

Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет информатики и
радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Жизненный цикл разработки программного обеспечения

Лабораторная работа №2
«Требования к проекту»

Проверил:
асс. каф. ЭВМ
О.М. Внук

Выполнил:
ст. гр. 350504
А.А.Арциеня

МИНСК 2025

СОДЕРЖАНИЕ

1 Введение.....	3
2 требования пользователя	4
2.1 Программные интерфейсы	4
2.2 Интерфейс пользователя.....	4
2.4 Предположения и зависимости.....	9
3 системные требования	11
3.1 Функциональные требования	11
3.2 Нефункциональные требования	12
4 Глоссарий	15
5 ДИАГРАММА КЛАССОВ	16
6 АРХИТЕКТУРА ВЗАИМОДЕЙСТВИЯ С СИСТЕМОЙ	17
6.1 Случаи использования	17
6.2 Диаграммы активностей	20
6.3 Диаграммы состояний	25
7 ЗАКЛЮЧЕНИЕ	26

1 ВВЕДЕНИЕ

В современном информационном пространстве, характеризующемся высокой скоростью распространения новостей и их обилием, пользователи сталкиваются с проблемой фрагментации контента и информационной перегрузкой. Стандартные новостные агрегаторы зачастую предлагают лишь пассивное потребление информации, оставляя в стороне растущую потребность аудитории в активном обсуждении, обмене мнениями и формировании сообщества по интересам. Именно эту нишу призван занять проект PulseNews.

Продукт PulseNews представляет собой современное веб-приложение, разработанное для решения этих задач. Он позиционируется как интерактивный новостной портал нового поколения, который не только агрегирует актуальные материалы из различных источников, но и предоставляет пользователям широкие возможности для социального взаимодействия. Ключевая концепция проекта заключается в синергии качественного новостного контента и развитых инструментов для его обсуждения, что превращает платформу из обычного источника информации в динамичную дискуссионную площадку.

Проект преследует несколько основных целей. Прежде всего, он нацелен на создание централизованного источника информации, предоставляя пользователям единую платформу для доступа к новостям из разных сфер с удобной навигацией по категориям и тегам. Наряду с этим, важнейшей задачей является формирование активного сообщества через построение среды, где пользователи могут не только получать информацию, но и делиться своим мнением, вступать в дискуссии и оценивать контент через систему комментариев, лайков и дизлайков. В дополнение, проект сфокусирован на обеспечении персонализированного опыта через инструменты для настройки ленты новостей и сохранения интересных статей, что позволит каждому пользователю формировать собственное информационное поле.

Границы проекта: На текущем этапе проект сфокусирован на разработке полнофункционального веб-приложения. Система позволит пользователям читать актуальные материалы, осуществлять поиск публикаций по тегам и ключевым словам, делиться новостями, а также взаимодействовать друг с другом. Для администраторов и авторов будет реализован защищенный интерфейс для управления контентом, включая создание, редактирование и удаление статей, а также настройку автоматического сбора новостей с внешних ресурсов. Настоящий документ определяет и детализирует функциональные и нефункциональные требования, предъявляемые к системе для достижения поставленных целей.

2 ТРЕБОВАНИЯ ПОЛЬЗОВАТЕЛЯ

2.1 Программные интерфейсы

Для обеспечения стабильной и эффективной работы веб-приложение PulseNews будет взаимодействовать с рядом ключевых внешних и внутренних систем. Архитектура проекта построена на взаимодействии специализированных сервисов, каждый из которых выполняет определенную функцию.

Основным хранилищем данных выступит реляционная СУБД PostgreSQL. В ней будет надежно храниться вся ключевая информация, включая новостные статьи, сведения о пользователях, категории и теги. Выбор этой системы обусловлен ее высокой надёжностью, строгой поддержкой ACID-транзакций и мощными возможностями для работы со структурированными данными, что гарантирует целостность и безопасность всей критически важной информации проекта.

С целью повышения производительности и минимизации нагрузки на основную базу данных будет интегрировано высокоскоростное хранилище Redis. Оно будет выполнять двойную функцию: во-первых, кэшировать часто запрашиваемые данные, такие как главная страница или популярные новости, для ускорения их загрузки. Во-вторых, Redis выступит в роли брокера сообщений для управления фоновыми задачами, например, для асинхронного парсинга новостей с внешних источников.

Для наполнения контентом система будет регулярно обращаться к внешним новостным сайтам и RSS-лентам. Это взаимодействие будет осуществляться через специализированные программные модули-парсеры.

В качестве веб-сервера будет задействован Nginx. Он будет отвечать за проксирование запросов к бэкенду приложения и быструю отдачу статических файлов, таких как CSS, JavaScript и изображения, что позволит оптимизировать скорость загрузки страниц для конечных пользователей.

2.2 Интерфейс пользователя

Система будет взаимодействовать с пользователем через веб-сайт. Основные элементы интерфейса и их поведение:

- Лента новостей: Центральный хаб портала. Отображает ленту последних новостей, отсортированных по дате публикации (сначала самые свежие).
- Страница статьи: Детальное представление выбранной новости.
- Страница категорий и тэгов: Страница, которая отображает все статьи, принадлежащие к определенной или помеченные определенным тегом.
- Поиск и фильтрация: Система для поиска и уточнения запросов пользователя.

- Административный интерфейс: Защищенный раздел для редакторов и администраторов.

