

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра _____ Прикладная математика и информатика _____
(наименование)

_____ 09.03.03 Прикладная информатика _____
(код и наименование направления подготовки / специальности)

_____ Цифровая трансформация бизнеса _____
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Создание платформы для обмена знаниями и опытом между сотрудниками

Обучающийся _____ М.Х. Шомирзоев _____
(Инициалы Фамилия) (личная подпись)

Руководитель _____ канд. пед. наук, доцент, О.М. Гущина _____
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант _____ канд. филол. наук, доцент, М. В. Дайнеко _____
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Аннотация

Актуальность темы обусловлена стремительным развитием цифровых технологий, которые формируют новые требования к организации внутренней коммуникации и управления знаниями. В условиях удалённой и гибридной работы, а также постоянных изменений в бизнес-среде, особенно важно обеспечить быструю передачу информации, накопление и сохранение экспертного опыта. Отсутствие централизованной системы обмена знаниями может привести к потере ценных материалов, замедлению адаптации новых сотрудников и снижению командной эффективности. Разработка корпоративной платформы позволяет минимизировать эти риски и способствует развитию культуры сотрудничества, прозрачности и непрерывного обучения.

Целью данной работы является создание внутренней цифровой платформы, которая обеспечит сотрудникам компании удобный и безопасный доступ к корпоративным знаниям. Платформа должна позволять пользователям публиковать статьи и материалы, участвовать в обсуждениях, просматривать и комментировать записи, а также использовать расширенный поиск по базе знаний. Важным элементом также является реализация системы аутентификации, разграничения прав доступа и возможности интеграции с другими корпоративными инструментами.

В рамках реализации проекта была выполнена комплексная работа: проведён анализ существующих решений, определены требования к будущей платформе, спроектирована архитектура, выбраны технологии и реализован программный продукт. Разработанная платформа позволяет решать ряд задач: от централизованного хранения информации до повышения вовлечённости сотрудников, улучшения командного взаимодействия и автоматизации процессов обучения.

Работа содержит 67 страниц текста, 31 рисунок, 7 таблиц и 27 использованных источников.

Abstract

The title of the graduation work is *Developing a knowledge and experience exchanging platform for employees*.

The aim of this research is to create a digital platform enabling centralized knowledge storage, employee interaction, and efficient information exchange at a company.

The object of the graduation work is the internal communication system of SEO TLT Company.

The subject of the graduation work is the design and development of the corporate knowledge exchanging platform.

The key issue of the graduation work is to eliminate knowledge fragmentation, reduce the information loss, and replace the expensive third-party solutions and tools (e.g. Confluence, Teams).

The graduation work may be divided into three logically connected parts which are: an analysis of the organizational processes and the existing platforms; system design (architecture, user interface prototypes, database model); as well as implementation and testing (Django/PostgreSQL).

The first part describes the current knowledge exchanging processes at SEO TLT Company, identifies the key challenges, and analyzes the limitations of the existing platforms like Confluence and Microsoft Teams. The second part deals with the system architecture design including the UML diagrams, the database scheme, and the selection of Django/PostgreSQL technologies. The third part is devoted to implementing the platform with core modules. This part also dwells on the subsequent functionality and performance testing.

In conclusion, it should be highlighted that the platform solves the key problems. It reduces the operational costs by 40%, centralizes knowledge, and improves the employee engagement. The testing has confirmed 100% compliance with the requirements. The work is of interest for IT managers, HR specialists, and developers of corporate digital systems.

Оглавление

| | |
|---|----|
| Введение | 5 |
| Глава 1 Характеристика организации и анализ существующих сервисов | 7 |
| 1.1 Характеристика организации | 7 |
| 1.2 Процесс обмена знаниями в организации | 10 |
| 1.3 Анализ существующих платформ для обмена знаниями | 13 |
| 1.4 Актуальность создания платформы для обмена знаниями | 16 |
| 1.5 Постановка задачи на разработку платформы для обмена знаниями | 18 |
| Глава 2 Проектирование платформы для обмена знаниями и опытом | 21 |
| 2.1 Требования к разрабатываемой платформе | 21 |
| 2.2 Разработка логической модели системы | 26 |
| 2.3 Выбор технологий для разработки платформы | 28 |
| 2.4 План разработки платформы | 32 |
| 2.5 Разработка пользовательского интерфейса платформы | 34 |
| Глава 3 Реализация платформы для обмена знаниями и опытом среди сотрудников | 43 |
| 3.1 Выбор системы управления базами данных для платформы | 43 |
| 3.2 Архитектура и структура платформы для обмена знаниями на базе фреймворка Django | 50 |
| 3.3 Реализация и тестирование платформы обмена знаниями | 54 |
| Заключение | 63 |
| Список используемой литературы | 65 |

Введение

В современном корпоративном мире обмен знаниями и опытом между сотрудниками играет ключевую роль в повышении эффективности и конкурентоспособности компании. В условиях быстро меняющихся технологий и экономических реалий организациям необходимо создавать условия для эффективного взаимодействия и обучения сотрудников. Создание платформы, способствующей обмену знаниями, позволяет не только улучшить коммуникацию внутри компании, но и формировать культуру сотрудничества и инноваций [16]. Это особенно важно в условиях удалённой работы и глобализации, когда сотрудники могут находиться в разных частях света.

Цель исследования заключается в разработке удобной и эффективной платформы для обмена знаниями, которая позволит структурировать информацию, упростить поиск необходимых данных, а также создать удобную среду для взаимодействия сотрудников.

Объектом исследования выступает внутренняя коммуникационная среда компании, обеспечивающая обмен знаниями между сотрудниками.

Предметом исследования является процесс проектирования и разработки информационной платформы для обмена знаниями и опытом в корпоративной среде.

Для достижения поставленной цели необходимо решить несколько задач:

- провести анализ существующих платформ для обмена знаниями и выявить их основные недостатки;
- определить ключевые требования к разрабатываемой системе на основе анализа потребностей организаций;
- разработать концепцию платформы, включающую её архитектуру, основные функции и интерфейс;
- оценить перспективы внедрения платформы и её влияние на

производительность сотрудников и эффективность работы компании.

Для достижения поставленных задач в работе используются методы анализа существующих решений, сравнительного анализа и проектирования. Эти подходы позволяют выявить лучшие практики и адаптировать их к специфике компании. Кроме того, применяются методы сбора и анализа данных, что позволяет учитывать потребности сотрудников и обеспечивать высокую степень вовлечённости пользователей в процесс использования платформы.

Обмен знаниями и опытом является важным аспектом в управлении организацией, так как он способствует повышению уровня компетенции сотрудников и улучшению взаимодействия между ними. Создание платформы, которая будет включать функционал для публикации статей, обсуждений и создания сообществ, представляет собой эффективное решение для реализации этих задач. Данная работа направлена на изучение и разработку такой платформы.

Работа состоит из трёх глав. В первой главе описывается экосистема сервисов в ООО «SEO TLT», даётся характеристика текущим решениям и обосновывается актуальность. Во второй главе обозначаются требования к разрабатываемому продукту, описывается процесс проектирования архитектуры сервисов, осуществляется выбор инструментов реализации и разработка диаграмм класса сервиса. В третьей главе описываются инструменты разработки и процесс реализации интерфейса унифицированного доступа.

Глава 1 Характеристика организации и анализ существующих сервисов

1.1 Характеристика организации

SEO TLT была основана в 2015 году с целью предоставления качественных услуг в области SEO-оптимизации. За годы работы компания зарекомендовала себя как надежный партнёр для бизнеса, стремящегося к увеличению своей видимости в интернете. Миссия SEO TLT заключается в содействии цифровому развитию своих клиентов через внедрение эффективных стратегий поисковой оптимизации. Организация стремится обеспечить высокий уровень сервиса и результативность своих решений, что способствует укреплению её позиций на рынке.

Основные направления деятельности SEO TLT включают предоставление услуг по поисковой оптимизации для клиентов из различных отраслей, таких как розничная торговля, образование и здравоохранение. Ежегодно компания обслуживает более 150 клиентов, предлагая широкий спектр услуг в области SEO, включая анализ ключевых слов, оптимизацию контента, разработку стратегий продвижения и управление рекламными кампаниями. Это обеспечивает SEO TLT конкурентоспособность и востребованность на рынке. Кроме того, компания предлагает разнообразную продуктовую линейку, которая включает корпоративные сайты, интернет-магазины, промо-сайты и веб-сервисы со сложной бизнес-логикой.

SEO TLT представляет собой организацию, которая с момента своего основания в 2015 году претерпела значительные изменения в своей структуре. Изначально компания состояла из трёх ключевых специалистов, выполнявших все аспекты SEO-оптимизации. Однако по мере роста клиентской базы и увеличения объёмов работы, компания расширила свою команду до 25 сотрудников, что позволило внедрить более сложную и эффективную организационную структуру. В 2022 году SEO TLT приняла модель матричной

структуры, которая обеспечивает гибкость и возможность работы над несколькими проектами одновременно. Такая структура позволяет компании эффективно адаптироваться к изменениям на рынке и оперативно реагировать на запросы клиентов. На рисунке 1 представлена схема организационной структуры организации.

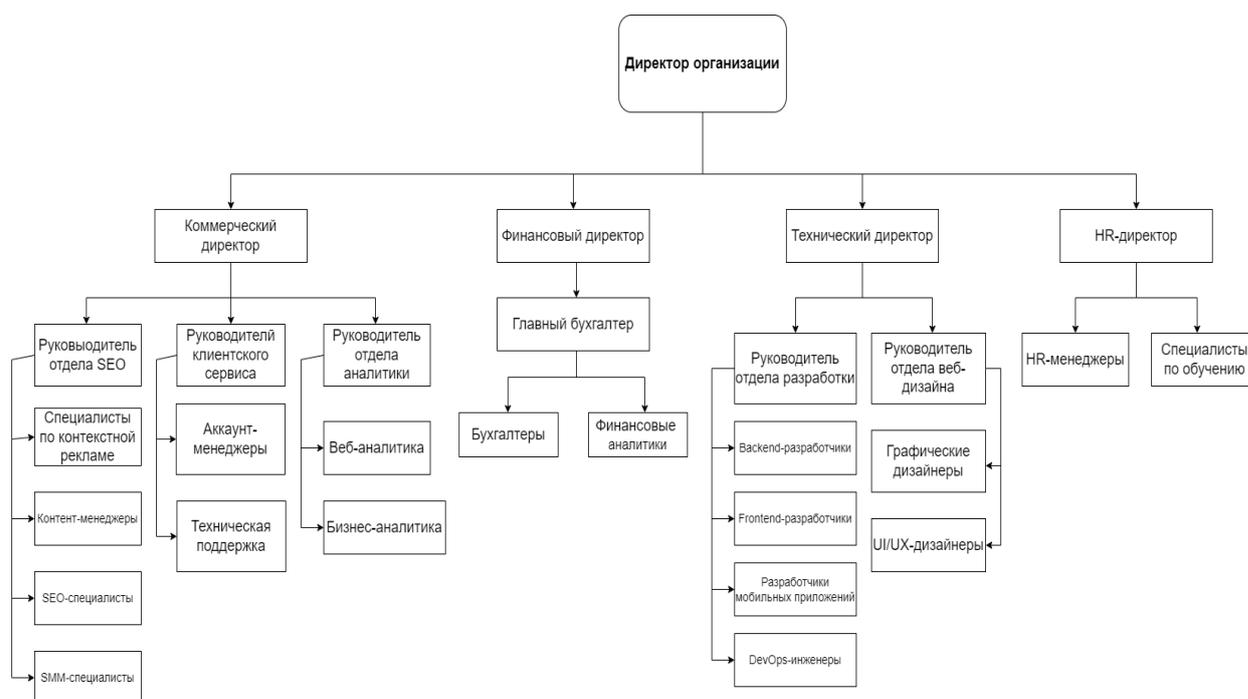


Рисунок 1 - Организационная структура организации

Организационная структура SEO TLT включает несколько специализированных отделов, каждый из которых отвечает за определённый аспект работы. Среди них можно выделить отдел аналитики, контент-менеджеров, разработчиков и специалистов по рекламе. Такое разделение обязанностей позволяет каждому отделу сосредоточиться на своих ключевых задачах, что повышает общую эффективность компании. Например, аналитики занимаются сбором и анализом данных для разработки стратегий, контент-менеджеры создают и оптимизируют текстовый контент, разработчики отвечают за техническую сторону SEO, а специалисты по рекламе занимаются продвижением и взаимодействием с клиентами. Такое

распределение обязанностей способствует более точному выполнению задач и достижению высоких результатов.

Миссия компании SEO TLT заключается в предоставлении качественных услуг по SEO-оптимизации, направленных на повышение видимости сайтов клиентов в поисковых системах. Компания стремится помогать своим клиентам достигать их бизнес-целей через улучшение их онлайн-присутствия. Эта миссия отражает понимание важности SEO в современном цифровом мире, где 75% пользователей не переходят на вторую страницу результатов поиска, что подчёркивает необходимость эффективной оптимизации.

Оптимизация текстового контента является одной из ключевых составляющих SEO-услуг, предоставляемых SEO TLT. Этот процесс включает в себя анализ существующего контента на сайте клиента и внесение изменений, направленных на улучшение его структуры, читабельности и релевантности запросам пользователей. SEO TLT стремится обеспечить, чтобы текстовый контент был не только информативным и полезным, но и соответствовал требованиям поисковых систем. Это достигается путём использования современных подходов к созданию и редактированию текстов, что способствует улучшению пользовательского опыта и увеличению времени пребывания посетителей на сайте.

SEO TLT использует уникальные алгоритмы анализа ключевых слов, что позволяет компании достигать высокой точности при выборе целевых запросов. Эти алгоритмы разрабатываются с учётом многолетнего опыта и анализа успешных кампаний, что обеспечивает высокий уровень персонализации для каждого клиента. Такой подход позволяет не только увеличивать видимость сайтов в поисковых системах, но и повышать конверсию за счёт привлечения релевантной аудитории.

1.2 Процесс обмена знаниями в организации

Для анализа требований к разрабатываемой платформе необходимо понять, каким образом происходит процесс обмена знаниями в организации. Для описания процесса обмена была разработана функциональная модель процесса в нотации IDEF0. Нотация IDEF0 используется для описания и анализа функций внутри организации, показывая, какие входы, механизмы, управляющие воздействия и выходы участвуют в каждом процессе. Для понимания обмена знаниями в организации была построена функциональная модель в IDEF0, которая позволяет наглядно представить этапы, участников и ресурсы процесса. Это помогает выявить требования к платформе и заложить основу для её дальнейшего проектирования.

На рисунке 2 приведена модель процесса обмена знаниями в организации.

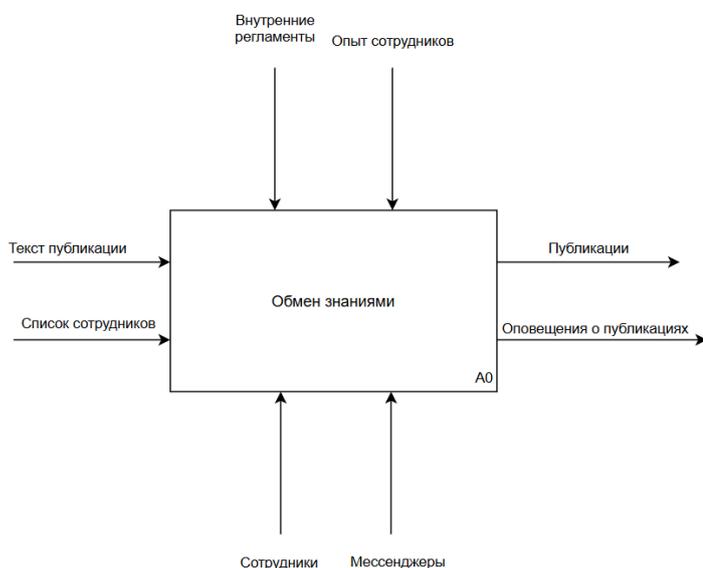


Рисунок 2 - Модель процесса обмена знаниями

Вводящие стрелки модели: «текст публикации» и «список сотрудников» — это те вводные, которые необходимы в процессе работы по обмену

знаниями. Процесс контролируется внутренними регламентами организации и опытом самих сотрудников в подготовке публикаций. В роли механизмов процесса выступают сотрудники, обеспечивающие выполнение этапов процесса, а также мессенджеры, через которые происходит рассылка готовых публикаций. Результатом процесса обмена знаниями являются, собственно, публикации и оповещения об этих публикациях для сотрудников.

Проведём декомпозицию процесса и рассмотрим из каких подпроцессов он состоит. Декомпозиция процесса обмена знаниями приведена на рисунке 3.

