

УДК 004.01

Хасаншин Ш.Р. / Khasanshin Sh.R.

xsham@inbox.ru

Казанский национальный исследовательский технический

университет им. А.Н. Туполева – КАИ,

Kazan National Research Technical Institute named after A. N. Tupolev – KAI

Научный руководитель: Тутубалин П.И. / Tutubalin P.I.

канд. техн. наук, доцент, доцент каф. ПМИ

университет им. А.Н. Туполева – КАИ,

Kazan National Research Technical Institute named after A. N. Tupolev – KAI

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА ПУБЛИКАЦИОННОЙ АКТИВНОСТИ КАФЕДРЫ

DEVELOPMENT OF AN AUTOMATED SYSTEM FOR RECORDING THE PUBLICATION ACTIVITY OF THE DEPARTMENT

Аннотация: В статье представлен обзор проектирования автоматизированной системы учета публикационной активности кафедры. Система предназначена для управления научными публикациями преподавателей кафедры университета. При этом, предполагается автоматизация процессов подачи, проверки, опубликования и анализа публикаций.

Ключевые слова: учет публикационной активности, автоматизированная система, кафедра, научные публикации, безопасность данных, Flask, React.

Abstract: The article provides an overview of the design of an automated system for tracking the publication activity of a university department. The system is intended for managing the scientific publications of the department's faculty members. It is assumed that the processes of submission, review, publication, and analysis of publications will be automated.

Keyword: Publication activity tracking, automated system, department, scientific publications, data security, flask, react.

Введение

Публикационная активность является ключевым индикатором эффективности работы кафедр в высших учебных заведениях. Согласно данным Минобрнауки России [1], в 2025 году в России насчитывается более 700 вузов, и каждая кафедра сталкивается с необходимостью учета и анализа научных публикаций своих сотрудников. Традиционные методы ручного управления этими процессами приводят к значительным времененным затратам, ошибкам и отсутствию оперативного доступа к данным. Разработка автоматизированной системы с интуитивно понятным интерфейсом и надежной серверной частью становится актуальной задачей для повышения эффективности работы кафедры.

Целью данной работы является описание процесса создания комплексной системы учета публикационной активности кафедры, включающей серверную часть для обработки данных и клиентскую часть для взаимодействия пользователей. В статье описаны этапы проектирования и реализации обеих частей системы, используемые технологии и функциональные возможности.

Основной задачей разработки системы является создание надежного и масштабируемого решения для учета публикационной активности кафедры. Система должна обеспечивать:

- Хранение данных о пользователях кафедры (администраторы, менеджеры);

- Управление научными публикациями (создание, редактирование, проверка, публикация);
- Формирование аналитических отчетов о публикационной активности кафедры;
- Обеспечение безопасности данных через аутентификацию и авторизацию;
- Удобный пользовательский интерфейс для всех категорий пользователей.

Технологии и архитектура

Рассмотрим доступные современные информационные технологии, которые могут быть использованы для создания подобной системы. Системы такого рода традиционно строятся на основе клиент-серверной архитектуры, что обеспечивает разделение логики обработки данных и интерфейса пользователя. Это позволяет добиться высокой производительности, масштабируемости и удобства сопровождения.

Для реализации серверной части системы выбраны следующие технологии:

- **Язык программирования Python с фреймворком Flask:** используется для серверной логики благодаря простоте синтаксиса, гибкости и широкой поддержке сообщества. Flask позволяет быстро разрабатывать REST API, обрабатывать запросы от клиентской части и интегрировать дополнительные библиотеки для расширения функциональности [2]. Например, с помощью Flask можно реализовать маршруты для загрузки файлов, аутентификации и получения аналитических данных.
- **База данных SQLAlchemy с поддержкой PostgreSQL:** SQLAlchemy обеспечивает объектно-реляционное отображение (ORM), что упрощает работу с данными через Python-объекты вместо прямых SQL-запросов. PostgreSQL выбран как надежная реляционная СУБД с поддержкой сложных запросов, транзакций и индексации для ускорения поиска [4]. Например, запросы для подсчёта публикаций по годам или фильтрации по статусу будут оптимизированы благодаря возможностям PostgreSQL [7].
- **Миграции базы данных Flask-Migrate:** инструмент для управления изменениями структуры базы данных. Это особенно полезно при добавлении новых полей (например, для поддержки метаданных публикаций) или изменении связей между таблицами, что обеспечивает гибкость системы в процессе разработки.
- **Аутентификация и сессии Flask-Login и Flask-Session:** Flask-Login управляет процессами входа и выхода пользователей, включая проверку ролей, а Flask-Session обеспечивает безопасное хранение сессий на стороне сервера, минимизируя риски компрометации данных.
- **Безопасность Flask-WTF с CSRF-защитой:** Flask-WTF используется для защиты форм от атак межсайтовой подделки запросов (CSRF) [9]. Это критически важно для защиты данных, например, при загрузке файлов или изменении статуса публикаций.

Серверная часть приложения будет построена по модульной архитектуре с использованием **Blueprint** из фреймворка Flask. Это позволяет разделить функциональность на независимые модули, упрощая масштабирование и поддержку кода. Возможные модули включают:

- **Модуль аутентификации и авторизации:** обеспечивает регистрацию, вход и управление ролями пользователей (user, admin, manager). Поддерживает проверку прав доступа для различных операций, таких как редактирование публикаций или утверждение планов. Например, администратор сможет назначать роли, а менеджер — просматривать отчеты.
- **Модуль управления публикациями:** обрабатывает создание, редактирование, удаление и проверку публикаций. Включает поддержку загрузки файлов (PDF, DOCX) с валидацией форматов и ограничением размера для предотвращения

перегрузки сервера. Также предусматривается возможность добавления метаданных (DOI, ISBN) для интеграции с внешними системами.

- **Модуль аналитики:** предоставляет данные о количестве публикаций по годам через функцию `get_publications_by_year`, которая возвращает агрегированные данные в формате JSON. Дополнительно может включать анализ по типам публикаций (статьи, монографии) и статусам (опубликовано, на проверке) [6].
- **Модуль отчетов:** позволяет менеджерам формировать и просматривать отчеты о публикационной активности. Отчеты могут включать графики (например, динамику публикаций за 5 лет) и экспорт данных в форматы CSV или PDF для дальнейшего использования.
- **Модуль управления планами:** поддерживает создание, утверждение и возврат планов публикационной активности на доработку. Например, менеджер сможет установить целевое количество публикаций на год и отслеживать выполнение плана в реальном времени.

Серверная часть будет поддерживать загрузку файлов через маршрут `/uploads/<filename>` с проверкой форматов (PDF, DOCX) и обеспечит безопасность через CSRF-токены и ролевую модель. Также планируется внедрение кэширования запросов (например, с использованием Redis) для ускорения обработки аналитических данных [8].

Для реализации клиентской части приложения будет использован фреймворк **React.js**, который обеспечит интуитивно понятный и динамичный интерфейс для пользователей кафедры. Основные технологии и компоненты включают:

- **Фреймворк React.js:** используется для построения динамического интерфейса с компонентным подходом. Это позволяет создавать переиспользуемые элементы, такие как формы или таблицы, и обновлять интерфейс без полной перезагрузки страницы [3].
- **Стилизация Material-UI:** применяется для создания современного и адаптивного дизайна. Компоненты Material-UI (кнопки, таблицы, диалоговые окна) обеспечивают единообразный стиль и удобство работы на разных устройствах (ПК, планшеты).
- **Контекстная аутентификация AuthContext:** управляет состоянием пользователя и CSRF-токенами для безопасного взаимодействия с сервером [5]. Например, после входа пользователь получит доступ только к разрешенным ему разделам.
- **HTTP-запросы Axios:** обеспечивает удобное выполнение асинхронных запросов к серверу (GET, POST, PUT, DELETE) с обработкой ошибок и поддержкой перехватчиков для автоматического добавления токенов аутентификации.
- Клиентская часть будет включать следующие ключевые компоненты:
- **Регистрация и вход:** компонент предоставит форму для регистрации новых пользователей с валидацией полей (например, проверка формата email) и интеграцией с сервером через API.
- **Публикации:** компонент позволит просматривать детали публикаций, включая предпросмотр PDF-файлов в браузере [10], скачивание и добавление комментариев с поддержкой вложенных ответов.
- **Панель менеджера:** компонент обеспечит управление планами публикаций, просмотр отчетов и утверждение/возврат работ на доработку с использованием интерактивных элементов (аккордеоны для планов, таблицы для отчетов, диалоговые окна для комментариев).