Реализация пользовательского интерфейса представлена на рисунках 2.2.1 – 2.2.6

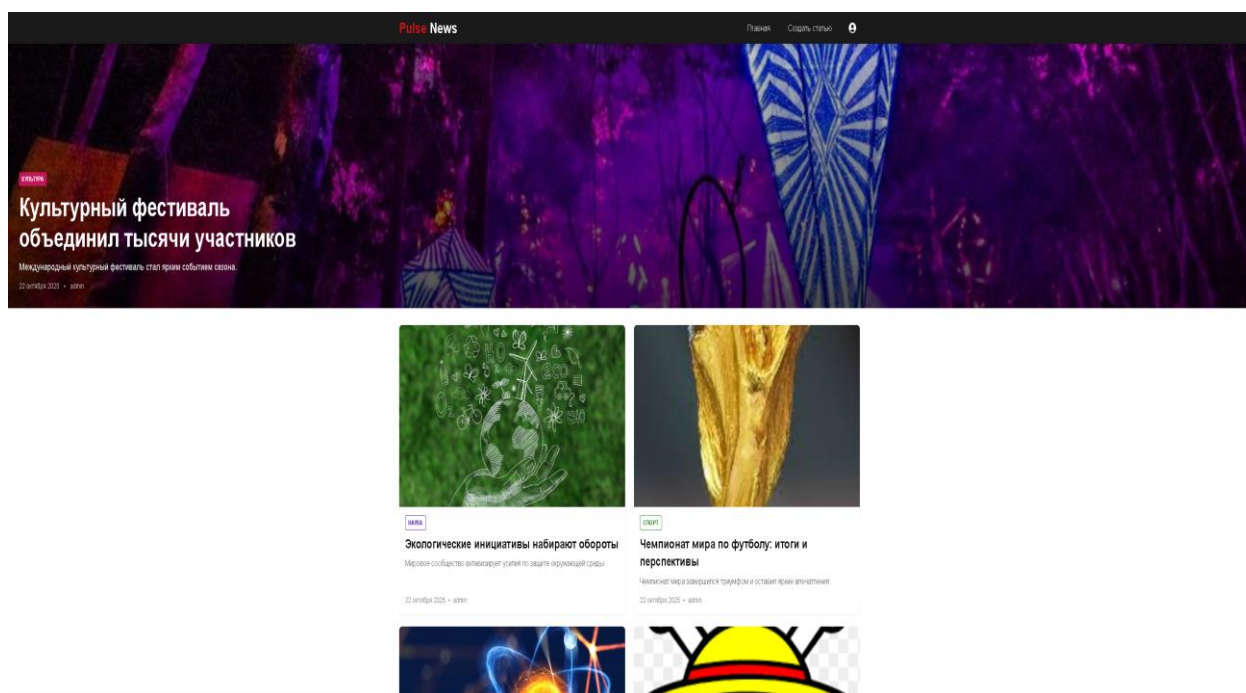


Рисунок 2.2.1 – Главная страница.

Рисунок 2.2.2 – Страница регистрации пользователя.

EventFlow

Pulse News

Главная

Создать статью

Admin Adminovich

@admin

Администратор

Дата регистрации

Н/Д

Публикации

0

Активность

0 действий

ПРОФИЛЬ

ИСТОРИЯ АКТИВНОСТИ

Личная информация

Имя *

Admin

Ваше имя

Фамилия *

Adminovich

Ваша фамилия

Email *

admin@pulsenews.com

Ваш адрес электронной почты

О себе

Краткое описание о вас

Телефон

Контактный телефон

Дата рождения

ДД . ММ . ГГГГ

Ваша дата рождения

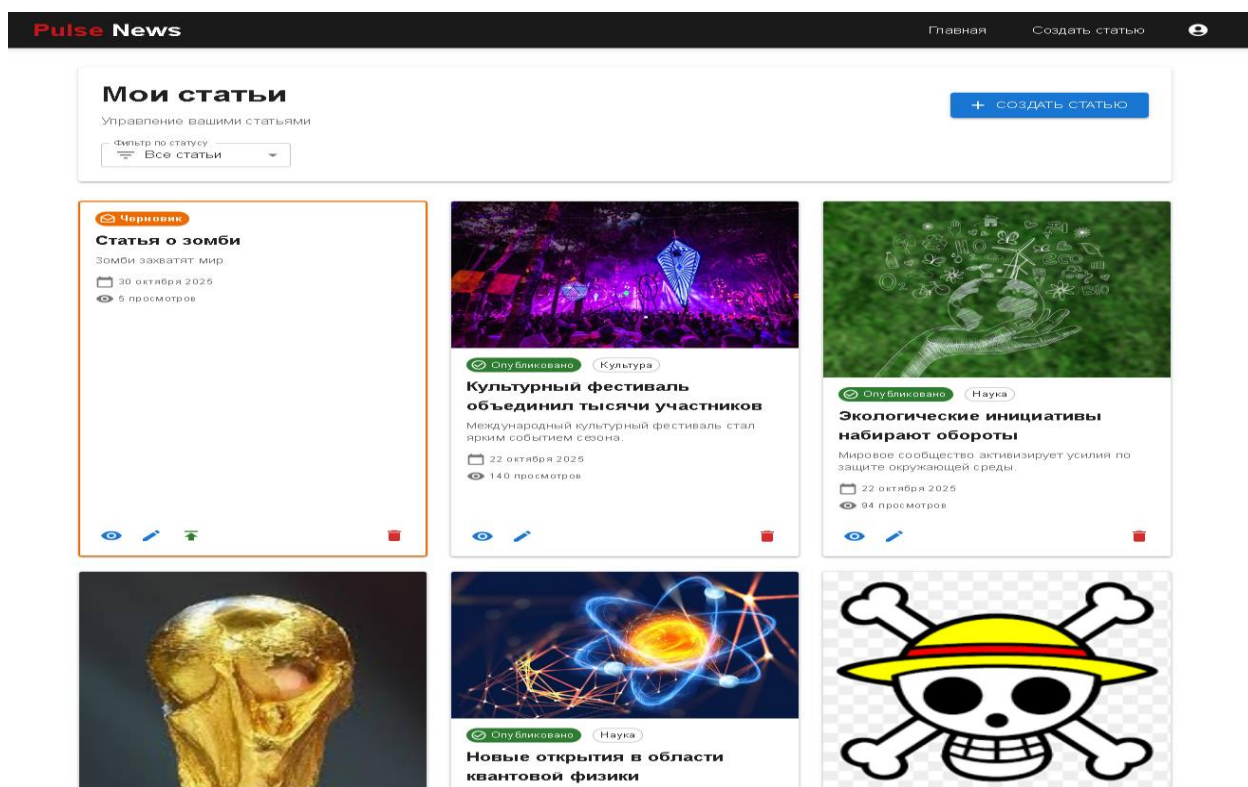
Нажмите "Редактировать" для изменения данных

УДАЛИТЬ АККАУНТ

РЕДАКТИРОВАТЬ

СМЕНИТЬ ПАРОЛЬ

Страница 2.2.3 – Страница личного кабинета пользователя.



EventFlow

Pulse News

Главная

Создать статью

Создать статью

Заголовок *

Краткое описание *

Краткое описание статьи для превью

Содержание *

Категория

Теги

ЗАГРУЗИТЬ ОБЛОЖКУ

Опубликовать сразу

СОХРАНИТЬ

ОТМЕНА

О нас

О проекте

Наша команда

Категории

Политика

Технологии

Правовая информация

Политика конфиденциальности

Условия использования

Социальные сети

Facebook

Twitter

Рисунок 2.2.5 – Страница создания статьи.

Pulse News

Главная

Создать статью

← НАЗАД К СТАТЬЯМ

Культура

Культурный фестиваль объединил тысячи участников

Аноним

22 октября 2025, 02:54

141 просмотров

новости

репортаж

Международный культурный фестиваль стал ярким событием сезона.

Культурный фестиваль в столице объединил тысячи ценителей искусства со всего мира

Международный культурный фестиваль, прошедший в столице, стал ярким событием сезона, привлекая тысячи участников и зрителей. Гостей ждала насыщенная программа, включавшая в себя авангардные театральные постановки, концерты классической и современной музыки, новаторские выставки и увлекательные мастер-

Рисунок 2.2.6 – Страница просмотра статьи.

2.3 Характеристики пользователей

2.3.1 Классы пользователей

Таблица 2.3.1.1 – Основные классы пользователей.

