

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

ДОПУЩЕНА К ЗАЩИТЕ

Заведующий кафедрой

_____ / Е.А. Пухова /

Руководитель образовательной
программы

_____ / М.В. Даньшина /

**СЕРВЕРНАЯ ЧАСТЬ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ГЕЙМИФИКАЦИИ
ПРОЦЕССА ОБУЧЕНИЯ СОТРУДНИКОВ ДИЛЕРСКИХ ЦЕНТРОВ
BMW GROUP**

Выпускная квалификационная работа
по направлению 09.03.03 Прикладная информатика
Образовательная программа (профиль) «Корпоративные информационные
системы»

Студент: _____ / Наконечный Павел Александрович, 191-361 /
подпись *ФИО, группа*

Руководитель ВКР: _____ / Змазнева Олеся Анатольевна, к.ф.н., доцент /
подпись *ФИО, уч. звание и степень*

Москва, 2023

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
по направлению 09.03.03 Прикладная информатика
Образовательная программа (профиль) «Корпоративные информационные
системы»

ТЕМА ВКР	Серверная часть веб-приложения для геймификации процесса обучения сотрудников дилерских центров BMW Group
ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ	
Назначение	Корпоративное обучение
Основные функции	Прохождение курсов, организация учебного процесса по группам, геймификация.
Используемые технологии и платформы	TypeScript, JavaScript, Node.JS, React, MySQL, SQLite
ВЫПОЛНЕНИЕ РАБОТЫ	
Решаемые задачи	1. Провести анализ предметной области, аналогов и конкурентов. 2. Разработать проект программного продукта системы корпоративного обучения. 3. Разработать БД и программные компоненты системы. 4. Провести тестирование системы. 5. Предложить методы анализа данных. 6. Оценить технико-экономическую целесообразность проекта. 7. Обеспечить информационную безопасность, надежность, отказоустойчивость и производительность системы.
Состав технической документации	1. Пояснительная записка. 2. Техническое задание.
Состав графической части	Контекстная диаграмма – 1 экз. IDEF0 диаграмма – 2 экз. BPMN диаграмма – 2 экз. DFD диаграмма – 2 экз. Ментальная карта – 1 экз. ER диаграмма – 1 экз. UML диаграмма классов – 2 экз. Диаграмма последовательности – 1 экз.

ПЛАН РАБОТЫ НАД ВКР

ЗАДАЧИ	НЕДЕЛИ																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Провести анализ предметной области, аналогов и конкурентов																		
Разработать проект программного продукта системы корпоративного обучения, демонстрирующий особенности реализации и функционирования ИС																		
Разработать базу данных системы, серверный модуль системы, панель администрирования																		
Предложить методы анализа данных																		
Оценить технико-экономическую целесообразность проекта																		
Провести тестирование компонентов системы																		
Обеспечить информационную безопасность, надежность, отказоустойчивость и производительность системы																		

РУКОВОДИТЕЛЬ ОП:

«__» _____ 2023, _____ / Даньшина Марина Владимировна /
подпись *ФИО, уч. звание и степень*

РУКОВОДИТЕЛЬ ВКР:

«__» _____ 2023, _____ / Змазнева Олеся Анатольевна, к.ф.н., доцент /
подпись *ФИО, уч. звание и степень*

СТУДЕНТ:

«__» _____ 2023, _____ / Наконечный Павел Александрович, 191-361 /
подпись *ФИО, группа*

АННОТАЦИЯ

Выпускная квалификационная работа «Серверная часть веб-приложения для геймификации процесса обучения сотрудников дилерских центров BMW Group» посвящена разработке веб-сервиса и административной панели для обучающего информационного веб-приложения для брендов BMW Group.

Эта работа актуальна, так как у Центра Обучения и Квалификации BMW нет информационной системы для организации дистанционного обучения, что привело к списыванию, низким вовлеченности учащихся и усвояемости материала.

Целью данной выпускной квалификационной работы является разработать серверную часть информационной системы для корпоративного обучения сотрудников дилерских центров BMW Group.

Работа состоит из 3 глав, Введения, Заключения и Приложения. Во введении обозначена актуальность, цель и задачи, объект и предмет исследования. В первой главе проведены анализ предметной области и проектирование ИС. Вторая глава посвящена технической реализации проекта. В Третьей главе раскрыты технические и экономические аспекты будущего внедрения и эксплуатации практического результата ВКР. В Заключении представлены выводы по поставленным задачам.

Работа изложена на 104 страницах, из них приложение на 7 листах. В работе представлены 44 рисунка, 9 таблиц. Список источников включает 53 источника. Приложение содержит 1 рисунок.

КЛЮЧЕВЫЕ СЛОВА: ВЕБ-СЕРВИС, ВЕБ-ПРИЛОЖЕНИЕ, BMW, ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ГЕЙМИФИКАЦИИ, ГЕЙМИФИКАЦИЯ, КОРПОРАТИВНОЕ ОБУЧЕНИЕ, КОРПОРАТИВНОЕ ОБРАЗОВАНИЕ, АДМИНИСТРАТИВНАЯ ПАНЕЛЬ, NODE.JS, REACT.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	7
Глава 1 ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ	9
1.1 Описание проекта	9
1.2 Актуальность темы, концептуальная база корпоративного обучения	10
1.3 Проблема предметной области.....	16
1.4 Анализ подобного ПО	18
1.5 Функционал системы.....	27
1.6 Выводы.....	39
Глава 2 ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ.....	41
2.1 Обзор современных технологий и принципов для создания веб- сайтов.....	41
2.2 Особенности реализации Бэкенда	43
2.2 Особенности реализации панели администратора	50
2.3 Особенности функционирования API Бэкенда.....	53
2.4 Особенности функционирования панели администратора	57
2.5 Подходы к проектированию аналитических дашбордов.....	65
2.6 Выводы.....	69
Глава 3 ВНЕДРЕНИЕ И ЭКСПЛУАТАЦИЯ	70
3.1 Оценка технико-экономической целесообразности.....	70
3.2 Источники контента.....	76
3.3 Аспекты тестирования программного продукта	77
3.4 Аспекты информационной безопасности.....	80
3.5 Установка и настройка программных компонентов	82
3.6 Аспекты надежности, отказоустойчивости и производительности программного продукта.....	84
3.7 Выводы.....	87

ЗАКЛЮЧЕНИЕ	89
СПИСОК ИСТОЧНИКОВ	92
ПРИЛОЖЕНИЕ А.....	98

ВВЕДЕНИЕ

BMW Group (далее – Организация) является одним из крупнейших международных автоконцернов. Одной из функций Центра квалификации и обучения BMW является подготовка и переподготовка сотрудников дилерских центров брендов BMW Group. Вследствие пандемии Центр квалификации и обучения BMW перешёл на дистанционный формат обучения.

Целевая аудитория системы – сотрудники Центра квалификации и обучения BMW, сотрудники дилерских центров и другие сотрудники офиса BMW.

Процесс обучения построен с применением геймификации – концепции обучения через игру, что увеличивает мотивацию и вовлеченность учащихся.

Для успешного воплощения проекта в жизнь, необходимо выполнить следующие задачи:

1. Провести анализ предметной области, аналогов и конкурентов.
2. Разработать проект программного продукта системы корпоративного обучения, демонстрирующий особенности реализации и функционирования ИС.
3. Разработать базу данных системы, серверный модуль системы, панель администрирования.
4. Предложить методы анализа данных, собираемых системой.
5. Оценить технико-экономическую целесообразность проекта.
6. Провести тестирование компонентов системы.
7. Обеспечить информационную безопасность, надежность, отказоустойчивость и производительность системы.

При разработке использовались: язык программирования TypeScript, JavaScript, платформа Node.JS, фреймворк React, СУБД MySQL, SQLite.

В процессе проектирования системы использовались анализ академических источников, методы бизнес-анализа, контекстные диаграммы, IDEF0 диаграммы, BPMN диаграммы, теория SDLC.

Практический результат системы состоит из нескольких компонентов:

1. Панели администратора.

2. Веб-сервиса (далее – Бэкенд), предоставляющий API для взаимодействия с другими компонентами.

Внедрение системы позволит повысить компетентность сотрудников, проходящих обучение в Центре Квалификации и Обучения BMW, и, как следствие, увеличить продажи автомобилей, сопутствующих товаров и услуг в дилерских центрах брендов концерна. Работа является практически значимой.

Глава 1 ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Описание проекта

1.1.1 Описание организации Заказчика

В России у BMW Group представлено более чем 90 аккредитованных дилерских центров [10, 22, 23, 32].

С развитием онлайн-технологий и переходом в онлайн-формат обучения вследствие пандемии для Организации стала актуальна цифровая трансформация процессов работы Центра Обучения.

1.1.2 Предметная область и предмет проекта

Предметной областью проекта являются информационные системы для корпоративного обучения сотрудников.

Предметом проекта является информационная система для корпоративного обучения, предоставляемого компанией BMW сотрудникам лицензированных дилерских центров.

1.1.3 Описание целевой аудитории

Целевой аудиторией проекта являются сотрудники дилерских центров (внешние организации), проходящие обучение в Учебном Центре BMW.

Дилерские центры обеспечивают продажи автомобилей конечным пользователям. Важно отметить, что организация-заказчик не может нанимать или увольнять сотрудников дилерских центров, но может при помощи обучения повысить их компетентность в продажах и знания ассортимента, что в конечном итоге приведёт к росту продаж автомобилей.

Другая группа пользователей – сотрудники офиса BMW.

В целях дальнейшего анализа сотрудники дилерских центров будут приравнены к собственным сотрудникам организации для задач корпоративного обучения.

Возраст целевой аудитории Заказчик определяет как 20-45 лет.

1.2 Актуальность темы, концептуальная база корпоративного обучения

1.2.1 Понятие геймификации

Геймификация – использование механик компьютерных игр в организации процессов использования неигрового прикладного программного обеспечения с целью привлечения пользователей, повышения их вовлеченности.

Для геймификации свойственны: системность, наличие чётких правил, целей, структуры, привязки к реальному миру.

Геймификация применяется широко как в сфере онлайн образования, так и в приложениях для поддержания здоровья (Nike+Run [42, 45], ВК шаги), в соц. сетях (Foursquare [33], DevHub [50], Reddit [40]), в маркетинге других видов бизнеса.

Ключевым аспектом геймификации является игровая механика – набор правил и способов, определяющих интерактивное взаимодействие игрока и игры.

Татаринов выделяет следующие элементы геймификации: очки, уровни, таблицы результатов, значки, задания [26]. Также можно выделить использование сторителлинга и аватаров.

Рассмотрим каждый из них подробнее:

1. Очки являются вознаграждением за прогресс или за желаемые действия (приближение к победе).

2. Уровни определяют статус игрока и показывают его место в рейтинге по отношению к другим игрокам, что мотивирует их к дальнейшей игре.

3. Таблицы результатов усиливают дальнейшее взаимодействие и позволяют игроку сравнивать себя с другими, а также хвастаться достигнутыми результатами.

4. Значки достижений показывают проблемы, с которыми столкнулся игрок, и то, чего он достиг.

5. Задания, которые игроку нужно выполнить, позволяют набрать очки или выйти на более высокий уровень игры.

6. Сторителлинг и хорошо описанный мир дают контекст игровым механикам, упрощая их понимание.

7. Аватары – это визуальное отображение пользователя в игре.

Как пишет Катарина Ян, аватары добавляют персональности опыту пользователя при работе с системой, увеличивая вовлеченность [36].

1.2.2 История геймификации

Первым применением геймификации в образовании можно назвать 1908 год, когда было основано движение Бойскаутов. Они получали значки как форму мотивации и признания их достижений.

Механика вознаграждения для поощрения лояльности клиентов началась с American Airlines, когда они выпустили первую программу лояльности в 1981 году, и эта инициатива был воспроизведена франшизой Holiday Inn, National Car Rental и многими другими. Эту модель до сих пор используют многие розничные компании.

Спустя почти 70 лет после образования Бойскаутов (в 1984 г.) Чарльз Кунрадт написал книгу «Игра в работу: как наслаждаться работой не меньше, чем игрой», чтобы подчеркнуть кризис производительности в Соединенных Штатах в то время, потому что сотрудники стали менее продуктивным в работе. Основная идея книги заключается в том, что стимулы, которые мотивируют нас в отдыхе, можно применить к работе.

Со временем геймификация стала применяться не только в бизнесе, но и других секторах, в частности, в образовании. В 80-х годах профессор Школы бизнеса Слоуна в Массачусетском технологическом институте по имени Томас У. Мэлоун выпустил книгу «Что делает вещи интересным для изучения: исследование внутренне мотивирующих компьютерных игр», документ, в котором он описал, как студенты могли учиться, играя в компьютерные игры, и как игровые функции могут быть использованы для того, чтобы сделать обучение интересным и приятным. Он также опубликовал «К теории

внутренне мотивирующего обучения» в 1981 г. и «Эвристика для разработки приятных пользовательских интерфейсов» в 1982 году.

По мере развития индустрии видеоигр, рос и интерес исследователей к ней. В 1996 г. Ричард Бартл разделил игроков на 4 уникальные категории (убийцы, достигаторы, исследователи и социализирующиеся), основываясь на том, как они подходят к игре. И в 1999 г. Стивен Драпер опубликовал работу «Анализируя развлечение как требование к ПО», в которой он пишет о важности удовольствия пользователя от работы с программой, и что оно должно быть включено как требование при разработке ПО.

По мере приближения миллениума, люди начали воспринимать игры и игровые механики как способы увеличить производительность на рабочем месте. Термин геймификация был рождён Ником Пеллингом лишь в 2002 г.

В 2007 г. компании как Badgeville, Bunchball и Ripple стали оказывать услуги геймификации. В 2010ых спикеры как Джесси Шелл, Джейн МакГонигал и дизайнеры видеоигр Кристофер Канингем и Гейб Зихерман популяризировали концепт и заставили общественность задуматься о потенциале геймификации в практически любой человеческой деятельности.

В 2017 г. первое событие под названием Gamification Day было проведено в Барселоне. Оно включило множество воркшопов, которые помогли применить техники геймификации существующим компаниям. С тех пор Gamification Day проводится ежегодно.

Сейчас геймификацию можно найти повсюду. Практически все приложения применяют геймификацию в той или иной форме, от Facebook до Duolingo. Теперь это не только бизнес и образование, но и спорт, медитация, список дел и многое другое.

Резюмируя, геймификация всегда была с нами, но у неё не было названия. История происхождения геймификации неясна, но будущее очевидно. Согласно данным Growth Engineering Report: «Геймификация имеет потенциал решать проблемы от изменения поведения отдельных людей до проблем глобального масштаба» [34, 48].

1.2.3 Необходимость организации корпоративного обучения

Андриянова пишет: «Обучение сотрудников является неотъемлемым элементом системы управления человеческими ресурсами современной организации» [6].

Большим компаниям свойственно иметь несколько категорий персонала с большим числом сотрудников, выполняющих одинаковые функции. Такими категориями могут быть операторы колл-центров, продавцы в отделениях, упаковщики и т.д.

Для улучшения работы таких специалистов, важно уделять внимание организованному обучению сотрудников [15].

В условиях рыночной конкуренции компании вынуждены постоянно адаптироваться под меняющиеся условия: меняются процессы, процедуры, организационная структура и т.д. [5].

Корпоративное обучение помогает организациям адаптироваться, добиваться эффективности и конкурентоспособности, помогает развивать организацию.

С увеличением размеров компании и ее присутствия на различных рынках возникают сложности в управлении внутренними процессами и соответствии внешним требованиям. Объем документов, связанных с проектами, нормами и требованиями, которые напрямую влияют на ежедневную работу, становится настолько большим, что каждый сотрудник не может самостоятельно его изучать и следить за изменениями.

Сотрудник или отдел больше не участвуют в подготовке многих документов, связанных с процессами, проектами, нормами и требованиями, которые непосредственно относятся к ежедневной работе.

Традиционный метод ознакомления сотрудников с новыми стандартами через роспись в журнале неэффективен. Документы, выпущенные региональными центрами, могут находиться без дела на полке многие годы, прежде чем будут внедрены в процессы на местах.

Инициативы, которые не подкреплены внутренней коммуникацией, неминуемо проваливаются [52]. Важным аспектом управления изменениями является трансформационное лидерство и стратегическое использование внутренней коммуникации.

Нельзя полагаться лишь на вертикаль власти и возлагать обязанности по изменению бизнес-процессов и донесению смысла изменений адресатам только на глав отделов и департаментов. Продуктивность менеджеров среднего звена может сильно проседать в отдельных частях организации, создавая операционные риски без полноценного внедрения изменений.

В то же время, компаниям не стоит передавать развитие персонала полностью на аутсорсинг. Выстроенные ими программы обучения обычно не совпадают с запросами компании.

Инструменты корпоративного обучения позволяют донести смысл новых требований, например, изменений законодательства в областях персональных данных и импортозамещения, до каждого сотрудника организации, минуя менеджеров как посредников.

Для обеспечения ожидаемых исходов при внедрении изменений, рекомендуется выделить корпоративное обучение как самостоятельную функцию в структуре организации.

Важной задачей корпоративного обучения работников выступает увеличение уровня квалификации сотрудников [9].

Резюмируя, корпоративное обучение представляет собой совокупность процессов, которые направлены на повышение квалификации и дополнительную подготовку сотрудников, развитие новых компетенций в связи с изменением их функций.

1.2.4 Типичные задачи, процессы и результаты корпоративного обучения

По данным Мыльниковой и Погосян, сегодня затраты на корпоративное обучение в прогрессивных компаниях стоят на втором месте после затрат на оплату труда [17].

Можно определить несколько направлений корпоративного обучения:

1. Приобщение новых сотрудников к процессам работы, знакомство с внутренними правилами и структурой компании.
2. Обучение персонала при внедрении новых технологий и практик, изменении требований контроля и регулирования работы.
3. Развитие общекорпоративных компетенций, основанных на ценностях компании и корпоративной культуре.

В области корпоративного обучения можно выделить несколько общих характеристик, таких как: сбор и обновление знаний, которые направлены на достижение целей организации; улучшение квалификации сотрудников; развитие системы корпоративных ценностей и культуры.

Модель Колба считается наиболее эффективной в практике среди всех теорий, описывающих и объясняющих процесс результативного обучения. В соответствии с этой моделью, люди обучаются в определенной последовательности: приобретение опыта в учебных условиях, рефлексия полученного опыта, выработка концепций на основе выводов и экспериментирование с новыми концепциями [29].

Следовательно, процесс корпоративного обучения для проектирования ИС и методических материалов может быть разделен на четыре этапа: повторение опыта других, его применение в обучении, использование для усвоения новых знаний и применение компетенций в практической деятельности. Таким образом полученный опыт может изменить подход специалиста к работе.