Клиентская часть предоставит интерактивный интерфейс: пользователи смогут подавать публикации, просматривать их статус и получать уведомления о проверке, а менеджеры — анализировать данные и формировать отчеты. Интерфейс будет адаптирован для различных ролей, включая удобные формы для регистрации и панели управления планами. Для повышения удобства планируется добавить фильтры (по году, типу публикации) и поиск по ключевым словам.

Для реализации приложения будет использована следующая модель данных, представленная в виде реляционной структуры с основными таблицами:

- **User:** данные о пользователях (id, username, password_hash, role, last_name, first_name, middle_name).
- **Publication:** информация о публикациях (id, title, authors, year, type, status, file_url, user_id).
- **Comment:** комментарии к публикациям с поддержкой вложенных ответов (id, publication_id, user_id, text, parent_comment_id).
- **Plan и PlanEntry:** планы публикационной активности и их записи (year, expectedCount, status, entries).

Модель данных поддерживает связи между таблицами (например, внешний ключ user_id в таблице Publication), что позволяет эффективно извлекать данные для отчетов и аналитики.

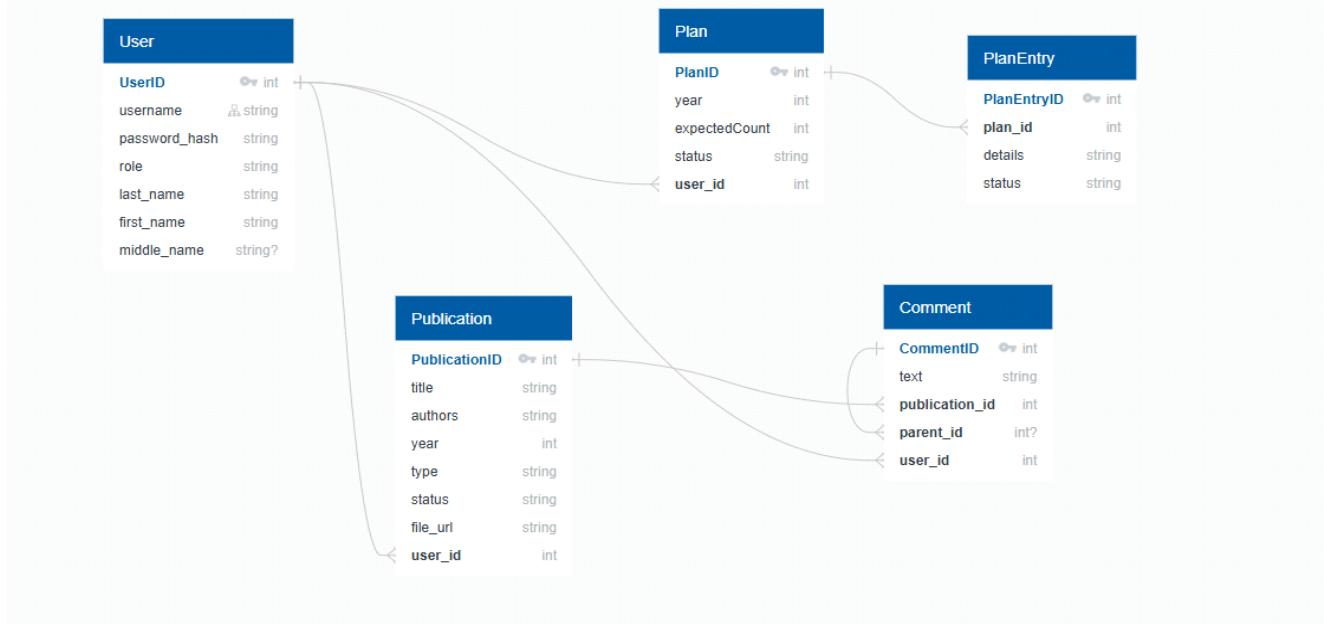


Рис. 1 Диаграмма сущностей и таблиц базы данных.