Класс пользователей	Описание
Посетители	Незарегистрированные пользователи, имеющие доступ к базовому функционалу. Они могут просматривать новостные статьи, осуществлять поиск и навигацию по категориям и тегам. Возможности для интерактивного взаимодействия (комментарии, реакции, закладки) для них недоступны.
Зарегистрированные пользователи	Пользователи, создавшие аккаунт и прошедшие аутентификацию. Обладают полным доступом ко всем пользовательским функциям: чтение и поиск новостей, оставление комментариев, выставление реакций (лайков/дизлайков) и сохранение статей в личные «Закладки» для последующего чтения.
Админы	Пользователи с максимальными привилегиями, отвечающие за администрирование всей платформы. В их обязанности входит управление учетными записями других пользователей (назначение ролей, блокировка), настройка системных параметров (например, управление парсерами новостей) и полный контроль над всем контентом.
Редакторы	Пользователи с расширенными правами, отвечающие за управление контентом. Имеют доступ к административному интерфейсу для создания, редактирования, публикации и удаления новостных статей, а также для управления категориями и тегами.

2.3.2 Аудитория приложения

2.3.2.1 Целевая аудитория

- Лица, достигшие возраста 16 лет, и пользователи, которые:
- Стремятся оперативно получать информацию о событиях в мире, стране или в конкретных интересующих их отраслях;
 - Ищут единую и удобную платформу, агрегирующую новости, чтобы избежать необходимости посещать множество сайтов;

– Ценят не только получение фактов, но и возможность участвовать в обсуждениях и знакомиться с мнениями других людей.

Создатели контента и администраторы – пользователи, которые:

- Нуждаются в удобном инструменте для публикации и управления новостными статьями, категориями и тегами;
- Отвечают за поддержание актуальности и качества контента на портале;
- Используют платформу как основной канал для распространения информации и модерации пользовательских дискуссий.

Активные потребители новостей – читатели и комментаторы, которые:

- Нуждаются в быстром поиске актуальной информации по интересующим их темам;
- Ищут платформу для активного взаимодействия: обсуждения новостей в комментариях и выражения своего мнения через реакции;
- Формируют свою персональную ленту и сохраняют важные статьи в «Закладки» для последующего чтения или анализа.

2.3.2.2 Побочная аудитория

Студенты профильных специальностей (журналистика, медиакоммуникации, политология, социология), которые:

- Используют платформу для учебных целей, например, для анализа новостной повестки и стилей подачи материала;
- Практикуются в анализе общественного мнения, изучая комментарии и дискуссии под резонансными статьями;
- Собирают и систематизируют материалы для курсовых работ, эссе или создания портфолио.

2.4 Предположения и зависимости

2.4.1 Предположения

Таблица 2.4.1.1 – Предположения

Предположение	Описание
Доступ к интернету	Приложение требует постоянного подключения к интернету для загрузки ленты новостей, изображений и взаимодействия с системой (комментарии, лайки).
Поддержка браузеров	Предполагается, что пользователи используют современные браузеры с поддержкой JavaScript (Chrome, Firefox, Safari, Edge последних версий).
Корректность данных	Предполагается, что контент, добавляемый редакторами или собираемый автоматически (парсинг), является валидным и корректно отображается.

Конфиденциальность данных	Пользователи предполагают, что их учетные данные (email, пароль) и личные списки «Закладок» надежно защищены от несанкционированного доступа.
Техническая грамотность	Предполагается, что пользователи обладают базовыми навыками веб-серфинга и взаимодействия с интерфейсами новостных порталов.

2.4.2 Зависимости

Внешние

зависимости:

- PostgreSQL – хранение всех структурированных данных проекта: новостных статей, учетных записей пользователей, комментариев, категорий и тегов.
- Внешние новостные источники – сторонние веб-сайты и RSS-ленты, от доступности и структуры которых напрямую зависит работа системы парсинга контента.
- Хостинг-провайдер – обеспечивает размещение и доступность серверной части веб-приложения и его базы данных.

Технические

зависимости:

- Работа приложения – невозможна без стабильного подключения к интернету как для пользователей, так и для серверных компонентов (парсеров).
- Доступ к интерактивным функциям – пользователь не может оставлять комментарии, ставить реакции или добавлять статьи в «Закладки» без успешной авторизации в системе.
- Актуальность контента – своевременное обновление ленты новостей зависит от корректной работы фоновых задач (управляемых Celery и Redis) и доступности внешних новостных ресурсов.

Функциональные

зависимости:

- Сохранение данных – любая операция, связанная с записью информации (регистрация, комментирование, создание статьи), невозможна без работающей базы данных PostgreSQL.
- Авторизация пользователей – требует исправной и безопасной работы системы аутентификации для предоставления доступа к личным функциям.
- Производительность и отзывчивость – высокая скорость загрузки страниц и обработки запросов напрямую зависит от корректной работы системы кэширования Redis.

3 СИСТЕМНЫЕ ТРЕБОВАНИЯ

3.1 Функциональные требования

3.1.1 Управление учетными записями

Регистрация нового пользователя:

- Система должна предоставлять форму регистрации с полями: имя пользователя, email и пароль;
- После успешной регистрации система должна автоматически авторизовать пользователя;
- Email пользователя должен быть уникальным в системе.

Авторизация пользователя:

- Система должна предоставлять форму входа с полями: email, пароль;
- При неуспешной попытке входа система должна отображать сообщение об ошибке;
- После успешной авторизации система должна создавать безопасную сессию

Выход из системы:

- Система должна предоставлять кнопку выхода из аккаунта;
- После выхода система должна прекращать сессию пользователя.

3.1.2 Управление новостным контентом

Поиск и просмотр новостей:

- Система должна предоставлять поле для поиска новостей по ключевым словам в заголовке и тексте.
- Результаты поиска должны отображаться в виде списка статей с основными данными (заголовок, краткое описание, дата публикации).

Управление «Закладками» (для зарегистрированных пользователей):

- Пользователь должен иметь возможность добавить любую новостную статью в свой личный список «Закладки».
- Система должна отображать все статьи, добавленные пользователем в «Закладки».
- Пользователь должен иметь возможность удалить статью из своего списка «Закладок».

Управление контентом (для авторов/администраторов):

- Авторизованные пользователи с соответствующими правами должны иметь возможность создавать, редактировать и удалять новостные статьи.
- Система должна позволять назначать статьям категории и добавлять к ним теги.
- Администраторы должны иметь доступ к настройке и запуску парсинга новостей с внешних источников.

3.1.3 Система фильтрации и поиска

- Полнотекстовый поиск: Система должна предоставлять пользователям возможность осуществлять поиск по ключевым словам. Поиск должен производиться по заголовкам и содержанию всех новостных статей.
- Фильтрация контента: Пользователь должен иметь возможность фильтровать новостную ленту по одному или нескольким параметрам одновременно (комбинированный фильтр), включая:
 - Категорию новости;
 - Присвоенные теги;
 - Диапазон дат публикации.
- Сортировка результатов: Пользователь должен иметь возможность сортировать отображаемый список новостей по дате публикации (сначала новые / сначала старые) и по популярности (например, по количеству комментариев или реакций).
- Поиск в «Закладках»: Для аутентифицированных пользователей должен быть реализован отдельный поиск по статьям, сохраненным в их личном списке «Закладки».