Участие в корпоративном обучении сотрудников стимулируется возможностями профессионального развития, улучшения личных навыков и учетом индивидуальных интересов [30].

Для улучшения обучающего персонала, систем и процессов обучения компаниям необходимо получать обратную связь о результатах обучения. Обратную связь можно организовать через индивидуальные консультации тренеров с обучающимися, централизованный сбор анкет обучающихся для

оценки качества восприятия информации, анализ успеваемости учащихся по тестам и анализ массива данных о прохождении обучения, таких как скорость выполнения заданий, количество ошибок и возвращение к пройденным материалам.

Однако последний источник обратной связи доступен только при цифровизации процессов обучения и аналитической работе с накопленными данными.

1.3 Проблема предметной области

1.3.1 Проблемы, с которыми столкнулась организация

С началом пандемии и введением ограничений, Учебный Центр перешёл на дистанционный формат работы, что привело к ряду проблем, таких как низкая вовлеченность учащихся, низкая усвояемость материала и списывание.

1.3.2 Как ИС может решить выявленные проблемы

Цифровая трансформация — неизбежный этап для компании, вызванный современным технологическим развитием в условиях коммерческой конкуренции [54].

В среднесрочной перспективе для HR департаментов необходимо увеличить производительность персонала и сократить расходы на управление [11]. Цифровые технологии – один из инструментов для достижения этой цели.

Благодаря цифровым технологиям, корпоративное обучение становится более доступным, систематизированным и стандартизированным процессом с меньшими затратами [37].

Внедрение ИС позволит использовать положительный эффект от микрообучения. По определению Корпоративного Университета Сбербанк, «микрообучение – метод обучения, когда контент подается небольшими частями, каждая из которых имеет одну конкретную цель, и ее выполнение занимает совсем немного времени» [21]. Регулярное взаимодействие с материалом в ненавязчивой форме – залог эффективности подхода.

Другая полезная методика, которая будет реализована через ИС – обучение на ошибках. Ошибка – сигнал о том, что обучающийся чего-то не знает. Правильно преподнесенная пользователю ошибка создаёт внимание и мотивацию исправиться.

Как объясняет этот феномен Шальнев, чем больше мы проверяем наши знания, тем больше прогрессируем [28]. При этом необходимо обеспечить:

- прозрачную обратную связь: что было правильно, а что – нет;
- тесты подходящие под уровень и профиль обучающегося.

Для повышения заинтересованности студентов и более тщательного освоения новых навыков мы будем применять методы геймификации. К ним относятся, например, использование больших целей и призывов, ощущение постоянного развития, значки достижений и списки лидеров, использование творчества, персонализация и социализация.

К методам геймификации можно также отнести управление чувствами и эмоциями пользователя: любопытством, страхом потери, завистью. Если друзья пробегают больше вас – вы им завидуете, что создаёт мотивацию приложить больше усилий. Этот механизм применяется в Nike Run Club.

Автоматизация процесса обучения позволяет без лишних усилий организовать асинхронное обучение: учащиеся будут выполнять задания в своём темпе без постоянного участия преподавателя. Гибкость подхода позволяет студентам самостоятельно выбирать, проходить ли длинные блоки материала или же делить их на небольшие ежедневные порции.

Для повышения уровня компетентности специалистов необходимо внедрение многокомпонентной системы непрерывного обучения [27]. Система корпоративного обучения – один из процессов непрерывного развития потенциала сотрудника.

Негативный побочный эффект электронного обучения — оно позволяет выполнить задание в системе за других. Списывание нельзя предотвратить без систем онлайн-прокторинга, которые влекут за собой большие расходы и стресс для учащихся.

Исследователи отмечают, что подача контента в форме мультимедийных презентаций гораздо эффективнее лекций и учебников [8, 25].

Для организации коммуникации учащихся и тренеров BMW предлагается проводить онлайн встречи учащихся с тренерами, использовать чаты.

Чтобы совершенствовать материалы курсов, BMW необходимо изучать данные предыдущих периодов. Дашборды успешно применялись для анализа внедрения систем обучения учебными организациями [24].

В итоге, при помощи системы учащиеся Центра смогут быстрее и эффективнее усваивать новые навыки, что решит ряд проблем организации, связанных с процессами корпоративного обучения.

1.4 Анализ подобного ПО

Организация не может использовать проприетарные решения на чужих серверах из-за корпоративных политик и процедур, а также локального законодательства РФ. Поэтому единственной альтернативой являются open-source решения, которые не всегда отвечают требованиям безопасности.

На сегодняшний день существует множество решений для обучения сотрудников. Однако они не предоставляют геймификационную компоненту. Moodle – лидер среди Open Source Learning Management систем, однако в нём не получится построить сложный трек обучения с древовидными зависимостями упражнений без сторонних компонентов.

Важно отметить, что для организации также важно обеспечить соответствие материалов тренингов немецким стандартам материнской организации.

Продукты, выполняющие аналогичные разрабатываемой Системе функции, являются внутренними разработками, защищёнными коммерческой тайной. Такие системы применяются в крупных международных компаниях.

Взамен были рассмотрены доступные массовому пользователю системы, которые не являются прямыми конкурентами разрабатываемой ИС.

В них выделены примеры удачного применения функционала и игровых механик для подачи образовательного контента.

1.4.1 Duolingo

Duolingo – самое скачиваемое образовательное приложение, которое использует более 500 млн человек [7]. Duolingo – сервис для изучения иностранных языков, доступный как в виде сайта, так и в мобильном приложении.

Весь учебный процесс в Duolingo разбит на пронумерованные модули – «Units», соотносящиеся с основными темами лексики и грамматики языка. Модули состоят из последовательно разблокируемых учебных блоков нескольких типов: тест, текст, практика. В начале модуля несколько тестовых заданий посвящено практике и «вспоминанию» прошлой темы. В конце модуля даётся комплексный экзамен.

Все учебные блоки представлены набором заданий. Пользователю недоступно перепрыгивать задания. В блоке типа «тест», если пользователь ошибается, задание перемещается в конец и повторяется, пока пользователь не ответит верно на все вопросы. Один тестовый блок развивает сразу множество навыков: говорение, слушание, грамматику (вставь в таблицу или предложение корректную форму) и построение предложений целиком.

Для закрепления уже пройденных тем пользователь может бесконечно повторять экзаменационное задание (с разными наборами вопросов) с нарастающей сложностью, обеспечиваемой не только сложностью вопросов, но и жёстким ограничением по времени прохождения. Взамен пользователь получает большое число опыта – в 4 раза больше чем за простое тестовое задание.

Преимущества игровых подходов в образовании изучаются с 80-ых годов. Поразительные результаты Duolingo в сфере онлайн изучения языков исследователи во многом связывают с геймификацией.

Ярким примером геймификации в Duolingo являются значки достижений и списки лидеров, которые поддерживают стремление к

прогрессу. В Duolingo пользователи выполняют задания по иностранному языку и набирают опыт, чтобы занимать высокие места в еженедельной таблице лидеров.

За выполнение заданий пользователю выдаётся опыт. Размер опыта зависит от скорости выполнения задания, числа ошибок, сложности задания и его типа. Одна из основных метрик успешности пользователя – общее количество опыта, набранное им за период обучения. Также к опыту привязаны ежедневные задания, задания «выполни с другом», достижения и прогрессия в лиге.

Чтобы ограничить возможность перебора и подстегнуть пользователя к покупке подписки Super, число жизней (=ошибок) ограничено до 3 и восстанавливается со временем. При достижении определенного порога длительности непрерывного обучения, жизни становятся бесконечными. Если в процессе выполнения блока у пользователя закончились жизни – ему предлагается их восполнить за внутриигровую валюту или закончить выполнение блока. Жизни можно получить повтором прошлых тем за удвоенный опыт.

Часть лексики и грамматики языка вводится и осваивается в процессе прохождения учебных блоков. Другая же, для систематизации, представлена в справочнике – «Guidebook» – модуля.

1.4.2 Coursera

Coursera – онлайн образовательная платформа, где университеты и крупные компании публикуют свои курсы, специализации. На платформе можно даже закончить университет и получить диплом государственного образца США, Соединенного Королевства или Индии.

В Coursera, в отличие от Udey, можно проходить не только тесты, но и более сложные задания с произвольным ответом – текстом или загруженным файлом. Чтобы максимально ускорить процесс проверки и не зависеть от авторов курса, такие задания проверяются одноклассниками, которые уже выполнили задание. Такой подход оптимален для популярных курсов и

позволяет получить оценку в течение пары часов. На менее популярных курсах оценки можно ждать несколько недель.

1.4.3 UdeMy

UdeMy – онлайн-платформа для независимого обучения через курсы. В отличие от Coursera, на UdeMy выкладывать курсы может любой желающий, что приводит к множеству материалов низкого качества и без какого-либо признания индустрией. Чего не скажешь о курсах Google или IBM на платформе Coursera.

Но на сайте и в приложении UdeMy можно увидеть как по-разному можно подать один и тот же курс, используя разные форматы контента: видео (Рис. 1), текст (Рис. 2), дополнительные файлы (Рис. 3).

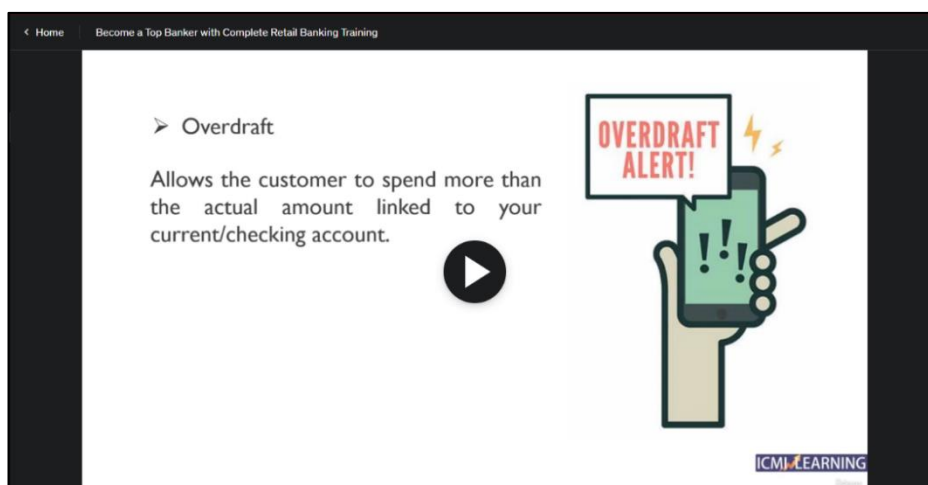


Рисунок 1 — Видеоматериал курса

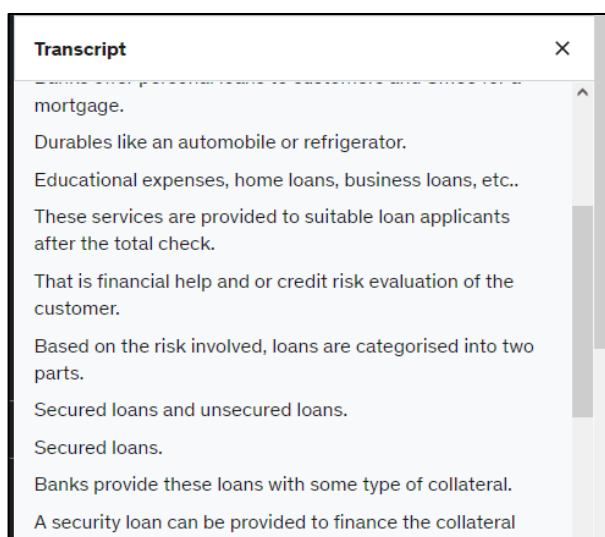


Рисунок 2 — Расшифровка лекции

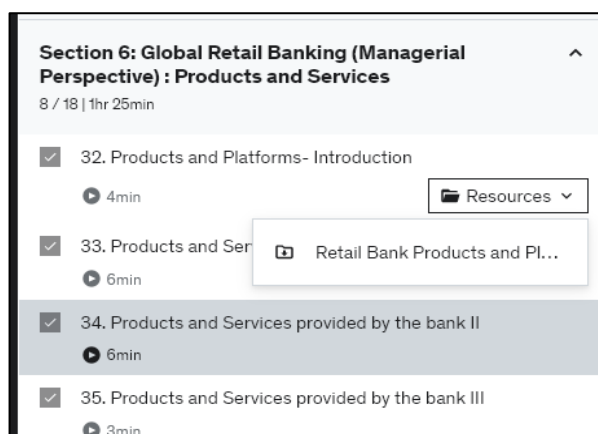


Рисунок 3 — Прикрепленные к лекции дополнительные материалы

Этот же приём можно найти и в СМИ. Многие крупные издатели подают одну и ту же статью в разных форматах: тексте, YouTube роликах, подкасте. Тренд дошёл и до консалтинговых агентств (Рис. 4). Udemu также использует небольшой приём из геймификации – на отражение прогресса и обещание награды в конце (Рис. 5).

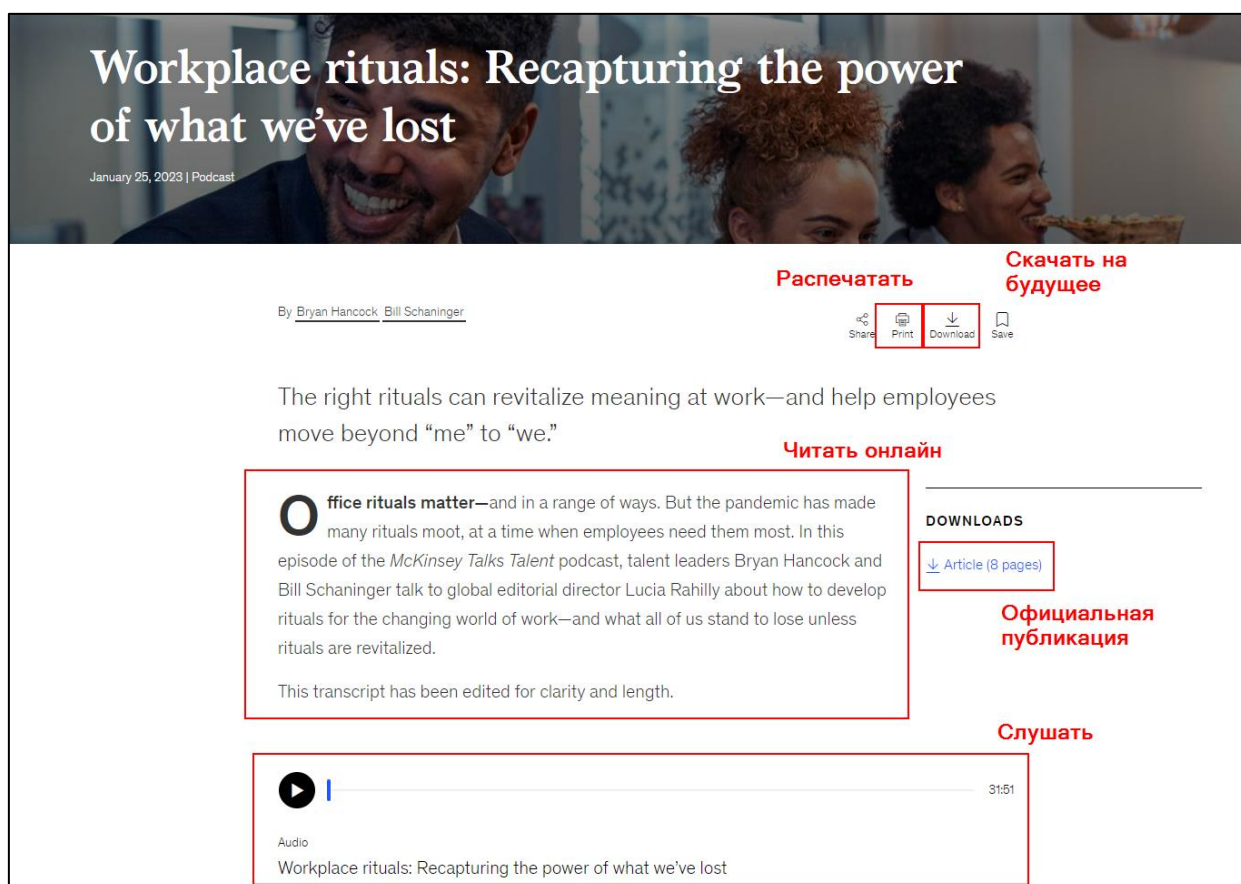


Рисунок 4 — Пример статьи с сайта McKinsey

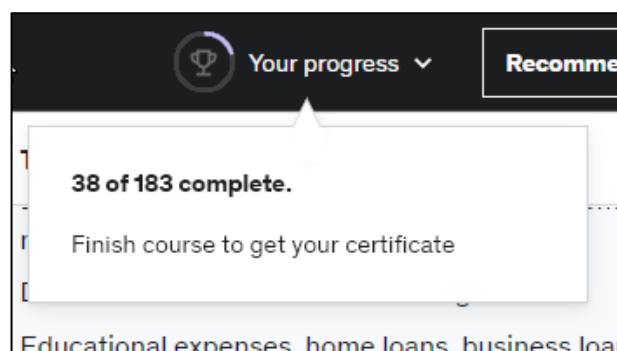


Рисунок 5 — Мотивация к действию через достижения

1.4.4 Infinite Spanish

Infinite Spanish – мобильное приложение, доступное в Google Play Store. Это скорее игра для изучения испанского, чем обучающее приложение с геймификацией. Пользователю предлагается на скорость выбирать правильные переводы испанских слов, чтобы уничтожить метеориты, летящие к земле. Если метеорит достигает земли – игрок проиграл.

В отличие от остальных рассматриваемых приложений, в Infinite Spanish основное ограничение пользователя – скорость ответа. Для простых вопросов, требующих интуитивного ответа, где ответ можно задать выбором из 2-3 вариантов, это хороший дизайн процесса обучения.

1.4.5 Magoosh IELTS

Magoosh IELTS – приложение для подготовки к экзамену IELTS по английскому языку, сосредоточенное на подаче контента, а не тестировании. Пользователю доступны статьи и видеоматериалы разбитые по темам экзамена: «Reading, Listening, Writing, and Speaking». Многие страницы приложения – лишь встроенные веб-страницы с сайта Magoosh. Тестовая часть системы чрезмерно упрощена и даже не близка по формату к экзаменационному.