Заключение

В работе отражен процесс проектирования автоматизированной системы учета публикационной активности кафедры. Можно сделать предположение, что реализация подобной системы с применением выбранных современных инструментов (Flask для серверной части и React.js для клиентской) будет успешной. Система обеспечит надежный контроль, аналитику и отчетность, что повысит эффективность работы кафедры. В будущем планируется интеграция с внешними базами данных (например, Scopus или eLibrary) для автоматического импорта публикаций и расширения функциональности.

Список литературы:

1. Министерство науки и высшего образования Российской Федерации. Подведомственные организации // Официальный сайт Минобрнауки России. — [Электронный ресурс]. — URL: <https://minobrnauki.gov.ru/about/podvedomstvennye-organizatsii/> (дата обращения: 28.03.2025).
2. Flask Documentation // Официальный сайт Flask. — [Электронный ресурс]. — URL: <https://flask.palletsprojects.com/en/3.0.x/> (дата обращения: 28.03.2025).
3. React.js Official Documentation.Concurrent // Официальный сайт React. — [Электронный ресурс]. — URL: <https://react.dev/> (дата обращения: 28.03.2025).
4. PostgreSQL Documentation // Официальный сайт PostgreSQL. — [Электронный ресурс]. — URL: <https://www.postgresql.org/docs/> (дата обращения: 28.03.2025).
5. Тутубалин, П. И. Основные задачи прикладной теории информационной безопасности АСУ / П. И. Тутубалин // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. – 2007. – № 39. – С. 63-72. – EDN JSPVFB.
6. Алгоритм формирования маршрута буровой бригады / Г. В. Спиридовон, В. В. Мокшин, А. П. Кирпичников [и др.] // Вестник Технологического университета. – 2018. – Т. 21, № 2. – С. 169-175. – EDN YTSEWN.
7. Модель оптимизации графового представления для выделения значимых структур на примере предобработки визуальных данных / И. Р. Сайфудинов, В. В. Мокшин, П. И. Тутубалин, А. П. Кирпичников // Вестник Технологического университета. – 2018. – Т. 21, № 5. – С. 121-129. – EDN XREPRB.
8. Метод выделения значимых областей в графической информации для систем поддержки принятия решений / И. Р. Сайфудинов, В. В. Мокшин, А. П. Кирпичников [и др.] // Вестник Технологического университета. – 2018. – Т. 21, № 6. – С. 146-152. – EDN XUALUT.
9. Рахманов, А. С. Автоматизация тестирования безопасности веб-приложений / А. С. Рахманов, П. И. Тутубалин // Молодежь и наука: актуальные проблемы фундаментальных и прикладных исследований : Материалы IV Всероссийской национальной научной конференции студентов, аспирантов и молодых ученых. В 4-х частях, Комсомольск-на-Амуре, 12–16 апреля 2021 года / Редколлегия: Э.А. Дмитриев (отв. ред.), А.В. Космынин (зам. отв. ред.). Том Часть 3. – Комсомольск-на-Амуре: Комсомольский-на-Амуре государственный университет, 2021. – С. 298-301. – EDN MHNKNA.
10. Visible structures highlighting model analysis aimed at object image detection problem / I. R. Saifudinov, V. V. Mokshin, P. I. Tutubalin [et al.] // CEUR Workshop Proceedings : IPERS-ITNT 2018 - Proceedings of the International Conference on Information Technology and Nanotechnology - Session: Image Processing and Earth Remote Sensing, Samara, 24–27 апреля 2018 года. Vol. 2210. – Samara: Без издательства, 2018. – P. 139-148. – DOI 10.18287/1613-0073-2018-2210-139-148. – EDN KDXALF.