УДК 004.45:005.8

АВТОМАТИЗАЦИЯ ПРОЦЕДУР ПРОВЕДЕНИЯ КОНКУРСНЫХ МЕРОПРИЯТИЙ

Демидова Е.А. студентка гр.281074

Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь

Парамонов А.И. – канд. техн. наук, заведующий кафедрой ИСиТ

Аннотация. В работе рассматривается проблема повышения эффективности и прозрачности проведения конкурсных мероприятий. Предложено программное решение, ориентированное на автоматизацию ключевых процессов обработки заявок и экспертной оценки. Разработанная платформа реализует формализованные методы расчёта итоговых результатов и механизмы разрешения конфликтных ситуаций, возникающих при равенстве оценок. Архитектурные и технологические решения направлены на обеспечение масштабируемости, управляемости и устойчивости функционирования. Результаты разработки подтверждают возможность практического применения платформы для организации конкурсных процедур в различных предметных областях.

Ключевые слова. Автоматизация, микросервисная архитектура, конкурсные мероприятия, ранжирование участников, ролевая модель.

Введение. Современные организации активно используют конкурсные процедуры как механизм отбора проектов, распределения ресурсов и принятия управленческих решений. Эффективность таких процедур во многом определяется прозрачностью процессов оценки, корректностью обработки данных и соблюдением установленных регламентов проведения.

Конкурсное мероприятие представляет собой регламентированную процедуру отбора участников на основании заранее определённых критериев оценки, направленную на достижение установленной цели [1].

На рисунке 1 отражены основные принципы, обеспечивающие прозрачность и объективность конкурсных мероприятий.

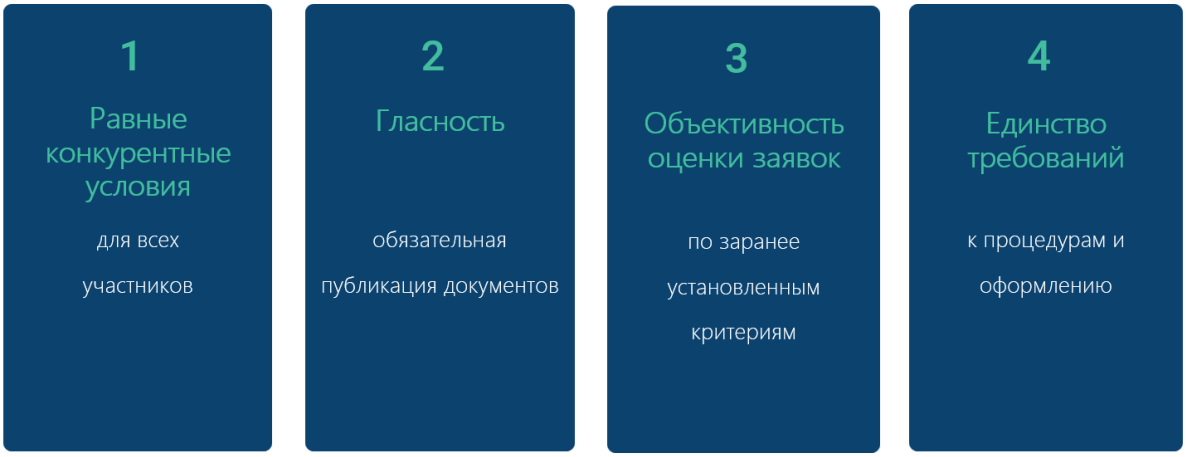

Рисунок 1 – Базовые принципы проведения конкурсных мероприятий

Конкурс может быть рассмотрен как управляемый процесс, состоящий из последовательности взаимосвязанных этапов, включающих подготовку условий проведения, подачу заявок, их административную проверку, экспертную оценку и формирование итоговых результатов.

На начальном этапе осуществляется подготовка конкурсного извещения, в котором формируются условия участия, сроки проведения, требования к конкурсным материалам и критерии оценки заявок. После публикации извещения начинается этап приёма заявок на участие в установленный срок. Каждая заявка проходит процедуру предварительной проверки, направленной на подтверждение соответствия установленным требованиям. По результатам проверки формируется перечень заявок, допущенных к экспертной оценке. Этап экспертной оценки предполагает анализ представленных материалов членами комиссии и выставление баллов по установленным критериям. На завершающем этапе осуществляется сбор и агрегирование результатов оценки, формирование рейтинга участников и формирование итогового протокола конкурса.