1.4.6 WordUp

WordUp – мобильное приложение для изучения слов английского языка всех уровней сложности. Основной акцент в приложении уделяется теоретическому материалу, а не самому тестированию. Для каждого слова

можно найти объяснение из словаря, транскрипцию, мужское и женское произношение в британском и американском английском, синонимы, примеры из цитат, ролики из YouTube, ссылки на новости с применением слова. Таким образом, приложение позволяет изучить термин со всех сторон в пределах одного приложения. В то же время пользователям Duolingo приходится отдельно искать даже базовые правила грамматики языка.

1.4.7 Finance Pocket Prep

Прохождение сертификаций FINRA, таких как CFP, Series 6, 7, 57, 63, 65, 66, 79, SIE требуется сотрудникам банков и брокеров, связанным с рынками капитала и предписано законодательством США [46]. Прохождение сертификаций Health Insurance и Life Insurance, к которым также помогает подготовиться приложение, требуется для продажи страховок. Finance Pocket Prep – мобильное приложение для подготовки к экзаменам FINRA.

Важное отличие этого приложения от остальных рассмотренных – его профессиональная направленность. Как и в случае с нашей ИС, приложением пользуются специалисты для профессиональных целей – сдачи экзаменов регулятора. Поэтому проанализировать используемые в нём приёмы геймификации особенно интересно.

В приложении доступно множество типов тестирования: вопрос дня, быстрый тест на 10 вопросов, тест на время, тест по слабой теме, тест по ранее неправильным ответам, «собери свой тест» (Рис. 6). Часть из них позволяет максимально приблизиться к условиям экзамена, другая – исправить собственные ошибки.

Для всех верных ответов пользователю отображается развёрнутое объяснение (Рис. 7). После завершения теста нам показывается средняя оценка всех пользователей с тем же набором вопросов (Рис. 8). На каждый вопрос можно даже посмотреть разбивку в % соотношении вариантов ответа выбранных другими пользователями, количество ответивших и долю ответивших верно (Рис. 9).

Основная метрика успеха для пользователя – доля верных ответов и доля пройденных вопросов из общего банка. Создатели даже предлагают завести обратный отсчет дней до записи пользователя на сертификационный экзамен.

Другая метрика – общее число пройденных тестов, число выполненных вопросов дня и общее время обучения.

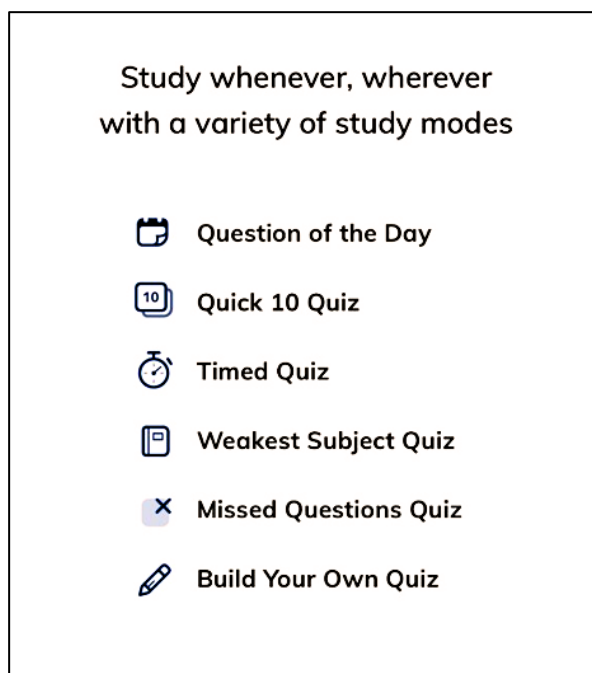


Рисунок 6 — Типы тестирования

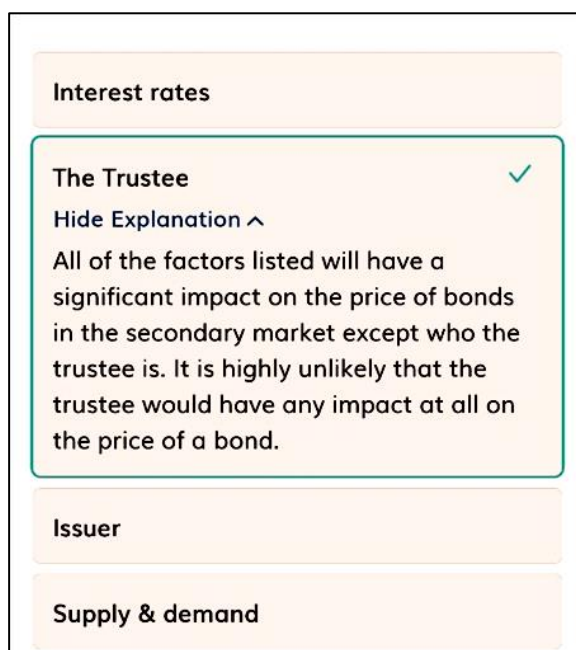


Рисунок 7 — Объяснения ответов

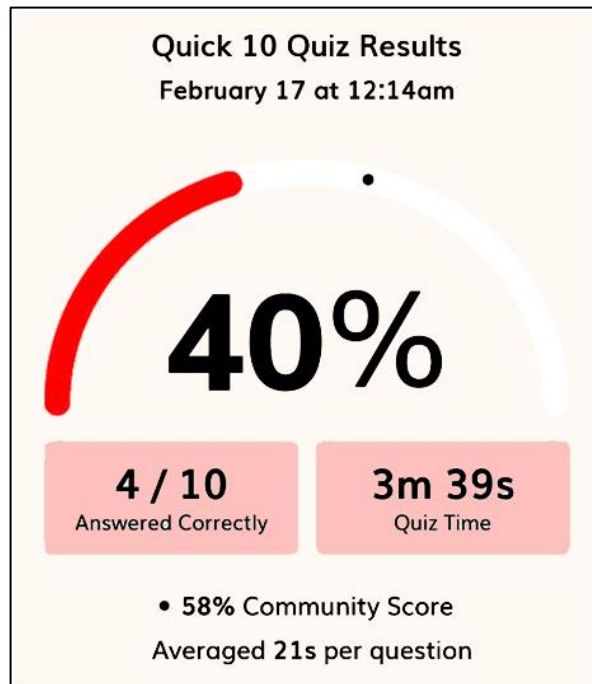


Рисунок 8 — Социальная статистика

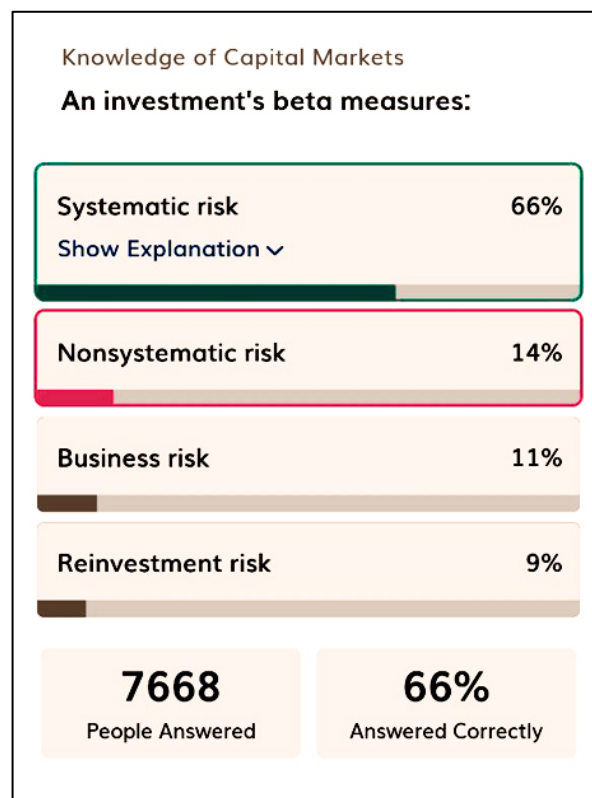


Рисунок 9 — Разбивка по выбранным вариантам ответа

На вкладке «Review» пользователь может вернуться к объяснениям вопросов, на которые он ранее отвечал неверно и увидеть, к какой теме экзамена они относятся. Агрегированной статистики по темам не представлено.

Пользователя мотивируют возвращаться в приложение накоплением непрерывной серии и вопросами дня. Чтобы не забывать учиться, пользователь может включить уведомления «Вопрос дня» и простые напоминания.

Всего в приложении доступно 4992 вопроса по 10 экзаменам. Ещё больше вопросов доступно с платной подпиской Premium.

1.4.8 Подытоги

Можно отметить, что во многих анализируемых продуктах значительная часть механик геймификации работает на ограничение пользователя:

- ограниченное количество жизней;
- медленная регенерация жизней;
- покупка доступа к материалам за внутреннюю валюту.

Автору представляется, что цель таких механик – побудить пользователя приобрести платные версии приложения или подписки на полный функционал, что не совпадает с целями организации по внедрению системы.

Другие же механики представляются скорее косметическими, например, еженедельные квесты с другом, если в приложении уже есть рейтинг по лигам и ежедневные индивидуальные квесты. Такие механики, хоть и обеспечат определенный эффект на результатах обучения, могут дать и обратный эффект (отвлечь пользователя от его задачи обучения), а потому также не будут включены в функционал разрабатываемой системы.

1.5 Функционал системы

Чтобы разработать ИС, соответствующую ожиданиям заказчика, необходимо составить целостное, системное представление будущей ИС, отобразить все аспекты функционирования.

Модель функционирования объекта ВКР можно представить при помощи контекстной диаграммы (Рис. 10).

Менеджмент компании не взаимодействует напрямую с системой, но пользуется аналитикой (дашбордами), построенными на основе данных системы.

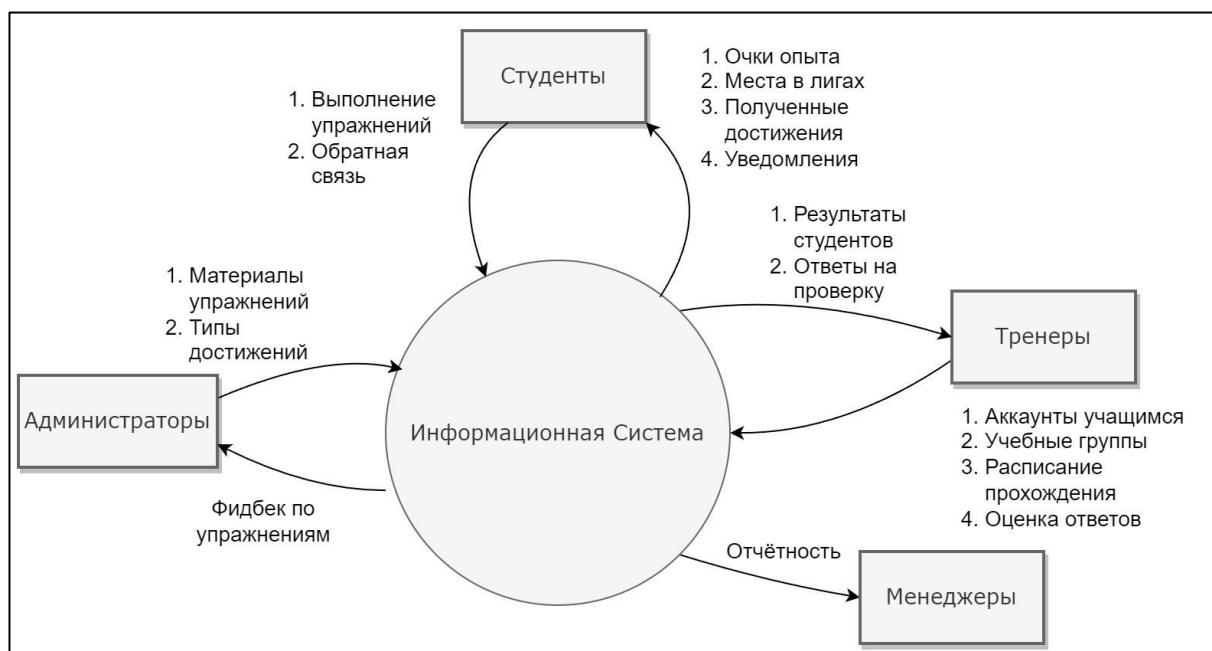


Рисунок 10 — Контекстная диаграмма системы корпоративного обучения

В разрабатываемом продукте выделены следующие роли: учащийся, тренер, администратор.

1.5.1 Функционал учащегося

Под ролью Учащийся подразумеваются сотрудники дилерских центров, направленные на курсы обучения в Центр Обучения BMW Group. Данная роль имеет следующий функционал:

1. Проходить упражнения, состоящие из шагов: теоретических (текстографические, видео), практических (тесты и задания с произвольным ответом).

Между упражнениями выстроена иерархия: от более ранних к более поздним.

Часть упражнений публичные – доступны всем пользователям системы без ограничения по времени. Такие упражнения полезны в целях Continuous Learning как сотрудников дилерских центров, так и сотрудников офиса BMW.

Другие же должны быть разблокированы тренером для прохождения конкретной учебной группой.

Для успешного выполнения упражнения пользователь должен пройти порог опыта, устанавливаемый тренером для каждой группы. Пользователь может проходить упражнения повторно, но количество попыток также ограничивается тренером. Тренер также может установить максимальный лимит опыта, при котором пользователю заблокировано повторное прохождение упражнения. Тренер ограничивает выполнение упражнения по срокам стартовой и конечной датой.

2. Конкурировать за места в лигах с одноклассниками. Просматривать таблицы рейтингов, обучающихся и рекордов.

К зачёту в лигах идёт весь опыт, набранный учащимися из группы за время, пока лига открыта. В конце заявленного периода Лиги закрываются автоматически и подводятся результаты – определяется победитель.

3. Получать достижения за прохождение определенных упражнений, победу в лигах или набор общего количества опыта.

4. Получать уведомления о приближающихся сроках закрытия упражнений, открытия новых лиг или упражнений.

5. Давать обратную связь по пройденным упражнениям.

1.5.2 Функционал тренера

Под ролью Тренер подразумеваются наставники и Тренеры Центра Обучения BMW Group.

Данная роль имеет следующий функционал:

1. Просмотр списков обучающихся и их успеваемости.
2. Установка сроков прохождения упражнений.

Тренер определяет расписание прохождения упражнений группами: сроки начала и окончания разрешенных периодов, минимальный опыт для успешного прохождения упражнения, число разрешенных попыток, верхний лимит опыта, ограничивающий повторное прохождение упражнений.

3. Оценка выполненных учащимися заданиями с произвольными ответами.

Тренер может оценивать результаты упражнений, заведённые учащимися из групп. В отличие от тестов, такие задания не оцениваются системой моментально и требуют ручной оценки задания тренером. К ответу могут прикрепляться файлы.

4. Открытие и закрытие Лиг, подведение итогов.

Лига – ограниченное по времени соревнование на сбор максимального количества опыта за выполнение упражнений. В одной лиге могут участвовать только пользователи одной группы.

Тренеры открывают Лиги с определенными датами начала и окончания. В этот временной промежуток учащиеся проходят упражнения для набора опыта. Участники ранжируются по убыванию Очков и выводятся на странице Лиги. Тренер может просматривать рейтинги Учащихся в Лиге в панели Администратора. После окончания Лиги подводятся итоги, назначается победитель лиги, ему выдаётся достижение (если предусмотрено).

5. Управление (создание, изменение и удаление) пользователями роли «Учащийся». Распределение пользователей типа «Учащийся» по группам.

6. Просмотр фидбека учащихся по упражнениям.

7. Выдача достижений вручную.

1.5.3 Функционал администратора

Под ролью администратор подразумеваются специалисты Центра Обучения BMW, осуществляющие проектирование методических материалов курсов. Данная роль имеет следующий функционал:

1. Доступный функционал роли «Тренер»

2. Управление содержанием упражнений.

Администратор может создавать, изменять и удалять упражнения. Каждое упражнение состоит из шагов разных типов. Эти шаги также создаются, редактируются и удаляются администраторами. Администратор устанавливает зависимость между упражнениями, указывая предыдущие

упражнения для редактируемого. Он определяет: является ли упражнение публичным или нет.

3. Просмотр, верификация, создание новых, удаление и редактирование пользователей.

Если тренер распределяет учащихся по группам, то администратор может их создавать, редактировать и удалять. Например, администратор может назначить новый пароль пользователю.

4. Управление достижениями.

Администратор создаёт, редактирует и удаляет достижения. Триггеры для выдачи достижений фиксированы в коде системы.

1.5.4 Моделирование информационной системы

Необходимо построить формальные модели для определения структуры и функционала проектируемой информационной системы.

В этом процессе входная информация – детали пользователя, данные о предыдущих попытках, данные о прохождении других упражнений.

Рассмотрим бизнес-процесс «Выполнить упражнение» после внедрения информационной системы. Для этого построим контекстную диаграмму TO BE в нотации IDEF0 (Рис. 11) и декомпозицию (Рис. 12). Процесс анализируется со стороны учащегося, которому необходимо пройти учебный курс в системе.

Как видно из диаграмм, разрабатываемая система должна автоматизировать процесс, исключив из него тренера. Лишь части задания, которые нельзя проверить автоматически, передаются тренеру на оценку.

Более детальные модели процесса отражены в виде BPMN диаграмм. На Рисунке 13— отражён процесс прохождения упражнения с текстовым и тестовым шагом, все шаги упражнения могут быть проверены автоматически. На рисунке 14 — упражнение состоит из единственного шага с произвольным ответом, который должен проверять человек – Тренер.

Объектно-ориентированная DFD модель позволяет детализировать проект системы и перейти от сценариев использования к отдельным объектам

и сущностям, задействованным в бизнес-функциях. Построим модель DFD в нотации Йордона для процесса выполнения упражнения (Рис. 15) и процесса настройки системы тренером (Рис. 16).

