

С. Н. Шергин, Н. В. Денисов, В. В. Лука

ВЕБ-ИНСТРУМЕНТ УПРАВЛЕНИЯ БАНКОМ ТЕСТОВЫХ ЗАДАНИЙ С СИНХРОНИЗАЦИЕЙ С СИСТЕМОЙ MOODLE

В статье рассматривается веб-инструмент, предназначенный для создания, управления и структурирования банков (наборов) тестовых заданий.

В ходе работы была предложена структурированная база данных, а также была проведена оптимизация работы веб-приложения.

Основной целью является взять за основу текущий модуль управления и создания тестовых заданий системы moodle и создать оптимальный, человекоориентированный интерфейс, с оптимизацией процессов в вузе, связанных с разработкой тестов.

Ключевые слова: фронтенд, бэкенд, разработка тестов, Moodle, GIFT

S. N. Shergin, N. V. Denisov, V. V. Luka

WEB-TOOL FOR MANAGING A SET OF TESTS WITH SYNCHRONIZATION WITH MOODLE

The article discusses a web application for creating, managing and structuring the bank (set) of tests. In the course of the work, a structured database was proposed, and the web application was optimized.

The main goal is to take as a basis the current module for managing and creating a bank of test tasks of the moodle system and optimize it.

Keywords: frontend, backend, Moodle, test development, GIFT

Введение

В большинстве учебных заведениях эксплуатируют одну из популярных платформ обучения – система управления обучением Moodle [1]. Также, в связи с коронавирусной инфекцией, в 2020 году большинство вузов разных стран приняли решение перейти на дистанционное обучение, и система стала неотъемлемой частью взаимодействия обучающихся с преподавателями и освоением образовательных программ.

Данная система разработана 2002 году и до сих пор поддерживается. С развитием технологий перегруженность подобных систем увеличивается, оптимизация снижается в связи с отказом разработчиков обновления технологий.

Проведя исследования внутри вуза (рис. 1), выяснилось, что модуль управления банка тестовых заданий является не оптимизированным и нагруженным в плане элементов взаимодействия пользователя с системой. Модуль требует современных решений проектирования и добавление различного функционала. Проведя тестирование, оказалось, что проблемой является блок работы с тестовыми заданиями для преподавателей.

Для устранения данной проблемы было решено разработать веб-приложение с структурированной архитектурой и использованием современных технологий проектирования. Применяя архитектуру, которая упрощает работоспособность системы, что позволяет оптимизировать работу приложения в дальнейшем.

Как вы оцениваете удобство создания тестов для Eluniver? (По 5-ти бальной шкале)

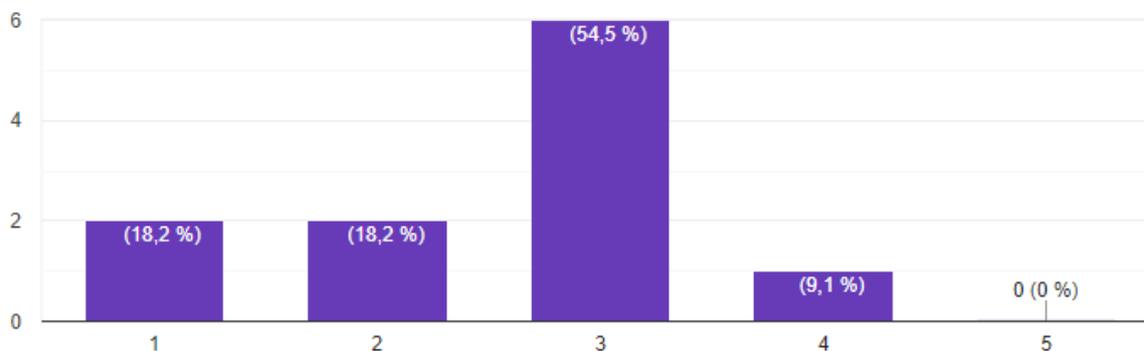


Рисунок 1. Удобство интерфейса по созданию вопросов

Требования к системе

При выборе реализации веб-инструмента работы с банками тестовых заданий были сформированы следующие требования:

- Автоматизировать создание паспорта банка тестовых заданий (в том числе отчетности).
- Обеспечить формирование вопросов банка тестовых заданий для загрузки на систему в moodle.
- Обеспечить быстродействие работы системы.
- Оптимизировать интерфейс взаимодействия с пользователем.