Актуальность разработки специализированного программного средства обусловлена тем, что конкурсные процедуры сегодня широко применяются в различных сферах: в образовательной и научной деятельности, при распределении грантов и субсидий, при кадровом отборе, а также в различных профессиональных и творческих соревнованиях. Вместе с тем на практике такие процессы

нередко остаются недостаточно структурированными: значительная часть операций выполняется вручную, информация хранится разрозненно, а контроль соблюдения нормативов во многом зависит от человеческого фактора.

Применяемый процесс проведения конкурса создаёт значительную нагрузку организаторов, усложняет координацию работы комиссии и снижает прозрачность процедур. Практический анализ типового процесса проведения конкурса наглядно выявил эти недостатки и подтвердил необходимость создания специализированного программного решения.

Предлагаемое программное средство ориентировано на формирование единой цифровой среды для обеспечения сквозной поддержки всех этапов конкурсного мероприятия, объединённых в логически выстроенный и управляемый процесс. Платформа обеспечивает централизованное хранение данных, автоматизацию расчётов, повышение прозрачности процедур и воспроизводимость получаемых результатов.

Архитектурное решение программной платформы.

В основу решения положена микросервисная архитектура, предусматривающая разделение платформы на независимые компоненты, каждый из которых отвечает за свою функциональную область: аутентификацию пользователей, управление конкурсами, обработку заявок, проведение экспертной оценки и формирование итоговых результатов [2].

В рамках архитектурной декомпозиции были выделены следующие основные сервисы:

- сервис управления пользователями и аутентификацией;
- сервис управления конкурсными мероприятиями;
- сервис обработки заявок участников;
- сервис экспертной оценки и формирования итоговых результатов.

На рисунке 2 представлена общая архитектура платформы.

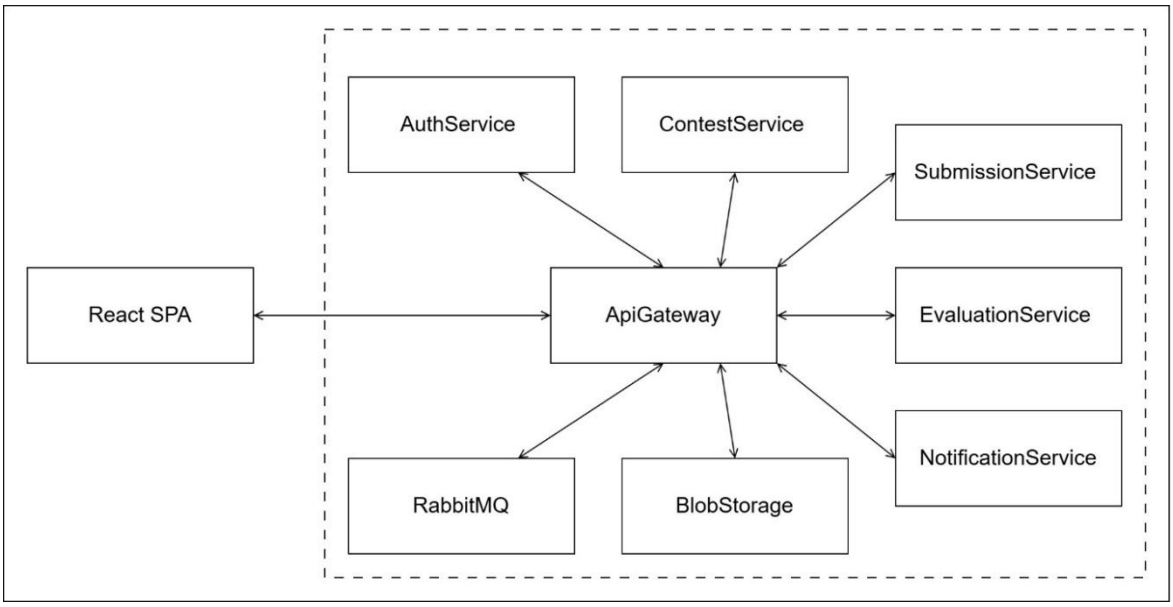

Рисунок 2 – Архитектура программного средства (платформы для проведения конкурсов)