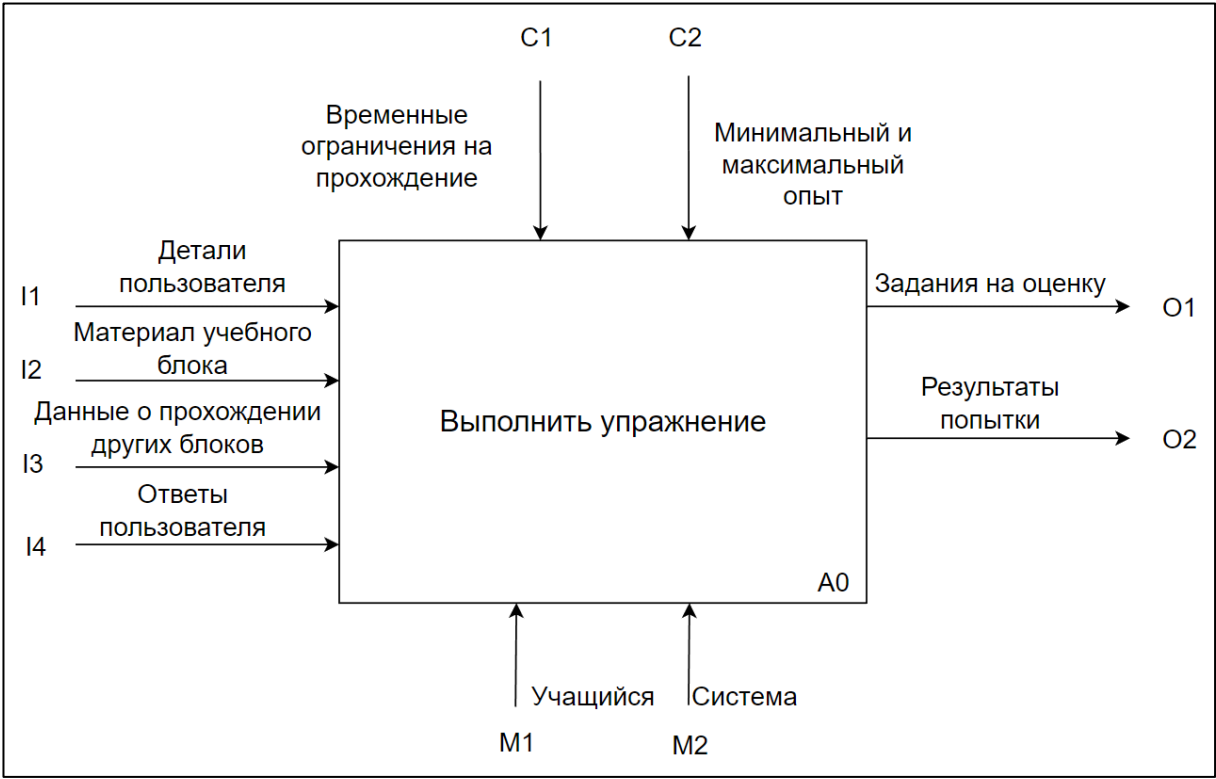


Рисунок 11 — IDEF0 диаграмма TO BE процесса «Выполнить упражнение»

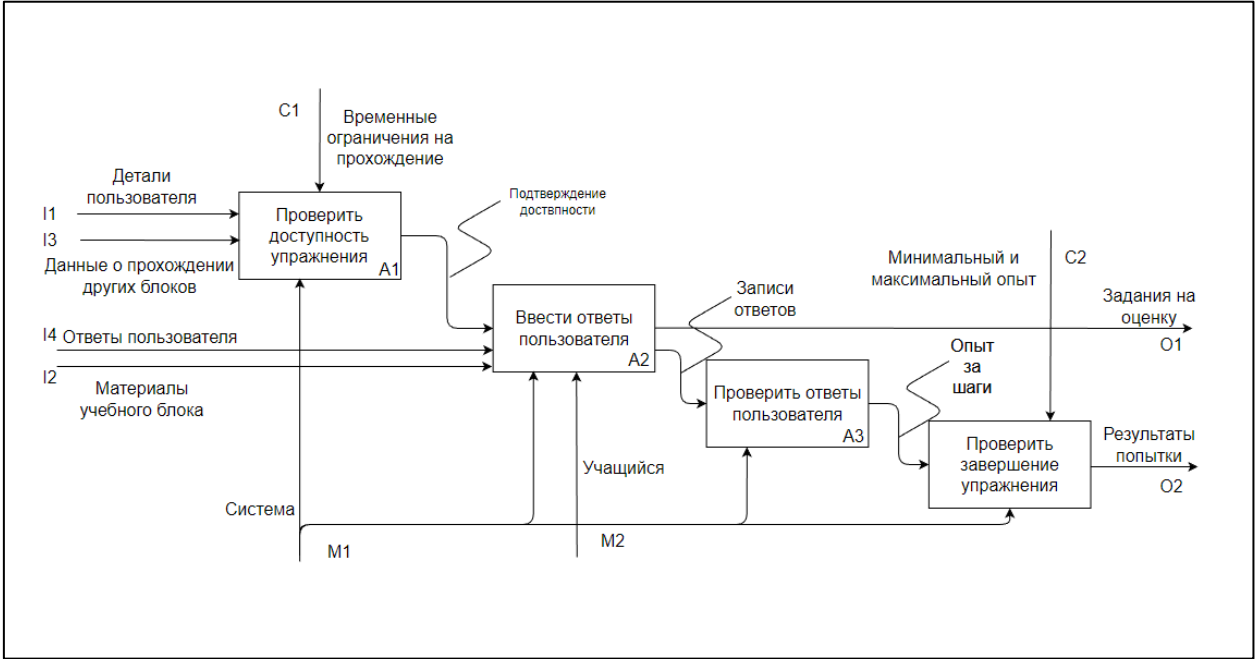


Рисунок 12 — Декомпозиция IDEF0 диаграммы TO BE процесса «Выполнить упражнение»

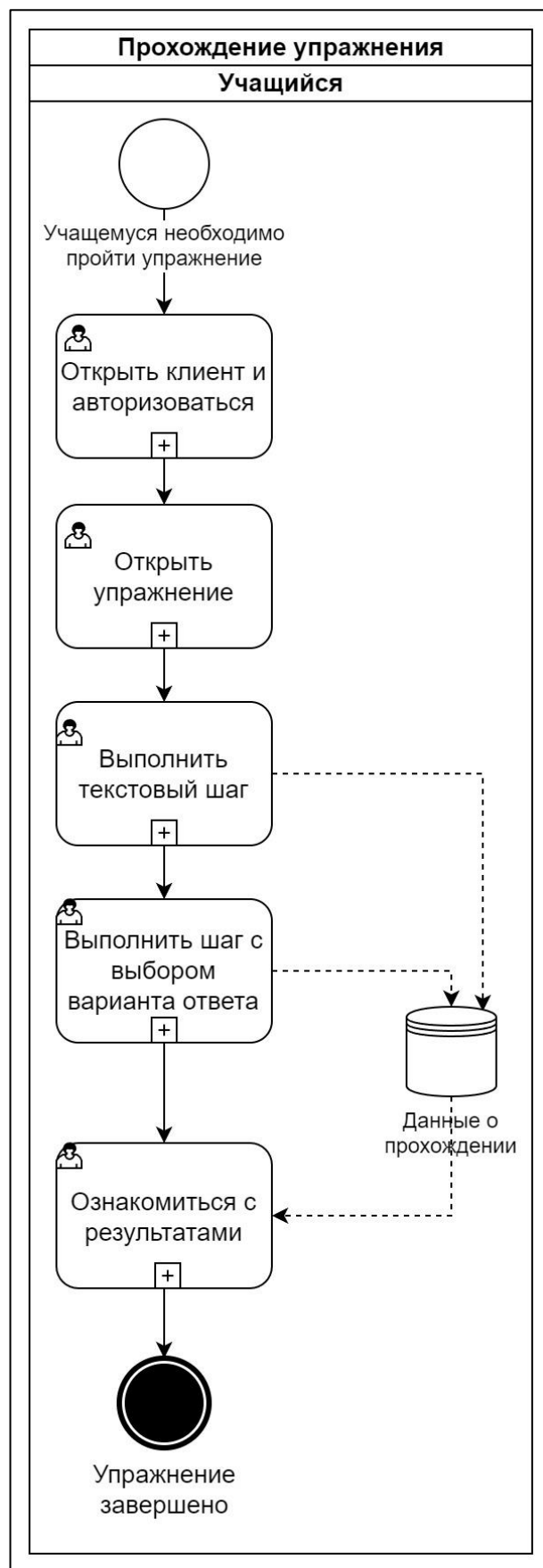


Рисунок 13 — BPMN диаграмма TO BE бизнес-процесса «Выполнить упражнение»

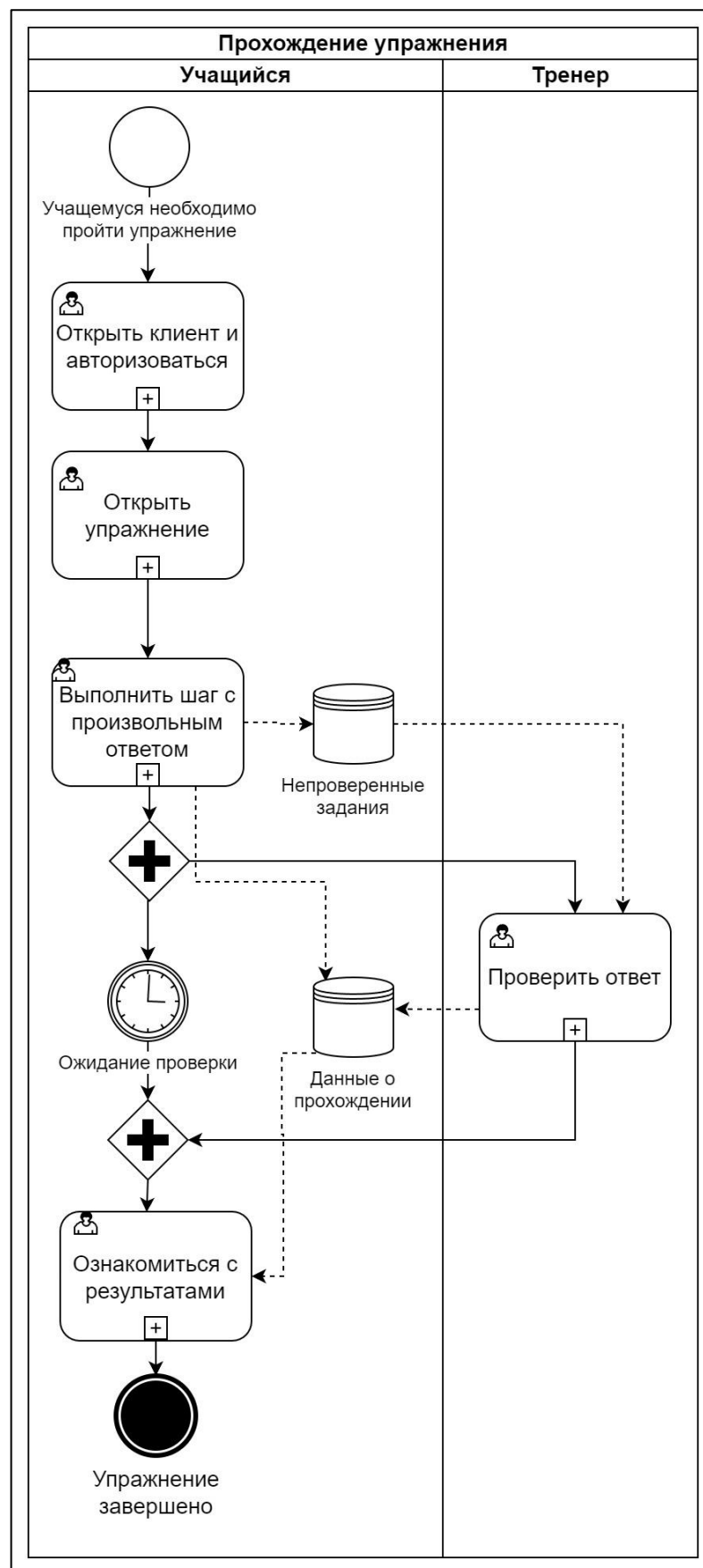


Рисунок 14 — BPMN диаграмма TO BE бизнес-процесса «Выполнить упражнение»

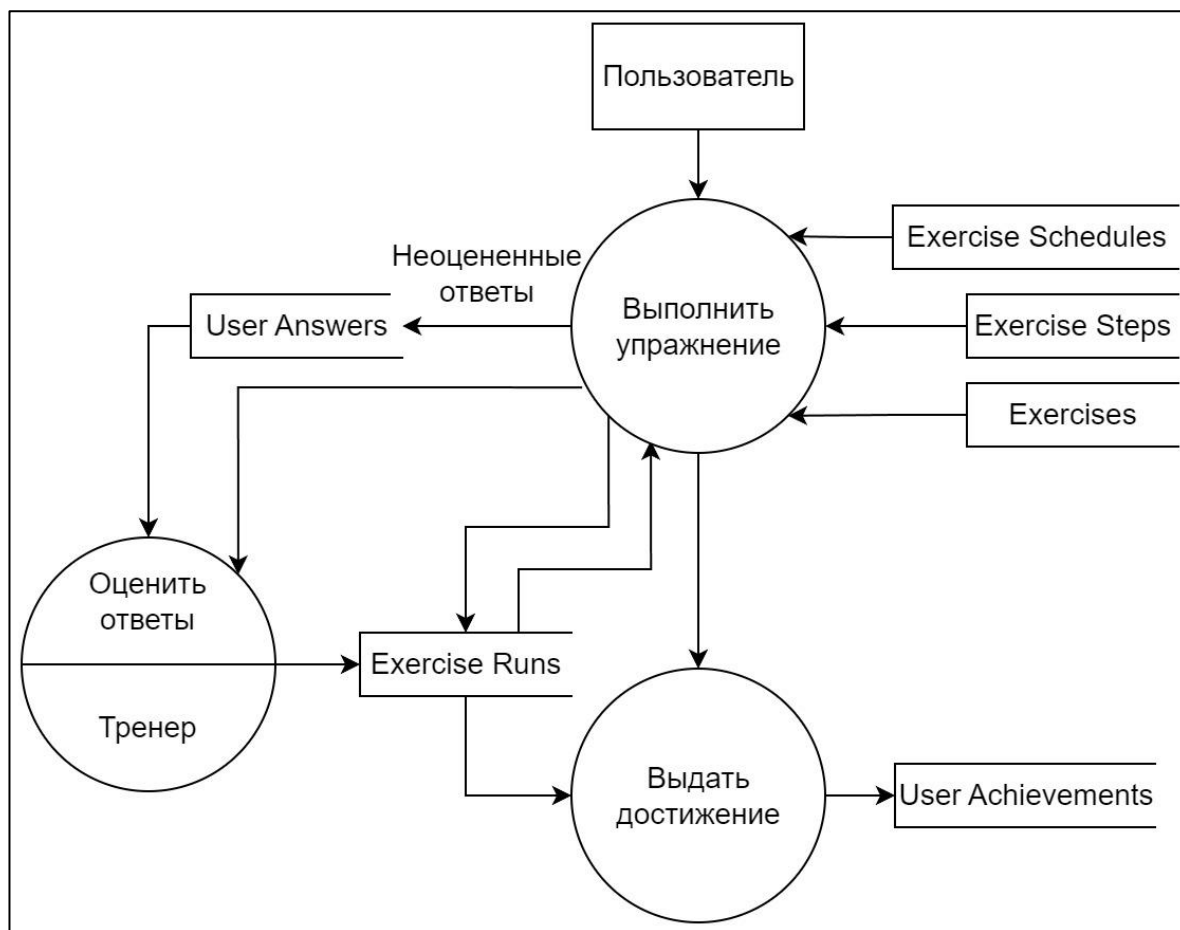


Рисунок 15 — DFD диаграмма процесса выполнения упражнения

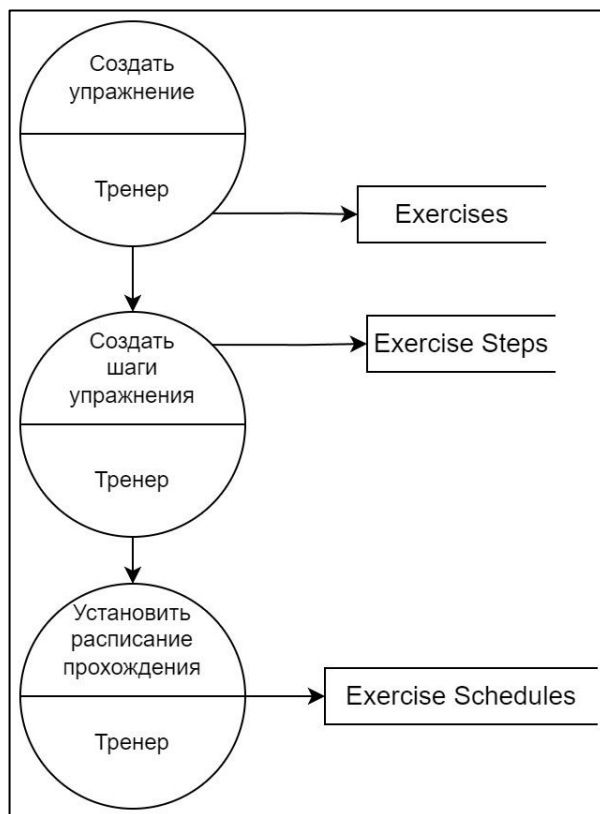


Рисунок 16 — DFD диаграмма процесса настройки системы тренером

1.5.5 Обзор функционала системы

Проанализировав предметную область, подобное ПО и современное состояние технологий для разработки веб-приложений, построив модели рассматриваемой системы в функциональном и объектном подходах, можно обозначить границы проекта, объем проекта, перечень планируемой функциональности.

Обзорное представление о планируемом функционале системы может дать ментальная карта функциональности системы (Рис. 17).

