

## **85. СИСТЕМА УПРАВЛЕНИЯ КОРПОРАТИВНЫМИ ОБУЧАЮЩИМИ ПРОГРАММАМИ С ЭЛЕМЕНТАМИ ГЕЙМИФИКАЦИИ ДЛЯ ПОВЫШЕНИЯ МОТИВАЦИИ И УРОВНЯ КОМПЕТЕНТНОСТИ СОТРУДНИКОВ**

*Крук Е.С., студент гр. 172302, Сторожев Д.А., ст. преподаватель кафедры ЭИ*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Комличенко В.Н. – канд. техн. наук, доцент каф. ЭИ*

**Аннотация.** В данной статье рассматривается разработка системы для управления корпоративным обучением и сертификацией сотрудников, направленной на оптимизацию процессов обучения и повышения квалификации персонала. Описаны ключевые этапы работы системы, включая создание и управление обучающими курсами, мониторинг прогресса сотрудников, организацию тестирования для оценки усвоения материала и формирование персонализированных сертификатов. Также в рамках системы реализованы функции сбора обратной связи от пользователей и администрирования, включая управление учетными записями сотрудников и преподавателей.

**Ключевые слова.** Корпоративное обучение, сертификация сотрудников, система управления, база данных, язык программирования Java, Spring Framework, архитектурные решения, PostgreSQL.

Обучение и сертификация сотрудников охватывает системы и процессы, которые направлены на развитие навыков, знаний и компетенций специалистов, а также на их объективную оценку и подтверждение. Основная цель — поддержание высокого уровня квалификации персонала, повышение производительности и качества выполнения работы, а также улучшение адаптации к изменениям в профессиональной среде.

Корпоративное обучение может принимать различные формы в зависимости от целей компании, особенностей ее отрасли и потребностей сотрудников. Современные подходы к обучению позволяют организациям разрабатывать гибкие программы, которые включают как традиционные методы, так и инновационные технологии.

Актуальность и значимость цифровизации образовательного процесса вызвана необходимостью адаптации системы образования и обучения к запросам цифровой экономики и цифрового общества, становление которых – глобальные тренды современной эпохи.

Таким образом, целью данного проекта является автоматизация и улучшение процессов управления корпоративными обучающими программами и аттестацией сотрудников. Данный процесс предполагает создание решения, способного устранить существующие проблемы, такие как низкая эффективность традиционных методов обучения, отсутствие прозрачности в процессе аттестации и сложности в анализе данных о результатах обучения.

Объектом исследования выступает сфера корпоративного обучения и сертификации сотрудников как важнейший элемент системы управления человеческими ресурсами.

Традиционные методы организации обучения и контроля знаний, такие как бумажная документация или использование разрозненных программных инструментов, зачастую являются недостаточно эффективными, трудоемкими и не отвечают современным требованиям бизнеса [1]. Создание унифицированного сервиса для управления корпоративными обучающими программами и аттестацией сотрудников позволяет централизовать данные, повысить прозрачность процессов и упростить анализ их эффективности.

Рассмотрим последовательность действий, поочередность которых позволит понять весь принцип работы приложения по корпоративному обучению. На начальном этапе преподаватель платформы должен опубликовать некоторый курс, оснащенный уроками в текстовом или видео формате, наполнив его всей необходимой информацией: название, сфера изучения, уровень навыков и описание курса.

Сотрудники, в свою очередь, должны иметь возможность просмотра опубликованных курсов после предварительного запроса на осуществление упомянутой операции. Как только создатель блока обучения одобрит полученную заявку – на экране сотрудника отобразится полный материал.

Помимо просмотра учебного материала, необходимо организовать формирование вопросов/ответов к курсу, с целью максимально подробного усвоения темы. Разбор возникающих вопросов как со своими коллегами, так и со специалистами заданной сферы позволит сотрудникам избежать пробелов в знаниях и расширить диапазон приобретаемых навыков.

Здесь, в личном кабинете, сотрудник должен иметь доступ к отслеживанию прогресса прохождения уроков, по итогам которого будет предоставлен проверочный тест.

Далее должно быть организовано тестирование на проверку усвоения предложенного материала. Как только все ответы будут предоставлены, на экране должен отобразиться результат, который послужит основанием для получения сертификата или повторного прохождения теста.

Для сбора мнений по улучшению курса, необходимо реализовать страницу отзывов, где пользователи смогут поделиться своим опытом изучения предоставленного учебного материала, выставляя при этом общую оценку по пятибалльной шкале. На основании выставленных значений сформируется общий рейтинг курса с детализированным отображением процентного соотношения баллов.