Использование микросервисной архитектуры позволяет обеспечить модульность и независимость компонентов системы, возможность масштабирования отдельных сервисов, упрощение сопровождения и развития программного комплекса, и повышение отказоустойчивости системы. Внутренняя структура сервисов реализована в соответствии с концепцией «Clean Architecture», предусматривающей разделение системы на уровни, что изолирует бизнес-логику от деталей реализации и упрощает изменение технологий хранения данных и интеграции без переработки прикладной логики [3].

При разработке программного средства соблюдались принципы «SOLID», что снижает связанность компонентов, повышает предсказуемость поведения системы и упрощает её развитие [4].

Проектирование сопровождалось построением архитектурных схем, диаграммы вариантов использования, деятельности, классов и описанием алгоритмов работы системы. Всё перечисленное позволило заранее формализовать бизнес-логику, проработать пользовательские сценарии и убедиться в логической согласованности всех процессов ещё на этапе проектирования.

Отдельно стоит отметить, что авторизация реализована на основе ролевой модели. В рамках платформы предусмотрены следующие категории пользователей: организатор; участник; эксперт; секретарь комиссии; администратор. Каждой роли соответствует набор допустимых операций, что обеспечивает разграничение доступа к функциональности системы.

Также дополнительно реализованы следующие механизмы защиты:

– передача данных по защищённым протоколам;
– валидация входных данных;
– журналирование действий пользователей.

Эти аспекты важны при построении современных цифровых платформ и требует детальной проработки, чтобы обеспечить надежное и стабильное функционирование.

Особенности реализации алгоритмов.

Отдельное внимание в работе уделено разработке вычислительных алгоритмов. Были реализованы различные стратегии расчёта итоговых баллов, а также формализованные методы разрешения ситуаций равенства результатов, что имеет принципиальное значение для обеспечения объективности и корректности конкурсных процедур [5].

Стратегия *TotalScore* (суммарный балл) обеспечивает расчёт итоговой оценки как суммы всех выставленных баллов:

$$TotalScore(submission) = \sum_{i=1}^{n} Score_{\text{i}}\,, \quad (1)$$

где $Score_i$— количество баллов, установленное при оценке экспертом по *i*-му критерию.

Преимущества:
– простая и прозрачная методика;
– высокая вычислительная эффективность (используются агрегирующие запросы без сложной постобработки);
– подходит для конкурсов с большим количеством критериев.

Особенности применения:
– все критерии имеют одинаковый вес;
– в случае равенства итоговых сумм применяются стратегии разрешения ничьих.

Стратегия *WeightedScore* (взвешенный балл) учитывает различную значимость критериев:

$$WeightedScore(submission) = \sum_{i=1}^{n} (Score_i \cdot Weight_i), \quad (2)$$

где $Weight_i$ — весовой коэффициент значимости критерия *i*-го критерия.

Преимущества:
– гибкость настройки под любой тип конкурса;
– позволяет учесть приоритетность различных аспектов конкурсных работ;
– применима в регламентированных и многоуровневых оценочных системах.

Особенности применения:
– требует корректной настройки весов на этапе подготовки мероприятия.
– позволяет уменьшить количество ничьих за счёт более точного ранжирования.
– используется преимущественно на финальных этапах.

Стратегия *MedianScore* (медианный балл) определяет итоговую оценку как медиану выборки:

$$MedianScore(submission) = median(Score_1, Score_2, \ldots, Score_n). \quad (3)$$

Преимущества:
– устойчивость к выбросам и аномальным значениям;
– снижает субъективность отдельных экспертов;
– хорошо применима при большом количестве экспертов.

Особенности применения:
– дополнительная логика вычислений;
– может давать идентичные медианы у разных участников, что повышает вероятность ничьих.

Стратегия *ExpertPriorityScore* (с приоритетом экспертов) учитывает значимость отдельных экспертов:

$$ExpertPriorityScore(submission) = \sum_{i=1}^{n} (Score_{expert(i)} \cdot ExpertWeight_i), \quad (4)$$

где $ExpertWeight_i$— весовой коэффициент значимости оценки *i*-го эксперта.