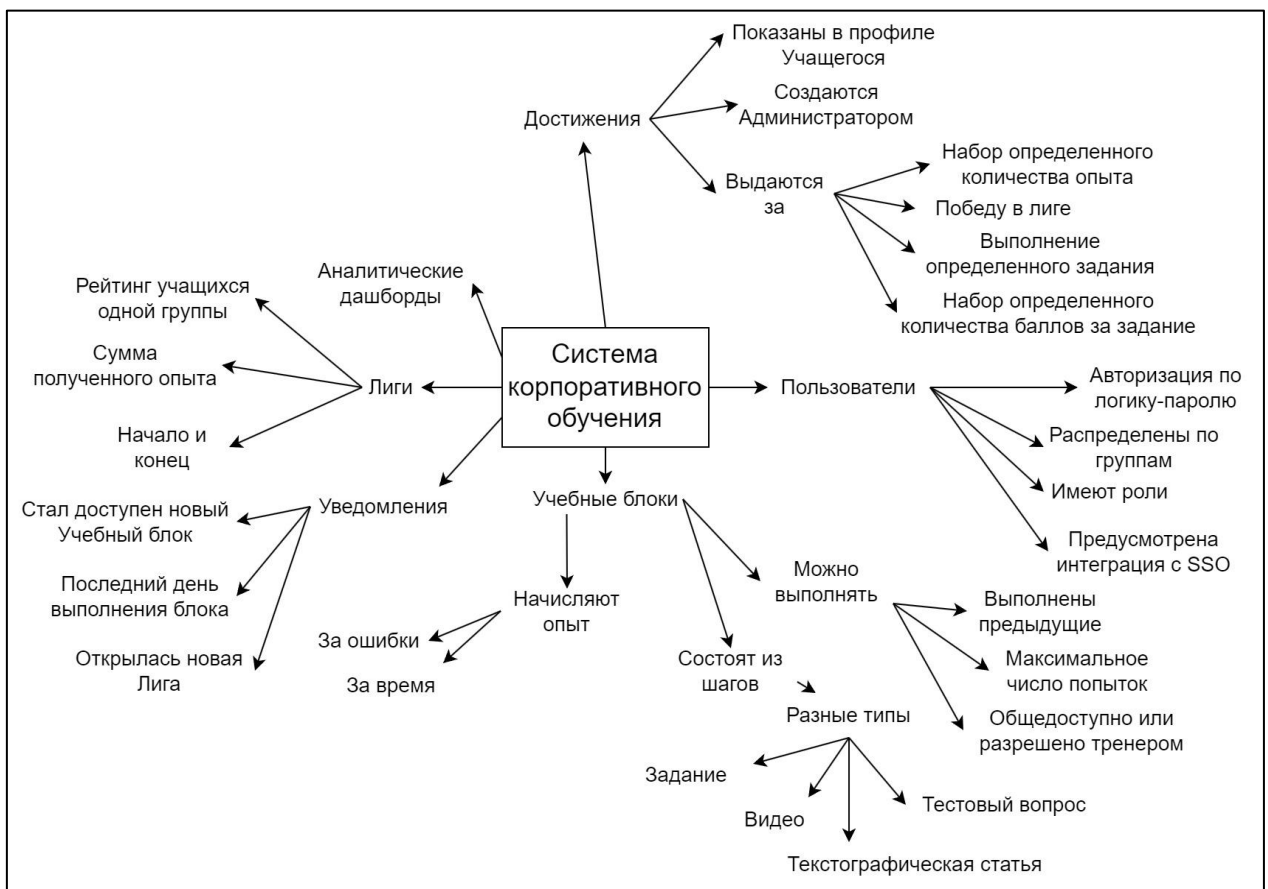


Рисунок 17 — Ментальная карта функциональности системы

Программный продукт должен реализовывать все бизнес-процессы системы:

- процессы авторизации;
- процессы, связанные с выполнением упражнений и оцениванием их выполнения;
- процессы, связанные с функционированием лиг и достижений;

- процессы, связанные с администрированием системы, созданием, изменением и удалением её сущностей;
- процессы, связанные с выводом различного рода информации о состоянии системы в других компонентах системы.

Говоря об увеличении вовлечения пользователей, мы неизбежно погружаемся в тему User Experience.

В индустрии для анализа UX системы применяется набор количественных метрик, за частью из которых необходимо следить регулярно. К ним относятся например:

1. System Usability Scale (SUS). После того, как пользователь совершит целевое действие (например, завершит упражнение) пользователя просят оценить опыт работы с системой численным показателем.

2. Индекс из 6 показателей Single Usability Metric (SUM). В нём анализируется, например, справился ли респондент с задачей, сколько потратил на это времени и сколько допустил ошибок. Все эти данные фиксируются в исторических таблицах БД при прохождении упражнения.

3. Net Promoter Score (NPS). Пользователя спрашивают: «С какой вероятностью вы порекомендуете этот продукт коллеге или другу?» по шкале от 0 до 10. NPS вычисляется как доля респондентов, давших оценку 9-10, за вычетом доли респондентов, давших оценку 0-6.

Из взаимодействия пользователя с системой происходит и набор User Centric метрик, таких как количество сессий на человека. Они измеряются на уровне продукта или компании в целом и позволяют оценить не только техническую реализацию системы, но и общую эффективность её внедрения.

Для дальнейшего вычисления этих метрик в целях анализа, мониторинга и усовершенствования системы, необходимо предусмотреть сбор фидбека пользователей в различных ключевых точках работы с системой.

При изменении состояния системы необходимо также сохранять изменения в исторические таблицы для дальнейшего анализа.

1.5.6 Предлагаемые технические решения

Чтобы обеспечить единое поведение и состояние системы, серверный модуль необходимо выделить как отдельный компонент. Разделение системы на отдельные программные компоненты, взаимодействующие между собой с помощью API позволит достигнуть легкого масштабирования, оптимизации нагрузки, ограничение ответственности частей информационной системы, быструю замену устаревших модулей, упрощенное обновление стека, и позволит дорабатывать программные продукты независимо друг от друга.

Так, разрабатываемая система состоит из 3 программных компонентов:

1. Портала, позволяющего проходить теоретические материалы и выполнять упражнения онлайн, агрегировать результаты этих упражнений, успеваемость учащихся в единой системе.

2. Панели администратора, позволяющей управлять сущностями системы, просматривать успеваемость учащихся.

3. Серверного модуля, обеспечивающего реализацию бизнес-процессов системы и интеграцию портала и панели администратора с единым источником правды.

Пользователи напрямую взаимодействуют с порталом и панелью администратора через веб-браузер.

Наиболее детальное описание системы представлено в виде Технического Задания (Приложение А).

Не менее важный аспект интеграции ИС в бизнес-процессы организации – отчётность. Руководство организации ожидает получать оперативную аналитику по выполнению KPI и достижению OKR Центром. Этот процесс может быть построен различными способами: через регулярные ETL скрипты для выгрузки данных из production СУБД в аналитическое хранилище (известное как Data Warehouse), через автоматизированную выгрузку необходимых данных в формате csv или json файлов, которые затем будут обработаны внешними скриптами, через запросы к API системы.

Для представления результатов применения системы в работе Центра Обучения BMW group будет предусмотрена возможность построения аналитических дашбордов. Эти дашборды будут обновляться автоматически на основе данных из СУБД системы. Необходимо, чтобы система сохраняла достаточный объем исторических данных для проведения анализа.

Лишь серверный модуль будет предоставлять данные для дашбордов.

1.6 Выводы

В первой главе работы по разработке серверной части веб-приложения для геймификации процесса обучения сотрудников дилерских центров BMW Group решались задачи анализа предметной области, аналогов и конкурентов, проектирования информационной системы.

В процессе решения поставленных задач использовались анализ академических источников, методы бизнес-анализа, контекстные диаграммы, IDEF0 диаграммы, BPMN диаграммы.

В итоге, обобщенные запросы заказчика к системе таковы:

- перевод материалов тренингов и упражнений в онлайн формат, с соблюдением при этом политики и процедуры организации;
- установка чётких сроков прохождения материалов и упражнений: сколько обучающийся в состоянии слушать одну тему, как часто ему необходимо менять вид активности в онлайн. В очном режиме это контролируется тренером, онлайн сложно отслеживать состояние аудитории;
- обработка предварительно созданного контента для построения единого образовательного трека.

В процессе работы над проектом было определено следующее:

- будут использоваться методы геймификации: лиги, опыт, достижения;
- система будет разделена на программные компоненты: Бэкенд (серверное приложение), панель администратора, Фронтенд (клиентское приложение);

- определены границы функционала ролей: пользователь упражнения может лишь выполнять, тренер устанавливает сроки и систему оценивания, а администраторы наполняют курсы контентом;

- для разработки системы будет применяться Waterfall подход, разработано техническое задание.

Были выполнены все поставленные задачи.

Для усовершенствования проекта системы в дальнейшем организации предлагается:

- провести анализ процессов AS IS и TO BE не только Центра Обучения, но и дилерских центров;

- доработать функционал уведомлений, внедрив, например, уведомления пользователей через Email или Telegram.

Глава 2 ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ

2.1 Обзор современных технологий и принципов для создания веб-сайтов

При разработке сайта ключевыми технологиям являются: HTML, CSS, JavaScript. Ни один современный сайт не может обойтись без них. Код на этих языках будет обрабатываться на стороне клиента в интернет-браузере.

Выбор технологий для серверной стороны же весьма широк. При желании можно реализовать собственный веб-сервер с нуля на C или C++, однако это целесообразно лишь в исключительных случаях по соображениям информационной безопасности или производительности.

В этом проекте резонно использовать более широко распространённые технологии, опираясь при выборе на факторы People, Process и Technology:

- People – насколько наша команда квалифицирована в этой технологии;
- Process – как это повлияет на итоговый результат;
- Technology – как выбранная технология соотносится с существующей инфраструктурой организации.

JavaScript – мультипарадигменный язык программирования, который поддерживает объектно-ориентированный, императивный и функциональный стили программирования.

С использованием TypeScript, JavaScript код становится компилируемым со статической типизацией, максимально близкой к C# или Java, чем не может похвастаться, например, Python. В то же время TypeScript допускает типовую гибкость JS, где это требуется. Функции могут возвращать разные типы в зависимости от параметров и окружения, типы возвращаемых значений таких функций могут проверяться при исполнении.

Node.JS – среда выполнения JavaScript, которая позволяет строить серверные приложения с использованием JavaScript. Она кроссплатформенна и легко масштабируется. Платформа построена на однопоточной событийно-

управляемой модели с неблокирующими операциями ввода-вывода, что делает ее эффективной для построения асинхронных приложений и легкой для разработчика.

Решающим фактором при выборе связки технологий Node.JS и TypeScript для серверной компоненты системы и панели администратора стал аспект People:

1. JavaScript – язык программирования, который в любом случае будет применяться в клиентской части системы.

2. JavaScript – крайне популярный язык программирования, найти новых разработчиков для поддержания системы будет сравнительно легко.

3. У обоих разработчиков был опыт использования JavaScript, использование этого ЯП поможет с взаимозаменяемостью разработчиков.

4. JavaScript можно развернуть как в Unix, так и Windows окружении, подключить к СУБД десятком способов и упаковать в Docker или Hybernates контейнер, что дает потрясающую гибкость.

5. У Node.JS богатая экосистема и сообщество, что поможет уменьшить стоимость разработки за счет использования готовых библиотек.

Аналогичным образом из клиентских SPA фреймворков был выбран React. Не имея значительных преимуществ или недостатков перед альтернативами – Vue.js и Angular – React является наиболее популярным из трёх, а разработчики с ним уже работали.

Для работы с данными была выбрана СУБД MySQL, широко используемая в отрасли. Она является универсальным решением среди реляционных БД для малых и средних приложений. Гибкость MySQL строится на поддержке двух систем хранения данных: MyISAM и InnoDB. MySQL распространяется по лицензии GNU General Public License, а потому будет всегда бесплатна для организации. Одновременно с этим прекращение поддержки MySQL в ближайшем будущем маловероятно, как часто случается с проприетарными или малопопулярными СУБД.

Серьезным конкурентом MySQL является PostgreSQL. Однако, высокая нагрузка на сервер при простейших операциях чтения плохо подходит для использования при разработке веб-приложения.

В процессе разработки тестирование проводилось с использованием SQLite в целях автоматизации отката состояния системы к состоянию по умолчанию при перезапуске сервера.

Для работы с СУБД используется библиотека Knex. Конструктор запросов в ней позволяет легко заменить СУБД на другую реляционную СУБД: PostgreSQL, CockroachDB, Amazon Redshift, MariaDB или MSSQL.

В процессе разработки использовалась система контроля версий git и удалённый репозиторий для хранения изменений на платформе Github.

2.2 Особенности реализации Бэкенда

2.1.1 Модель данных

В БД системы 16 таблиц:

1. Achievements – содержит типы получаемых достижений.
2. ReceivedAchievements – содержит достижения, полученные пользователями.
3. Groups – содержит учебные группы.
4. Notifications – содержит уведомления, полученные пользователями.
5. Feedback – содержит фидбек, предоставленный пользователями.
6. Users – содержит пользователей.
7. Leagues – содержит Лиги внутри учебной группы.
8. Answers – содержит произвольные ответы учащихся.
9. Exercises – содержит упражнения.
10. ExerciseSteps – содержит шаги упражнений.
11. ExerciseDependencies – содержит зависимости упражнений друг от друга.
12. ExerciseSchedules – содержит ограничения по прохождению, наложенные тренером.

13. Runs – содержит информацию о прохождении упражнений пользователями.

14. RunsHistory – историческая таблица содержит историю изменений записей в таблице Runs.

15. UserExperienceHistory – историческая таблица содержит историю изменения опыта пользователей.

16. Tokens – содержит действующие токены пользователей.

Для представления схемы БД, используемой системой, построена ER диаграмма (Рис. 18). На ней видны ключевые сущности и связи между ними. Была использована нотация Crow's Foot, сущность отображается в виде прямоугольника, а разные формы на конце стрелок обозначают связи между таблицами, все связи на схеме – один ко многим.

Все таблицы объединены несколькими свойствами: все первичные ключи отмечены Auto increment Primary Key id типа integer. В каждой сущности есть столбцы MIS_DT и UPDATED_DT типа integer – туда Бэкенд заносит время создания сущности и её обновления. Обработка этого механизма вынесена в специальный класс EntityFactory.

Все значения даты и времени вводятся в виде числа миллисекунд, прошедших с January 1, 1970, UTC. Это позволяет избавиться от множества проблем при работе с объектами дат конкретных СУБД. Для упрощения работы с датой и временем создан вспомогательный класс MIS_DT.

В целях нормализации БД до 1 нормальной формы, зависимость упражнений друг от друга (многие от многих) вынесена в отдельную таблицу ExerciseDependencies. Для пользователя панели администратора, предыдущие упражнения выведены как список id через запятую. Обработка происходит на уровне репозитория сущности Exercise.

Также таблица ExerciseDependencies позволяет работать с зависимостью упражнений как с деревом, например, при построении аналитики можно считать воронки конверсии для цепочки упражнений.

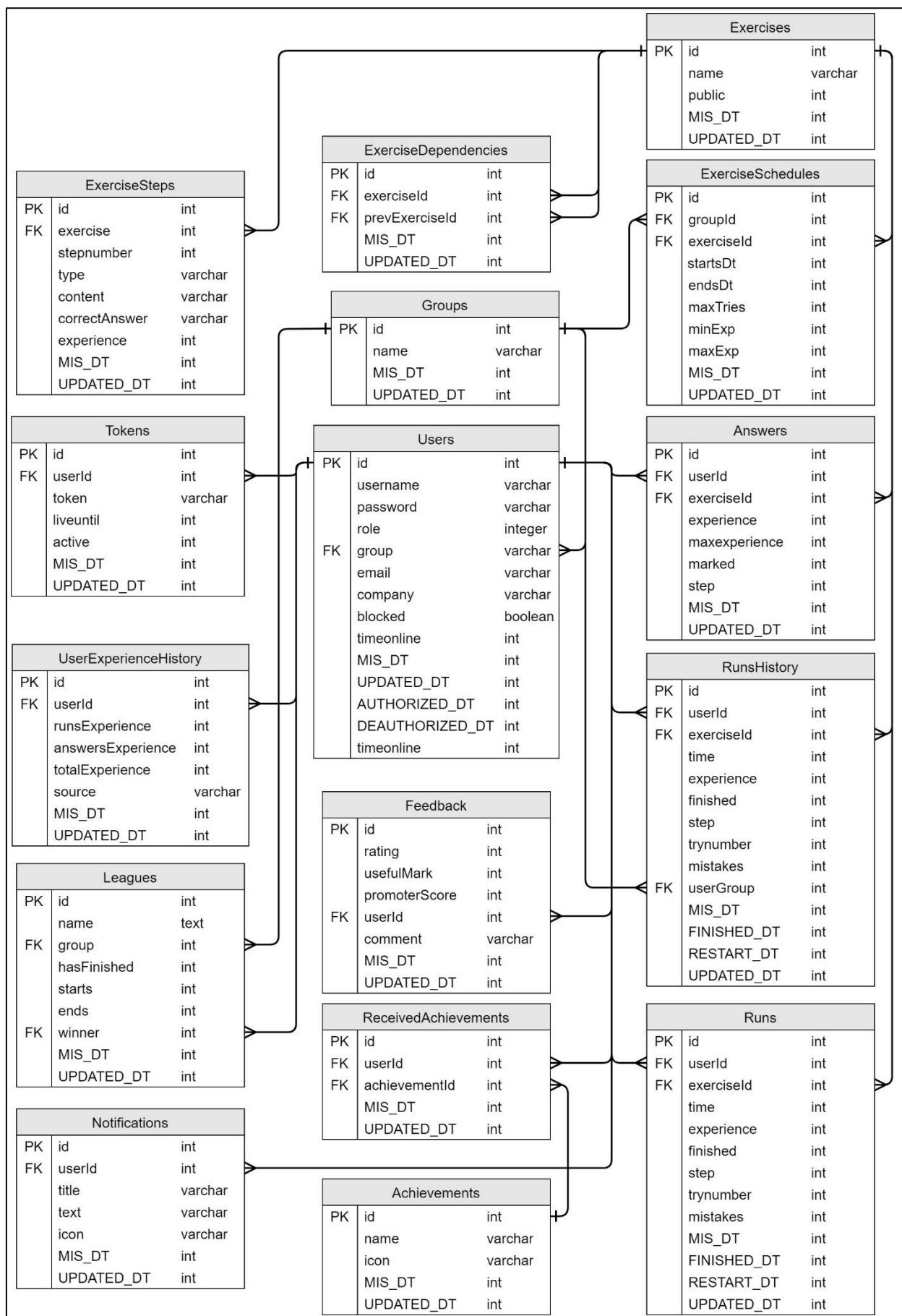


Рисунок 18 — Схема базы данных системы

Можно заметить, что в таблице ExerciseSteps столбец content представлен JSON объектом. В нём представлены данные, необходимые для вывода вопроса в клиентском приложении системы. Они не участвуют в механизмах Бэкенда. В то же время их состав зависит от типа шага. Так, для текстовых шагов или вопросов с произвольным вводом нет вариантов ответов. Чтобы эти данные можно было менять без доработки Бэкенда, было принято решение сохранить их в виде JSON объекта. Для обработки этого поля, в панели администратора разработан специальный редактор – он отображает и скрывает поля в зависимости от выбранного типа вопроса (раздел 2.4.3).

Схема БД поставляется вместе с исходным кодом программы в виде SQL файла для её воссоздания через запросы CREATE TABLE IF NOT EXISTS (файл DBSchema.sql).

2.1.2 Примененные паттерны и алгоритмы

Для создания веб-сервера применяется библиотека ExpressJS. ExpressJS – это библиотека для Node.js, которая предоставляет простой и гибкий способ создания веб-приложений и API. Она предоставляет множество функций для обработки HTTP-запросов и ответов, маршрутизации, обработки ошибок и многое другое. ExpressJS также поддерживает множество сторонних пакетов и плагинов, что делает ее еще более гибкой и мощной. Она является одной из самых популярных библиотек для создания веб-приложений на Node.js.

