УДК 331.101.64 DOI: 10.18324/2224-1833-2026-3-37-44

Разработка и практическая реализация автоматизированной системы «Абитур» для проведения вступительных испытаний в компьютерном классе ВУЗа

Н.А. Гончарова[1a], Т.М. Левченко[2b], С.А. Кириенко[2c]

[1] Братский государственный университет, ул. Макаренко, 40, Братск, Россия
[2] Сибирский Государственный Университет Телекоммуникации и Информатики, ул. Гурьевская, 51, Новосибирск, Россия
[a] goncharova-n@mail.ru, [b] Levchenko.64@mail.ru, [c] semenkirencko@yandex.ru
[a] https://orcid.org/0000-0002-4231-0272, [b] https://orcid.org/0009-0008-8123-1674
Статья поступила 14.09.2026, принята 14.10.2026.

В статье рассматриваются вопросы разработки и практической реализации автоматизированной системы «Абитур», предназначенной для проведения вступительных испытаний в компьютерном классе высшего учебного заведения. Актуальность исследования обусловлена необходимостью повышения оперативности, объективности и надёжности приёмной кампании, особенно при значительном количестве абитуриентов, ограниченных сроках проведения испытаний и большом объёме тестовых заданий. Традиционная бумажная организация экзаменов требует значительных трудозатрат, сопровождается риском ошибок при обработке ответов, переносе результатов и подготовке конкурсных списков, а также затрудняет рассмотрение апелляций. Целью работы является создание программного комплекса, обеспечивающего автоматизацию полного цикла вступительного испытания: от формирования банка заданий и подготовки индивидуальных вариантов до оценивания результатов, формирования отчётности и анализа качества тестовых материалов. Описана клиент-серверная архитектура системы, включающая административную панель, клиентское приложение абитуриента и сервер синхронизации. Рассмотрены возможности импорта заданий из таблиц, текстовых документов, PDF-файлов, сканов и формата QTI 2.x, а также поддержка различных типов заданий, включая тестовые, числовые, текстовые, формульные задания, эссе, загрузку файлов и программный код. Особое внимание уделено информационной безопасности и защите персональных данных. Система не передаёт правильные ответы на рабочее место абитуриента, контролирует время проведения испытания, фиксирует отклонения от установленного регламента, ограничивает использование буфера обмена, второго монитора и средств удалённого доступа. Серверное оценивание обеспечивает воспроизводимость результатов и снижает вероятность субъективных ошибок. Реализованные аналитические инструменты позволяют выявлять задания низкого качества, оценивать их различающую способность, анализировать совпадения ошибок и формировать конкурсные списки с подробной расшифровкой итоговых баллов. Результаты модульного, интеграционного и нагрузочного тестирования подтвердили работоспособность системы при одновременном проведении испытаний для большого числа пользователей. Практическая значимость разработки заключается в сокращении ручных операций, ускорении подведения итогов, повышении прозрачности приёмной кампании и создании условий для оперативного рассмотрения апелляций. Система позволяет существенно снизить временную нагрузку на сотрудников приёмной комиссии и повысить их производительность труда ориентировочно на 15–20 %. Рост производительности труда достигается за счёт исключения дублирующих действий и передачи типовых, повторяющихся операций автоматизированной системе. Сотрудники приёмной комиссии освобождаются от трудоёмкого ручного внесения данных, многократной проверки документов, ручного расчёта баллов и формирования промежуточных списков.

Ключевые слова: вступительные испытания; приёмная кампания; компьютерное тестирование; прокторинг; клиент-серверная архитектура; серверное оценивание; банк заданий; анализ качества тестовых заданий; защита персональных данных; NET.

Development and practical implementation of the automated system "Abitur" for conducting entrance exams in a computer classroom of a higher education institution

N.A. Goncharova[1a], T.M. Levchenko[2b], S.A. Kiriyenko[2c]

[1] Bratsk State University; 40, Makarenko St., Bratsk, Russia
[2] Siberian State University of Telecommunications and Informatics, Guryevskaya St., 51, Novosibirsk, Russia
[a] goncharova-n@mail.ru, [b] Levchenko.64@mail.ru, [c] semenkirencko@yandex.ru
[a] https://orcid.org/0000-0002-4231-0272, [b] https://orcid.org/0009-0008-8123-1674
Received 14.09.2026, accepted 14.10.2026.

The article discusses the development and practical implementation of the automated system "Abitur," designed for conducting entrance examinations in a computer classroom of a higher education institution. The relevance of the study is driven by the need to improve the efficiency, objectivity, and reliability of the admissions campaign, especially given a significant number of applicants, limited examination timeframes, and a large volume of test items. Traditional paper-based examination organization requires considerable labor costs,

is accompanied by the risk of errors in processing answers, transferring results, and preparing competitive lists, and also complicates the review of appeals. The aim of the work is to create a software package that automates the full cycle of the entrance examination: from forming a bank of tasks and preparing individual variants to evaluating results, generating reports, and analyzing the quality of test materials. The client-server architecture of the system is described, including an administrative panel, an applicant client application, and a synchronization server. The capabilities of importing tasks from spreadsheets, text documents, PDF files, scans, and the QTI 2.x format are considered, as well as support for various types of tasks, including multiple-choice, numerical, text, formula-based tasks, essays, file uploads, and program code. Particular attention is paid to information security and personal data protection. The system does not transmit correct answers to the applicant's workstation, controls the time of the examination, records deviations from the established regulations, and restricts the use of the clipboard, a second monitor, and remote access tools. Server-side grading ensures reproducibility of results and reduces the likelihood of subjective errors. The implemented analytical tools make it possible to identify low-quality tasks, assess their discriminating power, analyze error coincidences, and generate competitive lists with a detailed breakdown of final scores. The results of modular, integration, and load testing confirmed the system's operability during simultaneous examinations for a large number of users. The practical significance of the development lies in reducing manual operations, accelerating the summing up of results, increasing the transparency of the admissions campaign, and creating conditions for the prompt review of appeals. The system makes it possible to significantly reduce the time burden on admissions committee staff and increase their labor productivity by approximately 15–20 %. The growth in labor productivity is achieved by eliminating duplicate actions and transferring typical, repetitive operations to the automated system. Admissions committee staff are freed from labor-intensive manual data entry, repeated document verification, manual score calculation, and the generation of intermediate lists.