Преимущества:
– позволяет учесть компетентность и роль экспертов;
– обеспечивает более взвешенное распределение баллов;

– подходит для специализированных конкурсов (научных, инженерных).

Особенности применения:

– требует корректной настройки приоритетов.

– при неправильной конфигурации может привести к чрезмерной концентрации веса на одном эксперте.

– не всегда подходит для массовых конкурсов с равноправным голосованием.

После расчёта итоговых значений выполняется ранжирование заявок. Рейтинг формируется путём сортировки по убыванию итогового балла:

$$R = sort;\ k = 1, \dots, N. \quad (5)$$

Позиция заявки определяется её местом в упорядоченном списке:

$$P_k = rank(S_k),. \quad (6)$$

Параметр $Top\text{-}N$ задаётся на уровне этапа и используется для:

– отбора заявок в следующий этап;

– формирования списка финалистов;

– определения победителей.

При совпадении значений на границе $Top\text{-}N$ в результирующее множество включаются все заявки с одинаковым баллом.

Для разрешения ситуаций равенства реализованы следующие стратегии:

– *BySubmissionDate* — по времени подачи заявки;

– *ByHigherKeyCriteriaScore* — по ключевым критериям;

– *ByPrioritizedExpertScore* — по приоритетным экспертам;

– *Manual* — ручное решение.

Технологии и средства реализации программной платформы.

Выбор используемого технологического стека обусловлен требованиями к масштабируемости, надёжности и удобству сопровождения программного средства. Используемые решения охватывают серверную, клиентскую и инфраструктурную части системы и обеспечивают согласованную работу всех компонентов платформы.

Серверная часть реализована на базе платформы .NET 9 с использованием ASP.NET Core 9.0, что обеспечивает высокую производительность обработки запросов, встроенные механизмы маршрутизации и гибкую организацию веб-API. Данный стек позволяет эффективно реализовать сервисную архитектуру и обеспечить устойчивую работу при увеличении нагрузки. Для взаимодействия с базой данных используется Entity Framework Core совместно с провайдером Npgsql для PostgreSQL. Применение ORM-инструмента позволяет работать с данными на уровне объектной модели, упрощает разработку и сопровождение, а также обеспечивает согласованность структуры базы данных за счёт механизма миграций.

Обмен данными между сервисами организован с использованием RabbitMQ, выступающего в роли брокера сообщений. Такой подход обеспечивает асинхронную обработку задач, снижает связанность компонентов и позволяет перераспределять нагрузку между сервисами. Для работы с файловыми ресурсами используется Azure Blob Storage, обеспечивающее надёжное хранение файлов большого объёма и возможность масштабирования без увеличения нагрузки на основную базу данных.

Клиентская часть реализована с использованием React и языка TypeScript, что обеспечивает создание интерактивного и типобезопасного пользовательского интерфейса. Использование Vite ускоряет процесс сборки и разработки приложения. Для построения пользовательского интерфейса применяется библиотека Material UI, обеспечивающая единообразие визуального представления. Взаимодействие с серверной частью осуществляется через библиотеку Axios, которая позволяет централизовать выполнение HTTP-запросов, управлять заголовками и обрабатывать ошибки на уровне клиентского слоя.

Инфраструктурная часть построена с использованием контейнеризации на базе Docker и Docker Compose, что обеспечивает изолированное и воспроизводимое окружение для всех компонентов системы. В составе инфраструктуры используются PostgreSQL 16 в качестве основной СУБД и RabbitMQ 3 как брокер сообщений. Для локальной разработки и тестирования применяется Azurite, эмулирующий работу Azure Blob Storage. Клиентская часть развёртывается с использованием Node.js 20 и проксируется через nginx, что обеспечивает эффективную доставку статического контента.

Таким образом, выбранный технологический стек обеспечивает надёжную основу для разработки, масштабирования и эксплуатации программного средства, позволяя реализовать как вычислительные, так и инфраструктурные требования системы.

Аутентификация пользователей реализована на основе механизма JSON Web Token. После успешного входа пользователь получает токен, который используется при обращении к API, что

обеспечивает проверку подлинности пользователя при каждом запросе без повторной передачи учётных данных.