Для представления структуры ключевых классов Бэкенда построена UML диаграмма классов (Рис. 19). UML диаграмма позволяет описать классы, их атрибуты и методы, а также связи между классами. Данная диаграмма помогает разработчикам лучше понимать структуру системы и ее компонентов, а также облегчает коммуникацию между членами команды.

При реализации Бэкенда был применен паттерн Controller-Service-Repository. Такой паттерн широко распространён в разработке веб-приложений, например, он используется по умолчанию в Spring Boot или Laravel.

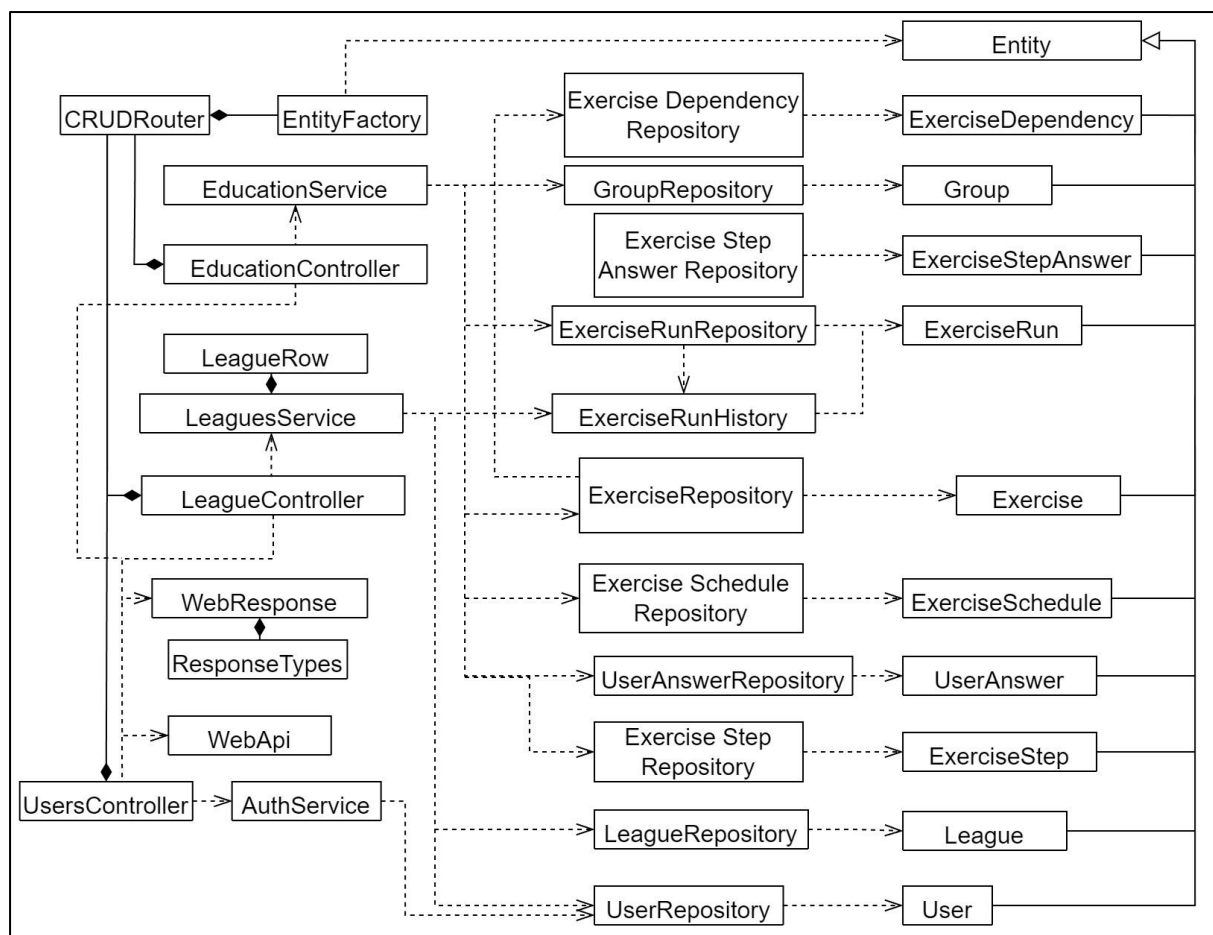


Рисунок 19 — Диаграмма основных классов Бэкенда

Паттерн состоит из уровней:

1. Controller – управляет REST интерфейсом для бизнес-логики.
2. Service – реализует бизнес-логику.
3. Repository – хранит сущности в системе.

В приложении уровни представлены следующим образом:

1. Controller – классы, такие как EducationController, LeagueController, UsersController. Композиционно используют роутеры класса CRUDRouter, которые обращаются к репозиториям для реализации базовых CRUD методов (раздел 2.3.2).

2. Service – статические классы как EducationService, LeaguesService и AuthService. Реализуют бизнес-логику приложения (пример на Рис. 20).

Repository – статические классы, наследующие от EntityFactory. Используют QueryBuilder к конкретной таблице или группе таблиц (объект

EducationService
<ul style="list-style-type: none"> + AdjustExperienceToTime(exercideld: number, experience: number, time: number) : number + GetTaskContent(userid: number, exerciseld: number) : WebResponse<IExerciseContent> + CanUserDoTask(userid: number, exerciseld: number) : WebResponse<boolean> + DidUserFinishTask(userid: number, exerciseld: number) : boolean + PassStep(userid: number, exerciseld: number, answer: string = "") : WebResponse + GetUserTotalExperience(userid: number): number + MarkAnswer(userid: number, exerciseld: number, stepno: number, experience: number) - EnsureRunObject(userid: number, exerciseld: number): ExerciseRun - StartTask(userid: number, exerciseld: number): ExerciseRun - RestartTask (run: ExerciseRun): ExerciseRun - FinalizeTask(run: ExerciseRun): WebResponse - CheckAnswer(userid: number, exerciseld: number, stepno: number, answer: string): WebResponse - CheckEducationScheduleOnStart(userid: number, exerciseld: number): WebResponse - CheckEducationScheduleOnRunSubmission(userid: number, exerciseld: number): WebResponse

Рисунок 20 — Описание класса EducationService

Connection) для работы с БД (select, update, insert, delete). Важно отметить, что не все классы репозитория напрямую соотносятся с таблицами в БД (Табл. 1).

Таблица 1 — Сопоставление репозитория и источников данных

Репозиторий	Источник данных
ExerciseDependencyRepository	ExerciseDependencies
ExerciseRepository	Exercises
ExerciseRunHistoryRepository	RunsHistory left join (select `id` as userId2, `group` as userGroup from Users) as a on a.userId2 = Runs.user
ExerciseRunRepository	Runs left join (select `id` as userId2, `group` as userGroup from Users) as a on a.userId2 = Runs.userId
ExerciseScheduleRepository	ExerciseSchedules
ExerciseStepAnswersRepository	ExerciseStepAnswers
ExerciseStepRepository	ExerciseSteps
GroupRepository	Groups
UserAnswerRepository	Answers
LeagueRepository	Leagues
FeedbackRepository	Feedback
AchievementRepository	Achievements
NotificationRepository	Notifications
ReceivedAchievementRepository	ReceivedAchievements
UserTokenRepository	Tokens
UserExperienceHistoryRepository	UserExperienceHistory

Приложение построено по транзакционной модели (stateless approach): в ответ на каждый запрос клиента происходит проверка данных, очистка,

обработка сервисами и взаимодействие с СУБД. Конкретный порядок зависит от типа запроса. Бизнес-сущности не хранятся в памяти приложения и не передаются между вызовами напрямую.

Для иллюстрации транзакционной модели приведена диаграмма последовательности запроса содержимого упражнения (Рис. 21).

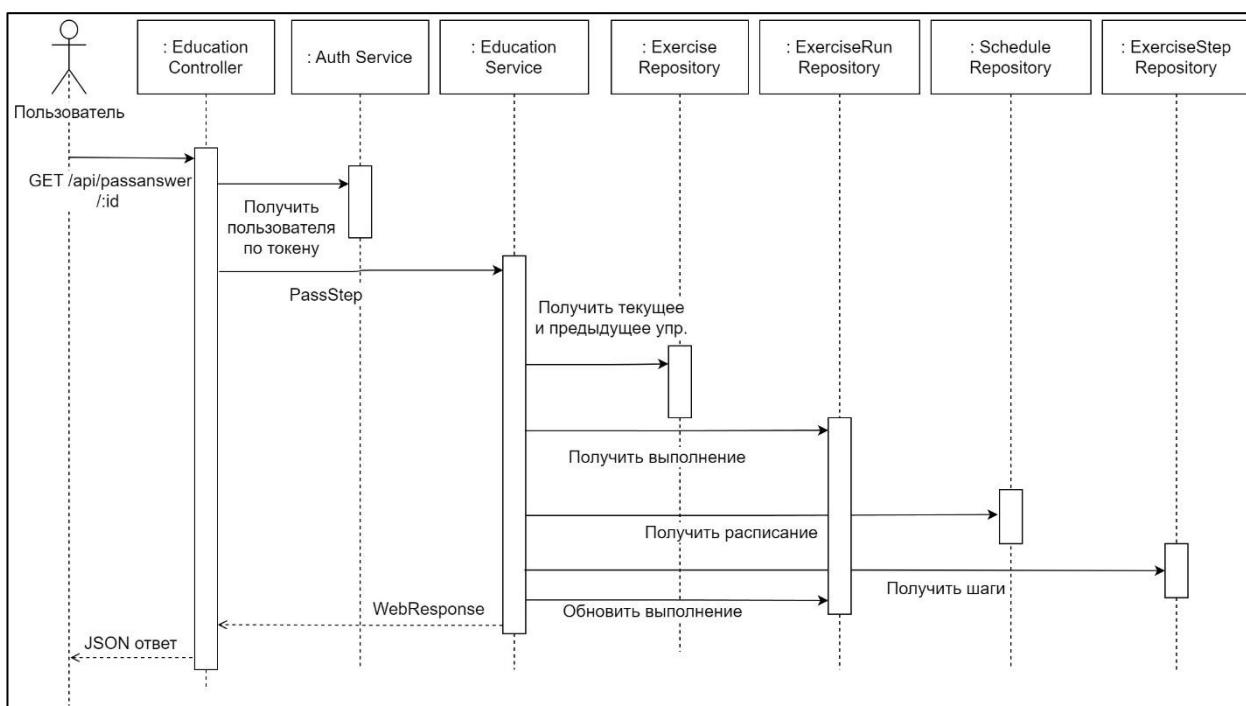


Рисунок 21 — Диаграмма последовательности получения упражнения

На схеме пользователем считается внешнее приложение, взаимодействующее через API с Бэкендом: панель администратора или Фронтенд. Для авторизации пользователей применяются токены. Токены генерируются случайно после авторизации пользователя по паре логин-пароль и действуют 15 минут с момента последнего обращения клиента.

Срок жизни токена хранится вместе с самим токеном и обновляется, когда контроллер или сервис получают объект пользователя по переданному с запросом токеном. Проверка сроков жизни токенов и удаление устаревших осуществляется по расписанию каждые 5 минут.

Токены пользователей тоже хранятся в БД. При горизонтальном масштабировании Бэкенда путём запуска копий приложения на одном или

разных серверах, клиентский токен действителен в каждом из процессов. При перезапуске Бэкенда пользователю не нужно повторно авторизовываться.

Для аутентификации пользователей используются md5 хеши паролей. Пароль находится в базе данных в зашифрованном виде. Токены пользователей также передаются клиентами в виде md5 хеша.

Авторизация пользователей производится за счёт проверки роли пользователя при обработке того или иного запроса.

Для сбора информации о результатах обработки запроса через все уровни приложения применяются объекты класса `WebResponse`. Они используются как «Value Object».

В нём методы `SetStatus` и `SetData` позволяют «нанизывать» новые данные и статусы на те, которые уже были добавлены глубже в приложении и возвращать объект вверх. Сам по себе объект `WebResponse` содержит: `boolean` атрибут `ok` – положительный и отрицательный статус вызова, `reason` – один из результатов выполнения в статическом классе `ResponseTypes`, `data` – любой передаваемый объект, `history` – массив `ResponseTypes`, куда добавляются новые результаты поверх старых. Сериализованный в JSON объект `WebResponse` возвращается на REST вызовы API для клиента.

Часть логики построена по событийной модели. Например, для выдачи достижений `EducationService` вызывает событие `OnTaskFinished`, которое прослушивает `AchievementsService`. Если было завершено одно из упражнений прописанных в коде (`id = 1` или `id = 6`), то пользователю выдается соответствующее достижение. Соответственно, хоть добавление новых типов достижений и возможно через панель администратора, для автоматической выдачи нового достижения необходима доработка программы.

Отдельного внимания заслуживает алгоритм обработки запросов, включающих в себя фильтрацию, сортировку и пагинацию (раздел 2.3.2).

2.2 Особенности реализации панели администратора

Панель администратора поставляется как отдельный программный компонент системы, взаимодействующий с Бэкендом через REST API.

Панель администратора реализована на языке TypeScript с использованием фреймворка React. Для разработки интерфейса использована библиотека готовых компонентов @mui. Компиляция производится при помощи vite.

Server-Side Rendering не используется, так как вопросы оптимизации не являются приоритетными для панели администратора. Зато панель администратора компилируется в набор статичных файлов HTML, CSS, JS, что позволяет при должной развернуть скомпилированную программу на любом веб-сервере (из популярных – Apache и nginx).

Коллекции сущностей системы, которые в Бэкенде реализованы репозиториями, здесь являются ресурсами (компонент типа Resource). Для каждого из них созданы подкомпоненты, позволяющие работать с сущностями конкретного типа:

1. List – вывод сущностей в табличной форме. В нём определяются выводимые столбцы, их форматирование, доступные с сущностями действия, сортировки, фильтрации.

2. Filter – параметры, которые можно применять для фильтрации сущностей на табличной форме.

3. Toolbar – доступные действия для сущности, которые зависят от роли.

4. Title – форматирование сущности до красивого уникального названия. По возможности должно заменять неинформативный числовой id.

5. Edit – форма редактирования сущности. В ней определяются выводимые, обязательные и скрываемые поля, типы редакторов, заблокированные от редактирования поля, уникальные редакторы, вывод зависимых сущностей.

6. Create – форма создания сущности. В ней определяются выводимые, обязательные и скрываемые поля, типы редакторов, заблокированные от редактирования поля, уникальные редакторы, вывод зависимых сущностей.

Для представления структуры зависимости основных компонентов Панели Администратора построена UML диаграмма классов для некоторых из ресурсов (Рис. 22). Эта диаграмма помогает отразить взаимосвязь между различными компонентами системы и облегчить решение проблем и создание новых функций и возможностей для пользователей.

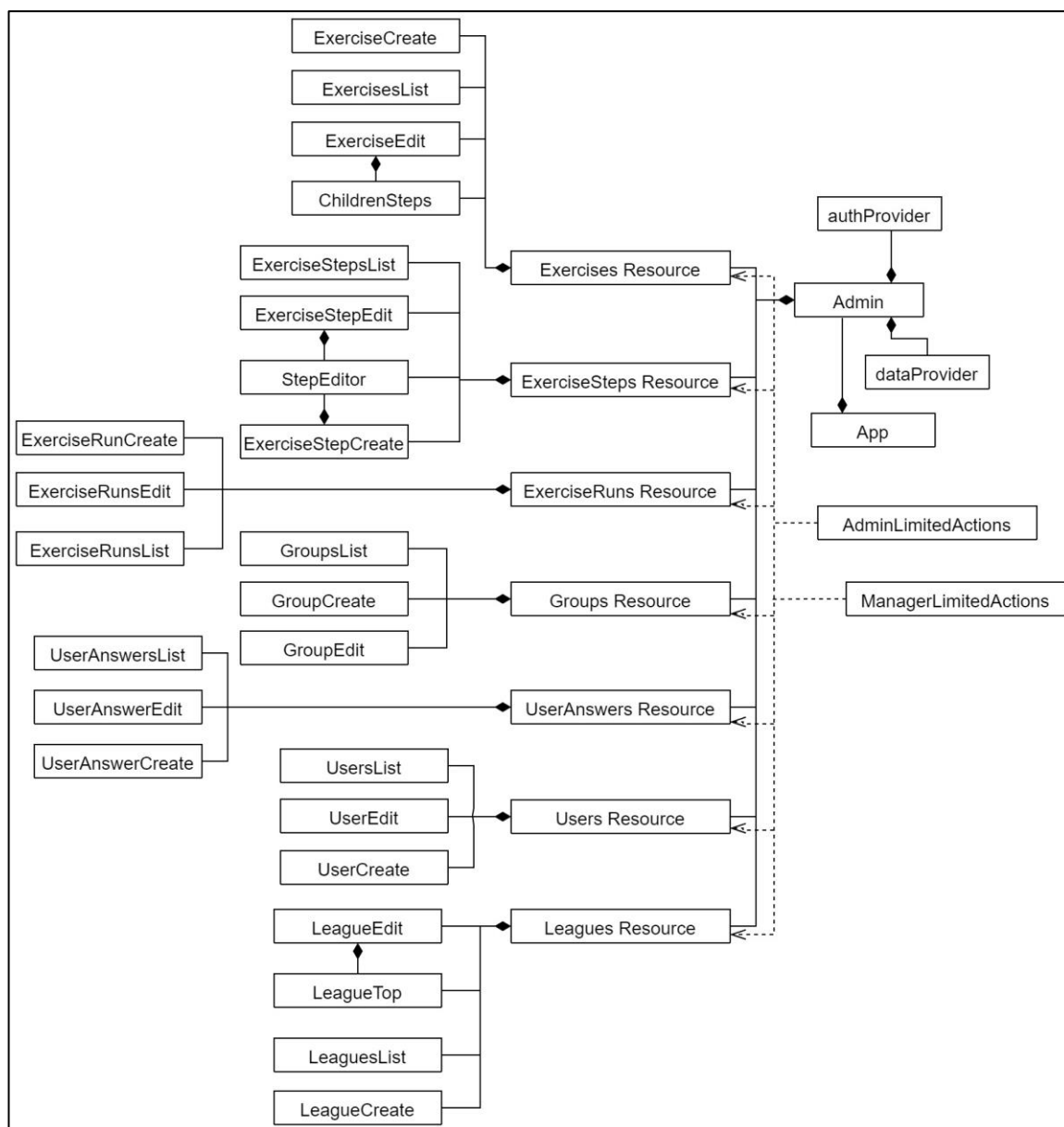


Рисунок 22 — Диаграмма основных классов панели администратора

В дополнение к этому можно выделить следующие уникальные компоненты:

1. Для Exercises, ChildrenSteps – список шагов, присвоенных этому упражнению. Доступен переход на страницу редактирования каждого из шагов и кнопка для создания нового (раздел 2.4.5).