Для отслеживания деятельности на платформе обучения, администратор должен иметь доступ к учетным записям пользователей приложения. Для удобства работы с ними нужно реализовать функции поиска, сортировки, редактирования данных, а также блокировки доступа к сервису.

Помимо управления аккаунтами сотрудников и преподавателей, зарегистрированных на платформе, именно администратор должен контролировать процесс авторизации новых пользователей посредством выдачи реферальных ключей. Генерация уникальных кодов должна осуществляться в соответствии с предоставленным при входе почтовым адресом, на который будет отправлено письмо с получившейся унифицированной строкой символов.

На рисунке 1 представлена страница с опубликованными материалами курса.



Рисунок 1 – Страница материалов курса

Разрабатываемое веб-приложение реализации управления корпоративным обучением и сертификацией сотрудников представляет собой клиент-серверное приложение. Серверная часть создана с применением фреймворка Spring и его модулей на языке программирования Java. Клиентская часть – с использованием JavaScript и TypeScript, а также других библиотек компонентов.

Spring — это фреймворк для разработки приложений на языке программирования Java. Его преимущества включают гибкость, модульность и удобство в использовании. Остановимся на каждом из них:

- Инверсия управления (IoC). Вместо того чтобы явно создавать и связывать объекты в коде, IoC делегирует эту ответственность контейнеру Spring, что упрощает управление зависимостями и делает код более гибким.

- Внедрение зависимостей (DI). Spring предоставляет механизм внедрения зависимостей, упрощая интеграцию компонентов приложения, что особенно полезно для тестирования и снижения связанности между классами.

- Аспектно-ориентированное программирование (AOP). Позволяет выносить аспекты, такие как логирование, транзакции и безопасность, из основного кода приложения, повышая модульность и читаемость кода.

- Обработка транзакций. Поддержка объединения, распределенных транзакций и настройка с использованием аннотаций или XML-конфигурации.

- Модульность. Фреймворк Spring состоит из различных модулей (Spring Core Container, Spring AOP, Spring Data, Spring MVC и др.), которые можно использовать по мере необходимости, что позволяет выбирать только те части Spring, которые необходимы при создании приложения.

- Управление жизненным циклом объектов. Spring обеспечивает удобное управление жизненным циклом объектов [2]. Есть возможность определения методов их инициализации и уничтожения, что полезно при подготовке и освобождении ресурсов.

- Простота тестирования. Благодаря использованию внедрения зависимостей и другим принципам, код, написанный с использованием Spring, более легко тестируется, что способствует созданию надежных приложений.

Используемые технологии для создания клиентской части разрабатываемой платформы корпоративного обучения, а именно JavaScript, TypeScript, React.js, Next.js, и Ant Design, в полной мере обеспечивают интерактивность, удобство разработки и масштабируемость.

Для хранения и удобного извлечения в ходе работы с приложением данных использовалась база данных PostgreSQL, которая представляет собой объектно-реляционную СУБД с открытым исходным кодом. Данная СУБД может обрабатывать большие объемы данных и обеспечивать высокую производительность благодаря поддержке параллелизма, оптимизации запросов и использованию индексов.

На рисунке 2 изображена диаграмма вариантов использования функционала сотрудника платформы. Ключевыми из них были определены:

- просмотр уроков различного формата;
- отслеживание прогресса обучения;
- управление личными запросами;
- управление баллами;
- просмотр и поиск курсов;



Для контроля прогресса предусмотрена таблица `user_progress`, которая фиксирует, какие курсы и на каком этапе проходит каждый пользователь. Также система поддерживает функциональность подачи заявок на получение статуса преподавателя через сущность `become_teacher`, что делает модель пригодной как для сотрудников, так и для экспертов, желающих делиться знаниями.