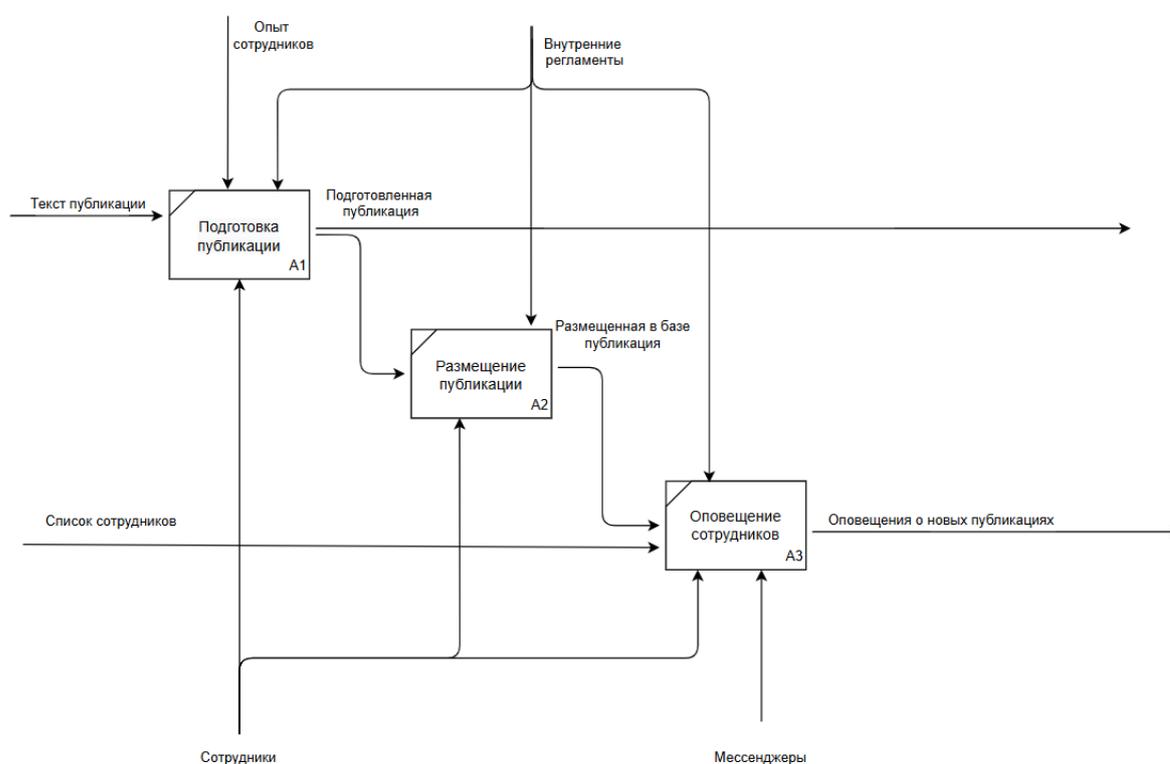


Рисунок 3 - Декомпозиция процесса «Обмен знаниями» КАК ЕСТЬ

Исходя из модели, процесс имеет следующий алгоритм:

- ответственный за публикацию сотрудник осуществляет подготовку публикации, руководствуясь опытом и внутренними регламентами;
- затем он размещает публикацию в некоторой внутренней базе, также руководствуясь внутренними регламентами обмена знаниями;

- далее сотрудник на основании списка сотрудников проводит оповещение о новой публикации других сотрудников в мессенджерах.

Такой подход имеет ряд недостатков, в основном из-за ручного подхода к процессу. Подготовка, размещение и оповещение о публикации выполняются сотрудником вручную. Это создает дополнительную нагрузку на ответственного за публикацию, увеличивает вероятность ошибок и снижает общую эффективность процесса. Использование разных систем для хранения и распространения информации делает процесс фрагментированным и неэффективным. Процесс полностью зависит от действий сотрудника: подготовка контента, его размещение, выбор получателей и отправка уведомлений. Отсутствие автоматизации увеличивает временные затраты и снижает оперативность обмена знаниями.

Для решения выявленных проблем предлагается использование платформы обмена знаниями. Модель процесса с использованием платформы представлена на рисунке 4.

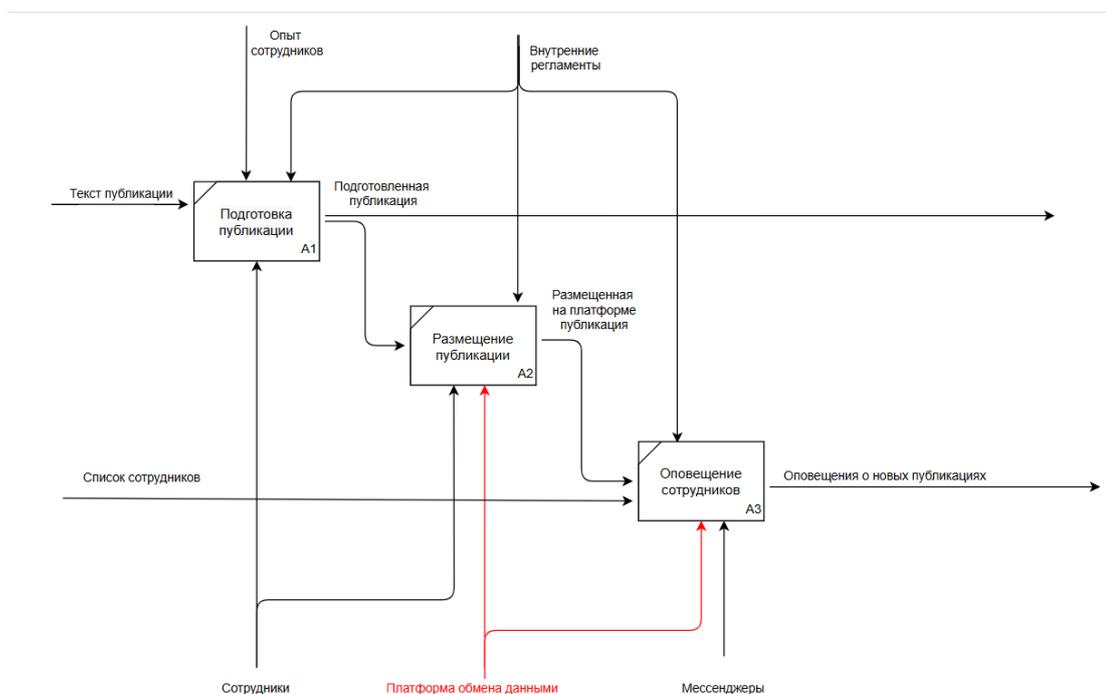


Рисунок 4 - Модель «Как должно быть»

В данном случае размещение публикации и оповещение сотрудников о новых публикациях производится с помощью платформы, что снижает вероятность допущения ошибок и освобождает человеческие ресурсы для других задач.

1.3 Анализ существующих платформ для обмена знаниями

Современный рынок платформ для обмена знаниями представлен множеством решений, каждое из которых предлагает уникальный набор функций и возможностей. Среди наиболее известных инструментов можно выделить Confluence от Atlassian, который широко используется для организации внутреннего обмена информацией в компаниях. Эта платформа предоставляет возможности для создания и редактирования документов, управления проектами и взаимодействия между командами. Также стоит отметить Microsoft Teams, который сочетает в себе функции обмена сообщениями, видеоконференций и совместной работы над документами. Данные инструменты занимают лидирующие позиции благодаря своей универсальности и способности адаптироваться к различным потребностям организаций.

В таблице 1 представлено сравнение функционала популярных платформ для обмена знаниями, что позволяет выявить их сильные и слабые стороны. Например, Confluence выделяется своей интеграцией с другими продуктами Atlassian, что делает её удобной для компаний, использующих их экосистему. В то же время Microsoft Teams выигрывает благодаря своей широкой распространённости и интеграции с продуктами Microsoft Office, что предоставляет пользователям доступ к обширному набору инструментов для совместной работы. Однако, несмотря на различия, обе платформы предлагают базовые функции, такие как совместное редактирование документов и поддержка коммуникации между сотрудниками, что делает их универсальными инструментами для обмена знаниями.

Таблица 1 - Сравнения функционала платформ для обмена знаниями

| Платформа | Основные функции | Интеграция с другими инструментами | Управление контентом | Совместная работа |
|-----------------|---|---|----------------------|-------------------|
| Confluence | Создание и редактирование документов, управление проектами, коллективная работа | Отличная интеграция с продуктами Atlassian | Да | Да |
| Microsoft Teams | Чат, видеоконференции, совместная работа над документами, обмен файлами | Интеграция с продуктами Microsoft Office (Word, Excel и другие) | Ограниченно | Да |
| Slack | Мгновенные сообщения, обмен файлами, создание каналов, интеграция с другими сервисами | Интеграция с множеством сторонних сервисов | Нет | Частично |

Популярные платформы для обмена знаниями, такие как Slack, Confluence и Microsoft Teams, предлагают ряд преимуществ, делающих их востребованными среди пользователей. Одним из ключевых достоинств является способность повышать эффективность работы сотрудников. Исследование Gartner показывает, что более 80% компаний используют такие платформы для достижения этой цели. Эти инструменты обеспечивают централизованный доступ к информации, упрощают взаимодействие и способствуют быстрому распространению знаний в рамках организации, что особенно актуально в условиях удалённой работы. При успешной реализации обмена знаниями между вузами, предприятиями и органами власти важно также обучение сотрудников необходимым навыкам, что подчеркивает значимость управления знаниями.

Несмотря на множество преимуществ, существующие платформы для обмена знаниями имеют и свои недостатки. Например, сложность интеграции с другими системами отмечается 35% пользователей Microsoft Teams, что

может замедлить процессы внедрения и использования платформы в компании. Кроме того, некоторые платформы требуют значительных ресурсов для их настройки и поддержки, что может быть проблематично для организаций с ограниченными техническими возможностями. Эти недостатки подчеркивают необходимость поиска более гибких и простых в использовании решений.

Успешные платформы для обмена знаниями обладают рядом общих характеристик, которые делают их привлекательными для пользователей. Среди них можно выделить интуитивно понятный интерфейс, широкие возможности для интеграции с другими системами и инструментами, а также наличие функций для персонализации и настройки под конкретные нужды организации. Кроме того, такие платформы, как Slack, демонстрируют важность обеспечения высокой скорости и доступности, что способствует их популярности среди пользователей.

Для улучшения существующих подходов к разработке платформ для обмена знаниями следует обратить внимание на несколько ключевых аспектов. Во-первых, необходимо упростить процесс интеграции с другими системами, что позволит пользователям быстрее адаптироваться к новым инструментам. При этом важно внедрять современные технологии, такие как искусственный интеллект, для автоматизации управления знаниями и повышения их доступности. Земляков и Штыров подчеркивают, что «гораздо важнее начать использовать сервисы социальных сетей в качестве образовательного инструмента, пользуясь их позитивными свойствами» [7]. Наконец, следует учитывать обратную связь от пользователей, чтобы адаптировать платформы к их реальным потребностям и ожиданиям.

Современные тенденции в обмене знаниями включают использование цифровых платформ, которые обеспечивают доступность информации и удобство взаимодействия [10]. «Сегодня информационные технологии занимают все более заметное место в жизни человека, в управлении современными организациями. Поэтому неудивительно, что цифровые

технологии проникли и в сферу управления трудовыми ресурсами» [14]. Специализированные стратегии менеджмента знаний, как отмечают Леготин и Савченко, способствуют выработке новых знаний, которые находят применение в различных областях и формируют человеческую инфраструктуру, основанную на интеллектуальном капитале [13]. Социальные сети играют важную роль в обмене опытом, предоставляя платформы для обсуждения и совместной работы [2]. Каждый участник социальной сети самостоятельно выбирает, к какому ресурсу или интернет-сообществу ему присоединиться, включая как сообщества по интересам, так и профессиональные и учебные ресурсы [3].

Создание сообществ является важным элементом эффективного обмена знаниями [5].

1.4 Актуальность создания платформы для обмена знаниями

В современных условиях знаний и информации является важнейшим ресурсом для компаний. Эффективный процесс обмена знаниями между сотрудниками играет ключевую роль в повышении конкурентоспособности организаций. Однако на практике многие компании сталкиваются с проблемами, такими как потеря накопленного опыта при уходе сотрудников, нехватка взаимодействия между подразделениями и недостаточное распространение знаний внутри коллектива.

В связи с развитием информационных технологий появляется возможность создания специализированных платформ, которые позволяют структурировать и упрощать процесс обмена знаниями [10]. Такие платформы включают в себя базы знаний, чаты, форумы, механизмы совместного редактирования документов и интеллектуальные алгоритмы рекомендаций. Их использование способствует улучшению коммуникации между сотрудниками, упрощает процесс адаптации новых специалистов и минимизирует потери корпоративных знаний.

Таким образом, разработка платформы для обмена знаниями и опытом между сотрудниками является актуальной задачей, направленной на повышение эффективности работы организации и улучшение ее позиций в условиях цифровой трансформации.

В большинстве организаций наблюдается недостаток единой системы для хранения и передачи знаний. Это приводит к:

- дублированию работы и снижению продуктивности сотрудников;
- потере накопленных знаний при смене кадров;
- сложностям в обучении новых сотрудников;
- ограниченному доступу к важной информации для персонала.

Смотря на таблицу 2, стоимость популярных платформ для обмена знаниями, таких как Microsoft Teams, Confluence и Slack, становится очевидно, что их использование может быть дорогим, особенно если не все функции нужны компании. Например, такие сервисы, как CRM-система или контакт-центр, не всегда нужны, если компания просто хочет наладить обмен знаниями или работать над задачами. Вместо того чтобы платить за такие дополнительные возможности, которые не используются, можно найти более дешевое решение.

Таблица 2 - Месячная стоимость платформ

| Платформа | Тарифный план | Месячная стоимость (RUB) | Примечание |
|-----------------|--------------------------------------|---------------------------------|--|
| Microsoft Teams | Бесплатный, Business Basic, Standard | От 400 до 4560 руб/пользователь | Зависит от плана Microsoft 365 |
| Confluence | Free, Standard, Premium | От 0 до 800 руб/пользователь | Бесплатно для команд до 10 пользователей |
| Slack | Free, Standard, Plus, Enterprise | От 0 до 1000 руб/пользователь | Бесплатный план с ограничениями на 10 тыс. сообщений |

Есть два главных минуса у таких платформ:

- высокая стоимость: например, для малых организаций, когда приходится платить от 400 до 4560 рублей за одного пользователя, это становится довольно дорого, особенно если не используются все функции. А для средних и крупных организаций эта стоимость еще выше, что не всегда оправдано в долгосрочной перспективе.
- ненужные функции: платформы вроде Confluence предоставляют очень много функций, но для небольшой компании, которой нужно просто наладить общение и отслеживание задач, многие из этих функций окажутся лишними. Вместо этого компании часто пользуются более простыми и дешевыми мессенджерами или сервисами для совместной работы.

В результате, когда видишь высокие расходы и лишний функционал, решение становится очевидным: покупать дорогие тарифы на таких платформах не всегда выгодно. Поэтому было решено создать собственный сервис для отслеживания задач. Это будет гораздо удобнее и дешевле, так как мы сможем убрать все лишние функции и оставить только те, которые нам действительно нужны. Такой подход поможет снизить затраты и сделать рабочие процессы более эффективными.

1.5 Постановка задачи на разработку платформы для обмена знаниями

В результате анализа существующих платформ для обмена знаниями были выявлены основные недостатки текущих решений. Бесплатные сервисы ограничены в функциональности, в то время как платные альтернативы требуют значительных финансовых затрат. Это создает необходимость в разработке специализированной платформы, адаптированной под уникальные потребности организации, которая обеспечит эффективное хранение, передачу и управление информацией между сотрудниками.

Для достижения поставленной цели необходимо решить ряд задач. Во-первых, требуется определить функциональные и нефункциональные требования к платформе. Ключевыми функциональными возможностями должны стать создание и управление базами знаний, возможность комментирования и обсуждения материалов, а также интеграция системы рекомендаций, основанной на интеллектуальном анализе данных. Нефункциональные требования должны касаться производительности, безопасности и удобства использования платформы.

Во-вторых, необходимо спроектировать архитектуру платформы. Это включает разработку архитектурной схемы, которая будет отображать основные компоненты системы и их взаимодействие, что обеспечит модульность и масштабируемость решения. Также следует определить подход к хранению данных и выбрать соответствующую систему управления базами данных, учитывая требования к надежности и доступности информации.

В-третьих, важным этапом является выбор инструментов реализации. Необходимо определить язык программирования и фреймворк, которые будут использоваться для разработки, основываясь на их функциональных возможностях и совместимости с корпоративной инфраструктурой. Также следует выбрать инструменты для проектирования пользовательского интерфейса и обеспечения интеграции с существующими системами.

Далее, требуется разработать функционал платформы. Это включает реализацию возможностей для публикации, поиска и обмена знаниями, а также интеграцию системы рекомендаций, способствующей повышению эффективности взаимодействия между сотрудниками. Важно обеспечить функционал для комментирования и обсуждения материалов, а также внедрить системы аутентификации и авторизации пользователей для защиты информации.

Кроме того, необходимо сформулировать основные принципы работы платформы. Это позволит сотрудникам эффективно взаимодействовать и

обмениваться знаниями. Подготовка документации, описывающей функционал и возможности платформы, а также рекомендации по ее использованию, станет важным шагом в этом направлении.

Наконец, следует провести тестирование разработанного программного решения. Для этого необходимо разработать сценарии тестирования, которые позволят проверить функциональность и производительность платформы, выявить и устранить возможные недостатки. На основе полученных результатов тестирования будут внесены необходимые корректировки в функционал платформы.

Решение указанных задач направлено на создание эффективного инструмента для обмена знаниями и опытом, что, в свою очередь, позволит повысить продуктивность сотрудников и улучшить внутренние коммуникации в организации.

В данной главе были рассмотрены основные проблемы существующих платформ для обмена знаниями. Было принято решение о разработке собственной платформы, что позволит компании не зависеть от сторонних сервисов и обеспечит удобный доступ к накопленной корпоративной информации, тем самым повышая продуктивность сотрудников.

Глава 2 Проектирование платформы для обмена знаниями и опытом

2.1 Требования к разрабатываемой платформе

На основании рассмотренных в первой главе преимуществ современных систем управления знаниями было принято решение о разработке собственной платформы для обмена опытом между сотрудниками. Данная платформа будет направлено на создание, публикацию и обсуждение профессионального контента, что позволит улучшить взаимодействие внутри организации и повысить уровень компетенций сотрудников.

