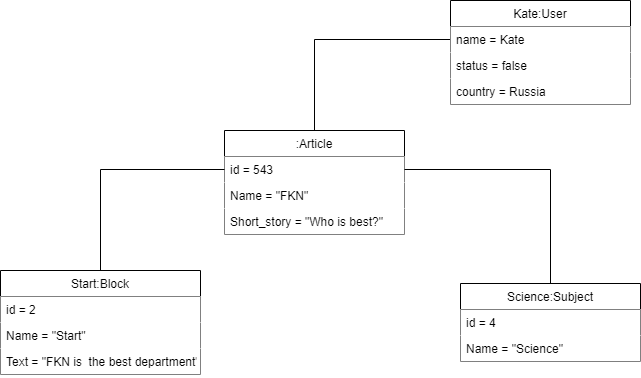


Рисунок 10 - Диаграмма объектов

На Рисунке 10 изображена диаграмма объектов, которая отражает множество экземпляров классов и отношений между ними в некоторый момент времени. На ней изображён экземпляр класса “Авторизованный пользователь”, экземпляр класса “Статья”, которая была создана пользователем и хранит в себе ссылку на него. В свою очередь экземпляр класса “Статья” содержит в себе поле Блоки, являющееся экземпляром класса «Блок» и поле Предметная область, являющееся экземпляром класса “Предметная область”, которые также отражены на диаграмме, и показана их связь с другими объектами.

## Сценарии воронок конверсии

1) Посетил главную страницу - Авторизовался - Перешел на страницусоздания статьи - Нажал кнопку "Добавить".

2) Посетил главную страницу - Авторизовался - Перешел на страницустатьи - Нажал кнопку с добавлением рейтинга.

3) Посетил главную страницу - Авторизовался - Выбрал нужные фильтры - Нажал кнопку "Отфильтровать".

## Графический интерфейс

# Тестирование

Для проверки работоспособности сайта, после его создания будет проведено 3 вида тестирования:

* дымовое тестирование;
* UI тесты;
* юзабилити тесты тесты.

# Дымовое тестирование

Для данного вида тестирования необходимо проверить работоспособность сайта на основных сценариях.

# UI тесты

Для UI тестирования необходимо выполнить тесты, охватывающие основные возможности сайта.

# Юзабилити тестирование

Для данного вида тестирования необходимы участники, не пользовавшиеся сервисом ранее, для проверки работоспособности данного проекта. Необходимо удостовериться в работе его основных сценариев взаимодействия с клиентами.

# Заключение

В результате работы был разработан сайт по размещению и редактированию статей. Были выполнены следующие задачи:

1. разработана Front-end часть сайта, находящаяся на компьютере/телефоне пользователя,
2. разработана Back-end часть сайта, развернутая на удаленном сервере,
3. была создана связь между Front-end и Back-end частями приложения,
4. разработана база данных, расположенная на удаленном сервере.

Приложение отвечает всем заявленным требованиям.