3.1.4 Системные требования

- Производительность API: Все конечные точки (endpoints) API должны возвращать ответ в течение приемлемого времени (см. раздел «Производительность»).
- Обработка ошибок: Ошибки API должны обрабатываться корректно и возвращаться клиенту в стандартизированном, документированном формате.
- Безопасность учетных данных: Пароли пользователей должны храниться в базе данных исключительно в хэшированном виде с использованием современных и стойких криптографических алгоритмов.
- Шифрование трафика: Весь обмен данными между клиентом и сервером должен осуществляться по защищенному протоколу HTTPS.
- Контроль доступа: Личные данные пользователей (email, список закладок) не должны быть доступны другим пользователям или неавторизованным лицам.

3.2 Нефункциональные требования

3.2.1 Атрибуты качества

3.2.1.1 Надежность

Система должна быть устойчива к сбоям. Так как архитектура распределенная, сбой одного микросервиса не должен приводить к полному отказу всей системы. Для этого важна правильная настройка

отказоустойчивости и мониторинга, измеряемая через показатель MTBF (Mean Time Between Failures).

Метрики измерения:

- Время безотказной работы (uptime): не менее 99.9% в месяц;
- Частота резервного копирования данных: ежедневно, с хранением копий не менее 7 дней.

3.2.1.2 Производительность

Приложение должно обрабатывать запросы пользователей быстро, особенно при поиске и загрузке статей. Это измеряется через время отклика для ключевых операций, таких как поиск и чтение.

Метрики измерения:

- Время отклика на запросы: не более 2 секунд для 95% запросов;
- Пропускная способность: обработка не менее 1000 запросов в минуту без деградации.

3.2.1.3 Безопасность

Защита пользовательских данных и целостности контента является фундаментальным требованием для новостного портала.

Метрики измерения:

- Защита от уязвимостей: отсутствие критических уязвимостей (согласно OWASP Top 10) по результатам регулярных аудитов безопасности (не реже одного раза в год).
- Шифрование: обязательное использование HTTPS для всего трафика.

3.2.1.4 Масштабируемость

Архитектура микросервисов должна позволять масштабировать каждый сервис независимо, чтобы справляться с ростом нагрузки и количества пользователей. Это можно измерять по максимальному количеству одновременных пользователей, которое система может поддерживать без деградации производительности.

Метрики измерения:

- Одновременные пользователи: поддержка не менее 10 000 активных пользователей одновременно без потери производительности.
- Эластичность: автоматическое масштабирование серверных ресурсов при увеличении нагрузки на CPU или память свыше 80%.

3.2.1.5 Удобство использования

Интерфейс должен быть интуитивно понятным для пользователей с разным уровнем технической грамотности. Это можно оценить через тестирование пользовательского опыта и сбор обратной связи.

Метрики измерения:

- Время освоения: новый пользователь должен освоить основные функции (поиск, чтение, комментирование, добавление в закладки) менее чем за 5 минут.

3.2.1.6 Поддерживаемость

Архитектура и кодовая база должны быть спроектированы таким образом, чтобы упростить внесение изменений, исправление ошибок и добавление нового функционала.

Метрики измерения:

- Время развертывания: процесс выкатки обновлений на продуктивный сервер не должен занимать более 1 часа.
- Качество кода: покрытие кода автоматическими тестами должно составлять не менее 80% для всех критически важных модулей.

3.2.1.7 Конфиденциальность

Приватность личных данных пользователей должна быть гарантирована на архитектурном уровне.

Метрики измерения:

- Доступ к данным: доступ к персональной информации пользователя (email, хэш пароля, закладки) должен быть строго ограничен самим пользователем и, в служебных целях, администраторами с ведением журнала аудита таких доступов.

3.2.1.8 Доступность

Сервис должен быть доступен для пользователей круглосуточно, независимо от их местоположения.

Метрики измерения:

- Доступность сервиса: 24/7, за исключением плановых технических работ.
- Отказоустойчивость: использование геораспределенной инфраструктуры для обеспечения бесперебойной работы в случае сбоя одного из серверов или дата-центров.

4 ГЛОССАРИЙ

Данный глоссарий определяет ключевые термины, используемые в проекте новостного онлайн-портала PulseNews. Он служит единым источником терминологии для разработчиков, тестировщиков, аналитиков и пользователей системы.

Роли пользователей

Пользователь – любое лицо, взаимодействующее с системой. Может быть неаутентифицированным или аутентифицированным.

Аутентифицированный пользователь – пользователь, вошедший в систему под своей учетной записью. Имеет доступ к дополнительным функциям — комментированию, реакциям и закладкам.

Автор / Редактор – привилегированный аутентифицированный пользователь с правами на создание, редактирование и удаление статей.

Администратор – пользователь с максимальными правами: управляет пользователями, контентом и настройками системы, включая запуск парсинга новостей.

Контент и его структура

Новостная статья (Статья) – основная информационная единица системы. Включает заголовок, текст, дату публикации, автора, категорию и теги.

Категория – раздел, к которому относится статья (например, Политика, Спорт, Технологии). У одной статьи может быть только одна категория.

Тег – ключевое слово или метка, описывающая тематику статьи. Статья может содержать несколько тегов.

Взаимодействие с контентом

Комментарий – текстовое сообщение, оставленное аутентифицированным пользователем под статьей.

Реакция (Лайк / Дизлайк) – оценка, которую пользователь может поставить статье или комментарию.

Закладка – функция, позволяющая сохранить статью для последующего прочтения.

Лента новостей – основная страница портала, отображающая список статей, отсортированных по дате публикации.

Технические термины

Парсинг – автоматический сбор новостных материалов с внешних источников для наполнения базы данных портала.

CRUD – акроним от Create, Read, Update, Delete — четыре базовые операции для работы с данными.

5 ДИАГРАММА КЛАССОВ

Диаграмма классов демонстрирует основные сущности системы PulseNews, их атрибуты, методы и связи между ними. Она является основой для проектирования структуры базы данных и бизнес-логики приложения.