Keywords: entrance exams; admissions campaign; computer-based testing; proctoring; client-server architecture; server-based assessment; task bank; analysis of test item quality; personal data protection; NET.

Введение. За приёмную кампанию через испытания проходит около 6000 человек. В отдельный день их бывает до 600, а банк заданий по одному предмету доходит до 2000 вопросов. Укладывается это в несколько недель июля.

На таких объёмах бумажная форма даёт предсказуемые потери. Печать и раскладка билетов, ручная проверка бланков, перенос баллов в ведомости – каждая операция отнимает время и допускает ошибку. Отдельная история начинается, когда человек приходит на апелляцию и его работу нужно найти. Итог испытания в лучшем случае появляется на следующий день.

Готовые системы дистанционного обучения закрывают задачу частично [1, с. 601]. Они рассчитаны на учебный процесс, а не на разовое испытание с юридическими последствиями. Им нужен устойчивый канал в интернет. Рабочее место абитуриента они не блокируют, внятного протокола нарушений комиссия от них не получает.

Требования к системе сформулированы вместе с приёмной комиссией:

1) наполнение банка заданий из тех файлов, которые у предметных комиссий уже есть, – таблиц, документов, сканов;

2) индивидуальный вариант каждому абитуриенту, восстановимый при разборе апелляции;

3) невозможность узнать правильные ответы на рабочем месте даже при полном доступе к компьютеру;

4) фиксация отклонений от регламента с точным временем;

5) автоматическое оценивание и отчёты в день испытания [2, С. 33].

По отдельности ни одно требование не ново. Работает именно их сочетание, причём в расчёте на изолированный компьютерный класс.

Архитектура. «Абитур» построен на платформе .NET 10 под Windows. Три программы и четыре слоя (рис. 1).

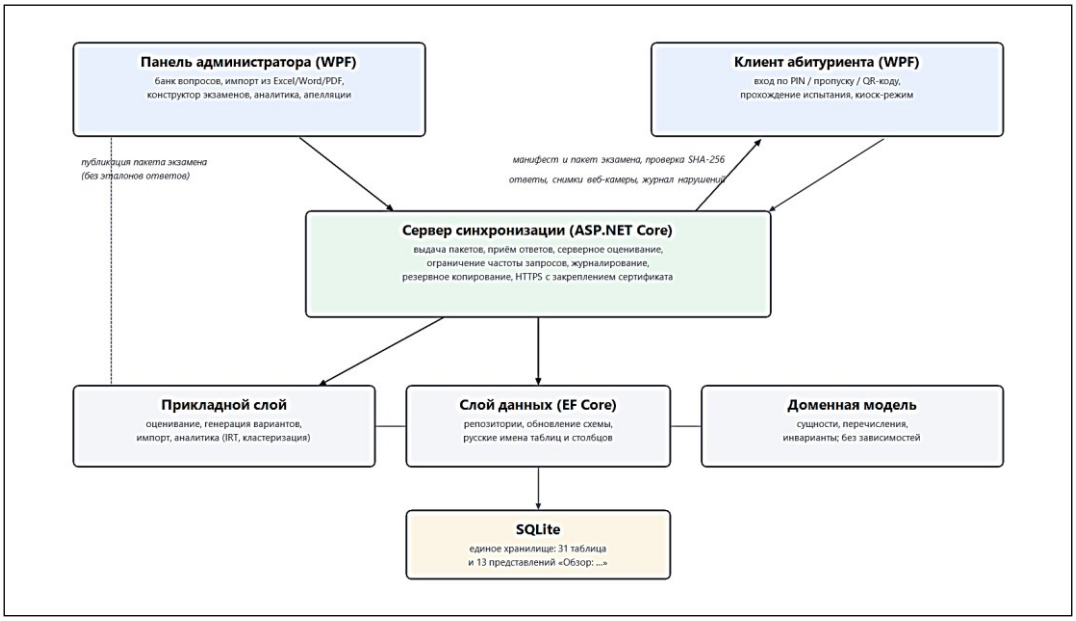


Рис. 1. Архитектура системы «Абитур»

В панели администратора работают методист и секретарь: банк заданий, импорт, конструктор испытаний, рассадка, результаты, апелляции, журнал аудита. Клиент абитуриента ставится на каждое место в классе. Сервер синхронизации раздаёт пакеты испытаний, принимает ответы и оценивает их. Ниже лежат прикладной слой, слой данных на EF Core и доменная модель, которая не зависит ни от чего. Зависимости идут только вниз, обратных вызовов нет.

Хранилище одно, SQLite. От сетевой базы Access, заложенной в первой редакции технического задания, пришлось отказаться: на 10–20 одновременных писателях она деградирует и повреждается, а речь идёт о сотнях рабочих мест [3, С. 15].

Поток данных односторонний. Содержимое испытания пишет только панель, ответы только клиент, и только добавлением. Конфликтов записи нет по построению, а не потому, что их кто-то разрешает. Версия содержимого помечена номером ревизии и контрольной суммой SHA-256, клиент сверяет её до начала испытания.