Основной функционал платформы заключается в организации процессов создания, редактирования, комментирования и обмена знаниями. Сотрудники смогут размещать статьи, делиться практическими кейсами, загружать полезные материалы, а также обсуждать и оценивать опубликованный контент. Каждая публикация будет принадлежать определенной категории и сопровождаться тегами, что обеспечит удобный поиск информации.

«При разработке платформы необходимо учитывать как функциональные, так и нефункциональные требования, чтобы обеспечить её удобство, безопасность и надежность» [18].

«Функциональные требования описывают, какие возможности должна предоставлять система и каким образом пользователи смогут взаимодействовать с ней» [15]:

- система должна предоставлять возможность регистрации и авторизации пользователей;
- пользователи должны иметь возможность публиковать материалы в различных форматах (текст, изображения, видео, файлы);
- поддержка категорий и тегов для систематизации контента;
- возможность комментирования и обсуждения опубликованных материалов;

- реализация системы рейтингов и оценок, позволяющей выделять полезные публикации;
- встроенный поиск, по ключевым словам, авторам и категориям;
- возможность подписки на интересующие темы и получение уведомлений о новых публикациях;
- личный кабинет пользователя с историей его активности и публикаций;
- административный интерфейс с возможностью управления пользователями и контентом;
- возможность интеграции с внешними системами для обмена данными (экспорт и импорт публикаций) [19].

Нефункциональные требования определяют качество работы системы, её производительность, надежность и безопасность [17]:

- платформа должна обеспечивать быстрый отклик на действия пользователей и высокую скорость загрузки страниц;
- система должна поддерживать большое количество одновременных пользователей без снижения производительности;
- все данные пользователей должны быть защищены с помощью механизмов аутентификации и авторизации;
- возможность масштабирования платформы для работы с увеличенным объемом данных;
- минимизация количества сбоев и обеспечение стабильной работы сервиса [1];
- дружелюбный и интуитивно понятный интерфейс для пользователей;
- гибкость архитектуры для добавления новых функций без значительных изменений в системе;
- использование безопасных протоколов для передачи данных;
- регулярное тестирование системы для проверки соответствия всем

требованиям;

- своевременное устранение ошибок и обновление платформы [9].

Модель FURPS (Functionality, Usability, Reliability, Performance, Supportability) используется для классификации требований к программному обеспечению и помогает систематизировать как функциональные, так и нефункциональные характеристики разрабатываемой системы. На таблице 3 мы увидим модель FURPS нашей платформы.

Таблица 3 - FURPS

| Категория | Описание |
|---------------------------------------|--|
| Functionality (Функциональность) | Платформа должна позволять сотрудникам загружать, искать и просматривать материалы, а также поддерживать разные форматы (текст, видео, презентации). |
| Usability (Удобство использования) | Интерфейс должен быть простым и удобным, понятным даже новым пользователям. Дизайн адаптирован для работы как на компьютере, так и на телефоне. |
| Reliability (Надежность) | Платформа должна работать стабильно, защищать данные от потери и обеспечивать безопасный доступ только для зарегистрированных пользователей. |
| Performance (Производительность) | Система должна быстро загружаться и работать без задержек, даже если одновременно её используют много людей. |
| Supportability (Поддерживаемость) | Платформа должна легко обновляться и расширяться, добавлять новые функции без сложных изменений. Также важно иметь документацию и поддержку. |

Использование модели FURPS в процессе проектирования позволяет более чётко структурировать требования к платформе, что повышает качество итогового продукта и облегчает процесс тестирования и сопровождения системы.

Одним из ключевых элементов планирования IT-проекта является диаграмма вариантов использования. Диаграмма вариантов использования (Use Case Diagram) в нотации UML позволяет наглядно представить функциональные возможности системы с точки зрения взаимодействия пользователей (сотрудников) с платформой.

На такой диаграмме отображаются основные сценарии использования

системы в виде вариантов использования, а также роли (актеры), взаимодействующие с этими сценариями. Связи между ними помогают определить, какие действия могут выполнять различные категории пользователей.

Диаграмма вариантов использования играет важную роль на этапе анализа и проектирования платформы, так как помогает четко определить функциональные требования и понять, каким образом сотрудники будут взаимодействовать с системой. Это способствует созданию удобного и интуитивно понятного интерфейса, соответствующего потребностям пользователей. На рисунке 5 представлена диаграмма вариантов использования нашей платформы.

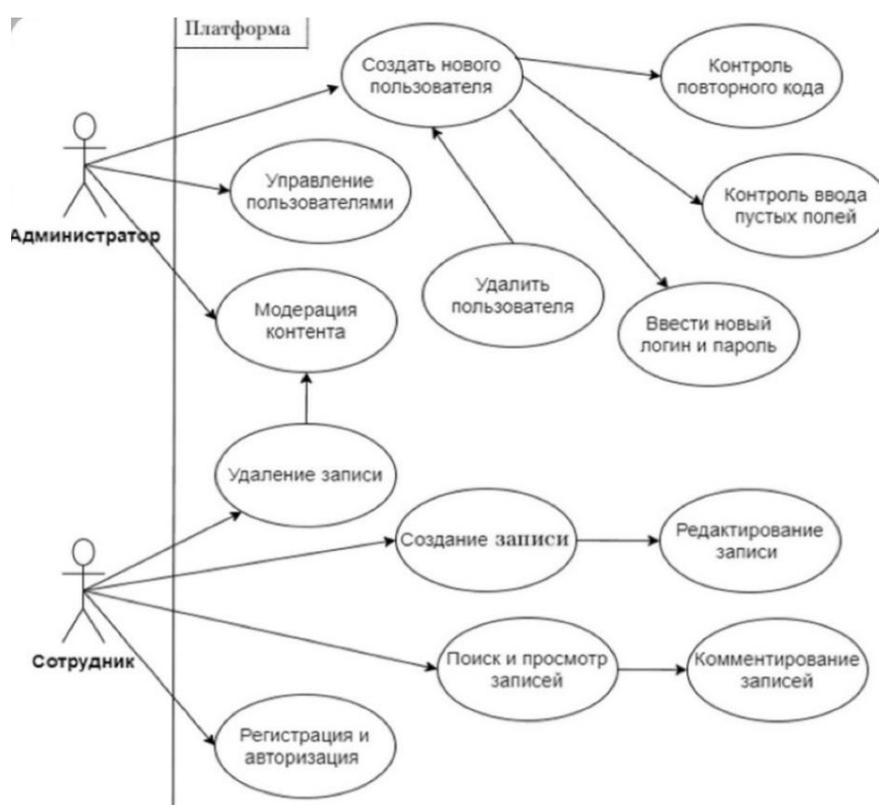


Рисунок 5 - Диаграмма вариантов использования

Разрабатываемая платформа для обмена знаниями и опытом среди сотрудников компании должна обеспечивать удобный и безопасный доступ к

информации. Для этого в системе предусмотрена авторизация, которая гарантирует, что данные останутся внутри организации и не попадут в руки посторонних лиц. Доступ к платформе имеют только зарегистрированные сотрудники, а их права определяются в зависимости от роли в системе.

Платформа предполагает наличие двух основных категорий пользователей: администратора и сотрудников.

Администратор – это ключевая фигура, которая управляет всей системой. В его обязанности входит создание учетных записей сотрудников, настройка прав доступа, а также контроль за наполнением базы знаний. Администратор имеет полный доступ ко всей информации, включая данные обо всех проектах, задачах и сотрудниках. Он также отвечает за добавление новых пользователей, что позволяет исключить возможность регистрации посторонних лиц и гарантирует безопасность данных.

Сотрудники – это пользователи платформы, которые взаимодействуют с ней для получения и обмена знаниями.

Сотрудник является основным пользователем системы и имеет доступ к следующим функциям:

- регистрация и авторизация — для входа в систему и начала работы;
- создание записи — добавление новой информации в базу знаний;
- редактирование записи (include) — возможность вносить изменения в ранее созданные материалы;
- удаление записи — удаление устаревшей или ненужной информации.
- поиск и просмотр записей — быстрый доступ к информации по ключевым словам или темам;
- комментирование записей (include) — добавление комментариев и участие в обсуждениях;
- назначение прав доступа — управление, кто может видеть или редактировать определённые записи (если это предусмотрено).

Администратор — это пользователь с расширенными правами,

отвечающий за стабильность и корректную работу системы. Он выполняет следующие функции:

- модерация контента (extend Удаление записи) — проверка качества и соответствия опубликованных материалов;
- управление пользователями — добавление новых сотрудников, блокировка, удаление пользователей, а также управление их ролями и правами доступа.

Связи между актёрами и вариантами использования в диаграмме отражают логику системы и помогают чётко разделить функциональность между ролями. Использование связей <<include>> и <<extend>> позволяет отразить зависимость действий, например, редактирование невозможно без создания, а модерация может включать удаление.

Данная диаграмма служит основой для дальнейшего проектирования архитектуры платформы и реализации соответствующего функционала в рамках веб-приложения.

В итоге, были сформированы основные требования к разработке платформы для обмена знаниями. Следующим этапом является выбор технологий и определение этапов реализации проекта.

2.2 Разработка логической модели системы

Логическая модель системы отражает основные сущности, которые будут реализованы в системе. Сущности представлены в таблице 4.

Таблица 4 – Основные логические сущности системы

| Название | Описание | Атрибуты |
|--------------|--|--|
| Пользователь | Представляет сотрудников компании, зарегистрированных в системе. | ID(Integer),Имя(String),Фамилия(String), Email(String),Роль(String),Дата регистрации(Date),Аватар(String),Статус активности(Boolean) |

Продолжение таблицы 4

| Название | Описание | Атрибуты |
|-------------|---|--|
| Публикация | Представляет материалы, созданные пользователями для обмена знаниями. | ID(Integer), Заголовок(String), Текст(Text), Автор(ForeignKey(User)), Дата создания(Date), Категория(String), Теги(String), Статус(String) |
| Обсуждение | Представляет темы для обсуждения, созданные пользователями. | ID(Integer), Название(String), Описание(Text), Автор(ForeignKey(User)), Дата создания(Date), Категория(String), Статус(String) |
| Комментарий | Представляет комментарии пользователей к публикациям или обсуждениям. | ID(Integer), Текст(Text), Автор(ForeignKey(User)), Дата создания(Date), Родительский объект(ForeignKey(Publication/Discussion)), Ответ на комментарий(ForeignKey(Comment)) |
| Уведомление | Представляет уведомления, отправляемые пользователям о новых публикациях, комментариях или обсуждениях. | ID(Integer), Тип(String), Получатель(ForeignKey(User)), Дата создания(Date), Статус(String) |
| Подписка | Представляет подписки пользователей на определенные категории или темы. | ID(Integer), Пользователь(ForeignKey(User)), Категория(String), Дата подписки(Date) |

Логическая схема представлена на рисунке 6.

Пользователь создает Публикацию и Обсуждение.

Пользователь пишет Комментарии к Публикациям и Обсуждениям.

Пользователь получает Уведомления.

Пользователь подписывается на Категории через сущность Подписка.

Публикация содержит Теги и может комментироваться.

Комментарий может быть ответом на другой Комментарий.

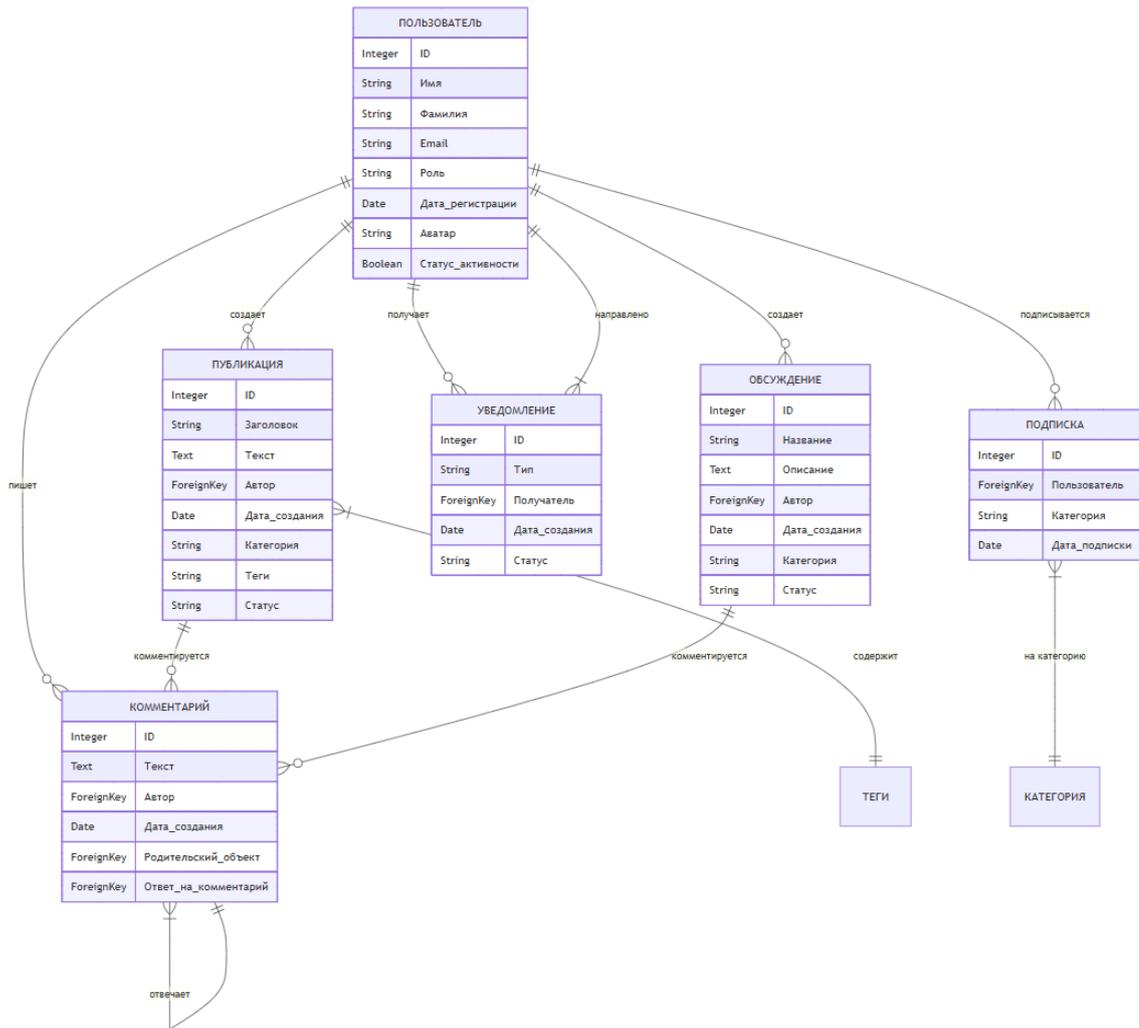


Рисунок 6 - Логическая схема платформы

Связи между сущностями обозначены стрелками: «один ко многим», «многие к одному», «многие ко многим».

2.3 Выбор технологий для разработки платформы

На современном рынке представлено множество инструментов, упрощающих разработку платформы. Поэтому одной из ключевых задач стало определение наиболее подходящего технологического стека, который будет соответствовать функциональным и нефункциональным требованиям нашей платформы.

«Одним из важнейших инструментов при создании платформы является фреймворк. Фреймворк — это готовый набор структур, библиотек и стандартов, который помогает ускорить процесс разработки программного обеспечения» [4]. «Он определяет архитектурные принципы приложения, предоставляет готовые модули для решения типовых задач и способствует стандартизации процесса разработки» [21]. «Использование фреймворка позволяет сосредоточиться на реализации ключевой логики, а не на создании базовых компонентов с нуля, что значительно упрощает работу разработчиков и снижает вероятность ошибок» [20].

Поскольку для нашей платформы важным аспектом является работа с базой знаний и публикациями, необходимо выбрать фреймворк, который обеспечит удобное управление данными. Рассмотрим несколько популярных решений:

«Django (Python) – полнофункциональный фреймворк на Python, включающий встроенную ORM (Object-Relational Mapping), административный интерфейс, маршрутизацию URL, шаблонизацию HTML и механизмы аутентификации» [12]. «Поддерживает работу с PostgreSQL, MySQL и SQLite, что делает его мощным инструментом для построения масштабируемых платформ» [6].

«Flask (Python) – минималистичный микрофреймворк, предоставляющий базовые возможности для веб-разработки, такие как маршрутизация и управление сессиями» [21]. Однако Flask не включает встроенную ORM, что может усложнить работу с базами данных и потребовать дополнительной интеграции сторонних решений.

«Ruby on Rails (Ruby) – удобный фреймворк с мощной встроенной ORM ActiveRecord, которая позволяет работать с базами данных без необходимости писать сложные SQL-запросы. Rails ориентирован на быстрое создание платформ и автоматизацию рутинных задач» [21].

«Laravel (PHP) – современный PHP-фреймворк с продвинутым ORM Eloquent, поддерживающий работу с различными базами данных. Laravel

предлагает широкий набор инструментов для удобного взаимодействия с данными, управления пользователями и создания API» [11].

«ASP.NET Core (C#) – кроссплатформенный фреймворк от Microsoft, предназначенный для создания высокопроизводительных платформ. Включает ORM Entity Framework Core, но требует более глубокого изучения и дополнительных ресурсов для освоения» [11].

Выбирая фреймворк для нашей платформы обмена знаниями, мы учитывали требования к удобству работы с базами данных, скорости разработки и простоте поддержки. Flask был исключен из-за отсутствия встроенной ORM, что может замедлить процесс интеграции базы знаний. Также было решено отказаться от ASP.NET Core, так как его освоение требует больше времени и ресурсов по сравнению с Django или Ruby on Rails.

Таким образом, наилучшими кандидатами для разработки нашей платформы являются Django, Ruby on Rails и Laravel, так как они предоставляют встроенные инструменты для работы с базами данных, обладают хорошей документацией и широко используются в разработке платформ. Окончательный выбор фреймворка будет сделан с учетом специфики проекта и требований к архитектуре системы.