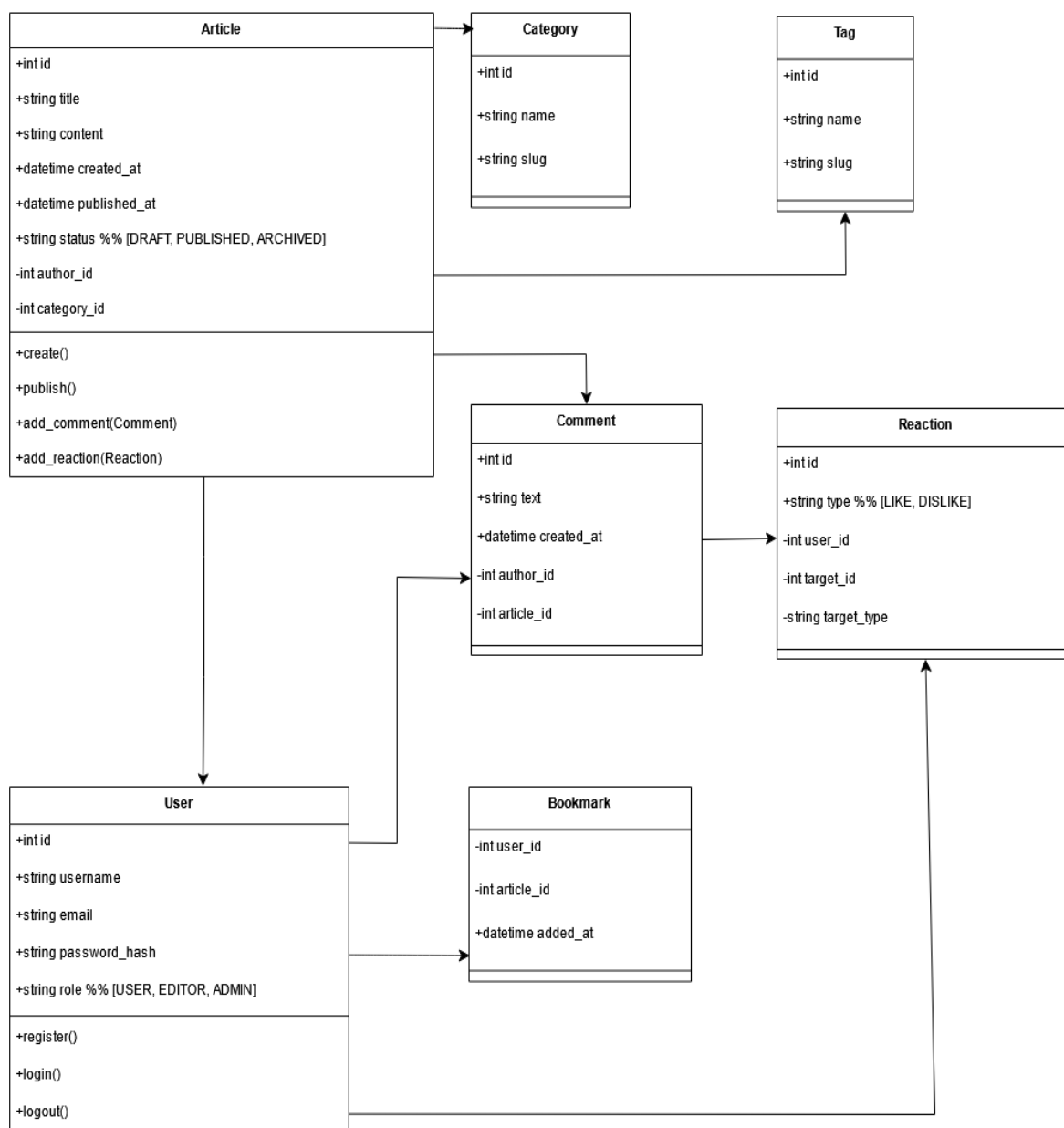


Рисунок 5.1 – Диаграмма классов

6 АРХИТЕКТУРА ВЗАИМОДЕЙСТВИЯ С СИСТЕМОЙ

В процессе анализа системы PulseNews выделены следующие основные акторы:

- Неаутентифицированный пользователь – любой посетитель сайта, который не вошел в систему. Имеет доступ к функциям просмотра контента и регистрации.
- Аутентифицированный пользователь – зарегистрированный пользователь, вошедший в систему. Может взаимодействовать с контентом (комментировать, ставить реакции, добавлять в закладки).
- Автор/Редактор – привилегированный пользователь с правами на управление новостными статьями.
- Администратор – пользователь с максимальными правами, управляющий всей системой, включая пользователей и контент.

6.1 Случаи использования

В данном разделе рассмотрены ключевые варианты использования приложения. Все варианты использования и взаимодействия акторов с системой представлены на рисунке 6.1.

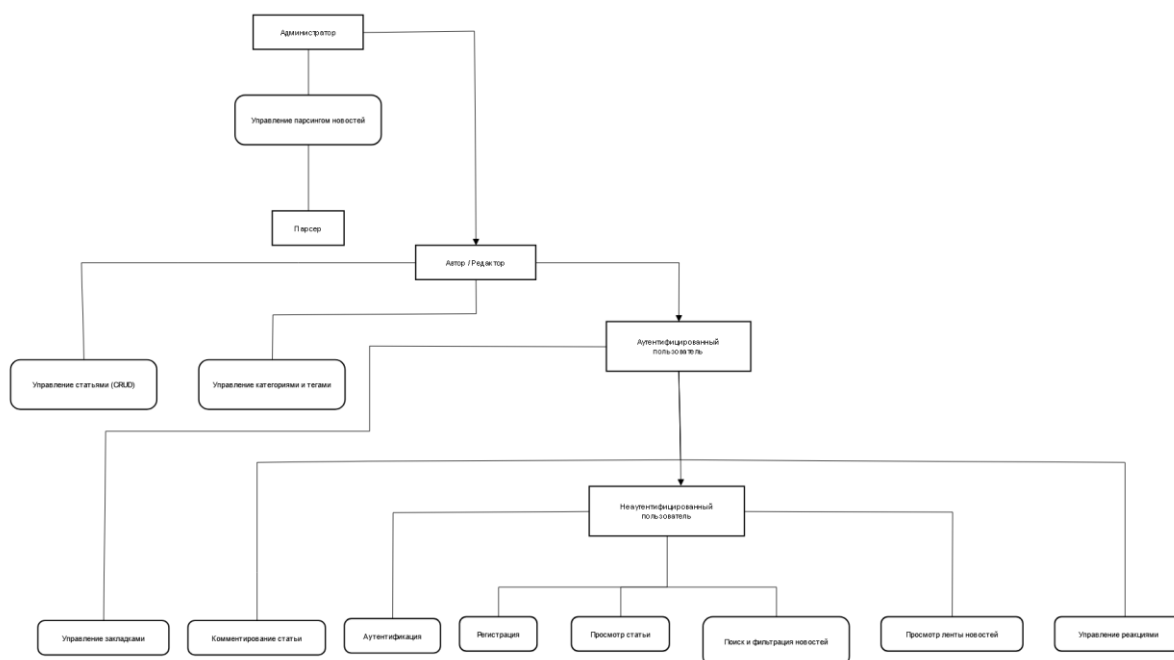


Рисунок 6.1 – Диаграмма вариантов использования.

6.1.1 Аутентификация пользователя

Акторы: Неаутентифицированный пользователь.
Описание: Позволяет пользователю войти в систему для доступа к

персонализированным функциям, таким как комментирование, реакции и управление закладками.

Предусловие: Пользователь имеет зарегистрированную учетную запись и находится на сайте.

Основной поток событий:

1. Пользователь выбирает опцию «Войти».
2. Система отображает форму входа с полями «Email» и «Пароль».
3. Пользователь вводит свои учетные данные и нажимает кнопку «Войти».
4. Система проверяет введенные данные в базе данных (сравнивает хэш пароля).
5. Система создает для пользователя безопасную сессию.
6. Система перенаправляет пользователя на главную страницу портала, отображая его как авторизованного (например, показывая его имя в шапке сайта).