Банк заданий. Поддерживаются десять типов заданий: одиночный и множественный выбор, числовой ответ, короткий текст, ввод формулы, сопоставление, упорядочивание, загрузка файла, эссе, программный код (рис. 2).

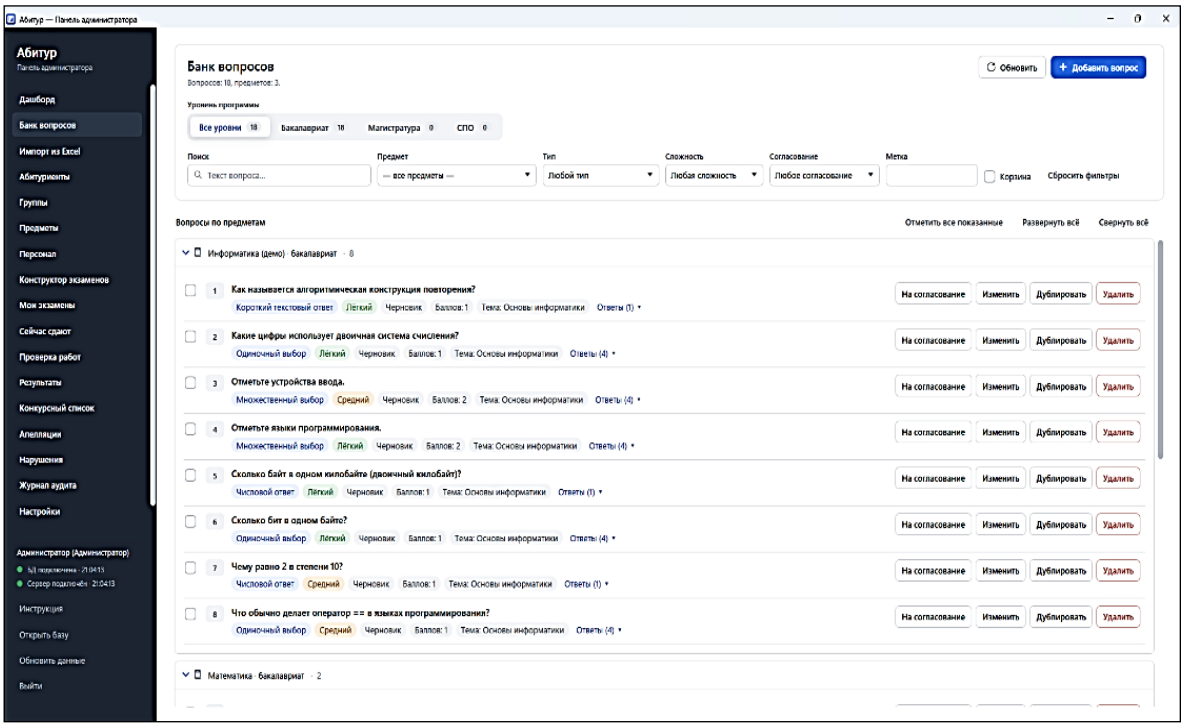


Рис. 2. Банк заданий в панели администратора

Набрать вручную 2000 вопросов по предмету невозможно, поэтому главным способом ввода сделан импорт. Читаются книги Excel, документы Word, файлы PDF, сканы страниц и пакеты QTI 2.x [14]. Разбирают их разные библиотеки: ClosedXML, OpenXml, PdfPig, штатное распознавание Windows. Предпросмотр при этом одинаковый, и дальше работает общий конвейер. Он проверяет строки, отсеивает дубликаты по ключу «предмет + текст», пишет протокол ошибок с номером строки и сохраняет вопросы в банк. На тысяче вопросов от выбора файла до записи в базу уходит около 16 секунд при нормативе 30. Вариант номер i строится по зерну seed(i) = master + i·7919 и по нему же пересобирается байт в байт в любой момент. Требование здесь процедурное, а не техническое: на апелляции комиссия обязана видеть ровно тот набор и порядок заданий, который видел абитуриент, а не похожий.

Проведение испытания. Войти можно по шестизначному PIN-коду, пропуск-коду, QR-коду с печатного пропуска или по логину и паролю. Пакет приходит на рабочее место без ключей правильных ответов. У заданий со свободным ответом в нём нет и перечня вариантов: эталоны и критерии эксперта остаются на сервере. Полный доступ к компьютеру и к кэшу клиента не даёт абитуриенту ничего.

Время контролирует сервер [4, С. 80]. Он назначает срок окончания сессии при старте, клиент лишь показывает обратный отсчёт. Перевод системных часов на рабочем месте времени не добавляет и сам попадает в журнал.

Клиент работает в полноэкранном защищённом режиме. Заблокированы Alt+Tab, Win, Alt+F4 и снимок экрана, окно скрыто от программ записи, отслеживаются буфер обмена, второй монитор и средства удалённого доступа. Каждое отклонение записывается с точным временем и показывается абитуриенту (рис. 3). Испытание при этом не прерывается: решение принимает комиссия, система только фиксирует факт. Черновик ответов уходит на сервер каждые пять секунд. Если связь пропала, ответы копятся в локальной очереди, а повторная доставка уже оценённой сессии обрабатывается как повтор [6, С. 119].