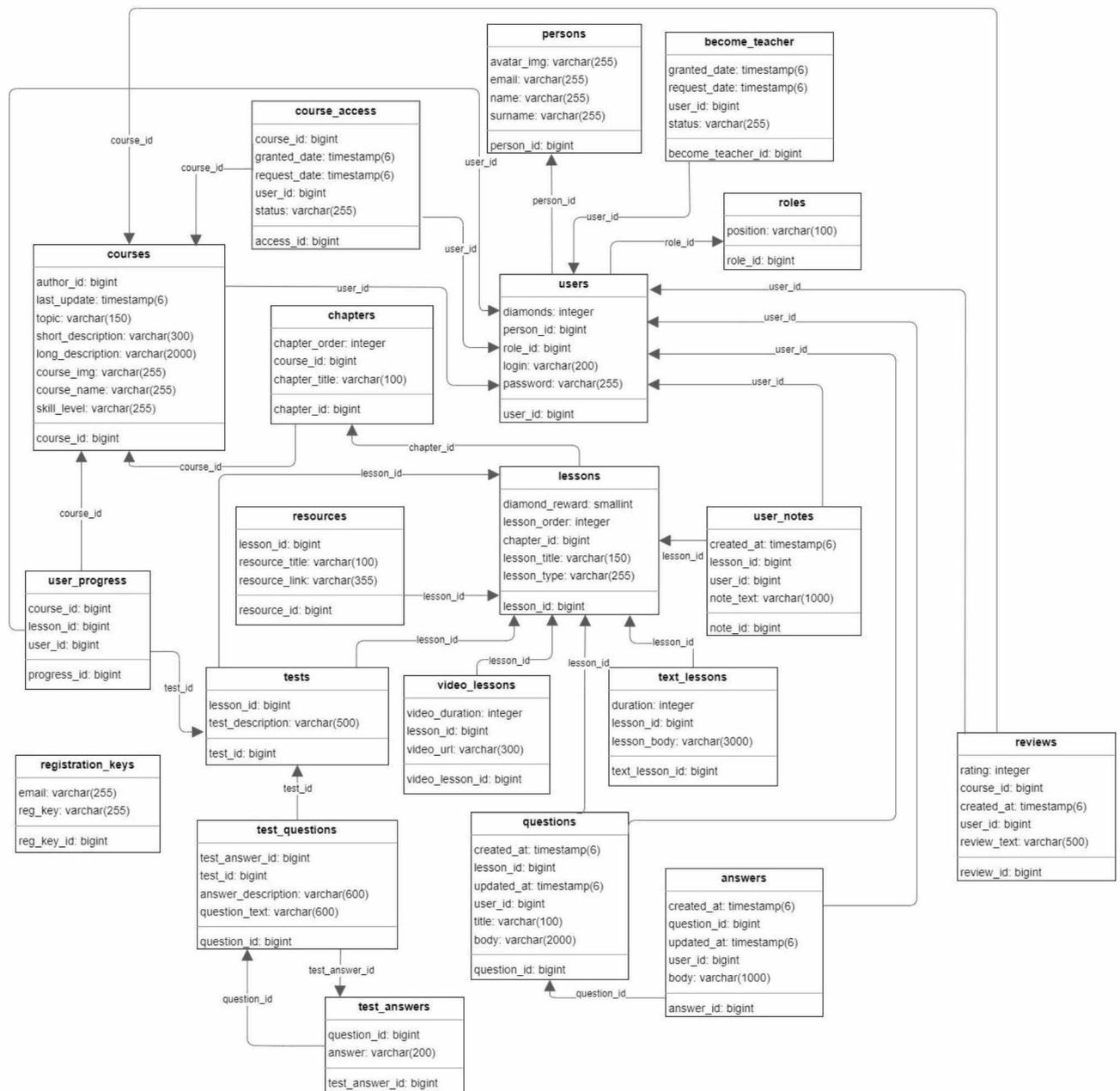


Рисунок 3 – Дatalogическая модель базы данных

Разработка дatalogической модели представляет собой важнейший этап в построении реляционной базы данных, поскольку именно от её логичности и структурной целостности зависит стабильность и эффективность всей информационной системы. Приведение структуры к нормальной форме позволяет избежать дублирования информации, снизить вероятность ошибок при обновлении данных и обеспечить простоту расширения функциональности в будущем.

Грамотно выстроенные связи между сущностями отражают ключевые бизнес-процессы системы, а тщательно продуманные справочники и механизмы контроля доступа делают модель не только технически корректной, но и практически удобной в использовании. Введённые вспомогательные таблицы, несмотря на увеличение количества сущностей, позволяют оптимизировать хранение и ускорить выполнение запросов.

В совокупности всё это формирует фундамент для надёжной и масштабируемой платформы, способной эффективно функционировать в условиях реальной нагрузки. А визуальное отображение компонентов системы и их взаимодействий на диаграмме развертывания даёт целостное представление о распределении программных модулей и аппаратных ресурсов, обеспечивая прозрачность архитектурных решений и упрощая последующую техническую реализацию.