Для автоматизации формирования паспорта необходимо использовать современные технологии формирования документов.

Реализация и функциональные особенности

Платформа moodle имеет очень широкий функционал. Было решено не оптимизировать конкретный модуль с работой над тестовыми заданиями внутри системы. Так, как его ограничения бы остались, а другие технологии имеют большую гибкость и удобство в использовании.

Для решения данной задачи было принято решение вынести модуль работы с тестовыми заданиями из платформы moodle в отдельное web- приложение. Особенности данного решения является то, что, абстрагируясь от текущей системы, мы получим возможность упростить взаимодействие пользователя с элементами взаимодействия системы, так же оптимизировать функциональную часть, автоматизировать формирование наборов вопросов и отчетностей для преподавателей и центра тестирования.

Клиент-серверная архитектура представляет собой иерархическую сеть (рис. 2), которая состоит из узлов-клиентов и центрального сервера[4], через который выполняется хранение и обработка данных, а также передача их в обоих направлениях.

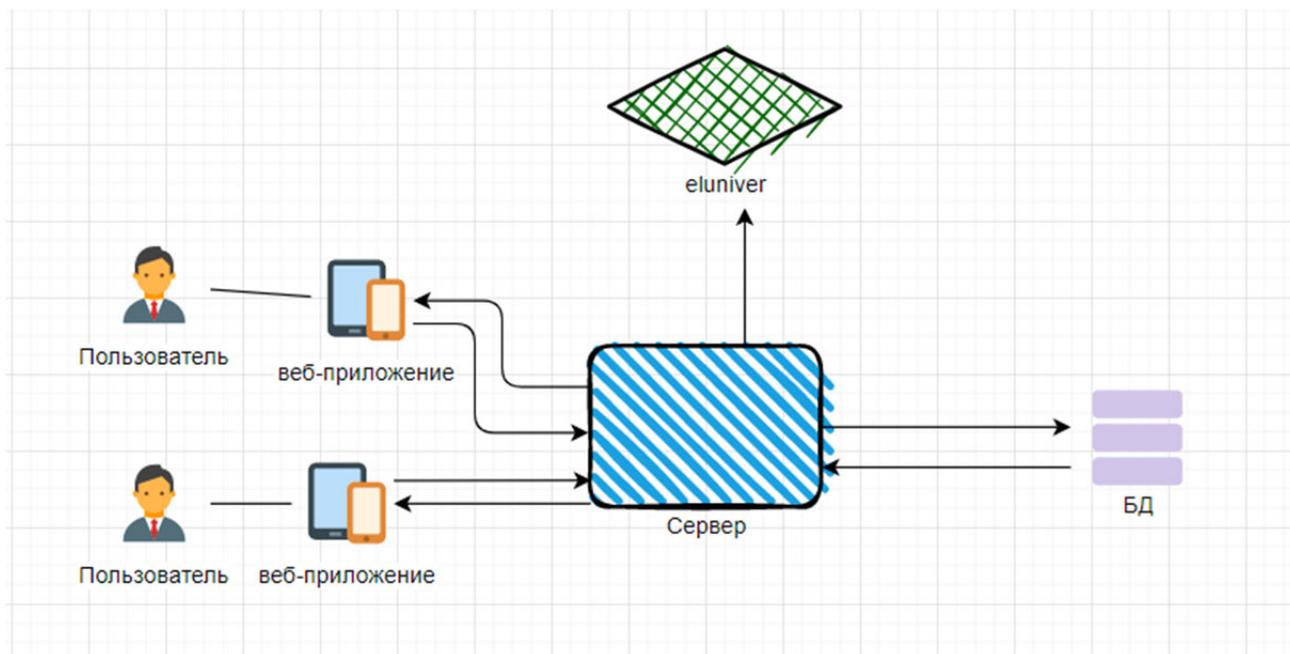


Рисунок 2. Структура ER-модели

На рисунке 3 продемонстрирована структура базы данных веб-сервера:

- Пользователь
- Банк тестовых заданий
- Секции
- Категории
- Вопросы
- Типы вопросов