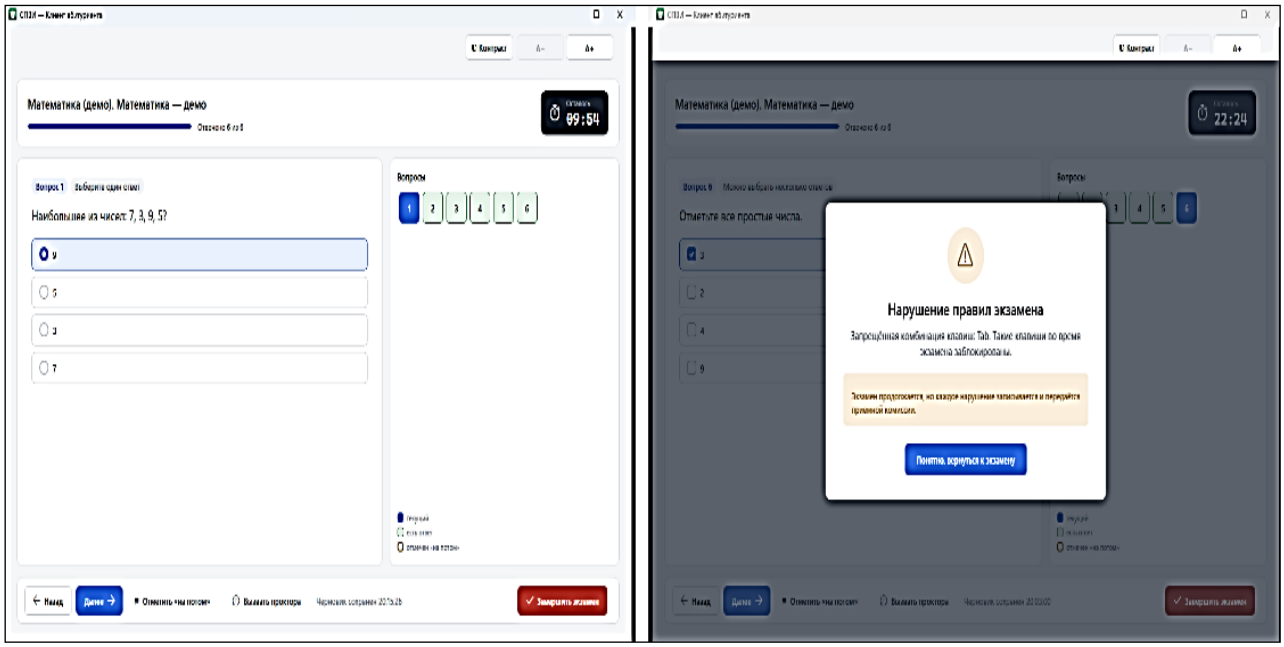


Рис. 3. Экран задания и реакция на отклонение от регламента

Оценивание и анализ. Оценивает сервер. Модуль оценивания написан как чистая функция: на входе ответы и параметры, на выходе баллы, обращений к базе нет. Такой модуль полностью покрывается тестами, и это важнее, чем кажется. Спорный балл на вступительном испытании становится предметом разбирательства, а не переписки [5, С. 39].

Числовой ответ сверяется с эталоном по допуску, короткий текст – без учёта регистра и лишних пробелов. Формулу разбирает собственный рекурсивный спуск, сторонний код при этом не выполняется. Эквивалентность двух формул проверяется сравнением значений в восьми случайных точках с фиксированным зерном, поэтому результат воспроизводим. Эссе, код и загруженные файлы уходят эксперту.

Поверх результатов работает аналитика: доля верных ответов и дискриминация задания, дихотомическая модель Раша [12, 13], детектор возможного сговора по совпадению ошибок, кластеризация методом k-средних. Отдельно помечается «бракованный вопрос» – такой, который при трёх и более попытках даёт 70 % и более неверных или пустых ответов. Комиссия видит это в день испытания, а не по итогам кампании (рис. 4).