Альтернативный поток A1: Неверные учетные данные

1. На шаге 4 основной последовательности система не находит пользователя с таким email или пароль неверный.
2. Система отображает сообщение об ошибке: «Неверный email или пароль».
3. Пользователь остается на странице входа с возможностью повторить попытку.

Постусловие: Пользователь успешно аутентифицирован в системе и имеет доступ к функциям для зарегистрированных пользователей.

6.1.2 Регистрация пользователя

Актеры: Неаутентифицированный пользователь.

Описание: Позволяет новому пользователю создать учетную запись в системе.

Предусловие: Пользователь находится на сайте и еще не имеет учетной записи.

Основной поток событий:

1. Пользователь выбирает опцию «Регистрация».
2. Система отображает форму регистрации с полями «Имя пользователя», «Email», «Пароль» и «Подтверждение пароля».
3. Пользователь заполняет поля и нажимает кнопку «Зарегистрироваться».
4. Система проверяет уникальность email и валидность введенных данных (например, совпадение паролей).
5. Система создает новую учетную запись в базе данных, сохраняя пароль в хэшированном виде.
6. Система автоматически аутентифицирует нового пользователя и перенаправляет его на главную страницу.

Альтернативный поток A1: Email уже существует

1. На шаге 4 система обнаруживает, что введенный email уже зарегистрирован.

2. Система отображает сообщение: «Пользователь с таким email уже существует. Пожалуйста, войдите в систему».

3. Процесс регистрации прекращается, пользователь остается на странице регистрации.

6.1.3 Поиск и фильтрация новостей

Актеры: Неаутентифицированный пользователь, Аутентифицированный пользователь.

Описание: Позволяет пользователю находить статьи по ключевым словам и применять фильтры по категориям, тегам и дате.

Предусловие: Пользователь находится на сайте.

Основной поток событий:

1. Пользователь вводит поисковый запрос в поле поиска.

2. Пользователь нажимает кнопку «Найти».

3. Система выполняет поиск по заголовкам и текстам статей в базе данных.

4. Система отображает страницу с результатами поиска – список статей, отсортированных по релевантности или дате.

5. Пользователь видит панель фильтров (категории, теги, диапазон дат).

6. Пользователь выбирает один или несколько фильтров (например, категорию «Технологии»).

7. Система немедленно обновляет список найденных статей в соответствии с фильтрами.

8. Пользователь выбирает статью из списка для просмотра.

Альтернативные потоки:
A1: Поиск не дал результатов

1. На шаге 4 система не находит статьи.

2. Система отображает сообщение: «По вашему запросу ничего не найдено. Попробуйте изменить формулировку».

6.1.4 Просмотр новостной статьи

Актеры: Неаутентифицированный пользователь, Аутентифицированный пользователь.

Описание: Позволяет пользователю открыть и прочитать полный текст выбранной новостной статьи, а также увидеть связанные с ней комментарии и реакции.

Предусловие: Пользователь находится на сайте (на главной странице или странице результатов поиска).

Основной поток событий:

1. Пользователь нажимает на заголовок или изображение статьи в ленте новостей.

2. Система загружает и отображает полную страницу статьи.

3. Страница содержит заголовок, полный текст, дату публикации, категорию и теги.

4. Под статьей отображается блок с комментариями и форма для добавления нового комментария (если пользователь аутентифицирован).

Альтернативный поток A1: Статья не найдена

1. На шаге 2 система не может найти статью по указанному идентификатору (например, если она была удалена).

2. Система отображает страницу ошибки 404 с сообщением «Статья не найдена».

6.2 Диаграммы активностей

Диаграммы последовательности детализируют взаимодействие между различными компонентами системы во времени. В контексте архитектуры PulseNews, они наглядно демонстрируют, как действия пользователя в клиентской части инициируют запросы к серверной части, которая, в свою очередь, обращается к базе данных.

Основная цель данных диаграмм — показать точный порядок обмена сообщениями и вызовов методов между объектами для реализации конкретных вариантов использования.

6.2.1 Аутентификация пользователя

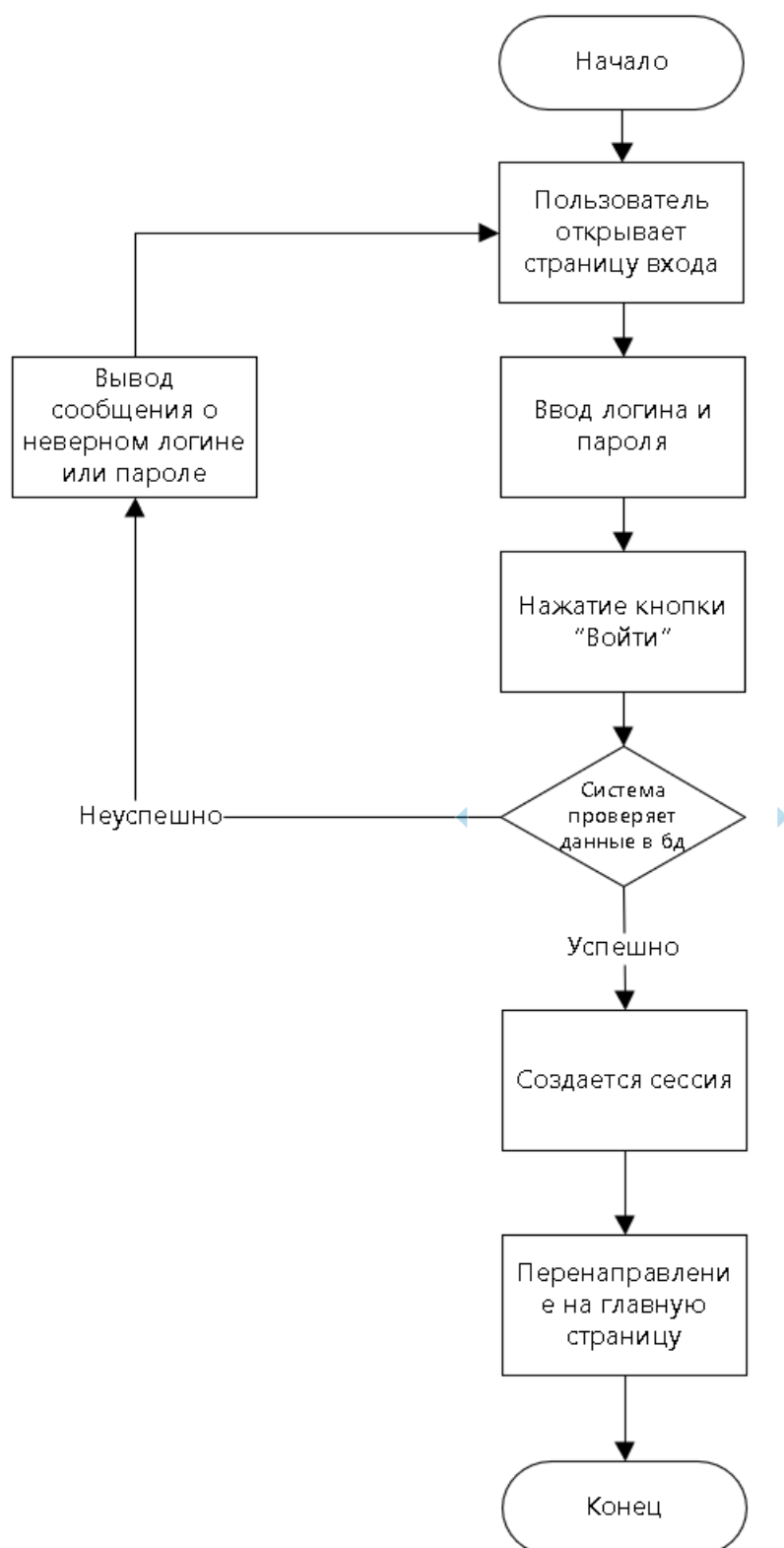


Рисунок 6.2 – Аутентификация пользователя.