Для наглядной демонстрации функционирования программной системы на уровне аппаратного и программного обеспечения, используется диаграмма развертывания и компонентов, представленная на рисунке 4.

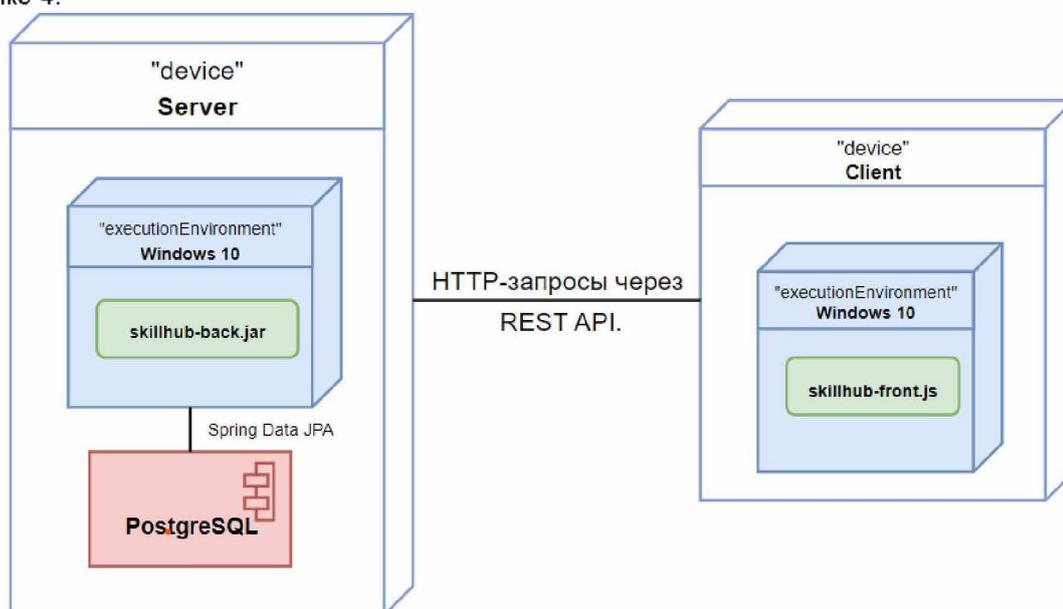


Рисунок 4 – Диаграмма развертывания и компонентов

Узлы диаграммы представлены следующим образом: Server и Client. Сервер, также как и клиент, включает в себя «executionEnvironment» с использованием Windows 10. В Server используется компонент skillhub-back.jar, а также база данных PostgreSQL, которая позволяет удобно хранить данные в безопасной форме, посредством использования модуля Spring Data JPA. Client оснащён компонентом skillhub-front.js, который отвечает за отображение веб-приложения на экране пользователя. Взаимодействие сервера с клиентом осуществляется с помощью протокола HTTP, посредством запросов через REST API.

Таким образом, данный вид диаграмм наглядно демонстрирует модульное разделение системы на клиентскую и серверную части, обеспечивая их независимость и удобство масштабирования.

С целью представления взаимодействия между элементами модели программной системы и сообщений между ними, применяется диаграмма последовательности. Применяется данный вид диаграмм для рассмотрения поведения группы объектов в рамках одного прецедента, последовательности действий объекта, зависимости между объектами и время жизни [3].

На рисунке 5 отображена диаграмма последовательности, отражающая взаимодействие объектов системы в рамках варианта использования «регистрация пользователя».

Можно заметить, что диаграмма включает в себя актера «Пользователь» и компоненты системы, взаимодействие с которыми позволяет осуществить рассматриваемую операцию. После того, как пользователь отправил запрос на регистрацию, осуществляется проверка на его наличие в соответствующем репозитории, после чего необходимо сверить его реферальный ключ с тем, что хранится в базе данных. В случае положительной проверки генерируется JWT-токен через соответствующий сервис. При успешной регистрации реферальный ключ из базы данных удаляется.