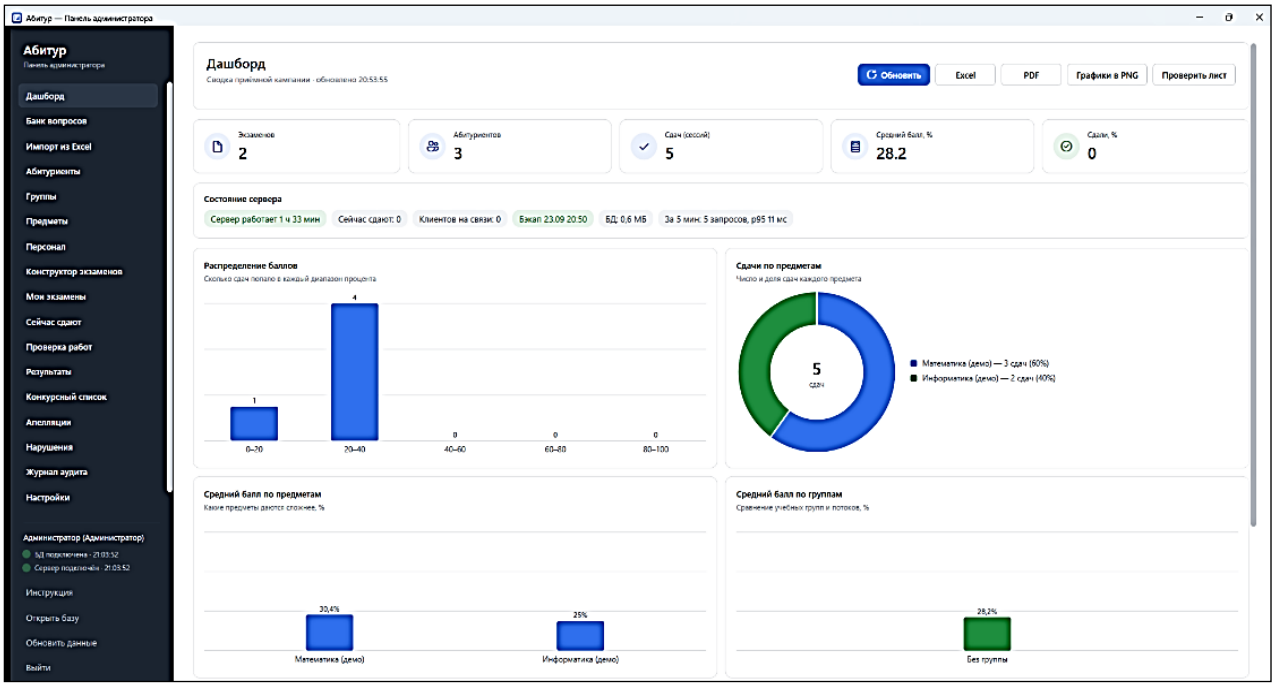


Рис. 4. Аналитический дашборд приёмной комиссии

Данные и защита. Рабочая база – один файл SQLite: 31 таблица и 13 представлений. Имена таблиц, столбцов и значения перечислений сделаны русскими. Поверх них собраны представления «Обзор: …», где вместо идентификаторов стоят фамилии и названия, а даты переведены в местное время (рис. 5). Для программиста решение спорное. Зато приёмная комиссия перестаёт зависеть от разработчика: секретарь открывает базу и читает её сам.

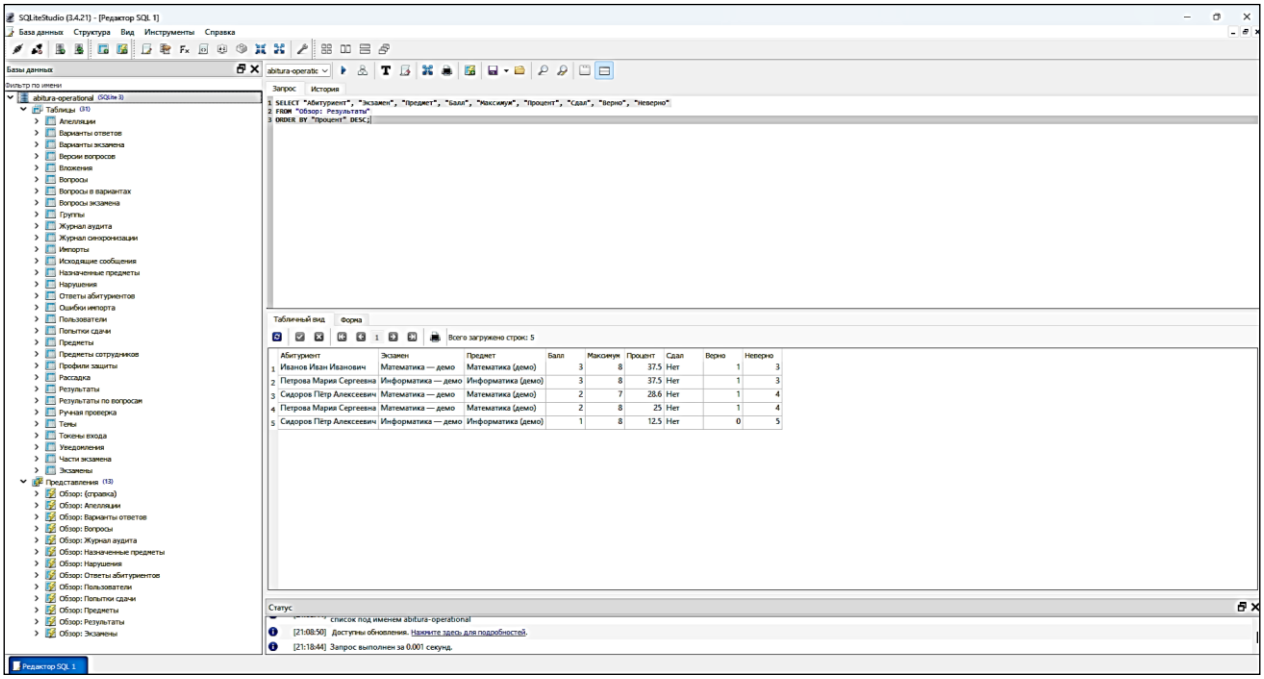


Рис. 5. Рабочая база данных: русские таблицы и представления «Обзор: …»

Персональные данные обрабатываются по требованиям 152-ФЗ [9]. СНИЛС и паспорт скрыты в списках и открываются отдельным действием, каждый такой просмотр попадает в журнал аудита. По требованию абитуриента карточка обезличивается, баллы при этом остаются в архиве приёма уже без имени [10].

Тестирование на проникновение дало четыре исправления. Самое существенное касалось разбора формулы. Ответ из четырёх тысяч вложенных скобок переполнял стек, а переполнение стека в .NET не перехватывается: падал весь сервер испытания и вместе с ним все сессии в классе. Введены пределы на длину формулы и глубину разбора. Закрыты также утечка эталонов в пакет, отсутствие ограничения на число загружаемых файлов и «zip-бомба» при импорте QTI.

Конкурсный список и правила приёма. Место в списке определяется не одним баллом за внутреннее испытание. В сумму входят ещё внешние баллы – ЦТ, ЕГЭ, олимпиады, испытания, сданные в другом вузе, – индивидуальные достижения и средний балл аттестата, умноженный на коэффициент. [11] Слагаемые система считает по отдельности и выводит прямо в строке абитуриента (рис. 6). Сделано это ради одного: когда человек спрашивает, откуда взялись его 184 балла, ответ есть сразу, без выгрузок и пересчётов. Предел за достижения и коэффициент аттестата вынесены в настройки сервера. По умолчанию коэффициент нулевой, предел – десять баллов. Так сделано намеренно: обновление программы не должно само по себе переставить людей в уже опубликованном списке. На рисунке коэффициент выставлен в десять.