Проведём сравнение оставшихся фреймворков для веб-разработки в таблице 5.

Сравнивать будем по следующим критериям: сложность начала работы, гибкость и расширяемость, экосистема и сообщество, производительность и масштабируемость.

В качестве основного фреймворка для разработки платформы обмена знаниями и опытом среди сотрудников был выбран Django. Он включает в себя все необходимые компоненты для создания полноценной платформы, что упрощает процесс разработки и сокращает время на его реализацию. «Django предлагает мощный инструментарий для работы с базами данных, в том числе встроенную ORM (Object-Relational Mapping), которая позволяет удобно управлять данными без написания сложных SQL-запросов» [22]. Это особенно

важно для нашей платформы, так как она предполагает хранение и обработку большого количества информации.

Таблица 5 - Сравнение выбранных фреймворков

| Критерии | Django | Ruby on Rails | Laravel |
|---------------------------------------|--|--|---|
| Сложность начала работы | Средняя. Требуется знание Python и структуры Django, но есть подробная документация и встроенный административный интерфейс. | Низкая. Имеет удобный генератор кода и следование принципу «конвенция вместо конфигурации». | Средняя. Требуется знание PHP, но встроенные инструменты облегчают разработку. |
| Гибкость и расширяемость | Очень большое и активное сообщество, множество обучающих материалов и сторонних библиотек. | Активное сообщество, но меньше, чем у Django и Laravel | Очень популярный фреймворк среди PHP-разработчиков, богатая документация и множество пакетов. |
| Экосистема и сообщество | Высокая. Поддерживает асинхронные задачи, кэширование и балансировку нагрузки. Подходит для крупных проектов. | Средняя. Хорошо работает для средних веб-приложений, но при большой нагрузке требует дополнительной оптимизации. | Высокая. Хорошо масштабируется, поддерживает кэширование и распределенные системы. |
| Производительность и масштабируемость | Высокая, особенно подходит для крупных проектов | Высокая, подходит для крупных проектов | Хорошая, поддерживает крупные проекты |

Помимо Django, для разработки платформы потребуются и другие ключевые технологии:

«JavaScript (JS) — язык программирования, который позволяет создавать интерактивные элементы на веб-страницах» [21]. Благодаря JavaScript наша платформа сможет реагировать на действия пользователей, обрабатывать события и взаимодействовать с сервером в реальном времени.

«HTML (HyperText Markup Language) — язык разметки, который

определяет структуру веб-страниц» [21]. «С его помощью будет создан каркас платформы, включая разделы, кнопки, формы и другие элементы интерфейса» [21].

«CSS (Cascading Style Sheets) — язык стилей, который отвечает за внешний вид страниц» [20]. «CSS позволит сделать интерфейс платформы удобным, стильным и интуитивно понятным для пользователей» [20].

В результате, для реализации платформы будут использоваться Django, JavaScript, HTML и CSS. Эти технологии обеспечат удобный и функциональный пользовательский интерфейс, а также стабильную работу платформы на серверной и клиентской стороне.

2.4 План разработки платформы

Перед началом создания платформы для обмена знаниями и опытом между сотрудниками необходимо разработать поэтапный план, который поможет организовать процесс и последовательно реализовать все ключевые задачи. План включает в себя основные этапы разработки, начиная от подготовки требований и проектирования архитектуры до тестирования готового решения.

Для наглядного представления плана разработки составлена диаграмма Ганта на рисунке 7, которая позволяет визуализировать сроки и взаимосвязь между этапами, обеспечить контроль выполнения задач и соблюдение графика.

Первым этапом является сбор и анализ требований. На этом этапе проводится изучение существующих решений, выявление потребностей сотрудников и формулирование целей, которым должна соответствовать разрабатываемая платформа. Анализ включает как функциональные, так и нефункциональные требования, отражающие технические, пользовательские и организационные ожидания.

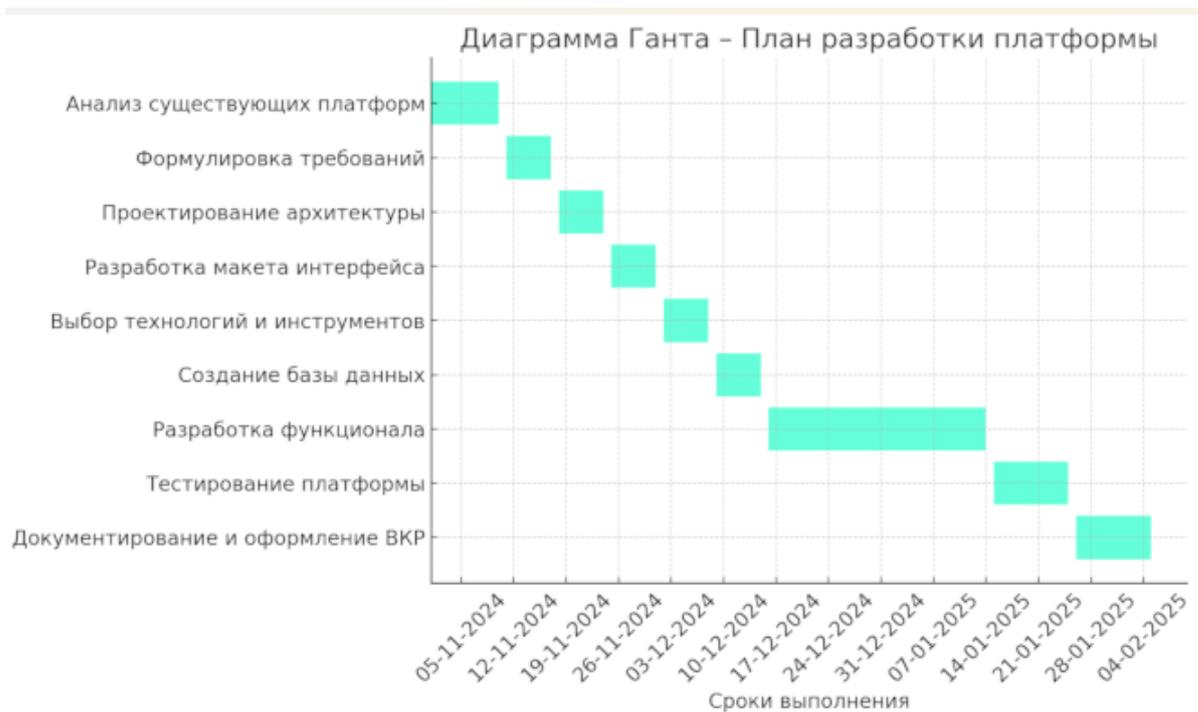


Рисунок 7 – Диаграмма Ганта

Следующим этапом является проектирование архитектуры системы. В рамках данного шага определяются ключевые компоненты платформы, их структура и взаимосвязь. Разрабатываются предварительные диаграммы и модели, описывающие базу данных, логику работы сервиса и взаимодействие пользователя с системой. Это обеспечивает основу для последующих этапов разработки.

Далее планируется выбор технологий и инструментов разработки. На этом этапе оцениваются подходящие языки программирования, фреймворки, системы управления базами данных, а также средства визуального и функционального проектирования. Выбор осуществляется на основе требований к производительности, удобству поддержки и масштабируемости платформы.

После подготовки архитектуры и выбора технологий следует этап прототипирования. Разрабатываются макеты пользовательского интерфейса, создаются шаблоны экранов для разных ролей пользователей, таких как сотрудники и администраторы. Это позволяет заранее продумать логику

взаимодействия с системой и повысить удобство использования.

Затем планируется основной этап разработки, в ходе которого будут создаваться все ключевые модули платформы: авторизация, рабочие экраны сотрудников и администраторов, разделы публикаций, обсуждений и взаимодействия. Каждому этапу соответствует свой временной промежуток, что позволяет отслеживать ход выполнения задач.

Завершающим этапом является тестирование платформы. На данном этапе проверяется работоспособность всех функций, корректность интерфейсов, стабильность системы и соответствие требованиям. Также возможны корректировки по результатам выявленных недочётов и замечаний.

В результате был сформирован предварительный план разработки платформы для обмена знаниями и опытом между сотрудниками. На следующем этапе предстоит определить подходящие технологии и приступить к последовательной реализации проекта.

2.5 Разработка пользовательского интерфейса платформы

При создании платформы для обмена знаниями и опытом между сотрудниками SEO TLT необходимо учитывать ключевые принципы проектирования пользовательского интерфейса. Эти принципы определяют удобство взаимодействия сотрудников с системой и обеспечивают её эффективность. Ниже представлены основные принципы и их требования.

Принцип интуитивно понятного и доступного интерфейса. Элементы интерфейса должны быть организованы таким образом, чтобы пользователь мог легко ориентироваться в системе, выполнять необходимые действия и взаимодействовать с коллегами.

Требования:

- интерфейс должен быть простым и удобным, без лишних элементов, с доступом к основным функциям без дополнительных сложностей;
- все кнопки и элементы управления должны иметь понятные

текстовые или графические обозначения, исключаящие возможность неправильного использования.

Принцип логичной структуры интерфейса. Система должна обеспечивать последовательное и удобное представление информации, что способствует быстрому доступу к нужным данным.

Требования:

- структура интерфейса должна быть логичной и последовательной, обеспечивая удобную навигацию по платформе;
- важные разделы и функции должны быть выделены, а интерфейс – разделён на логические блоки, такие как база знаний, управление задачами и взаимодействие между сотрудниками.

Принцип визуальной привлекательности и удобочитаемости. Эстетически продуманный интерфейс повышает комфорт работы с системой и облегчает восприятие информации.

Требования:

- цветовая схема должна быть сбалансированной, не создавая излишнего напряжения для глаз и обеспечивая удобное восприятие данных;
- важные элементы, такие как кнопки действий и ключевые функциональные зоны, должны быть визуальным образом выделены для удобства использования.

Все эти принципы направлены на создание удобного, понятного и эстетичного интерфейса, который обеспечит комфортную работу сотрудников с платформой.

С учётом данных принципов были разработаны основные шаблоны рабочих экранов пользователей.

На рисунке 8 представлен экран авторизации, который является входной точкой в систему. Он включает поле для ввода логина и пароля, а также кнопку входа. Так как доступ к платформе предназначен исключительно для

сотрудников компании, процесс регистрации осуществляется администратором, и ссылка на самостоятельную регистрацию отсутствует.

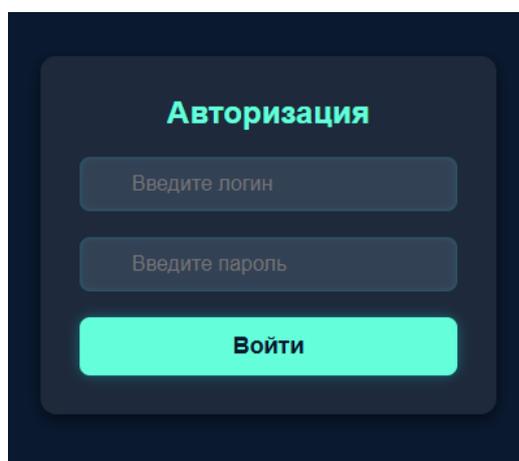


Рисунок 8 – Экран авторизации

На экране авторизации сотрудник вводит свои учетные данные: логин(ФИО) и пароль. После того как сотрудник нажимает кнопку «Войти», данные передаются на сервер для проверки. Если введенная информация является корректной, сотрудник получает доступ к платформе и автоматически перенаправляется в свой личный кабинет, где он может начать работу.

Этот дизайн выполнен с использованием темно-синего фона и приглушенных оттенков для комфортного восприятия информации. Вся платформа будет следовать этому стилю для обеспечения единого и гармоничного интерфейса.

Страница администратора представляет собой центральный узел управления платформой, предоставляя доступ к различным разделам и функциям, необходимым для эффективного администрирования. В разделе управления сотрудниками администратор может просматривать полный список сотрудников, их должности и статус, а также управлять данными сотрудников и отслеживать их активность. Раздел управления проектами позволяет просматривать текущие проекты, их статус и сроки выполнения, а

также добавлять новые проекты и редактировать существующие. В разделе задач администратор имеет возможность отслеживать статус выполнения задач, назначать ответственных и устанавливать приоритеты, что способствует эффективному управлению рабочим процессом.

Аналитика и отчеты предоставляют доступ к различным отчетам и аналитическим данным, позволяющим оценивать производительность сотрудников и эффективность проектов, а также генерировать отчеты по различным параметрам для принятия обоснованных решений. Управление правами доступа позволяет настраивать права пользователей платформы, обеспечивая безопасность данных и контроль над действиями пользователей. Наконец, раздел настроек платформы дает возможность конфигурировать общие параметры работы, включая настройки уведомлений и интеграции.

Страница администратора обеспечивает управление платформой и доступ ко всем функциям. Администратор может добавлять сотрудников через форму регистрации с обязательными полями. Данные отправляются на сервер и сохраняются в базе. На рисунке 9 мы увидим страницу администратора.

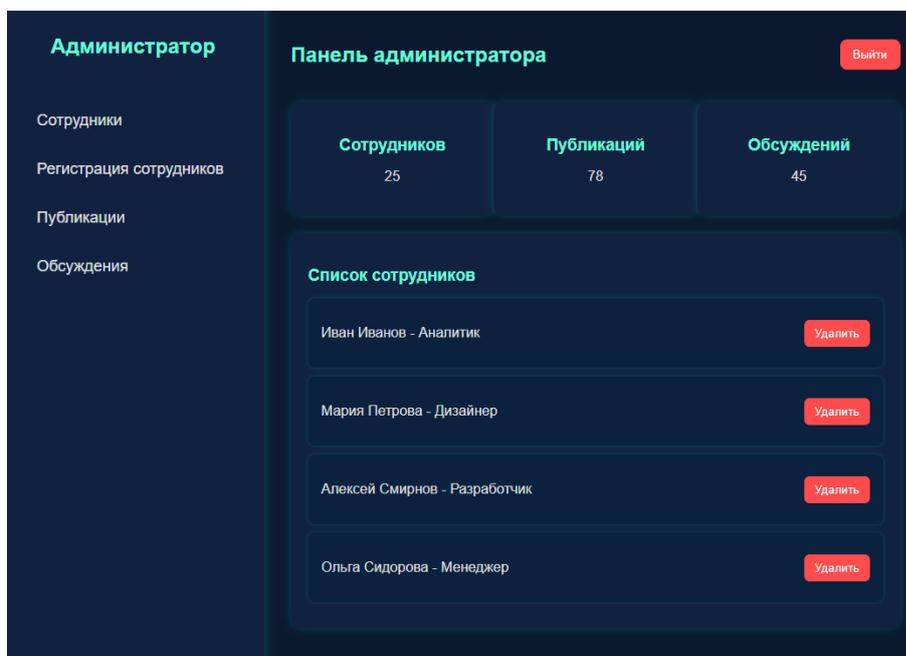
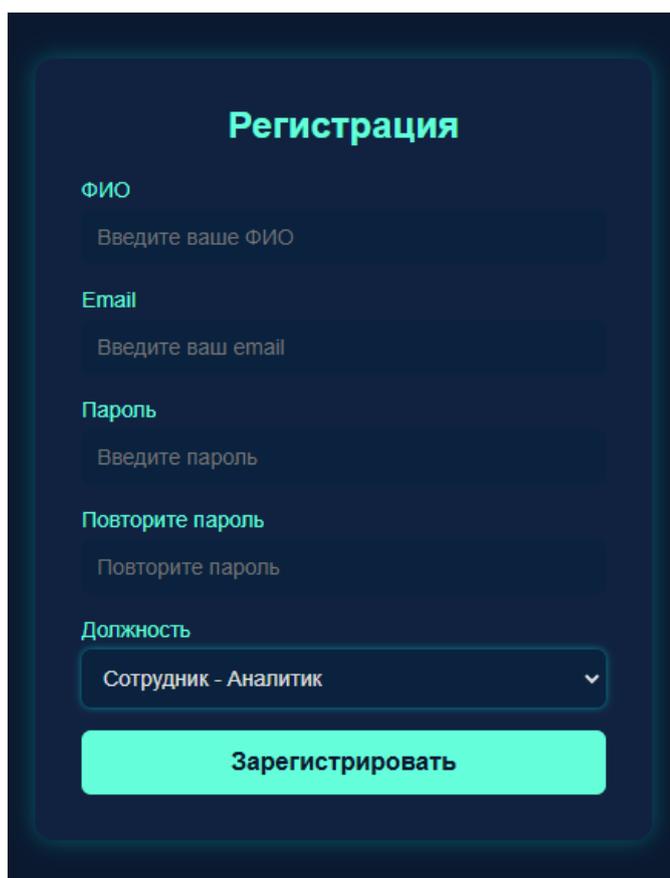


Рисунок 9 – Страница администратора

Интерфейс страницы администратора интуитивно понятен и удобен для навигации, с боковым меню, содержащим кнопки на все ключевые разделы, что упрощает доступ к необходимой информации. Основной контент организован в четкие блоки, позволяя быстро находить нужные функции и данные. Страница администратора является важным инструментом для управления платформой, обеспечивая полный контроль над всеми аспектами работы и позволяя эффективно организовывать деятельность команды.

Также доступен список сотрудников с их должностями и статусами, что помогает эффективно управлять командой и отслеживать статус текущих проектов и задач. На рисунке 10 изображена страница в которой администратор регистрирует новых пользователей.