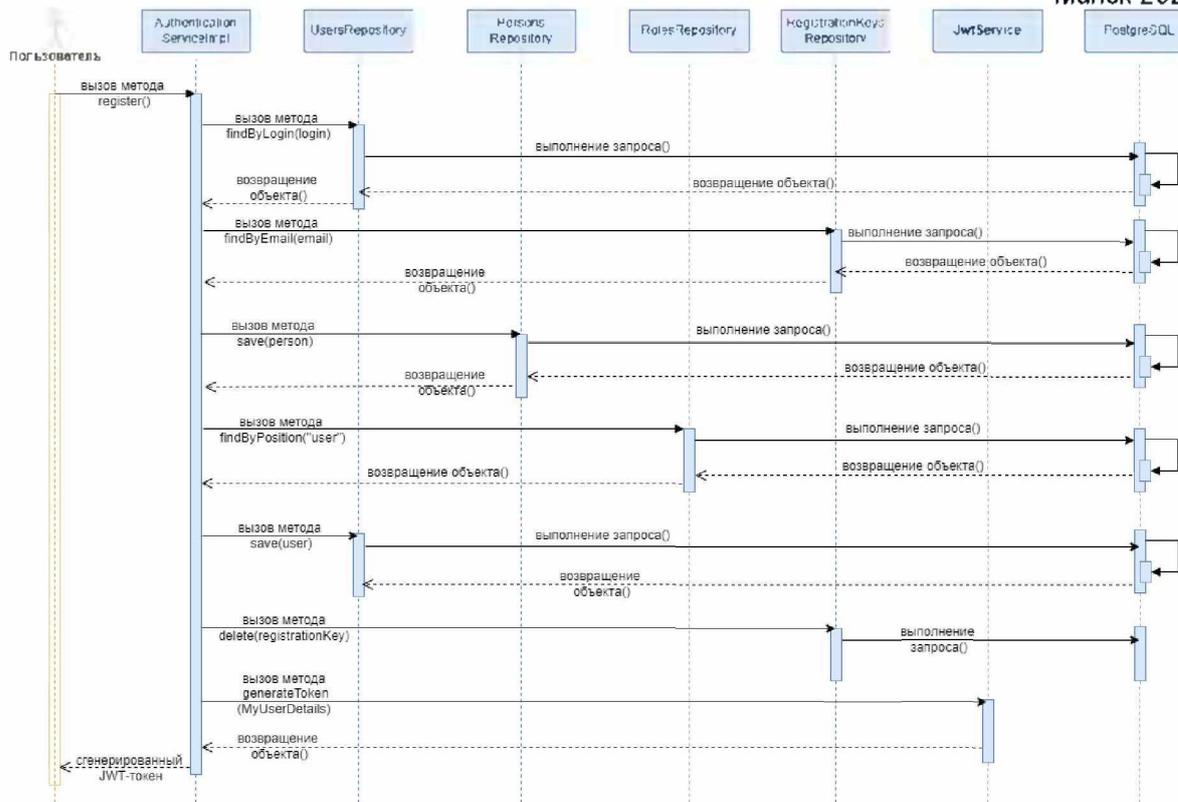


Рисунок 5 – Диаграмма последовательности

Таким образом, была разработан сервис для управления корпоративным обучением сотрудников с элементами геймификации, направленный на повышение мотивации и уровня компетентности персонала. В процессе работы была создана устойчивая программная архитектура, спроектирована и реализована база данных, а также сформирована клиент-серверная инфраструктура взаимодействия. Система прошла тестирование, показала стабильную работу и была внедрена в эксплуатацию, что подтвердило её готовность к дальнейшему использованию в рамках корпоративной образовательной среды.

**Список использованных источников:**

1. Баранов, И.Н., Юрченко, В.И. "Корпоративное обучение для цифрового мира". (3-е издание). — М.: Изд-во Альпина Паблишер, 2020. — 304 с.
2. Крейг, У. Spring в действии. — М.: Изд-во Манн, Иванов и Фербер, 2022. — 480 с.
3. Петров, М. П. Инструменты моделирования бизнес-процессов: переход от теории к практике. — М.: Юрайт, 2018. — 400 с.

UDC 004.455.1:005.963

## CORPORATE TRAINING PROGRAM MANAGEMENT SYSTEM WITH GAMIFICATION ELEMENTS TO INCREASE MOTIVATION AND COMPETENCE LEVEL OF EMPLOYEES

Kruk E.S., Storozhev D.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Komlichenko V.N. – Candidate of Technical Sciences, Associate Professor

**Annotation.** This article discusses the development of a system for managing corporate training and employee certification aimed at optimizing the processes of training and professional development of personnel. The key stages of the system's operation are described, including the creation and management of training courses, monitoring employee progress, organizing testing to assess the assimilation of material, and generating personalized certificates. The system also implements functions for collecting feedback from users and administration, including managing employee and teacher accounts.

**Keywords.** Corporate training, employee certification, management system, database, Java programming language, Spring Framework, architectural solutions, PostgreSQL.