Минимальный балл задаётся отдельно по каждому предмету. Абитуриент, не добравший его, из списка никуда не девается, а уходит в конец, и рядом написано, по какому предмету и сколько баллов не хватило. Сравнивают при этом лучшее из двух: балл внутреннего испытания и внешний балл по тому же предмету. Удачный ЦТ вытягивает неудачную сдачу в вузе. Бывает и обратное [7, С. 15].

Вводить внешние баллы можно двумя путями. Первый – вручную, в карточке абитуриента. Второй – таблицей, где на строку приходится ФИО или СНИЛС, предмет, вид испытания, балл, год и номер сертификата. Если тот же файл загрузить повторно, записи обновятся, а не удвоятся. Строки, по которым абитуриент не нашёлся, оператор получает обратно отдельным списком и разбирает руками.

Отчётность приёмной комиссии. Экран результатов собирает сдачи по выбранному испытанию. Видно балл, максимум, процент, сколько ответов оказалось верными, частично верными и неверными, когда работа проверена (рис. 7). Соседние вкладки отвечают за анализ: бракованные вопросы, калибровка, возможный сговор, разбор по темам.

Выгружается всё в Excel и PDF, отдельная кнопка собирает файл для 1С и ФИС ГИА, а протоколы для подписи печатаются пачкой, а не по одному. Удовлетворили апелляцию – балл правится здесь же, с указанием причины, и правка ложится в журнал аудита.