6.2.2 Управление статьями

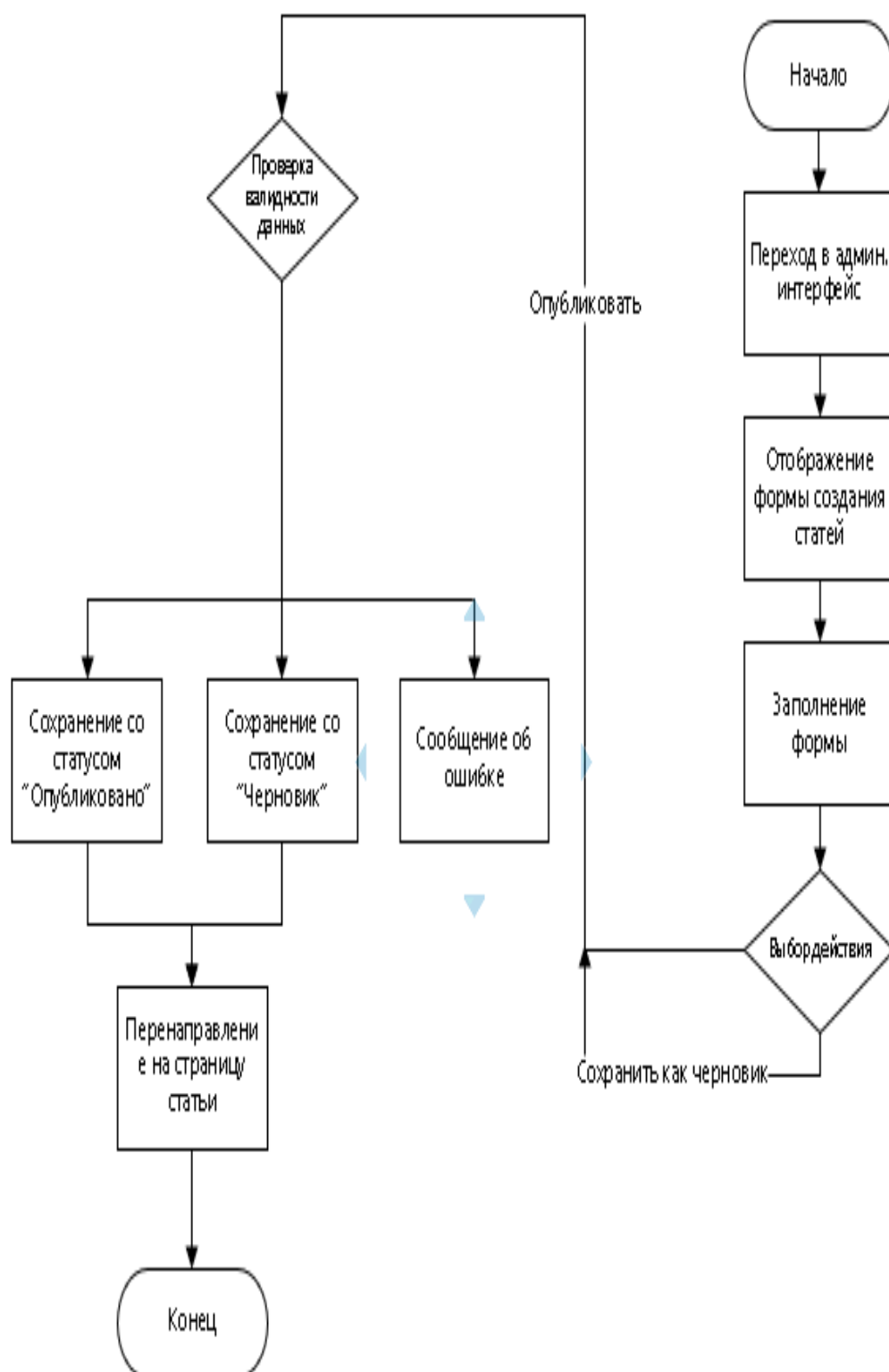


Рисунок 6.3 – Управление статьями.

6.3 Диаграммы последовательности

Диаграммы последовательности детализируют взаимодействие между различными компонентами системы во времени. В контексте архитектуры PulseNews, они наглядно демонстрируют, как действия пользователя в клиентской части инициируют запросы к серверной части, которая, в свою очередь, обращается к базе данных.

Основная цель данных диаграмм — показать точный порядок обмена сообщениями и вызовов методов между объектами для реализации конкретных вариантов использования.

6.3.1 Комментирование статьи

Данная диаграмма последовательности иллюстрирует процесс, описанный в варианте использования UC6: «Комментирование статьи». Она пошагово демонстрирует взаимодействие между аутентифицированным пользователем, клиентским приложением, сервером и базой данных в момент отправки и сохранения нового комментария. Особое внимание уделено HTTP-запросам (POST), проверке токена аутентификации и SQL-запросам на запись данных.