2. Для ExerciseSteps, StepEditor – редактор содержимого шага упражнения. В нём выводимые пользователю поля меняются в зависимости от выбранного типа шага, а данные сериализуются и десериализуются из JSON для поля content шага.

Локализация панели администратора на русский и английский осуществляется по ключам.

Для части ключей заданы как единственное, так и множественное число. Переменные, такие как цифры, вставляются при помощи фигурных скобок на заданные позиции. Библиотека автоматически выбирает правильное множественное число в зависимости от числа сущностей: «1 лисичка, 2 лисички, 5 лисичек», и т.д. Адаптация падежных окончаний кроме случаев множественного числа не поддерживается. В интерфейсе переключатель языка находится в правой верхней части страницы.

Для упрощения механизма локализации применяется библиотека polyglot, разработанная Airbnb. Логика и строки локализации доступны в папке src/i18n. На русский язык переведено около 90% интерфейса. Для адаптивности страниц панели администратора на разных устройствах, на экранах списков сущностей технические столбцы, такие как дата создания или обновления сущности скрываются на малых разрешениях экрана. Данные представлены полностью на экранах с разрешением от 1440px в ширину. На экранах редактирования все параметры выводятся последовательно.

2.3 Особенности функционирования API Бэкенда

Для обеспечения совместной работы всех компонентов системы, был разработан контракт API.

Все запросы к API осуществляются через REST – с использованием JSON в теле запроса. Определение типа запроса ведётся на основании кода HTTP Request (GET, PUT, UPDATE и т.д.) и конечного адреса (endpoint).

CORS запросы разрешены.

2.3.1 API авторизации

С каждым запросом к API должен поставляться временный токен пользователя, полученный путём авторизации (Табл. 2). Для GET запросов токен передаётся в HTTP Query string, для POST/PUT/UPDATE и т.д. – в body.

Таблица 2— Запросы, связанные с процессом авторизации

Запрос	Данные	Ответ
Авторизация GET /api/auth	{ login: string, password: string }	{ reason: { id: number, text: string }, data:{ token: string undefined } }

Token возвращается в виде md5 хэша. Система ожидает получить токен в таком же виде на вход. Пароль пользователя ожидается получить также в виде md5 хэша. Бэкенд допускает передачу пароля в raw виде, но это небезопасно. Пример исходящего запроса авторизации приведен на Рисунке 23, пример ответа — на Рисунке 24.

```
1 {
2   ... "login": "test83",
3   ... "password": "3b712de48137572f3849aabd5666a4e3"
4 }
```

Рисунок 23 — Пример запроса авторизации

```
1 {
2   "reason": {
3     "id": 20,
4     "text": "OK"
5   },
6   "data": {
7     "token": "0b8854ad38f0a6c65807928d28195609"
8   },
9   "history": [
10    {
11      "id": 20,
12      "text": "OK"
13    }
14  ],
15   "ok": true
16 }
```

Рисунок 24 — Пример ответа на запрос авторизации

2.3.2 API сущностей

Для базовых запросов к сущностям используется единый стандартизированный CRUD API по каждой коллекции (Табл. 3). Эти запросы используются преимущественно в панели администратора. Для запросов `getList` вместе с ответом отправляется заголовок `X-Total-Count`, отражающий общее число сущностей в коллекции, что необходимо для пагинации.

В системе представлены следующие коллекции, доступные с этим API:

- `/api/exercises;`
- `/api/steps;`
- `/api/runs;`
- `/api/schedules;`
- `/api/groups;`
- `/api/answers;`
- `/api/userexperiencehistory;`
- `/api/exerciserunhistory;`
- `/api/users;`
- `/api/leagues;`
- `/api/achievements;`
- `/api/receivedachievements;`
- `/api/notifications;`
- `/api/feedback.`

Таблица 3 — Схема API сущностей

Действие	Запрос к API
<code>getList</code>	GET http://my.api.url/posts?_sort=title&_order=ASC&_start=0&_end=24&title=bar
<code>getOne</code>	GET http://my.api.url/posts/123
<code>getMany</code>	GET http://my.api.url/posts?id=123&id=456&id=789
<code>create</code>	POST http://my.api.url/posts
<code>update</code>	PUT http://my.api.url/posts/123
<code>delete</code>	DELETE http://my.api.url/posts/123

Для построения пагинации на клиентской стороне вместе с ответами передаётся заголовок Content-Range вида «posts 0-24/319». Тогда, панель администратора может понять какие индексы записей были получены и сколько записей всего в коллекции.

Благодаря применению стандартизированного API, грамотно выстроенной абстракции сущностей и использованию готовых библиотек, вся обработка фильтров, сортировок и пагинаций заняла менее 100 строк (Рис. 25). Для поиска неполных строк используется SQL оператор LIKE.

```
export function ConvertAdminQuery(input: any, query: knex.QueryBuilder) {
  if (input.range) {
    const range = JSON.parse(input.range);
    if (range.length === 2) {
      const amount = range[1] - range[0] + 1;
      query.offset(range[0]);
      query.limit(amount);
    }
  }
  if (input.filter) {
    const obj = JSON.parse(input.filter);

    if (Object.keys(obj).length > 0) {
      const filters = Object.entries(obj);

      for (const filter of filters) {
        const key = filter[0][0].toUpperCase() + (filter[0] as string).substring(1);

        if (isArray(filter[1])) {
          for (let value of filter[1]) {
            value = valueCleanup(value);
            query = query.orWhere(key, value);
          }
        } else {
          const value = valueCleanup(filter[1]);
          query = query.andWhere(key, "LIKE", "%" + value + "%");
        }
      }
    }
  }
  if (input.sort) {
    const sort = JSON.parse(input.sort);
    const sortvalue = sort[0][0].toUpperCase() + (sort[0] as string).substring(1);

    query.orderBy(sortvalue, sort[1]);
  }

  return query;
}
```

Рисунок 25 — Алгоритм обработки сложных запросов

Для примера, запрос GET «/api/posts?sort=["title","ASC"]&range=[0,24]&filter={"title":"bar", author: "foo"}» будет преобразован в SQL запрос на Рисунке 26.

```
SELECT * from Posts
where title = "bar" and author = "foo"
order by title ASC
limit 25 offset 0
```

Рисунок 26 — Пример типового запроса на выборку элементов

2.3.3 API процессов

Часть запросов не укладывается в стандартный API сущностей. Для них разработаны endpoints и соглашения по использованию API (Табл. 4).

Таблица 4 — Схема API процессов

Действие	Запрос к API	Ответ
Топ пользователей лиги	GET /api/leaguestop/:id	[{userId: number, username: string, experience: number, lastFinishedDate: number}]
Получить содержимое текущего шага упражнения для пользователя	GET /api/getcontent/:id	{ reason:{ id:number, text:string }, data:{ text:string, answers:string[] } }
Отправить ответ на упражнение	POST /api/passanswer/:id body: { answer: string }	{ reason:{ id:number, text:string } }

2.4 Особенности функционирования панели администратора

2.4.1 Material Design

Интерфейс панели администратора построен с использованием Material Design 2.

Material Design – это дизайн-система, разработанная компанией Google в 2014 году для унификации интерфейсов всех продуктов и сервисов

корпорации, включая приложения Android. Теперь сайты и приложения компании воспринимаются как единая цифровая экосистема, что обеспечивает проникновение сервисов корпорации в жизнь пользователей [35].

Основные элементы Material Design:

1. Глубина теней – иерархия элементов, расположенных в «3D» друг над другом за счёт отбрасывания тени. Чем выше подъем объекта, тем больше его тень.

2. Использование средств и подходов полиграфического дизайна в интерфейсах ПО.

3. Встраивание анимации в логику системы.

4. Адаптивный дизайн. Интерфейс должен отображаться корректно на всех разрешениях и типах устройств.

5. Контрастная типографика между заголовками и наборным текстом.

6. Модульная сетка создает отступы и структурирует информацию.

7. Яркие цвета. В палитрах Material Design определяют основной цвет и акцентный для выделения и привлечения внимания пользователя к ключевым деталям.

8. Реактивная анимация. Анимация возникает только в ответ на действия пользователя.

9. Переход от общего к частному. С уменьшением размера экрана должно уменьшаться количество информации на экране. Например, на больших экранах можно выводить большие таблицы, а на малых преобразовать их в списки из карточек.

К преимуществам Material Design можно отнести:

1. Демонстрацию взаимосвязей элементов.

2. Фокусировку внимания.

3. Выразительность.

4. Интуитивность интерфейса.

5. Привлекательность интерфейсов.

В качестве шрифта панели администратора был выбран Roboto, являющийся шрифтом по умолчанию для ОС Android, а потому он привычен пользователям. Он нейтрален и хорошо вписывается в Material Design (Рис. 27).

В дизайне активно применяются иконки. На светло-сером фоне (#fafa fb) основное содержимое выделено белым фоном и нависающей тенью. Для верхней панели навигации выбран светло-голубой (#2196f3). Интерактивные элементы — ссылки, кнопки, переключатели — выделены голубым (#1976d2).

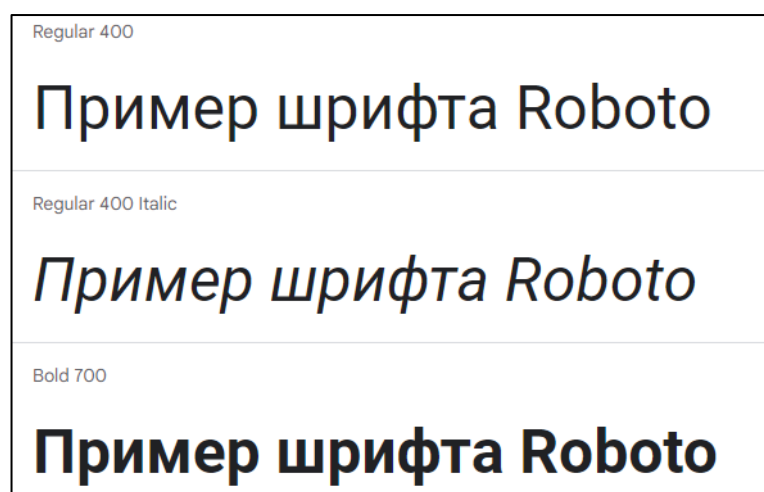


Рисунок 27 — Примеры шрифта Roboto

2.4.2 Интерфейс панели администратора

Вход осуществляется по паре логин-пароль. Внешний вид формы авторизации представлен на Рисунке 29.

Интерфейс панели администратора включает:

- блок основного содержимого;
- левое меню со списком коллекций (Рис. 30);
- в правом углу — имя пользователя, кнопку выхода и кнопку «Обновить» (Рис. 28).

После входа пользователь оказывается на экране первой из коллекций (Рис. 31) На Рисунке 32 представлен экран коллекции «Ответы для проверки», на Рисунке 33 — коллекции «Выполнения».

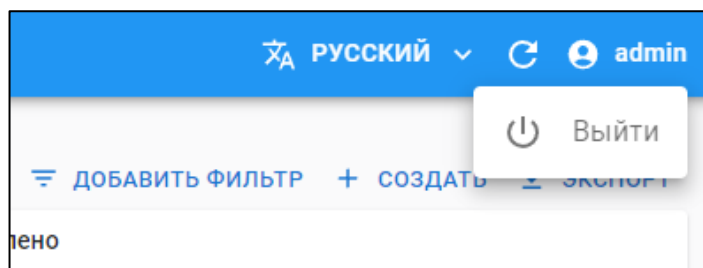


Рисунок 28 — Кнопка выхода из аккаунта

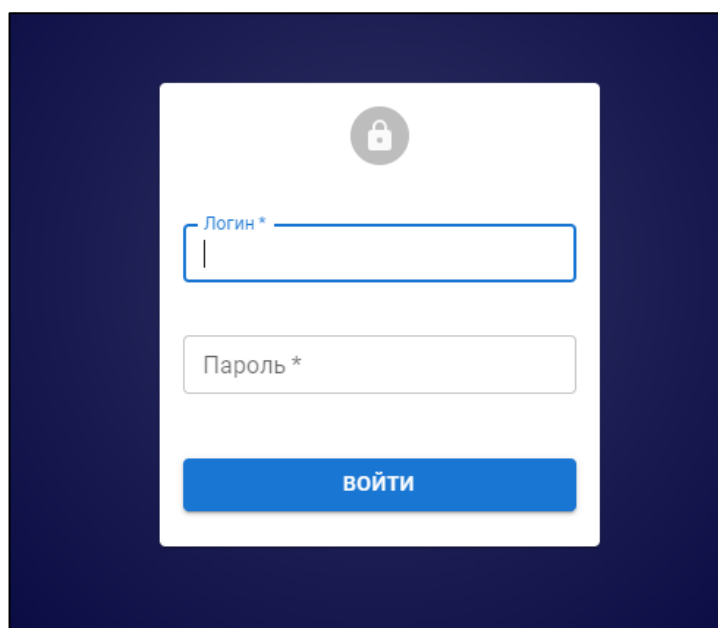


Рисунок 29 — Форма авторизации

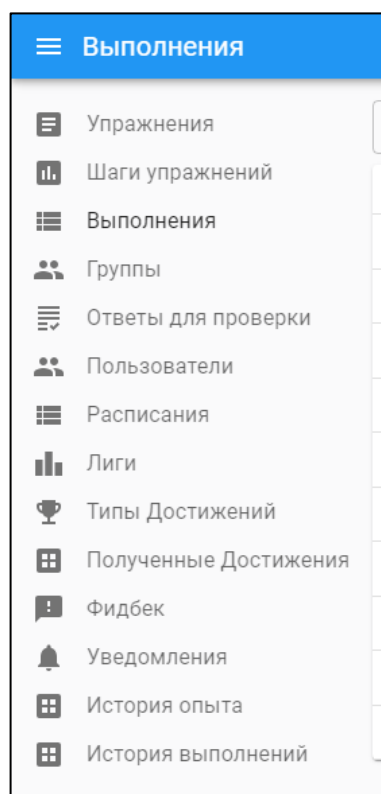


Рисунок 30 — Левое навигационное меню

Название

☰

ДОБАВИТЬ ФИЛЬТР

+

СОЗДАТЬ

⬇

ЭКСПОРТ

<input type="checkbox"/>	Id ↑	Название	Открыто	Создано	Обновлено	
<input type="checkbox"/>	1	first	✕	22.02.2023	22.02.2023	<div><div>✎</div>ИЗМЕНИТЬ</div>
<input type="checkbox"/>	2	test2	✓	22.02.2023	03.04.2023	<div><div>✎</div>ИЗМЕНИТЬ</div>
<input type="checkbox"/>	3	multipledependencies	✓	22.02.2023	03.04.2023	<div><div>✎</div>ИЗМЕНИТЬ</div>
<input type="checkbox"/>	4	4th	✕	22.02.2023	22.02.2023	<div><div>✎</div>ИЗМЕНИТЬ</div>
<input type="checkbox"/>	5	testExerciseCreation	✕	03.05.2023	03.05.2023	<div><div>✎</div>ИЗМЕНИТЬ</div>
<input type="checkbox"/>	6	history	✕	23.05.2023	23.05.2023	<div><div>✎</div>ИЗМЕНИТЬ</div>
<input type="checkbox"/>	7	history2	✕	23.05.2023	23.05.2023	<div><div>✎</div>ИЗМЕНИТЬ</div>
<input type="checkbox"/>	8	history3	✕	23.05.2023	23.05.2023	<div><div>✎</div>ИЗМЕНИТЬ</div>
<input type="checkbox"/>	9	history4	✕	23.05.2023	23.05.2023	<div><div>✎</div>ИЗМЕНИТЬ</div>

Строк на странице:

10

▼

1-9 из 9

Рисунок 31 — Экран списка сущностей коллекции «Упражнения»

<input type="checkbox"/>	Id ↑	Упражнение	Пользователь	Опыт	Проверено?	Устарело?	Номер шага	
<input type="checkbox"/>	1	first	test83	5	✓	✓	2	изменить
<input type="checkbox"/>	2	first	test12	0	×	×	2	изменить
<input type="checkbox"/>	3	first	test15	5	✓	×	2	изменить
<input type="checkbox"/>	4	first	test13	5	✓	✓	2	изменить
<input type="checkbox"/>	5	first	test13	5	✓	✓	2	изменить
<input type="checkbox"/>	6	first	test83	0	×	✓	2	изменить

Рисунок 32 — Табличная часть коллекции «Ответы для проверки»

<input type="checkbox"/>	Id ↑	Упражнение	Учебная группа	Пользователь	Опыт	Закончено?	
<input type="checkbox"/>	1	first	test group	test83	15	✓	изменить
<input type="checkbox"/>	2	test2	test group	test83	0	×	изменить
<input type="checkbox"/>	3	test2	test group	test83	5	✓	изменить
<input type="checkbox"/>	4	first	test group	test12	15	✓	изменить
<input type="checkbox"/>	5	first	test group	test15	4	✓	изменить
<input type="checkbox"/>	6	test2	test group	test83	5	×	изменить
<input type="checkbox"/>	7	test2	test group	test83	5	✓	изменить
<input type="checkbox"/>	8	4th	test group	user75	0	×	изменить
<input type="checkbox"/>	9	4th	test group	test94	0	×	изменить

Рисунок 33 — Табличная часть коллекции «Выполнения»

2.4.3 Стандартные коллекции сущностей

Работа с панелью администратора построена вокруг коллекций сущностей Системы, которые можно просматривать, создавать, редактировать и удалять.