The image shows a registration form titled "Регистрация" (Registration) on a dark blue background. The form contains the following fields and elements:

- ФИО** (Full Name): A text input field with the placeholder "Введите ваше ФИО".
- Email**: A text input field with the placeholder "Введите ваш email".
- Пароль** (Password): A text input field with the placeholder "Введите пароль".
- Повторите пароль** (Repeat Password): A text input field with the placeholder "Повторите пароль".
- Должность** (Position): A dropdown menu with the selected option "Сотрудник - Аналитик" and a downward arrow.
- Зарегистрировать** (Register): A large, prominent red button at the bottom of the form.

Рисунок 10 – Регистрация новых пользователей

Страница «Обсуждения» (рисунок 11) является важным компонентом

платформы, предоставляющим пользователям удобный инструмент для обмена знаниями, идеями и опытом. Он включает в себя несколько ключевых функций, обеспечивающих эффективное взаимодействие и структурирование обсуждений.

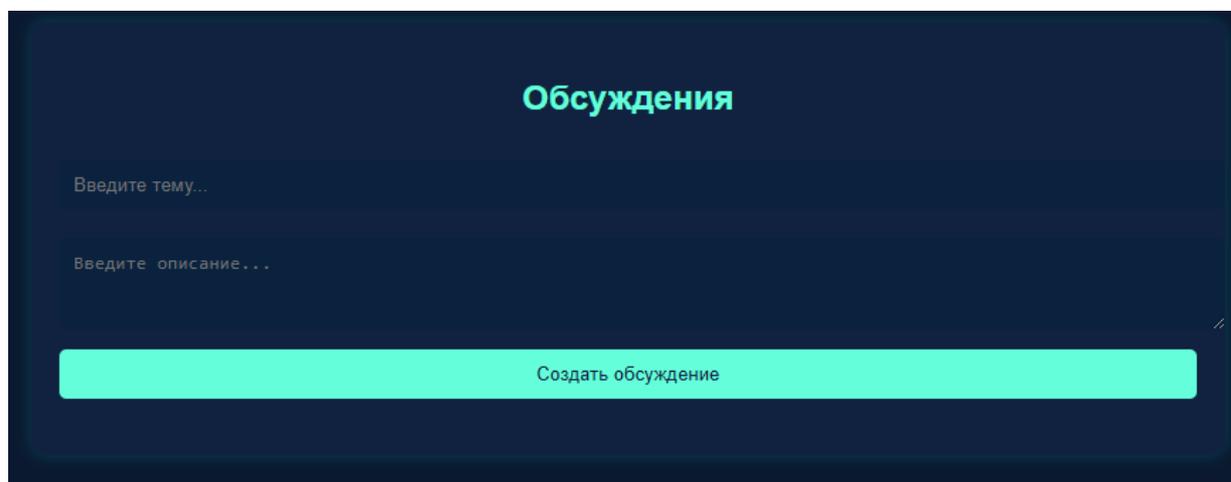
The image shows a dark-themed user interface for creating a new discussion. At the top, the word "Обсуждения" (Discussions) is written in a light blue font. Below it, there are two text input fields: the first is labeled "Введите тему..." (Enter topic...) and the second is labeled "Введите описание..." (Enter description...). At the bottom of the form, there is a prominent, wide, light blue button with the text "Создать обсуждение" (Create discussion).

Рисунок 11 – Создание новых обсуждений

Пользователи могут создавать новые темы обсуждений, заполняя специальную форму, которая включает обязательные поля: заголовок темы, описание, возможность прикрепления файлов и изображений. После публикации обсуждение становится доступным для других участников, которые могут оставлять комментарии, предлагать решения и делиться своим мнением.

Все обсуждения представлены в виде списка на рисунке 12, где отображаются название темы, автор, дата создания и количество комментариев. Это позволяет легко ориентироваться в имеющихся дискуссиях и быстро находить нужную информацию. Для удобства предусмотрены фильтрация и сортировка обсуждений по дате создания, популярности и активности пользователей.

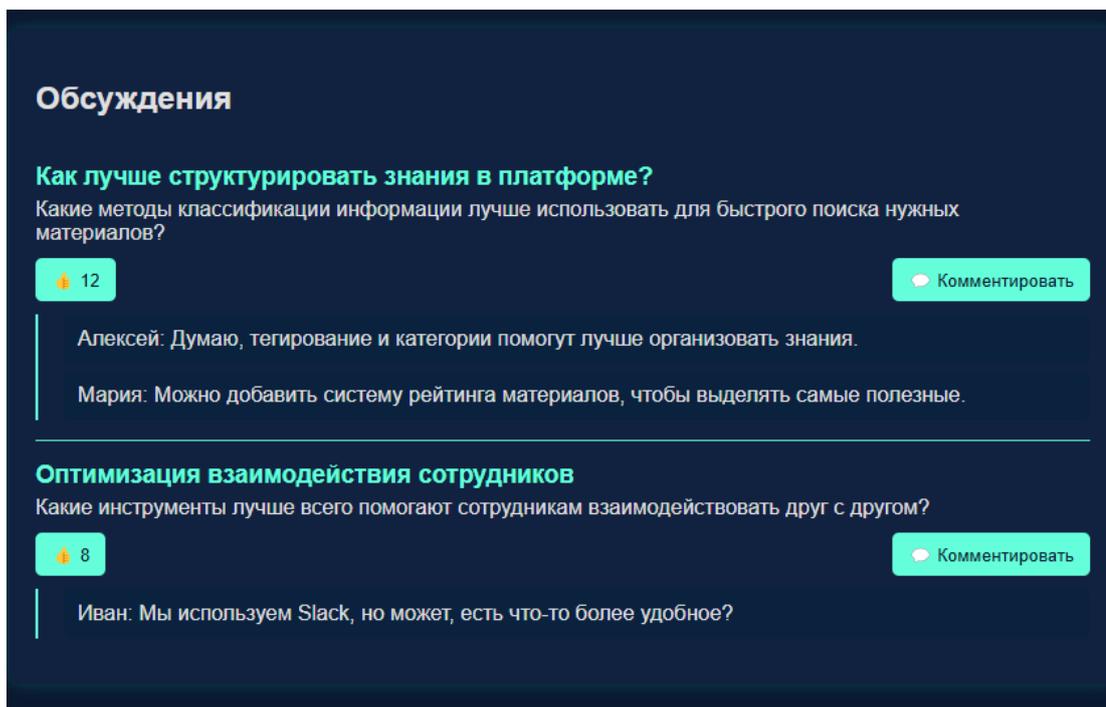


Рисунок 12 – Созданные обсуждения

Раздел также оснащен системой оценок, позволяющей отмечать полезные комментарии и темы, а также функцией подписки, благодаря которой пользователи могут получать уведомления о новых ответах и обновлениях в интересующих их обсуждениях.

Администраторы платформы имеют дополнительные возможности управления, включая модерирование контента, удаление некорректных или нежелательных сообщений, а также управление активностью пользователей. Это помогает поддерживать высокое качество обсуждений и предотвращает распространение нерелевантной информации.

На рисунке 13 представлена страница «Сотрудники» в администраторской панели, предназначенная для управления пользователями платформы. Интерфейс страницы включает список всех зарегистрированных сотрудников, отображая их ФИО, должность и статус. Для удобства навигации реализованы функции поиска и фильтрации, позволяющие быстро находить нужных пользователей.

| ID | ФИО | Должность | Отдел | Дата приёма | Действия |
|---------|-----------------------------|--------------|-----------|-------------|---|
| EMP-001 | Иванов Алексей Владимирович | Team Lead | Backend | 2025-01-15 | Профиль Удалить |
| EMP-002 | Петрова Мария Сергеевна | UX Designer | Дизайн | 2025-02-01 | Профиль Удалить |
| EMP-003 | Смирнов Дмитрий Игоревич | Data Analyst | Аналитика | 2025-03-10 | Профиль Удалить |
| EMP-004 | Козлова Анна Викторовна | HR Manager | HR | 2025-01-22 | Профиль Удалить |
| EMP-005 | Фёдоров Артём Олегович | Frontend Dev | Frontend | 2025-02-18 | Профиль Удалить |

Рисунок 13 – Страница «Сотрудники»

Администратор имеет возможность добавлять новых сотрудников через специальную форму, содержащую обязательные поля: ФИО, email, пароль, повторный ввод пароля и выбор должности. После заполнения и отправки формы данные сохраняются в базе.

Кроме того, предусмотрена возможность редактирования информации о сотрудниках, включая обновление их должности и контактных данных. В случае необходимости администратор может удалить сотрудника, после чего он исчезает из списка. Все изменения автоматически синхронизируются с базой данных, обеспечивая актуальность сведений.

На рисунке 14 показаны публикации сотрудников в табличном виде.

На рисунке 15 представлена главная страница сотрудника, которая предоставляет доступ к основным функциям платформы. Интерфейс включает боковое меню для навигации, раздел уведомлений, список задач и кнопку выхода. Сотрудник может отслеживать актуальные обсуждения, публиковать материалы, взаимодействовать с коллегами и управлять своими задачами.

Публикации

Поиск по заголовку или содержанию...

Все категории ▼ Все статусы ▼

| ID | Заголовок | Категория | Автор | Дата | Действия |
|---------|--|--------------------------|--------------|------------|---------------------------|
| PUB-015 | Руководство по безопасности API Полное руководство по реализации OAuth 2.0... | Техническая документация | Иванов А.В. | 2025-02-10 | Опубликовано |
| PUB-016 | Анализ рынка IT-2025 Сравнительный анализ облачных решений... | Исследования | Петрова М.С. | 2025-03-01 | Опубликовано |
| PUB-017 | Отчет Q1 2025 Финансовые показатели и ключевые метрики... | Отчёты | Смирнов Д.И. | 2025-03-05 | Черновик |

Рисунок 14 – Страница «Публикации»

Меню

- Главная
- Профиль
- Обсуждения
- Публикации
- Мои задачи
- Уведомления
- Выход

Добро пожаловать!

Иван Иванов

Последние уведомления

- Новое сообщение в обсуждении "Оптимизация процессов"
- Добавлена новая публикация "Эффективные методы обмена знаниями"
- Обновлен статус вашей задачи "Анализ документации"

Мои задачи

- Подготовить отчет по проекту В процессе
- Обсудить стратегию с коллегами Ожидает
- Заполнить форму обратной связи Выполнено

Рисунок 15 – Главная страница сотрудника

Рассмотрев все страницы и разделы нашей платформы, можно отметить, что разработанный интерфейс полностью подходит под основные принципы.

В этой главе были установлены основные требования к разрабатываемому сервису. Выбрали основные технологии для разработки нашего веб-приложения, также и план. Разработали пользовательский интерфейс приложения с соблюдением основных принципов.

Глава 3 Реализация платформы для обмена знаниями и опытом среди сотрудников

3.1 Выбор системы управления базами данных для платформы

Для эффективного хранения и управления данными, связанными с обсуждениями, сотрудниками и базой знаний, в платформе обмена знаниями используется система управления базами данных (СУБД). Django предоставляет встроенную ORM (Object-Relational Mapping), которая значительно упрощает работу с базой данных, позволяя использовать объектно-ориентированный подход вместо написания сложных SQL-запросов. Это делает код более читаемым, удобным для сопровождения и ускоряет процесс разработки.

При выборе СУБД для платформы были рассмотрены несколько наиболее распространенных решений:

- PostgreSQL – мощная реляционная СУБД с поддержкой сложных запросов, индексации, транзакций и расширенных механизмов защиты данных. Она является одним из лучших вариантов для веб-приложений с высокой нагрузкой [27];
- MySQL – популярная реляционная СУБД, обладающая высокой производительностью и простотой настройки. Подходит для большинства веб-приложений, однако в некоторых случаях уступает PostgreSQL в функциональности [25];
- SQLite – легковесная встроенная СУБД, не требующая развертывания серверной инфраструктуры. Отличается простотой, но не подходит для многопользовательских систем с высокой нагрузкой [24];
- Microsoft SQL Server – полнофункциональная СУБД от Microsoft, обеспечивающая мощные аналитические инструменты, поддержку транзакций и глубокую интеграцию с экосистемой Microsoft [26].

Сравним СУБД в таблице 6.

Таблица 6 – Сравнение СУБД

| Критерий | MySQL | SQLite | PostgreSQL |
|----------------------------|-------|--------|------------|
| Удобство эксплуатации | + | + | + |
| Гибкость настройки | + | + | + |
| Высокая производительность | + | + | + |
| Широкий функционал | + | + | + |
| Опыт работы | + | + | + |
| Простота реализации | - | + | - |
| Итого: | 5 | 6 | 5 |

Проведя анализ возможных вариантов систем управления базами данных (СУБД) для платформы обмена знаниями, приходим к выводу, что целесообразно использовать PostgreSQL. Данная СУБД обладает высокой надежностью, поддерживает работу с транзакциями, индексацию, масштабируемость и обеспечивает необходимую производительность для хранения данных сотрудников, обсуждений и других элементов системы [23].

После выбора СУБД необходимо разработать физическую модель базы данных, основанную на ранее спроектированной логической модели. Физическая модель базы данных описывает, каким образом данные будут фактически храниться, а также включает информацию о структуре таблиц, типах данных, индексах, ключах и ограничениях целостности. Это важно для обеспечения оптимальной работы платформы и быстрого доступа к информации. На рисунке 16 мы увидим физическую модель базы данных нашей платформы.

Физическая модель базы данных разработана с учетом требований к платформе для обмена знаниями и опытом между сотрудниками компании SEO TLT. Структура базы данных реализована в среде PostgreSQL и включает основные таблицы, соответствующие ключевым сущностям системы: пользователи, публикации, обсуждения, комментарии, уведомления, подписки, теги и промежуточные связи.

Таблица `users` содержит информацию о зарегистрированных сотрудниках, включая их уникальный идентификатор, имя, фамилию, электронную почту, роль, дату регистрации, аватар и статус активности. Для обеспечения уникальности учетных записей поле `email` имеет ограничение уникальности.

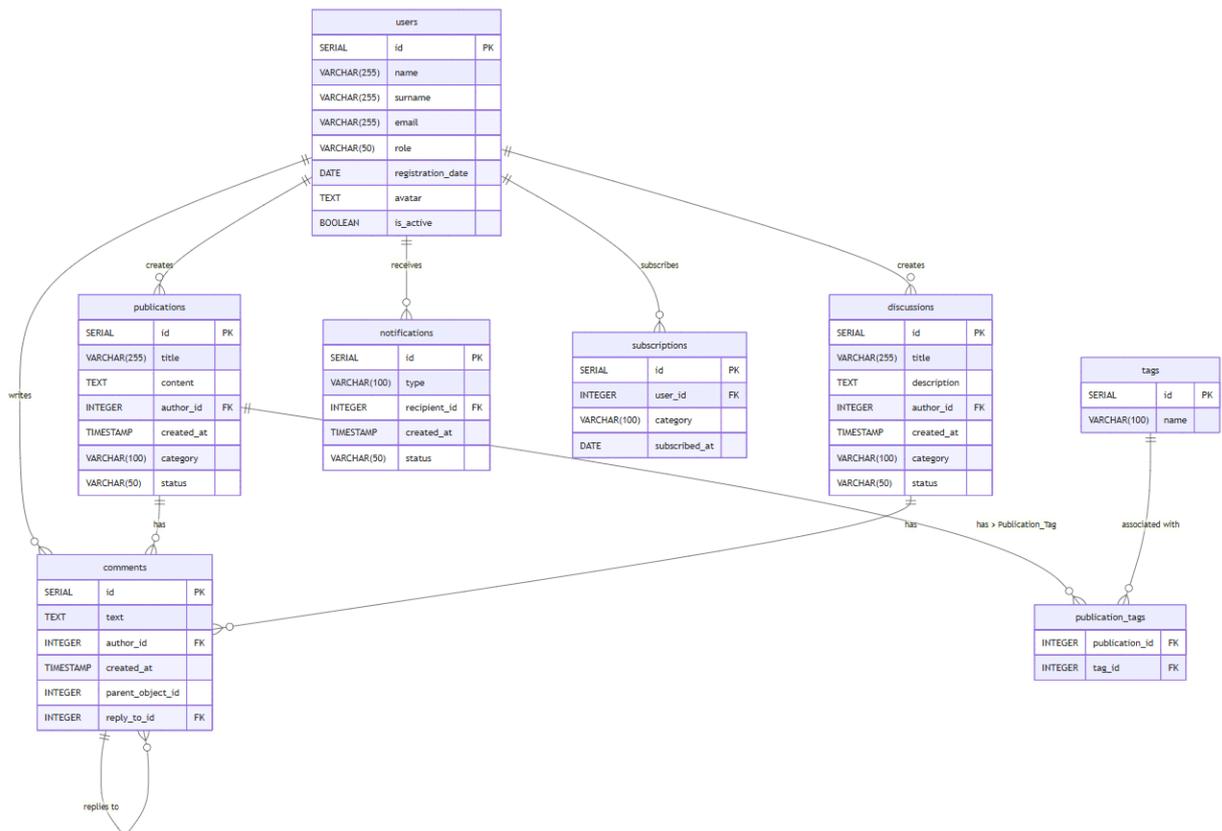


Рисунок 16 – Физическая модель базы данных

Таблица `publications` предназначена для хранения данных о публикациях, созданных пользователями. Каждая запись включает заголовок, текстовое содержание, ссылку на автора, дату создания, категорию и статус публикации. Связь с таблицей `users` реализуется через внешний ключ `author_id`, что позволяет однозначно определить авторство материалов.

Таблица `discussions` отражает темы для обсуждений, создаваемые пользователями. Её структура аналогична таблице `publications`, но с акцентом

на описание темы и её категории.

Таблица `comments` используется для хранения комментариев к публикациям и обсуждениям. В ней предусмотрены поля для текста комментария, ссылки на автора, даты создания, родительского объекта и ответа на другой комментарий. Самосвязь реализуется через поле `reply_to_id`, что позволяет организовать древовидную структуру комментариев.