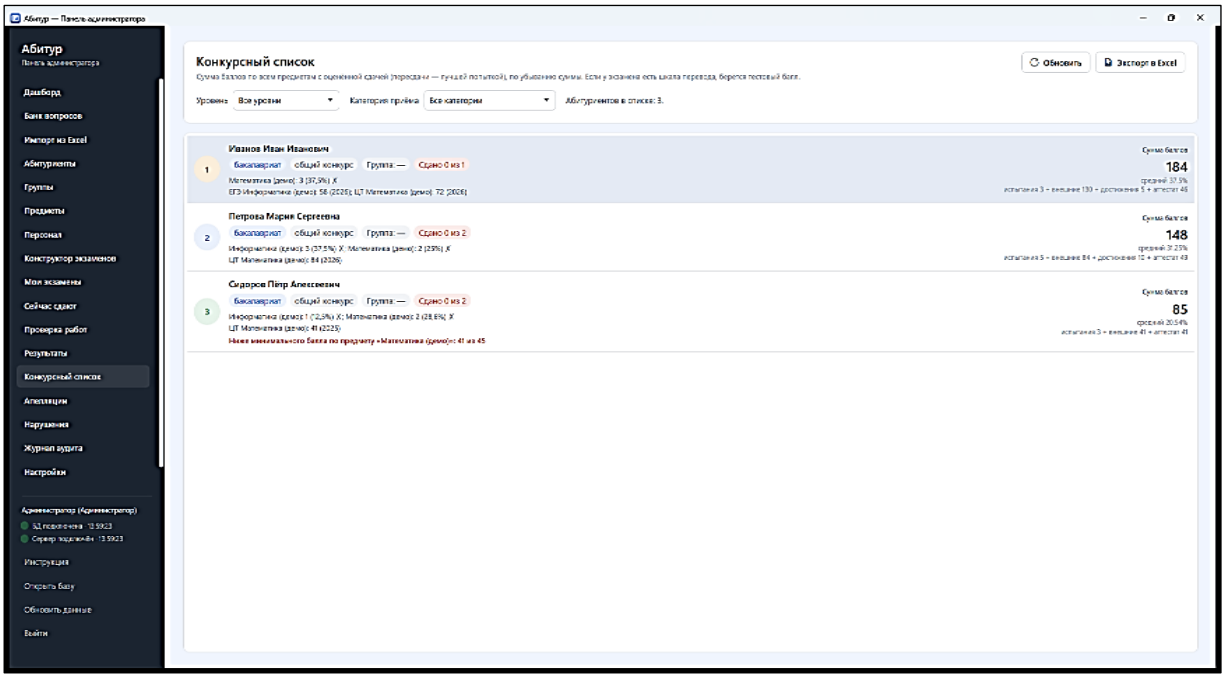


Рис. 6. Конкурсный список с разбором суммы баллов

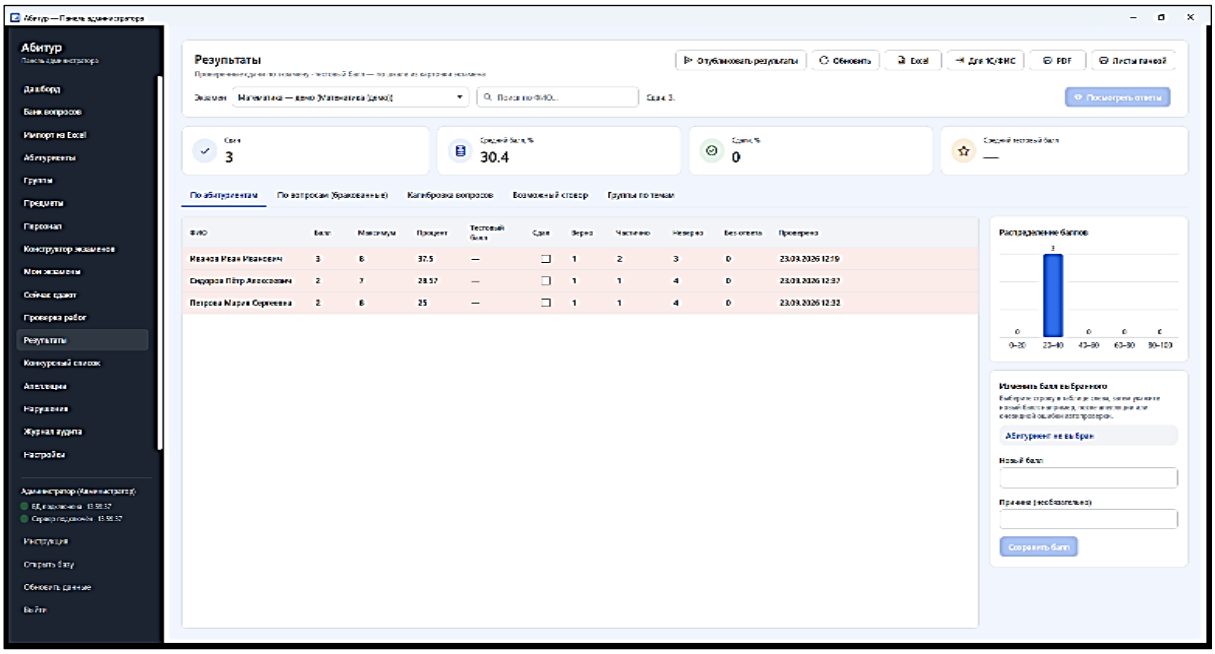


Рис. 7. Результаты испытания и вкладки анализа

Проверка. Прикладной слой покрыт 856 модульными и интеграционными тестами, покрытие 75 %. Сквозной сценарий из 50 проверок прогоняется на сборке из установщика, а не на отладочной.

Нагрузочное тестирование шло на изолированном сервере: банк 2000 вопросов, 20 заданий в варианте, автосохранение каждые 15 секунд, все сессии стартуют одновременно. При 600 сессиях отказов нет, при 1000 тоже; сжатый пакет занимает 1,5 и 2,3 МБ. До исправления шесть сессий из шестисот завершались ошибкой блокировки базы. Причину устранили переводом соединений в режим PRAGMA synchronous=NORMAL и ожиданием освобождения блокировки до 30 секунд. Отклик интерфейса на штатном оборудовании класса пока не замерялся, это ближайшая задача.

Поверх модульных тестов работает ещё один слой, интеграционный. Тест поднимает настоящий конвейер сервера прямо в своём процессе и стучится к нему по HTTP. Проверяется то, чего по отдельным функциям не увидишь: порядок обработчиков, ключ администратора, ограничение частоты запросов, весь сценарий испытания от входа до результата [8, С. 50]. Код сервера, кроме того, читали атакующим взглядом. Пригодных к использованию дыр не нашли. Токены сессий длиной 128 бит выдаёт криптографический генератор, в базе хранится

только их хэш, ключ администратора сверяется за постоянное время, а баллы, присланные клиентом, сервер просто игнорирует и считает сам.

Эксплуатация и восстановление. Журнал аудита связан цепочкой хэшей: каждая запись несёт хэш предыдущей. Вырезать строку из середины или подменить её незаметно уже не получится. Контрольную точку цепочки сервер держит отдельно от базы и при расхождении поднимает тревогу. После восстановления из резервной копии её снимают отдельной командой, сама она не погаснет.

На случай отказа оборудования предусмотрен тёплый резерв. Второй сервер забирает снимки базы и копию сертификата, а переключение делает оператор, вручную: решение, кого считать главным, автоматике не доверили. Отпечаток сертификата после переключения тот же, поэтому в классе ничего не перенастраивают. Сам сертификат меняется заранее – новый выпускается за два месяца до истечения срока, его отпечаток публикуется, и панель с клиентами подхватывают замену сами.

Новизна и практическая значимость. Ключи правильных ответов не передаются на рабочее место в принципе, а у заданий со свободным ответом отсутствует и сам перечень вариантов. Распространённая схема, когда содержимое теста синхронизируется целиком и защищается правами доступа к локальной базе, здесь не применяется. Вариант абитуриента воспроизводится по зерну, поэтому апелляция перестаёт быть спором о том, что было на экране. Анализ качества заданий доступен сразу после испытания, а не отдельным исследованием по следам кампании [3]. Добавим к этому рабочую базу, которую читает человек без участия разработчика.

Практический эффект тоже понятен. Итоги появляются в день испытания. Ручной перенос баллов и поиск бланков исключены. Класс работает без интернета, а проверенный запас по нагрузке перекрывает дневной поток приёмной комиссии. [15]

Заключение. Представлена система «Абитур», клиент-серверный комплекс для проведения вступительных испытаний в компьютерном классе вуза. Закрыт полный цикл: от наполнения банка заданий до анализа результатов.

Версия проверена тестами, нагрузкой и сквозными сценариями на сборке из установщика. Ближайший шаг – развернуть её на стенде класса и попробовать в приёмную кампанию. К конкретному вузу архитектура не привязана: перенос сводится к настройке предметов и правил приёма.

При использовании автоматизированной системы «Абитур» произойдёт сокращение времени выполнения стандартных операций, которое обеспечит возможность обработки большего объёма заявлений без пропорционального увеличения численности персонала приёмной комиссии. В результате произойдёт повышение пропускной способности приёмной комиссии, сократятся очереди и время ожидания для абитуриентов, а пиковые периоды приёмной кампании пройдут более равномерно и организованно. Производительность труда возрастёт не только за счёт ускорения отдельных операций, но и вследствие снижения количества ошибок, исправлений, повторных проверок и согласований.