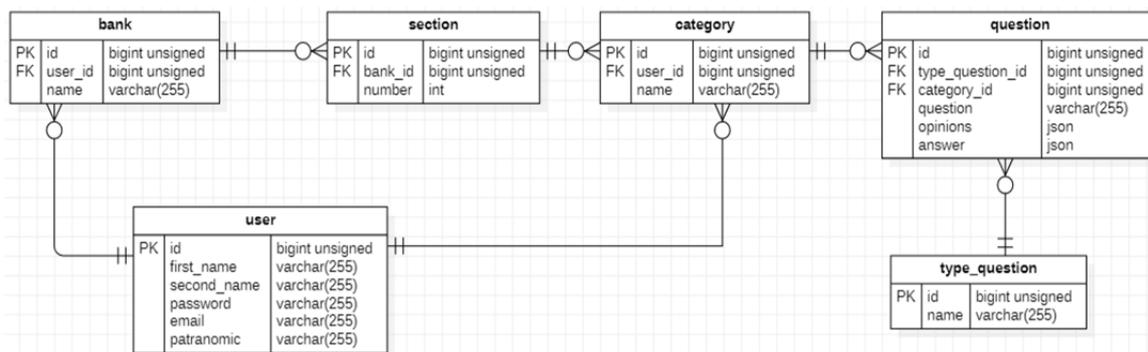


Рисунок 3. ER-диаграмма веб-сервиса

Пользователь имеет возможность регистрироваться в системе и использовать следующий функционал: создавать и изменять набор тестовых заданий. Каждый набор состоит из секций, которая состоит из категорий. Каждая категория содержит вопросы разных типов.

Система позволяет пользователям продолжать вести разработку тестов если до этого они работали в moodle. Пользователю необходимо предоставить возможность сохранить, редактировать набор имеющихся тестов. Взаимодействия с платформой mododle осуществляется путем загрузки банка тестовых заданий в формате Moodle XML. Это специфический формат экспорта вопросов Moodle в простой XML формат. Позволяет импортировать вопросы в другую категорию и использовать вопросы для других процессов, например XSLT трансформации. XML формат экспортирует также изображения, прикрепленные к вопросам; они кодируются в base64 [3]. Данный формат загружается как обыкновенный XML, декодируется в требуемый формат, который в дальнейшем используется для заполнения базы данных.

Выгрузка осуществляется в формате GIFT[2]. GIFT – формат данных для экспорта и импорта вопрос для платформы moodle. Она имеет структурированную схему, которая позволяет иметь разделы категории и вопросы. Каждый вопрос имеет свой определенный формат написания вопросов. В платформе moodle можно использовать для взаимодействия с пользователями формат данных GIFT. Для взаимодействия с форматом GIFT используется алгоритм, представленный на рисунке 4.