Таблица `notifications` содержит данные об уведомлениях, отправляемых пользователям. Каждое уведомление характеризуется типом, получателем, датой создания и статусом прочтения.

Таблица `subscriptions` отражает подписки пользователей на категории. Она включает ссылку на пользователя, название категории и дату подписки.

Для управления тегами создана таблица `tags`, содержащая уникальные идентификаторы и названия тегов. Связь между публикациями и тегами реализуется через промежуточную таблицу `publication_tags`, которая обеспечивает поддержку отношения «многие ко многим».

Типы данных таблиц:

- `SERIAL`: Автоинкрементируемый числовой тип для первичных ключей.
- `VARCHAR(n)`: Строка фиксированной длины (n символов).
- `TEXT`: Текст произвольной длины.
- `TIMESTAMP`: Дата и время.
- `BOOLEAN`: Логический тип данных (true/false).

Создадим физическую модель классов, которая представлена на рисунке 17.

Диаграмма классов представляет собой модель платформы для обмена знаниями и опытом, разработанной для компании SEO TLT. Она включает следующие основные классы: `User` (Пользователь), `Publication` (Публикация), `Discussion` (Обсуждение), `Comment` (Комментарий), `Notification` (Уведомление), `Subscription` (Подписка), `Tag` (Тег), `Publication_Tag` (Промежуточная таблица для связи «многие ко многим»).

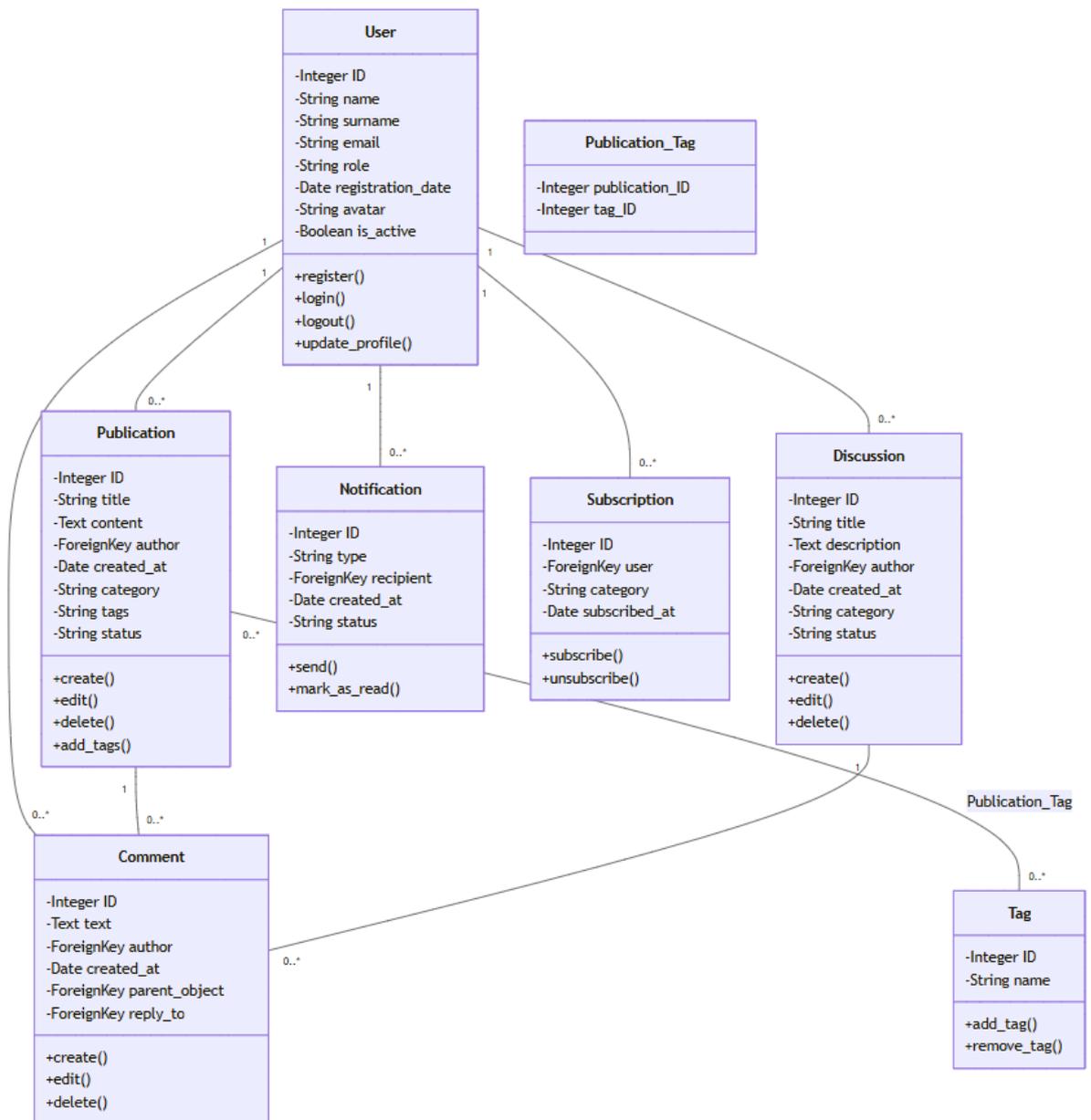


Рисунок 17 - Диаграмма классов разрабатываемой платформы

Каждый класс описывает сущность системы, её атрибуты, методы и взаимосвязи с другими классами. В таблице 7 представлено подробное описание классов.

Таблица 7 – Описание классов

| Название класса | Атрибуты | Методы |
|---|---|---|
| <p>Класс: User</p> <p>представляет сотрудников компании, зарегистрированных в системе</p> | <ul style="list-style-type: none"> – ID (Integer): Уникальный идентификатор пользователя. – name (String): Имя пользователя. – surname (String): Фамилия пользователя. – email (String): Электронная почта пользователя. – role (String): Роль пользователя (например, «сотрудник» или «администратор»). – registration_date (Date): Дата регистрации. – avatar (String): URL аватара пользователя. – is_active (Boolean): Статус активности пользователя. | <ul style="list-style-type: none"> – +register(): Регистрация нового пользователя. – +login(): Авторизация пользователя. – +logout(): Выход из системы. – +update_profile(): Обновление профиля пользователя. |
| <p>Класс: Publication</p> <p>представляет публикации, созданные пользователями для обмена знаниями.</p> | <ul style="list-style-type: none"> – -ID (Integer): Уникальный идентификатор публикации. – -title (String): Заголовок публикации. – -content (Text): Текст публикации. – -author (ForeignKey): Ссылка на автора публикации (класс User). – -created_at (Date): Дата создания публикации. – -category (String): Категория публикации. – -tags (String): Теги, связанные с публикацией. – -status (String): Статус публикации (например, «опубликовано», «черновик»). | <ul style="list-style-type: none"> – +create(): Создание новой публикации. – +edit(): Редактирование существующей публикации. – +delete(): Удаление публикации. – +add_tags(): Добавление тегов к публикации. |
| <p>Класс: Discussion</p> <p>представляет темы для обсуждения, созданные пользователями.</p> | <ul style="list-style-type: none"> – -ID (Integer): Уникальный идентификатор обсуждения. – -title (String): Название обсуждения. – -description (Text): Описание обсуждения. – -author (ForeignKey): Ссылка на автора обсуждения (класс User). – -created_at (Date): Дата создания обсуждения. – -category (String): Категория обсуждения. – -status (String): Статус обсуждения (например, «открыто», «закрито»). | <ul style="list-style-type: none"> – +create(): Создание нового обсуждения. – +edit(): Редактирование существующего обсуждения. – +delete(): Удаление обсуждения. |

Продолжение таблицы 7

| Название класса | Атрибуты | Методы |
|--|---|---|
| Класс: Comment представляет комментарии пользователей к публикациям или обсуждениям. | <ul style="list-style-type: none"> – -ID (Integer): Уникальный идентификатор комментария. – -text (Text): Текст комментария. – -author (ForeignKey): Ссылка на автора комментария (класс User). – -created_at (Date): Дата создания комментария. – -parent_object (ForeignKey): Ссылка на родительский объект (класс Publication или Discussion). – -reply_to (ForeignKey): Ссылка на комментарий, на который отвечает текущий комментарий. | <ul style="list-style-type: none"> – +create(): Создание нового комментария. – +edit(): Редактирование существующего комментария. – +delete(): Удаление комментария. |
| Класс: Notification представляет уведомления, отправляемые пользователям. | <ul style="list-style-type: none"> – -ID (Integer): Уникальный идентификатор уведомления. – -type (String): Тип уведомления (например, «новая публикация», «новый комментарий»). – -recipient (ForeignKey): Ссылка на получателя уведомления (класс User). – -created_at (Date): Дата создания уведомления. – -status (String): Статус уведомления (например, «прочитано», «не прочитано»). | <ul style="list-style-type: none"> – +send(): Отправка уведомления. – +mark_as_read(): Пометка уведомления как прочитанного. |
| Класс: Subscription представляет подписки пользователей на категории. | <ul style="list-style-type: none"> – -ID (Integer): Уникальный идентификатор подписки. – -user (ForeignKey): Ссылка на пользователя (класс User). – -category (String): Категория, на которую подписывается пользователь. – -subscribed_at (Date): Дата подписки. | <ul style="list-style-type: none"> – +subscribe(): Подписка пользователя на категорию. – +unsubscribe(): Отмена подписки. |
| Класс: Tag представляет теги, используемые для категоризации публикаций. | <ul style="list-style-type: none"> – -ID (Integer): Уникальный идентификатор тега. – -name (String): Название тега. | <ul style="list-style-type: none"> – +add_tag(): Добавление нового тега. – +remove_tag(): Удаление тега. |
| Класс: Publication_Tag - промежуточная таблица для реализации связи «многие ко многим» между публикациями и тегами. | <ul style="list-style-type: none"> – -publication_ID (Integer): Ссылка на публикацию (класс Publication). | отсутствуют |

Диаграммы классов играют важную роль в проектировании информационных систем, так как позволяют наглядно представить структуру данных и взаимосвязи между основными сущностями системы. Они помогают разработчикам и аналитикам понять, какие объекты существуют в системе, какие у них есть свойства (атрибуты) и каким образом эти объекты взаимодействуют между собой. Это особенно важно при создании масштабируемых и поддерживаемых программных решений, таких как платформа для обмена знаниями, разрабатываемая для компании SEO TLT.

Использование диаграммы классов позволяет заранее выявить возможные недочёты в архитектуре и избежать ошибок при реализации. Она также облегчает коммуникацию между членами команды, упрощает документирование системы и служит основой для генерации кода и построения базы данных.

3.2 Архитектура и структура платформы для обмена знаниями на базе фреймворка Django

При создании платформы для обмена знаниями и опытом между сотрудниками SEO TLT использование фреймворка Django значительно упрощает процесс разработки. Django обеспечивает удобную организацию кода, разбивая проект на логически обособленные модули. Такая структура ускоряет разработку и делает поддержку приложения более удобной.

Фреймворк предоставляет готовую архитектуру, включающую в себя все необходимые инструменты — от маршрутизации запросов до работы с базой данных, что особенно важно при создании внутренней корпоративной платформы [8].

На рисунке 18 представлена схема, иллюстрирующая архитектуру и принцип работы платформы, разработанной с использованием Django.

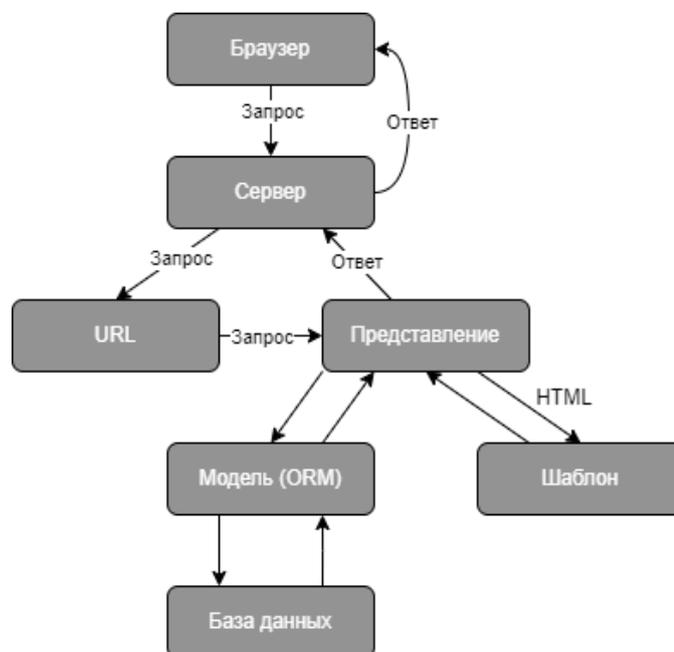


Рисунок 18 – Схема работы Django-приложения

Работа платформы для обмена знаниями, разработанной с использованием фреймворка Django, построена на архитектурной модели MVT (Model-View-Template) — разновидности более широкой концепции MVC. Эта структура помогает логично организовать код и упростить разработку.

Наша платформа будет использовать концепцию MVT (Model-View-Template).

Model (Модель) отвечает за взаимодействие с базой данных, хранит всю основную информацию — например, данные о пользователях, обсуждениях, сообщениях и загруженных файлах. Кроме того, модели реализуют бизнес-логику и правила валидации данных, что особенно важно для корректной работы корпоративного приложения.

View (Представление) обрабатывает пользовательские запросы, обращается к нужным моделям за данными и формирует ответ. В случае нашей платформы это может быть, например, вывод списка обсуждений, просмотр сообщений в конкретной теме или отображение профиля пользователя.

Template (Шаблон) отвечает за внешний вид страниц платформы. Шаблоны — это HTML-страницы с элементами шаблонизатора Django, которые позволяют динамически вставлять информацию, полученную из представлений. Благодаря этому сотрудник компании видит уже готовую страницу с нужными данными, будь то список обсуждений, сообщение коллеги или загруженный документ.

Использование подхода MVT позволяет чётко разделить работу с данными, обработку логики и отображение интерфейса. Это делает разработку платформы более удобной и масштабируемой, а также упрощает её поддержку и развитие в будущем.

Структура нашего проекта разработки веб-приложения представлена на рисунке 19.

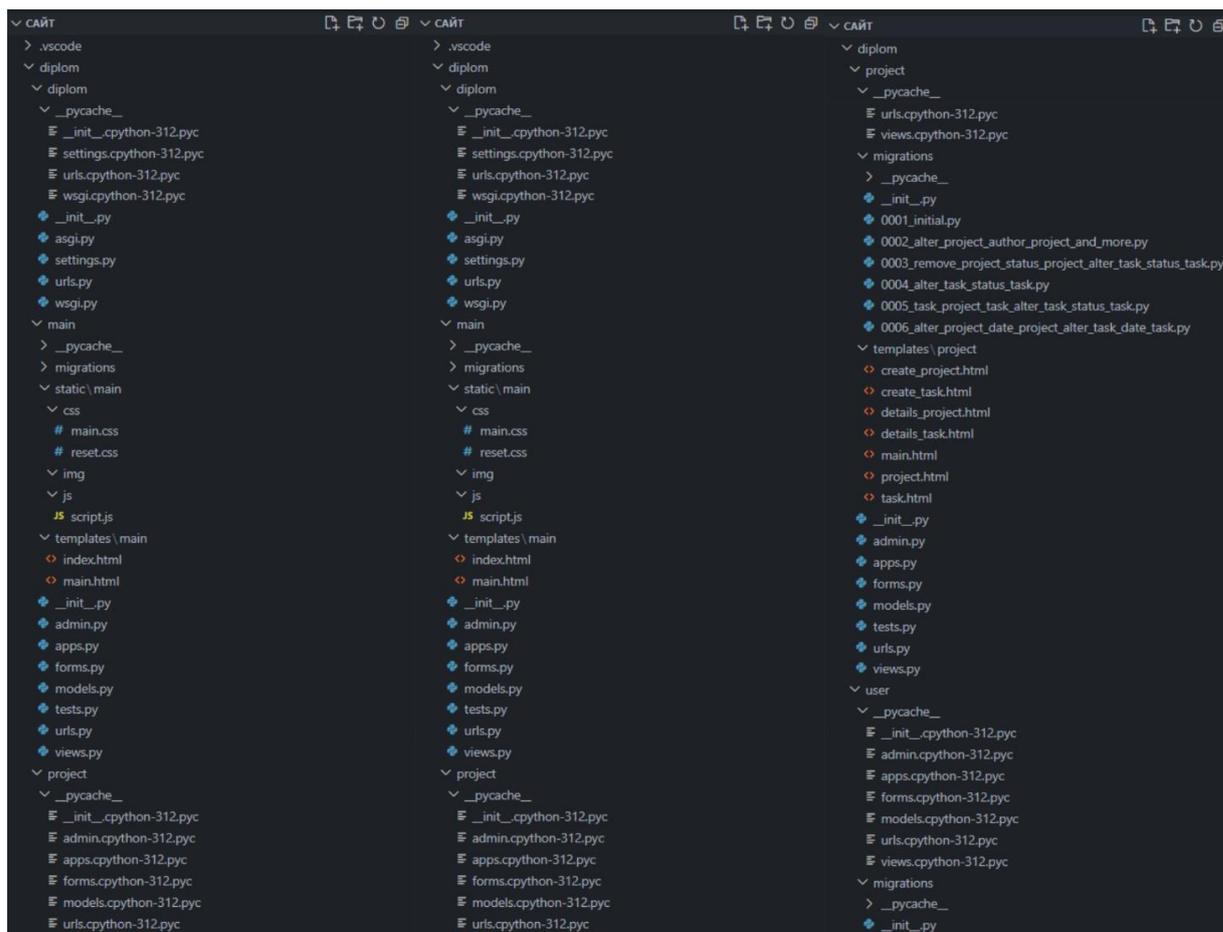


Рисунок 19 – Реализованная структура проекта

При разработке платформы для обмена знаниями и опытом между сотрудниками компании SEO TLT с использованием фреймворка Django была построена чёткая и логичная структура проекта. Django предоставляет удобные инструменты, которые позволяют организовать код по отдельным функциональным модулям, что упрощает поддержку и развитие системы.