Литература

1. Дорохов Д.С. Вступительные испытания в высшие учебные заведения // Экономика и социум. 2016. № 5–1 (24). С. 600–604.
2. Кошкарова Л.Ф. Инструментальные средства поддержки принятия решений при проведении приемной кампании // Вестник КИГИТ. 2013. №7 (37). С. 31–35.
3. Ляликова В.И., Цехан О.Б. Оценка баллов вступительных испытаний абитуриентов, обеспечивающих их успешное обучение на первом курсе // Вестник Гродненского государственного университета имени Янки Купалы. Серия 5. Экономика. Социология. Биология. 2014. № 2 (177). С. 12–18.
4. Никитина А.И., Барахсанова Е.А. Применение информационной системы в работе приемной комиссии ФГБОУ ЧГИФКИС / Современное образование: традиции и инновации. 2022. № 2. С. 77–80.
5. Сундукова Г.М. Умный ВУЗ – экономный ВУЗ: цифровые сервисы как инструмент финансовой устойчивости // Восточно-европейский научный вестник. 2026. Т. 22. № 2. С. 38–40.
6. Башмакова В.И. Суперсервис «Поступление в ВУЗ онлайн» как новая форма административной процедуры / Юридическая наука. 2026. № 5. С. 118–120.
7. Белышева А.Н. Возникновение у абитуриентов различных функциональных психических состояний при прохождении вступительных испытаний в ВУЗ // Вестник Вятского государственного гуманитарного университета. 2010. № 3, С. 10–15.
8. Тимирясов В.Г. Инновационный ВУЗ: приоритеты развития // Высшее образование в России. 2007. № 9. С. 46–52.
9. Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных» (с изм. и доп.). – Режим доступа: http://pravo.gov.ru (дата обращения: 21.09.2026).
10. Приказ Минобрнауки России от 21.08.2020 № 1076 «Об утверждении Порядка приёма на обучение по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры» (с изм.).
11. Нейман Ю.М., Хлебников В.А. Введение в теорию моделирования и параметризации педагогических тестов. М.: Прометей, 2000. 168 с.
12. Rasch G. Probabilistic Models for Some Intelligence and Attainment Tests. Copenhagen: Danish Institute for Educational Research, 1960. 184 p.
13. IMS Question and Test Interoperability (QTI) Specification, version 2.2 / 1EdTech Consortium. Режим доступа: https://www.imsglobal.org/question (дата обращения: 21.09.2026).
14. Documentation for Entity Framework Core and Microsoft. Data.Sqlite / Microsoft Learn. – Режим доступа: https://learn.microsoft.com/ef/core (дата обращения: 21.09.2026).
15. Гончарова Н.А., Левченко Т.М., Афанасьев А.С. Формирование аутсорсинга в инновационном менеджменте / Монография. 2013.

References

1. Dorokhov D.S. Entrance examinations for higher education institutions / "Economics and Society". 2016. No. 5–1 (24). Pp. 600–604.
2. Koshkarova L.F. Decision support tools for the admissions campaign / "Bulletin of KIGIT". 2013. No. 7 (37). Pp. 31–35.

3. Lyalikova V.I., Tsekhan O.B. Assessment of applicants' entrance exam scores ensuring successful first-year studies / "Bulletin of Yanka Kupala State University of Grodno. Series 5: Economics. Sociology. Biology". 2014. No. 2 (177). Pp. 12–18.
4. Nikitina A.I., Barakhsanova E.A. Application of an information system in the work of the admissions committee of ChGIFKiS / "Modern Education: Traditions and Innovations". 2022. No. 2. Pp. 77–80.
5. Sundukova G.M. Smart University – Cost-Efficient University: digital services as a tool for financial sustainability / "Eastern European Scientific Bulletin". 2026. Vol. 22. No. 2. Pp. 38–40.
6. Bashmakova V.I. The "Online University Admission" superservice as a new form of administrative procedure / "Legal Science". 2026. No. 5. Pp. 118–120.
7. Belysheva A.N. Emergence of various functional mental states in university applicants during entrance examinations / Bulletin of Vyatka State University of Humanities. 2010.
8. Timiryasov V.G. Innovative University: Development Priorities / Higher Education in Russia. 2007. No. 9. P. 46–52.
9. Federal Law No. 152-FZ of July 27, 2006, "On Personal Data" (as amended). – Access mode: http://pravo.gov.ru (date of access: September 21, 2026).
10. Order of the Ministry of Science and Higher Education of the Russian Federation No. 1076 of August 21, 2020, "On Approval of the Procedure for Admission to Study under Higher Education Programs – Bachelor's, Specialist, and Master's Programs" (as amended).
11. Neiman Yu.M., Khlebnikov V.A. Introduction to the Theory of Modeling and Parameterization of Pedagogical Tests. Moscow: Prometey, 2000. 168 p.
12. Rasch G. Probabilistic Models for Some Intelligence and Attainment Tests. Copenhagen: Danish Institute for Educational Research, 1960. 184 p.
13. IMS Question and Test Interoperability (QTI) Specification, version 2.2 / 1EdTech Consortium. Access mode: https://www.imsglobal.org/question (date of access: September 21, 2026).
14. Documentation for Entity Framework Core and Microsoft. Data.Sqlite / Microsoft Learn. Access mode: https://learn.microsoft.com/ef/core (accessed: September 21, 2026).
15. Goncharova N.A., Levchenko T.M., Afanasyev A.S. The Development of Outsourcing in Innovation Management / Monograph. 2013.