Заключение. В работе описана архитектура и особенности реализации полнофункционального программного средства для обеспечения автоматизации процедур проведения конкурсных мероприятий. Серверная архитектура обеспечивает обработку бизнес-логики, хранение данных и файлов, а также защиту и разграничение доступа. Клиентская часть предоставляет удобные средства взаимодействия для всех категорий пользователей. Использование документированного программного интерфейса делает систему пригодной для интеграции с внешними информационными системами организаций. Принятые архитектурные и технологические решения обеспечивают масштабируемость, расширяемость и практическую применимость разработанного программного средства.

Для подтверждения корректности работы ключевых компонентов были разработаны и проведены модульные тесты, направленные на проверку алгоритмов расчёта и разрешения ничьих. Результаты тестирования продемонстрировали соответствие фактических значений ожидаемым, что подтверждает корректность реализованной логики и готовность к практическому использованию.

Разработанное решение ориентировано на повышение прозрачности, объективности и управляемости конкурсных процедур и может быть эффективно использовано в организациях, где данные характеристики являются критически важными.

Дальнейшее развитие системы целесообразно ориентировать на расширение и углубление реализованных механизмов управления конкурсом. В первую очередь это касается системы уведомлений: помимо базовых оповещений, может быть реализована событийная модель с гибкой настройкой триггеров (изменение статуса заявки, завершение этапа, необходимость оценки), а также поддержка различных каналов доставки (электронная почта, push-уведомления, интеграция с корпоративными мессенджерами). Рассматривается возможность реализации конструктора конкурсных извещений с параметризацией условий участия и автоматической генерацией документации. Также перспективным направлением модернизации является развитие аналитического слоя системы, включающего расширенные средства визуализации результатов: распределение оценок, сравнительный анализ участников, выявление отклонений в экспертных оценках. Это позволит использовать систему не только как инструмент проведения конкурса, но и как средство анализа его результатов. По итогам эксплуатации будет рассмотрена целесообразность развития механизма апелляций с поддержкой многоэтапного рассмотрения и фиксацией всех изменений, что обеспечит прозрачность и воспроизводимость принимаемых решений. С точки зрения масштабирования и нагрузочной эксплуатации предполагается совершенствование механизмов отказоустойчивости, что позволит адаптировать систему к росту числа пользователей и конкурсных мероприятий.

Реализация указанных направлений развития позволит трансформировать текущую версию платформы из инструмента автоматизации отдельных процедур в полноценную универсальную платформу управления конкурсными мероприятиями, адаптируемую под различные предметные области и масштабы применения.

Список использованных источников:

1.Постановление Министерства образования Республики Беларусь 12 апреля 2021 г. № 65 [Электронный ресурс] / Национальный правовой Интернет-портал Республики Беларусь, 26.05.2021, 8/36706. Режим доступа: https://pravo.by/. Дата обращения: 15.04.2026.

2.Richardson, C. Microservices Patterns (2nd Edition) / C. Richardson — Manning Publications, — 2024. — 592 p.

3.Martin, R. C. Clean Architecture / R. C. Martin — Pearson Education, — 2018 — 432 p.

4.Martin, R. C. Agile Software Development, Principles, Patterns, and Practices / R. C. Martin. — Upper Saddle River: Prentice Hall, 2003. — 552 p.

5.Kendall, M. G. The Advanced Theory of Statistics / M. G. Kendall, A. Stuart. — London: Charles Griffin & Company, 1969. — 585 p.

UDC 004.45:005.8

AUTOMATION OF COMPETITIVE PROCEDURES

Demidova E. A.

Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Paramonov A.I. – Candidate of Technical Sciences, Associate Professor

Annotation. This paper addresses the problem of improving the efficiency and transparency of competitive procedures. A software solution focused on automating key processes of application processing and expert evaluation is proposed. The developed platform implements formalized methods for calculating final results and mechanisms for resolving conflicts arising from tied scores. The architectural and technological solutions are aimed at ensuring scalability, manageability, and operational reliability. The development results confirm the possibility of practical application of the platform for organizing competitive procedures in various subject domains.

Keywords. Automation, microservice architecture, competitive procedures, participant ranking, role-based model.