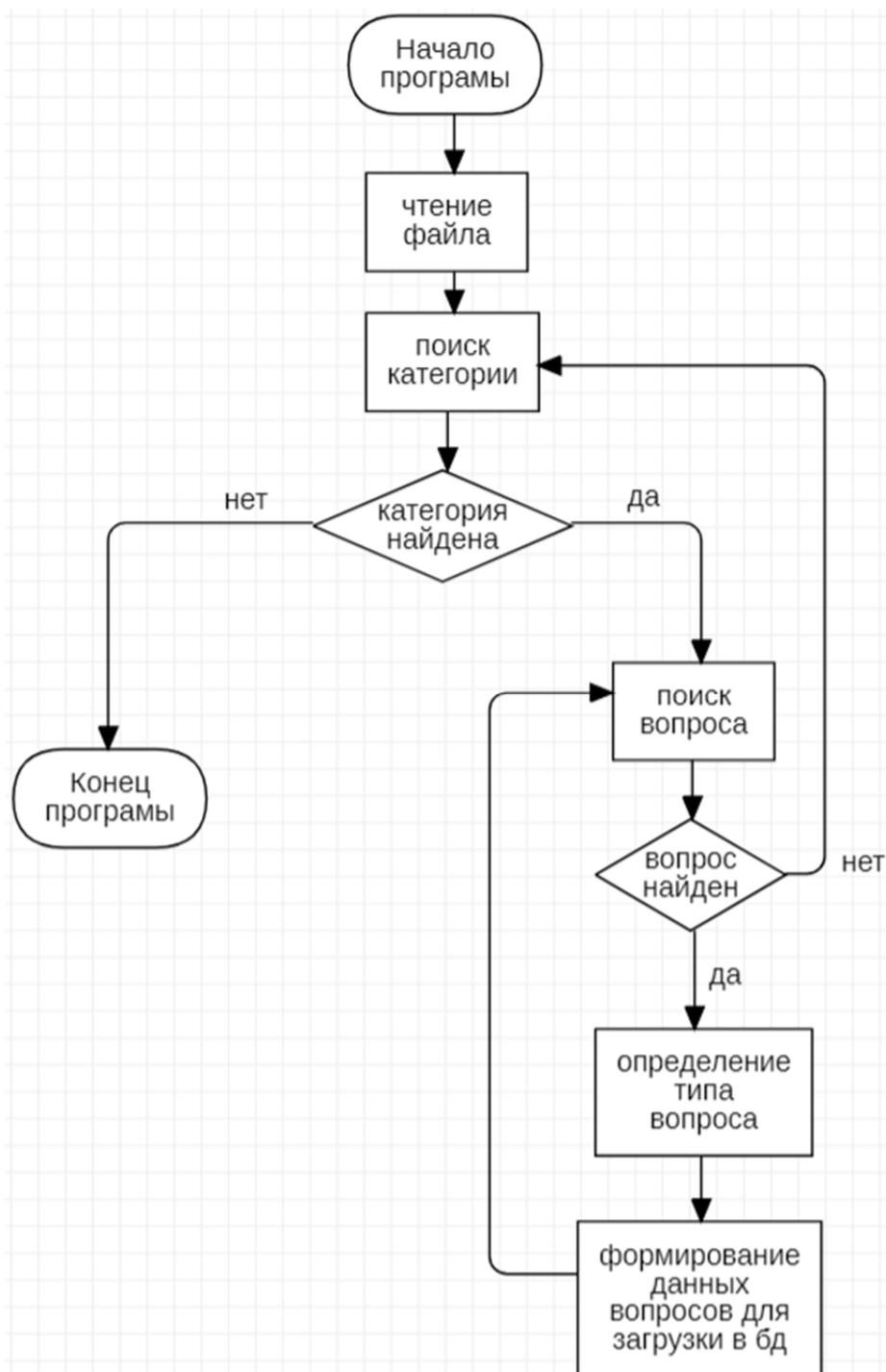


Рисунок 4. Блок схема алгоритма чтение файла GIFT

Также имеется выгрузка паспорта банка тестовых заданий. Паспорт подразумевает под собой ряд отчетностей о составе, наполненности набора тестов для проведения контрольных мероприятий для студентов. Для создания отчета требуется указать ряд параметров, характе-

ризующих контрольное мероприятие (форма проведения, время, количество участвующих вопросов, дата проведения и другие). Формирование файла происходит в формате docx.

Заключение

В результате проделанной работы представлен вариант реализации модуля работы с банком тестовых заданий платформы moodle. Данный вариант был успешно апробирован сотрудниками Югорского государственного университета (г. Ханты-Мансийск). В дальнейшем планируется внедрить совместную разработку тестов (с сохранением авторского права), реализовать более полную интеграцию с платформой moodle (без участия преподавателя), провести оценку сервиса для сравнения результатов с предыдущим исследованием.

Литература

1. Документация по Moodle 2.5 / перевод с английского В. А. Тунда ; под редакцией Ф. П. Тарасенко. – Текст : электронный // Электронная библиотека (репозиторий) Томского государственного университета. – 2014. – URL: <http://vital.lib.tsu.ru/vital/access/manager/Repository/vtls:000480285> (дата обращения: 12.04.2021).
2. Викулин, М. А. Разработка шаблонизатора для поточной загрузки вопросов в системы электронного обучения / М. А. Викулин, И. С. Старчаус, Л. Л. Хорошко. – Текст : непосредственный // Южно-Сибирский научный вестник. – 2019. – № 1 (25). – С. 166–169.
3. MOODLE : [сайт]. – 2021. – URL: <https://moodle.org> дата обращения: 12.04.2021). – Текст : электронный.
4. Смирнова, Г. Н. Проектирование экономических информационных систем : учебник / Г. Н. Смирнова, А. А. Сорокин, Ю. Ф. Тельнов ; под ред. Ю.Ф. Тельнова. – Москва : Финансы и статистика, 2002. – 512 с. – Текст : непосредственный.