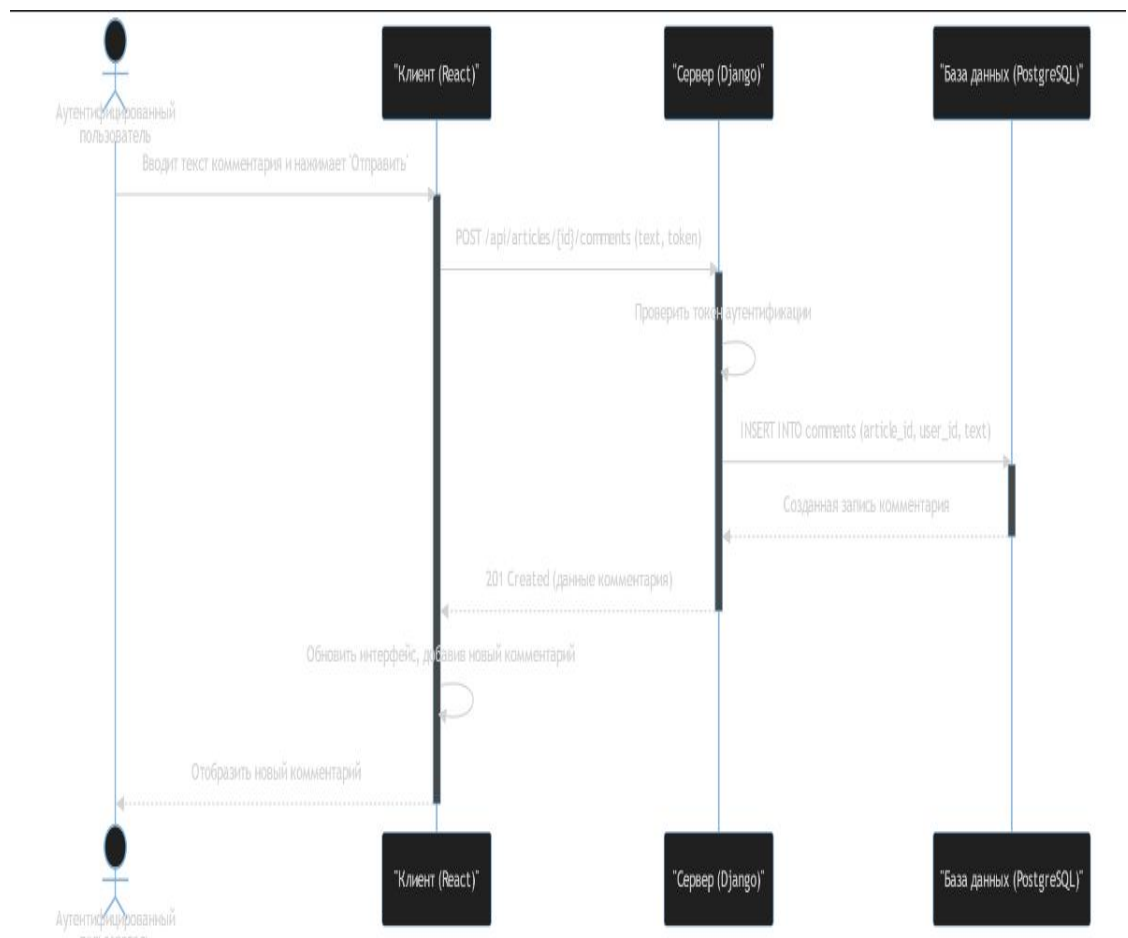


Рисунок 6.4 – Комментирование статьи

6.3.2 Добавление статьи в закладки

Эта диаграмма детализирует шаги, выполняемые системой, когда аутентифицированный пользователь добавляет статью в свой личный список закладок. Она показывает, как нажатие на иконку закладки в интерфейсе приводит к отправке запроса на сервер, который верифицирует пользователя и сохраняет соответствующую запись в базу данных. Диаграмма также охватывает обратный процесс — удаление статьи из закладок.



Рисунок 6.5 – Добавление статьи в закладки.

6.4 Диаграммы состояний

6.4.1 Состояния статьи

Ниже представлена диаграмма активности, которая визуализирует пошаговый сценарий входа пользователя в систему PulseNews, начиная от открытия страницы входа и заканчивая перенаправлением в личный кабинет.

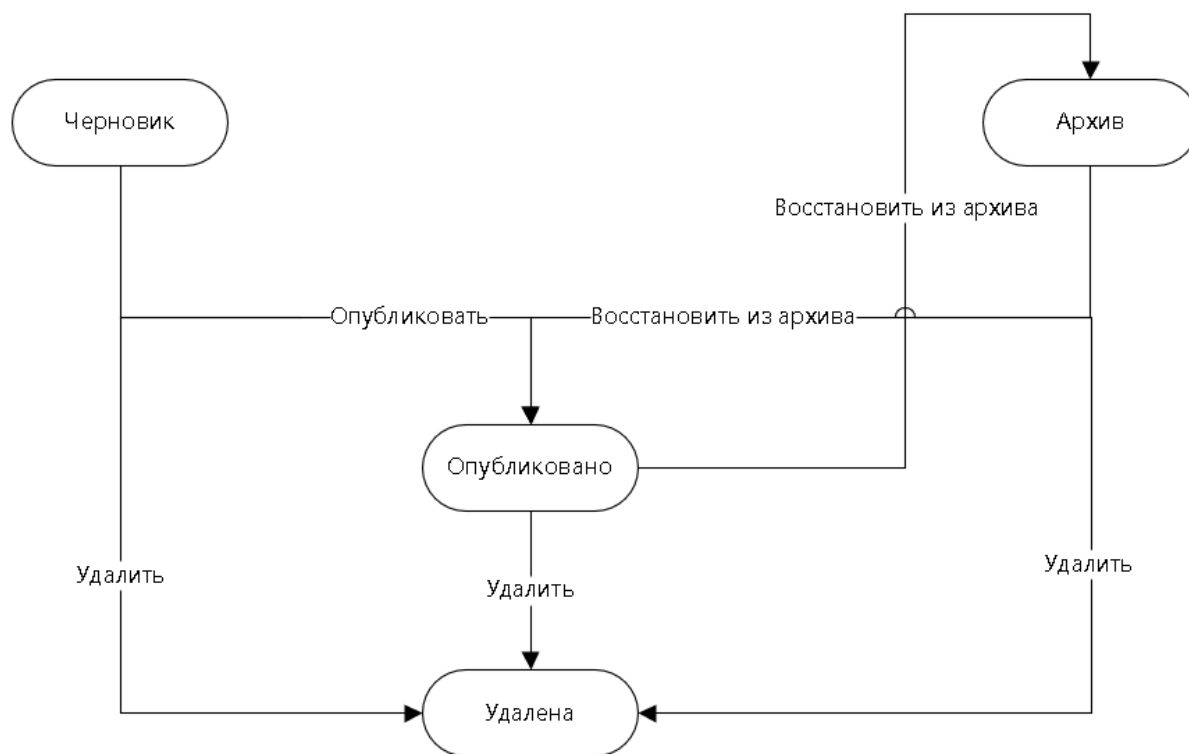


Рисунок 6.6 – Состояния статьи

7 ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы была проведена комплексная работа по анализу и проектированию веб-приложения PulseNews — современного новостного онлайн-портала, ориентированного на создание интерактивного сообщества. Основной целью проекта является предоставление пользователям не только инструмента для потребления новостей, но и платформы для их активного обсуждения и персонализации.

В рамках работы были успешно решены все поставленные задачи. Были детально определены функциональные и нефункциональные требования к системе, что позволило сформировать четкое видение конечного продукта. Были идентифицированы ключевые роли пользователей и подробно описаны сценарии их взаимодействия с системой через модели и диаграммы вариантов использования.

Особое внимание было уделено проектированию архитектуры и структуры данных. Разработанная диаграмма классов определяет основные сущности системы и связи между ними, закладывая основу для проектирования базы данных. С помощью диаграмм активностей и последовательностей были визуализированы ключевые бизнес-процессы, такие как аутентификация, публикация и комментирование статей, что позволило выявить и устранить потенциальные логические проблемы на раннем этапе.

В результате проделанной работы был создан полный пакет проектной документации, который служит исчерпывающим техническим заданием и руководством для команды разработчиков. Предложенная архитектура, основанная на современных технологиях, обеспечивает надежность, масштабируемость и высокую производительность будущей системы.