В корне проекта находятся несколько ключевых файлов:

- `settings.py` — содержит общие настройки платформы, включая параметры подключения к базе данных, пути к статическим и медиафайлам, список установленных модулей, а также настройки безопасности. Здесь же могут задаваться переменные окружения для защиты конфиденциальной информации, такой как секретный ключ проекта;
- `urls.py` — отвечает за маршрутизацию. Он связывает URL-адреса с соответствующими представлениями, которые обрабатывают запросы пользователей. Благодаря этому обеспечивается переход между различными страницами платформы;
- `wsgi.py` и `asgi.py` — обеспечивают взаимодействие приложения с веб-сервером. Эти файлы используются при развертывании платформы и позволяют обрабатывать как синхронные, так и асинхронные запросы.

При создании проекта с помощью команды `django-admin startproject <project_name>` все эти файлы формируются автоматически и размещаются в основной папке с именем проекта.

В рамках платформы реализовано три основных приложения:

- `main` — отвечает за авторизацию и вход пользователей в систему;
- `project` — содержит функционал, связанный с проектами, задачами и обсуждениями;
- `user` — управляет информацией о сотрудниках, их ролями и профилями.

Каждое из этих приложений включает набор основных компонентов:

- `models.py` — содержит описание моделей данных, таких как пользователи, обсуждения, сообщения и т.д.;
- `views.py` — реализует логику обработки пользовательских запросов и взаимодействие с моделями;
- `urls.py` — определяет маршруты внутри приложения и связывает их с соответствующими представлениями;
- `templates/` — папка с HTML-шаблонами, которые используются для отображения данных. Внутри неё создаются подпапки, совпадающие по названию с приложениями, чтобы Django корректно определял, к какому модулю относятся шаблоны;
- `forms.py` — содержит формы для ввода данных (например, авторизация, создание обсуждения и пр.);
- `migrations/` — содержит файлы миграций, которые фиксируют изменения в структуре базы данных.

Помимо этого, в проекте используются статические файлы — стили, скрипты и изображения, обеспечивающие внешний вид и взаимодействие пользователя с интерфейсом.

Таким образом, структура платформы разработана так, чтобы обеспечить надежную работу системы, её возможность масштабирования и удобство как для разработчиков, так и для пользователей. Это позволяет эффективно обмениваться знаниями и опытом внутри компании SEO TLT.

3.3 Реализация и тестирование платформы обмена знаниями

Финальный этап разработки включает демонстрацию готового продукта, в ходе которой осуществляется проверка стабильности работы системы. Тестирование проводится на запущенном сервере, где проверяется функционирование платформы в реальных условиях. Первым действием, которое должен выполнить пользователь, является авторизация. В зависимости от своей роли, сотрудник или администратор попадает на

соответствующий ему рабочий интерфейс. Такой подход к разграничению прав доступа и внешнего вида интерфейса также был заложен при проектировании архитектуры системы во второй главе.

Для входа в систему пользователю необходимо ввести логин(ФИО) и пароль, которые были заранее созданы администратором при регистрации учетной записи. После успешной авторизации платформа автоматически перенаправляет пользователя на соответствующую страницу в зависимости от его роли — сотрудник или администратор. Таким образом, можно сделать вывод, что механизм авторизации работает корректно. Пример экрана авторизации показан на рисунке 20.

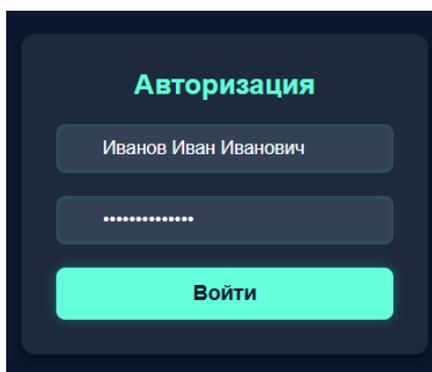


Рисунок 20 – Экран авторизации

Далее рассмотрим, как выглядит и функционирует рабочий экран, предназначенный для сотрудников платформы обмена знаниями. Личный кабинет сотрудников показан на рисунке 21.

Личный кабинет сотрудника предоставляет доступ к базе знаний, возможность публиковать и комментировать материалы, участвовать в обсуждениях, находить экспертов по интересующим темам, а также получать уведомления о новых публикациях и событиях. Это способствует эффективному обмену знаниями внутри организации и профессиональному развитию сотрудников. Далее мы переходим на страницу публикации и создаем новую публикацию.

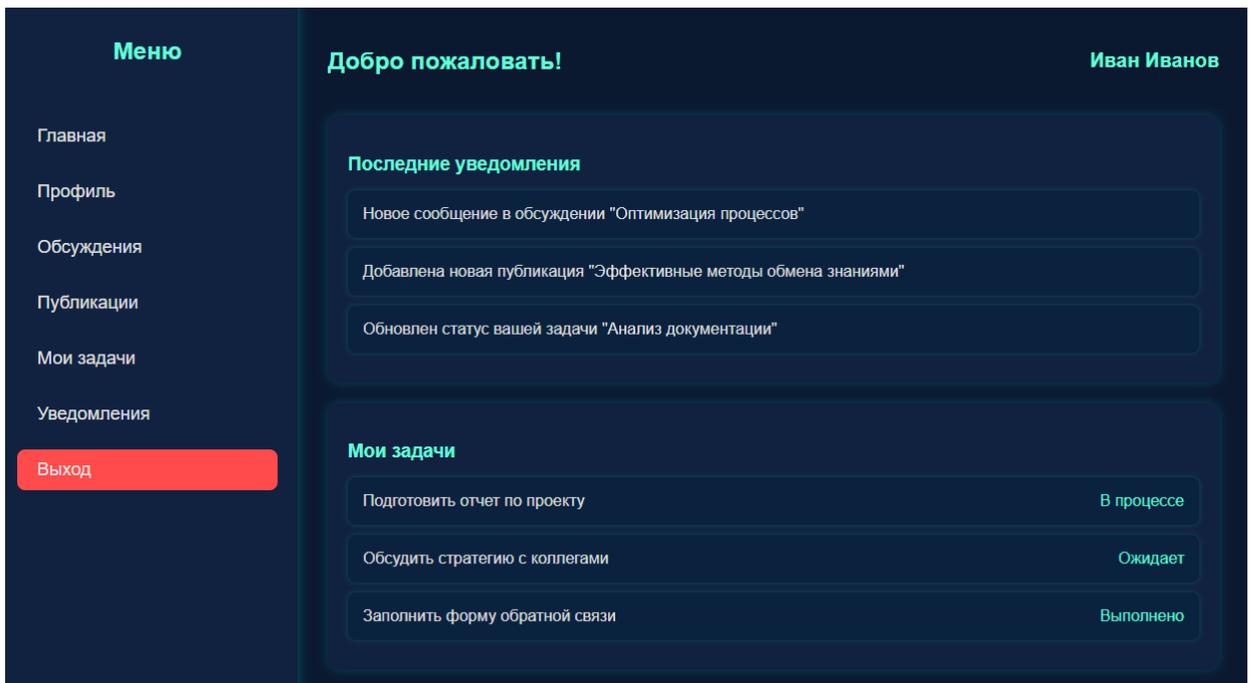


Рисунок 21 – Личный кабинет сотрудника

На рисунке 22 показан процесс до выполнения функции «Опубликовать», а на рисунке 23 — после завершения этой функции.

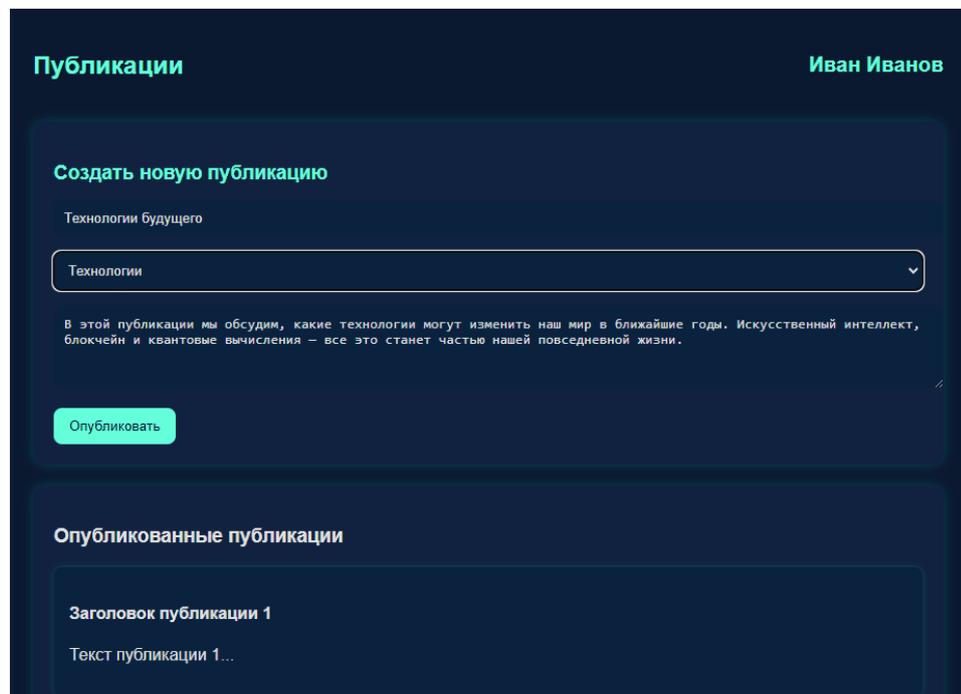


Рисунок 22 – Страница публикации до выполнения функции «Опубликовать»

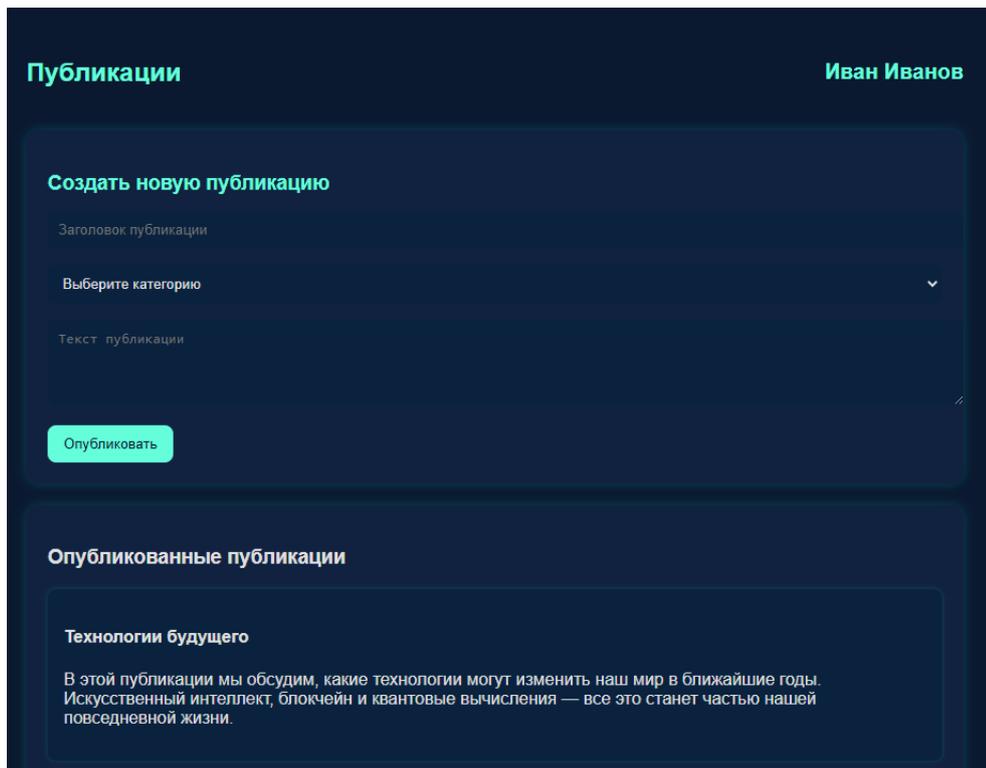


Рисунок 23 – Страница публикации после выполнения функции «Опубликовать»

Функция успешно выполнена и создала новую публикацию для нас. Затем перейдем в раздел обсуждений, который показан на рисунке 24 и создадим новое обсуждение.

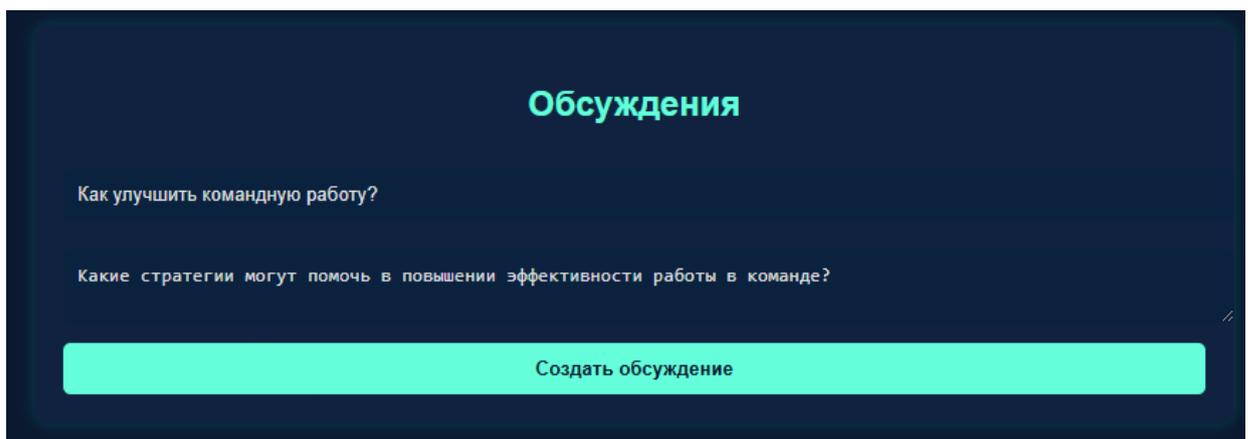


Рисунок 24 – Создание обсуждений

После выполнения функции «Создать обсуждение» на рисунке 25 мы увидим обсуждение, которое было создано ранее.

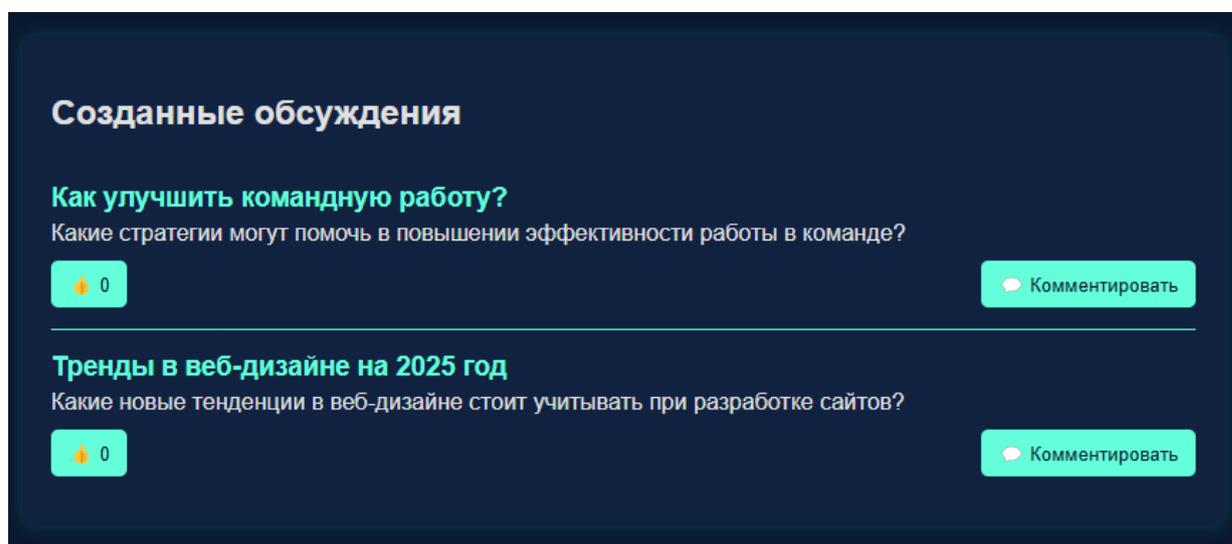


Рисунок 25 – Созданные обсуждения

Обсуждение успешно создано, что свидетельствует о корректной работе основных функций в личном кабинете сотрудника. Перейдем к следующему типу пользователей — администраторам (Project Manager). Контент для сотрудников и администраторов пересекается, но начнем с основ. После авторизации администратора платформа перенаправляет его на главный экран для сотрудников Project Manager, который недоступен и защищен от обычных сотрудников. Этот рабочий экран представлен на рисунке 26.

Страница администратора является ключевым элементом управления платформой и предоставляет администратору доступ ко всем необходимым функциям для эффективного администрирования. Она разделена на несколько основных блоков, что позволяет легко ориентироваться и выполнять задачи. Администратор может добавлять новых сотрудников в систему, заполняя форму регистрации, которая включает обязательные поля: ФИО, email, пароль, повторный ввод пароля и выбор должности.

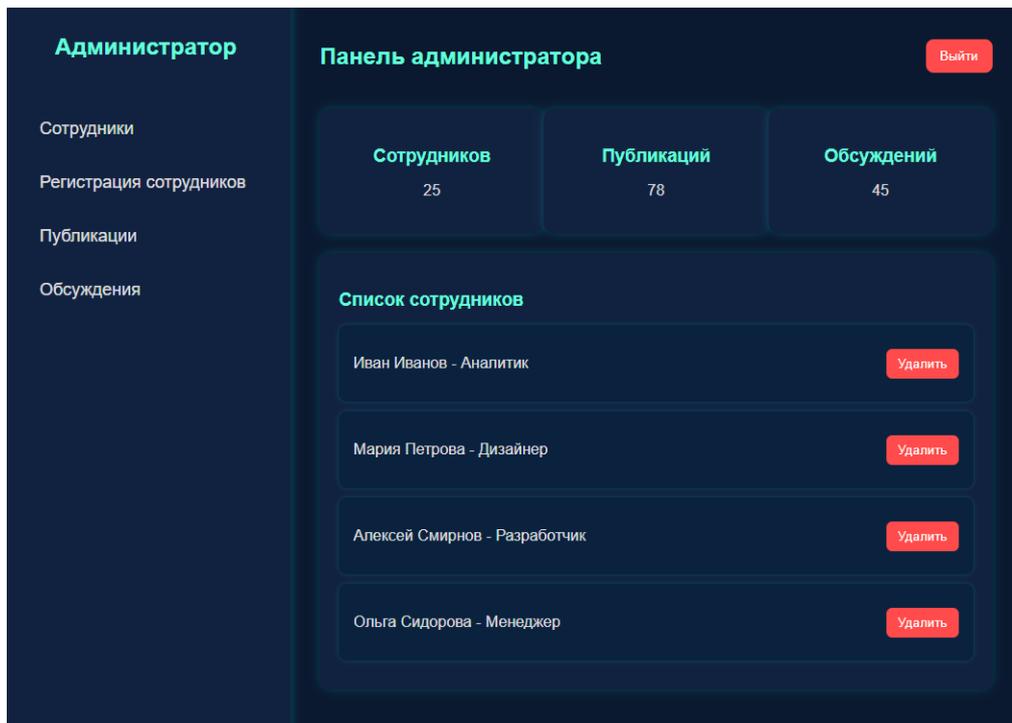


Рисунок 26 – Главный рабочий экран для администраторов

На рисунке 27 представлена «Регистрация» до выполнения функции.

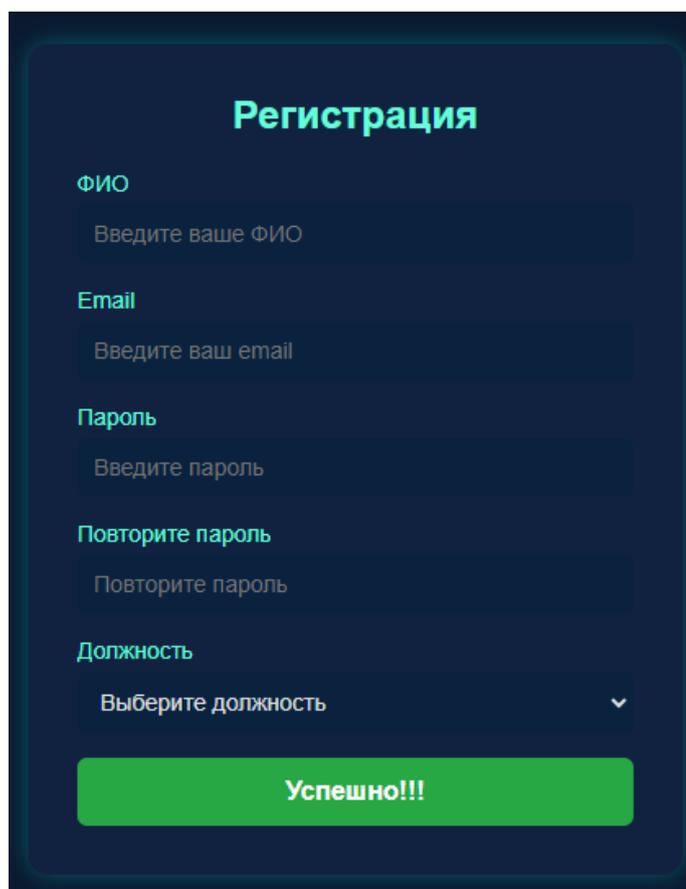
The image shows a registration form titled 'Регистрация' on a dark background. The form contains the following fields:

- ФИО**: Иван Иван Иванович
- Email**: ivan@seotit.com
- Пароль**: masked with dots
- Повторите пароль**: masked with dots
- Должность**: dropdown menu with 'Сотрудник - Разработчик' selected

A large red button labeled 'Зарегистрировать' is positioned at the bottom of the form.

Рисунок 27 – «Регистрация» до выполнения функции

После нажатия на кнопку «Зарегистрировать» информация о новом сотруднике отправляется на сервер и сохраняется в базе данных. Для того чтобы проверить данную важную функцию, создадим нового пользователя, у которого будет должность «Разработчик». Результат продемонстрирован на рисунке 28.



The image shows a registration form on a dark blue background. At the top, the word "Регистрация" is written in a light blue font. Below it are five input fields, each with a label in light blue and a placeholder text in a lighter shade: "ФИО" (placeholder: "Введите ваше ФИО"), "Email" (placeholder: "Введите ваш email"), "Пароль" (placeholder: "Введите пароль"), "Повторите пароль" (placeholder: "Повторите пароль"), and "Должность" (placeholder: "Выберите должность" with a dropdown arrow). At the bottom of the form is a prominent green button with the text "Успешно!!!" in white.

Рисунок 28 – «Регистрация» после выполнения функции

После нажатия кнопки «Зарегистрировать» появляется сообщение «Успешно!!!», что подтверждает успешное выполнение функции и создание нового сотрудника без ошибок. Кроме того, администратор имеет доступ к полному списку сотрудников, включая их должности и статус, что позволяет отслеживать активность сотрудников и управлять командой более эффективно. Страница администратора также предоставляет возможность просматривать всю информацию о текущих публикациях и обсуждениях, что

помогает быстро оценивать их статус и прогресс. Администратор может видеть, какие задачи выполнены, а какие находятся в процессе выполнения. На рисунках 29, 30 и 31 мы можем увидеть разделы «Сотрудники», «Обсуждения» и «Публикации»

| ID | ФИО | Должность | Отдел | Дата приёма | Действия |
|---------|-----------------------------|--------------|-----------|-------------|---|
| EMP-001 | Иванов Алексей Владимирович | Team Lead | Backend | 2025-01-15 | Профиль Удалить |
| EMP-002 | Петрова Мария Сергеевна | UX Designer | Дизайн | 2025-02-01 | Профиль Удалить |
| EMP-003 | Смирнов Дмитрий Игоревич | Data Analyst | Аналитика | 2025-03-10 | Профиль Удалить |
| EMP-004 | Козлова Анна Викторовна | HR Manager | HR | 2025-01-22 | Профиль Удалить |
| EMP-005 | Фёдоров Артём Олегович | Frontend Dev | Frontend | 2025-02-18 | Профиль Удалить |

Рисунок 29 – Раздел «Сотрудники»

| ID | Заголовок | Категория | Автор | Дата | Действия |
|---------|--|--------------------------|--------------|------------|---|
| PUB-015 | Руководство по безопасности API Полное руководство по реализации OAuth 2.0... | Техническая документация | Иванов А.В. | 2025-02-10 | 📄 🗑️ Опубликовано |
| PUB-016 | Анализ рынка IT-2025 Сравнительный анализ облачных решений... | Исследования | Петрова М.С. | 2025-03-01 | 📄 🗑️ Опубликовано |
| PUB-017 | Отчёт Q1 2025 Финансовые показатели и ключевые метрики... | Отчёты | Смирнов Д.И. | 2025-03-05 | 📄 🗑️ Черновик |

Рисунок 30 – Раздел «Публикации»

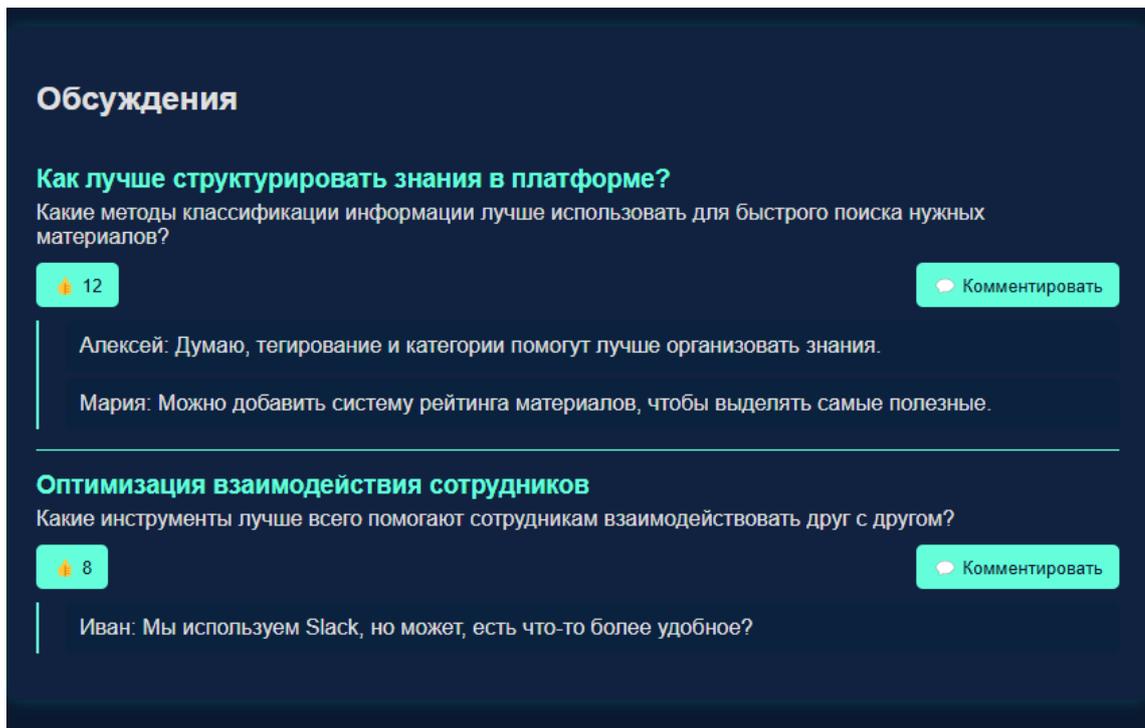


Рисунок 31 – Раздел «Обсуждения»

Таким образом, после тщательного тестирования всех функций платформы можно заключить, что они работают корректно и без ошибок. Все запланированные прецеденты, представленные в диаграмме вариантов использования UML, успешно реализованы и соответствуют заявленным требованиям.

Платформа демонстрирует стабильную работу в стандартных и нестандартных сценариях взаимодействия, обеспечивая надёжный доступ к знаниям и удобство для пользователей. Реализованный функционал подтверждает, что система готова к внедрению в корпоративную среду и способна решать поставленные задачи: способствовать эффективному обмену опытом, ускорять адаптацию новых сотрудников и повышать общую продуктивность работы коллектива.

Заключение

В рамках данной выпускной квалификационной работы была успешно разработана платформа для обмена знаниями и опытом между сотрудниками компании. Созданная платформа представляет собой удобный инструмент для совместного обучения, обсуждения профессиональных вопросов и накопления корпоративных знаний.

На стадии анализа объекта исследования были изучены существующие методы обмена знаниями внутри компании, выявлены основные проблемы текущих подходов, такие как отсутствие единого информационного пространства, сложность доступа к актуальной информации и недостаточная вовлеченность сотрудников в процесс обмена знаниями. Проведен анализ аналогичных решений, на основе которого была сформирована оптимальная модель взаимодействия, направленная на устранение данных недостатков.

На стадии проектирования были определены функциональные и нефункциональные требования к платформе, а также проведен анализ существующих технологий для разработки. В качестве основы для реализации выбран фреймворк Django, который обеспечивает надежность, масштабируемость и гибкость системы. Разработана структура сервиса, включающая такие модули, как публикация материалов, система комментариев и обсуждений, возможность поиска и фильтрации контента. Созданы шаблоны пользовательского интерфейса, обеспечивающие удобную и интуитивно понятную навигацию.

Реализация платформы с использованием Django позволила достичь поставленных целей. Особое внимание было уделено архитектуре системы, механизму управления пользователями, безопасности данных и удобству взаимодействия сотрудников. В сервисе предусмотрена возможность публикации материалов, обсуждения вопросов в тематических разделах, а также поиска необходимой информации с учетом ключевых слов и категорий.

Тестирование платформы подтвердило её функциональность и удобство

использования, продемонстрировав соответствие заявленным требованиям. Проверка различных сценариев работы системы показала ее устойчивость, корректность обработки данных и удобство взаимодействия пользователей.

Таким образом, созданный сервис вносит значительный вклад в развитие корпоративных информационных систем, предоставляя удобную платформу для обмена знаниями и взаимодействия сотрудников. Разработанная платформа способствует улучшению коммуникации, накоплению корпоративных знаний и повышению эффективности работы персонала.

Дальнейшие направления развития проекта могут включать интеграцию с внутренними системами компании, расширение функционала персонализации контента, а также внедрение аналитических инструментов для оценки вовлеченности сотрудников и эффективности использования платформы.

Список используемой литературы

1. Александров, Д. В. (2021). Проектирование и разработка веб-приложений. СПб: Питер.
2. Аснович Н.Г. Использование социальных сетей в образовательном процессе // [б. и.]. — [б. м.], [б. г.]. — [б. с.].
3. Бенеш, И. (2018). Веб-разработка для начинающих. Практическое руководство. Москва: Эксмо.
4. Васильев, В. И., & Петров, С. А. (2020). Программирование веб-приложений: от теории к практике. Москва: Издательство «РТ Софт».
5. Волкодавова Е.В., Погорелова Е.В. Управление трансфером знаний на базе технологической платформы // Фундаментальные исследования. — 2014. — № 12. — С. 1952–1953.
6. Гранчин Кияев. Информационные технологии в управлении. 2011.
7. Земляков Д.В., Штыров А.В. Система дистанционного взаимодействия музейных педагогов в социальных сетях // Электронный научно-образовательный журнал ВГСПУ «Грани познания». — 2014. — № 7(34). — С. 81–82.
8. Иванов В. Разработка веб-приложений: от архитектуры до реализации. — СПб.: Питер, 2020.
9. Исследование различных направлений современной науки: сборник материалов ЛШ-ой международной очно-заочной научно-практической конференции, в 3 т., том 2 / под ред. НИЦ «Империя». — М.: Издательство НИЦ «Империя», 2024. — 196 с.
10. Коммуникативные стратегии информационного общества : труды XIII Междунар. науч.-теор. конф., 22–23 октября 2021 г. — СПб. : ПОЛИТЕХ-ПРЕСС, 2021. — 340 с.
11. Кринчиян Н.А., Катаев А.В. Алгоритм SMM-ориентированного продвижения бренда в социальной интернет-платформе «ВКонтакте» // Южный Федеральный Университет. — [б. г.]. — [б. м.]. — [б. и.].

12. Кричевский В. Ю. Искусство быть наставником // Непрерывное образование. — 2017. — Вып. 3 (21). — С. 6.
13. Леготин И.А., Савченко А.П. Виды стратегий в управлении знаниями // Научно-практический электронный журнал Аллея Науки. — 2019. — № 8(35). — С. 110, 169, 46, 77.
14. Михайлов, О. А. (2020). Рынок веб-приложений и их развитие в современных условиях. Журнал «Информационные технологии», 35(4), 45-52.
15. Никаноров М.С. Индивидуальные особенности сотрудников как фактор мотивации: методы персонализированного подхода // Московская международная академия. — 2024. — [б. и.]. — [б. м.]. — [б. и.].
16. Новиков А.В. Субкультура виртуальной среды: социально-педагогический аспект // Педагогический журнал. — 2023. — Т. 13. — № 9А. — С. 568-574. — DOI: 10.34670/AR.2023.51.36.074.
17. Общество, интеллект, инициатива в контексте междисциплинарных исследований: сборник статей Международной научно-практической конференции / под ред. коллектива авторов. — Уфа: OMEGA SCIENCE, 2025. — 220 с.
18. Петров А. Современные веб-фреймворки и их применение. — М.: Наука, 2019.
19. Руденко, Е. И. (2019). Основы разработки веб-сервисов. Санкт-Петербург: Научная книга.
20. Сидоров Д. Введение в Django и создание веб-приложений. — СПб.: Питер, 2021.
21. Тельнов Ю. Ф. Инжиниринг предприятий и управление знаниями (ИПапр;УЗ-2024) : сборник научных трудов XXVII Российской научной конференции. 28–29 ноября 2024 г. / под науч. ред. Ю. Ф. Тельнова. — Москва : ФГБОУ ВО «РЭУ им. Г. В. Плеханова», 2024. — Т. 1. — 392 с.
22. Django Software Foundation. Django ORM Documentation. [Электронный ресурс]. URL: <https://docs.djangoproject.com/en/stable/topics/db/> (дата обращения: 20.02.2025).

23. Microsoft Corporation. SQL Server Documentation.
24. Michael Owens. The Definitive Guide to SQLite. Apress, 2011.
25. Oracle Corporation. MySQL Reference Manual. [Электронный ресурс].
URL: <https://dev.mysql.com/doc/> (дата обращения: 20.02.2025).
26. PostgreSQL Global Development Group. PostgreSQL Documentation.
[Электронный ресурс]. URL: <https://www.postgresql.org/docs/> (дата обращения: 20.02.2025).
27. Richard Stones, Neil Matthew. Beginning Databases with PostgreSQL.
Apress, 2017.