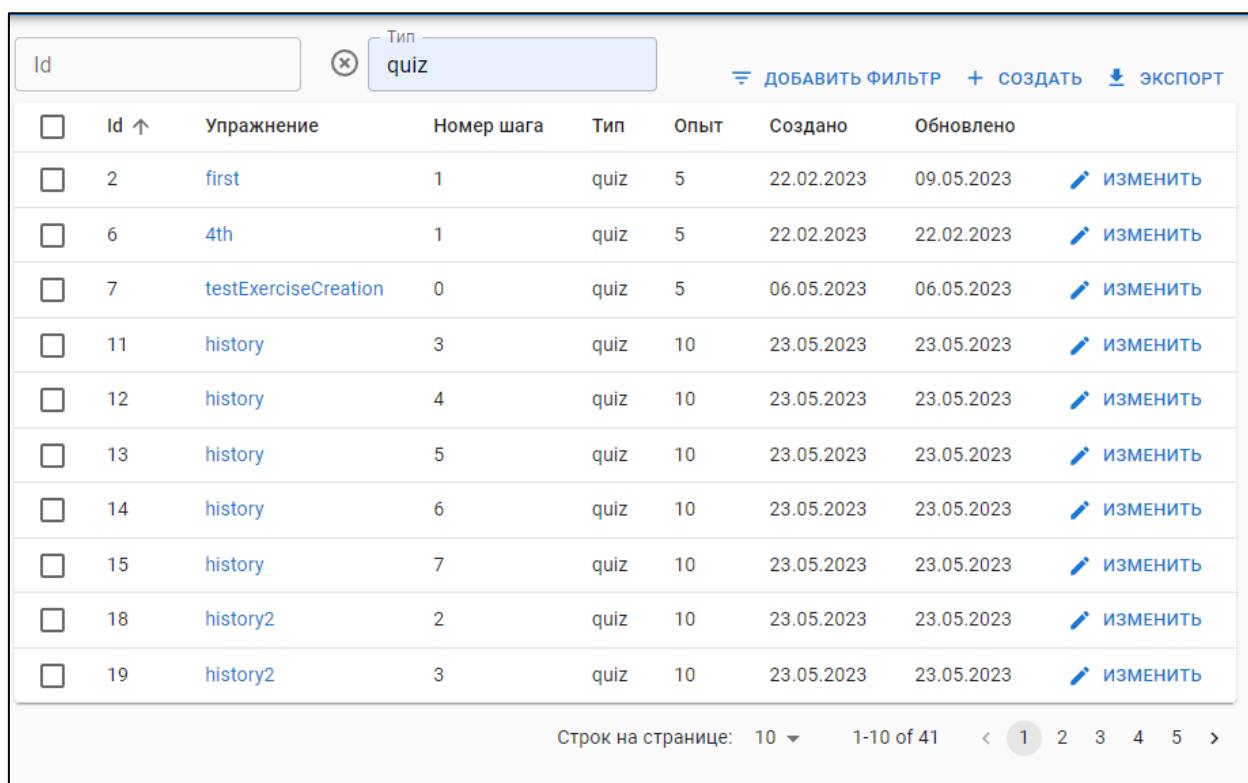
Таковыми сущностями являются:

1. Упражнения (Exercises) – основные учебные единицы в системе.
2. Шаги упражнений (Steps) – шаги, из которых состоят упражнения.
3. Выполнения (Runs) – выполнения упражнений учащимися, как текущие, так и уже завершённые.
4. Группы (Groups) – группы учащихся.
5. Ответы для проверки (Answers) – ответы на упражнения, требующие ручной проверки.
6. Пользователи (Users) – пользователи.
7. Лиги (Leagues) – текущие и завершённые лиги.
8. Расписания (Schedules) – установленные тренерами расписания прохождения упражнений для групп.
9. Типы достижений (Achievements) – достижения, которые могут быть выданы пользователю системой.
10. Полученные достижения (Received Achievements) – уже полученные пользователями достижения.
11. Фидбек (Feedbacks) – обратная связь учащихся на упражнения.
12. Уведомления (Notifications) – уведомления, полученные пользователями.
13. История опыта (User Experience History) – история начисления опыта пользователям.
14. История выполнений (Exercise Runs History) – история прохождения упражнений пользователями.

Для всех сущностей доступны:

- Динамическая сортировка по столбцам – при клике по названию столбца таблица будет отсортирована сначала по возрастанию, затем по убыванию этого атрибута.
- Фильтрация и поиск по выбранным атрибутам – часть сортировок всегда видима на экране, часть скрыта под кнопкой «Добавить фильтр».
- Пагинация – переключение страниц находится в правом нижнем углу таблицы, доступ переход как на следующую/предыдущую страницу, так и на произвольную по номеру, доступен выбор сколько строк отображать на странице.

Все эти операции выполняются на серверной стороне (Рис. 34) Так, панель загружает лишь небольшую часть сущностей из самой коллекции и связанных с ней коллекций, и достигается максимальная скорость работы приложения.



The screenshot shows a web application interface for managing exercises. At the top, there is a search bar with the text 'Id' and a filter button labeled 'Тип: quiz'. To the right of the filter button are three buttons: 'ДОБАВИТЬ ФИЛЬТР', '+ СОЗДАТЬ', and 'ЭКСПОРТ'. Below the search bar is a table with columns: 'Id', 'Упражнение', 'Номер шага', 'Тип', 'Опыт', 'Создано', 'Обновлено', and an action column. The table contains 10 rows of data. The first row is highlighted. The action column contains a pencil icon and the text 'ИЗМЕНИТЬ' for each row. At the bottom of the table, there is a pagination bar showing 'Строк на странице: 10', '1-10 of 41', and a series of page numbers: 1, 2, 3, 4, 5, with '1' being the active page.

<input type="checkbox"/>	Id ↑	Упражнение	Номер шага	Тип	Опыт	Создано	Обновлено	
<input type="checkbox"/>	2	first	1	quiz	5	22.02.2023	09.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	6	4th	1	quiz	5	22.02.2023	22.02.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	7	testExerciseCreation	0	quiz	5	06.05.2023	06.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	11	history	3	quiz	10	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	12	history	4	quiz	10	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	13	history	5	quiz	10	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	14	history	6	quiz	10	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	15	history	7	quiz	10	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	18	history2	2	quiz	10	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	19	history2	3	quiz	10	23.05.2023	23.05.2023	ИЗМЕНИТЬ

Строк на странице: 10 1-10 of 41 < 1 2 3 4 5 >

Рисунок 34 — Пример работы фильтрации и сортировки

2.4.4 Редактор связанных сущностей

Для связанных сущностей (например, пользователь и его учебная группа) отображается ссылка на связанную сущность (Рис. 35). При клике по ней можно быстро перейти к просмотру или редактированию связанной сущности.

<input type="checkbox"/>	Id ↑	Упражнение	Учебная группа	Пользователь	Опыт	Закончено?
<input type="checkbox"/>	1	first	test group	test83	15	✓
<input type="checkbox"/>	2	test2	test group	test83	0	✗
<input type="checkbox"/>	3	test2	test group	test83	5	✓
<input type="checkbox"/>	4	first	test group	test12	15	✓
<input type="checkbox"/>	5	first	test group	test15	4	✓
<input type="checkbox"/>	6	test2	test group	test83	5	✗

Рисунок 35 — Ссылки на связанные сущности

2.4.5 Редактор вложенных сущностей

Для сущностей, которые физически размещены в отдельных коллекциях (например, Exercises и Steps), но логически имеют родительскую связь (шаг упражнения – упражнение) разработан специальный редактор. На экране редактирования родительской сущности в редакторе виден список всех зависимых сущностей. Через редактор можно:

- создавать (кнопка «Create step»)
- изменять (кнопка «Изменить»)
- удалять (кнопка доступна на экране редактирования) вложенные сущности (Рис. 36).

Это позволяет значительно упростить работу с вложенными сущностями.

Из вложенной сущности доступна ссылка обратно на родительскую.

Создано
23.05.2023

Обновлено
23.05.2023

Exercise Steps

CREATE STEP

<input type="checkbox"/>	Id ↓	Упражнение	Номер шага	Тип	Опыт	Создано	Обновлено	
<input type="checkbox"/>	15	history	7	quiz	10	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	14	history	6	quiz	10	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	13	history	5	quiz	10	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	12	history	4	quiz	10	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	11	history	3	quiz	10	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	10	history	2	text	5	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	9	history	1	text	5	23.05.2023	23.05.2023	ИЗМЕНИТЬ
<input type="checkbox"/>	8	history	0	text	5	23.05.2023	23.05.2023	ИЗМЕНИТЬ

СОХРАНИТЬ

УДАЛИТЬ

Рисунок 36 — Редактор вложенных сущностей

2.5 Подходы к проектированию аналитических дашбордов

Применение системам Business Intelligence, таких как Power BI и Tableau, в ежедневной работе – де факто стандарт индустрии. Любой человек, умеющий обращаться с ПК, может просматривать готовые дашборды, экспериментировать с фильтрами и сортировками, а потому внедрение BI не требует затрат на переподготовку бизнес-пользователей.

Из-за широкого функционала и гибкости существующих BI систем, выбрано использовать существующие BI системы.

При построении аналитики на основе данных систем организации рекомендуется настроить автоматическую выгрузку данных в аналитическое хранилище при помощи ETL процессов. Так, одной выгрузки данных ночью достаточно для решения большинства аналитических задач. Такой подход

позволит минимизировать нагрузку на СУБД системы и предотвратить замедление работы системы для пользователей.

Используя Anaconda Python, Pandas и Numpy, сгенерируем тестовый CSV датасет, имитирующий данные таблицы Runs (Рис. 37). Предположим, что 100 пользователей сделали 1000 выполнений упражнений. Всего упражнений 5. Прохождение заняло у каждого от 1 до 5 минут с синусоидальной зависимостью от id упражнения. За каждое упражнение начислялось 10, 50 или 100 баллов, делённое на время выполнения. Каждый совершил 0, 1 или 2 ошибки с весами 5, 2 и 1 соответственно и тангенсоидальной зависимостью от id упражнения.

Для анализа наших данных используем бесплатную версию Tableau Public 2022.4.

К примеру, для оценки прогрессии группы нас может интересовать количество пользователей, выполнивших каждое конкретное упражнение (Рис. 38). Также сравнение доли выполнивших необязательное упражнение можно использовать как метрику его полезности и привлекательности для пользователя.

# transformed (2).csv F1	# transformed (2).csv id	# transformed (2).csv userId	# transformed (2).csv exerciseld	# transformed (2).csv time	# transformed (2).csv experience	# transformed (2).csv mistakes	# transformed (2).csv finished
0	0	69	4	4	25	0	1
1	1	40	1	4	2	2	1
2	2	76	2	2	5	0	1
3	3	49	3	4	12	0	1
4	4	25	2	5	20	2	1
5	5	14	1	2	50	0	1
6	6	29	4	5	20	0	1
7	7	42	3	5	20	1	1
8	8	95	3	3	3	0	1
9	9	60	1	5	2	0	1
10	10	68	4	3	3	0	1
11	11	97	2	4	2	0	1
12	12	25	3	4	2	0	1
13	13	66	2	2	25	0	1

Рисунок 37 — Пример тестовых данных

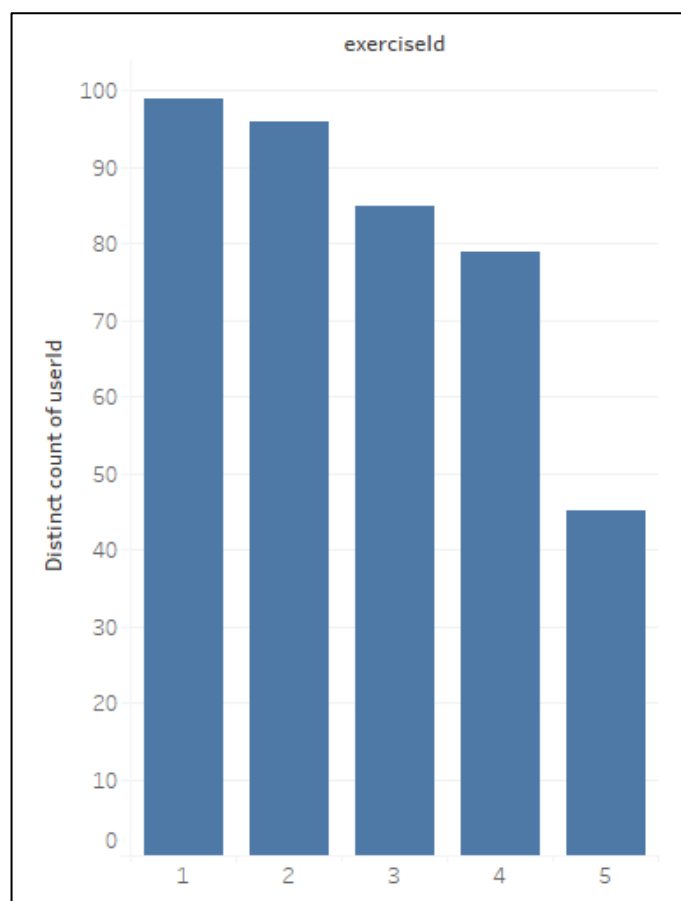


Рисунок 38 — Возможный вариант графика для визуального анализа числа пользователей выполнявших упражнение

Другая метрика, за которой должен следить автор упражнений – количество ошибок, совершаемых учащимися при прохождении. Слишком высокое число ошибок может указывать на ошибки в заданиях или непонятный теоретический материал.

Ожидается, что медиана для этого значения всегда будет 0, поэтому проанализируем среднее арифметическое (Рис. 39)

Аналогично предлагается анализировать значения выбранных перцентилей (например, 25 и 75), а также выбросы (например, рассчитав Z score для величины).

При оценке трудоёмкости курса и загруженности учащихся, полезно опираться на исторические данные о времени, которое занимает выполнение каждого конкретного упражнения (Рис. 40).

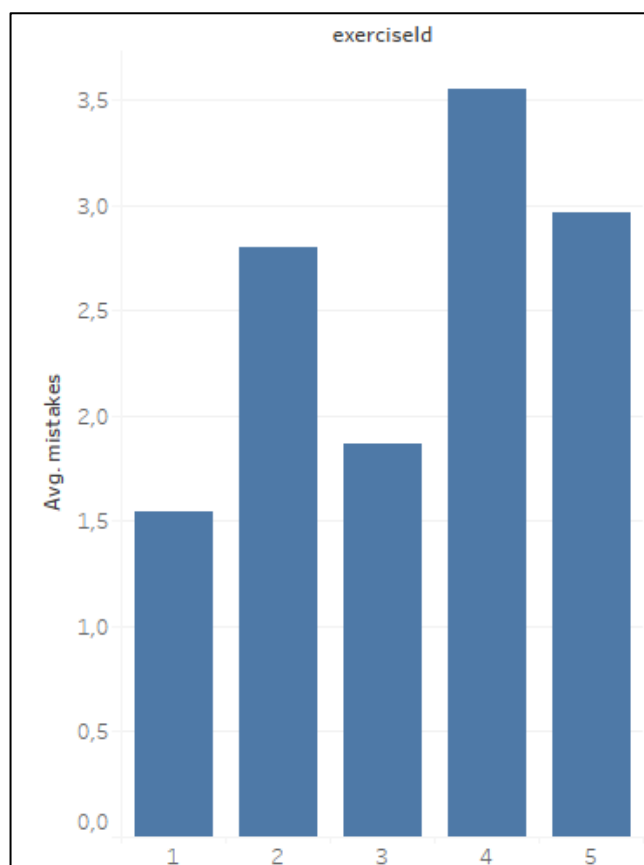


Рисунок 39 — График количества ошибок при прохождении упражнений

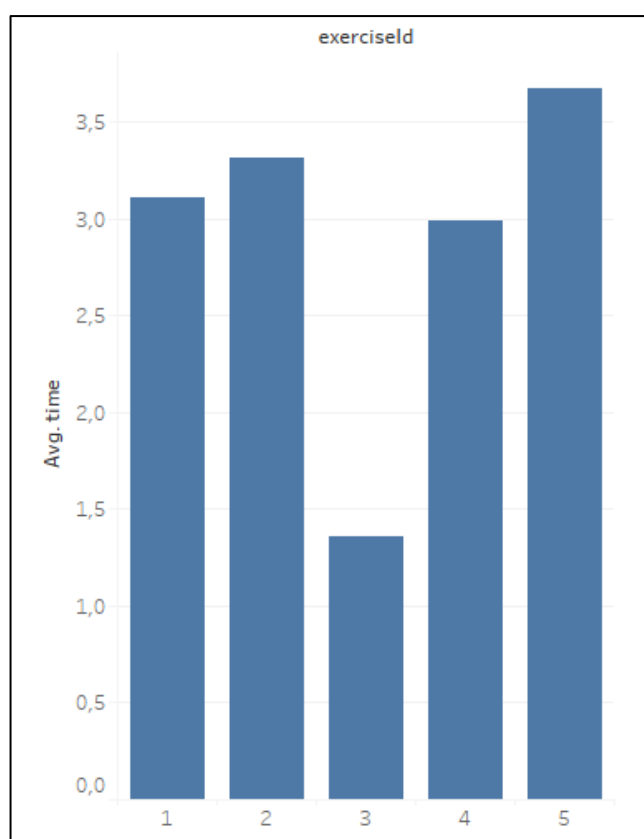


Рисунок 40 — График времени прохождения упражнений

2.6 Выводы

Во второй главе работы по разработке серверной части веб-приложения для геймификации процесса обучения сотрудников дилерских центров BMW Group решались задачи разработки серверного модуля системы, БД системы и панели администрирования, предложены методы анализа данных, собираемых системой.

В процессе решения поставленных задач использовались: язык программирования TypeScript, платформа Node.JS, СУБД SQLite.

В процессе разработки программной системы было сделано:

- разработана схема БД на 16 таблиц;
- разработан Бэкенд системы с применением архитектурного паттерна Controller-Service-Repository;
- разработана Панель Администратора с использованием технологии React и стиля Material Design;
- оба компонента интегрированы при помощи RESTful API, разработан RESTful API для Фронтенда;
- предложены методы анализа результатов использования системы для задач операционного мониторинга, улучшения содержимого курсов и стратегического планирования организации при помощи Python и Tableau.

Все поставленные задачи были выполнены.

Для усовершенствования системы организации предлагается:

- интегрировать уведомления пользователей через Email и бота Telegram;
- ограничить возможность пропускать текстовые и видео-шаги без ознакомления с контентом на стороне сервера;
- добавить более сложные типы вопросов, например, на сопоставление элементов;
- внедрить более точные и сложные инструменты аналитики.

Глава 3 ВНЕДРЕНИЕ И ЭКСПЛУАТАЦИЯ

3.1 Оценка технико-экономической целесообразности

3.1.1 Сравнение уровня качества разрабатываемого продукта с аналогом

По запросу Заказчика для определения целесообразности внедрения разработанной системы и выделения ресурсов для дальнейшей доработки был произведен анализ уровня качества разрабатываемого продукта по сравнению с аналогом.

Для анализа была выбрана методика, предложенная Миньковым и Томским государственным университетом систем управления и радиоэлектроники. Она адаптирована для обучающихся в ВУЗах, выполняющих выпускную квалификационную работу по специальности 08.08.01 «Прикладная информатика в экономике» [16]. Автору настоящей работы она показалась практичной, поскольку она предлагает беспристрастную систему оценки качества программного обеспечения.

Разработанный программный продукт будет сравниваться с Open Source LMS Moodle. Такой выбор был сделан по нескольким причинам:

- смежный профиль;
- закрывает широкий перечень задач корпоративного обучения;
- доступность реальной версии программы для изучения.

Показатели качества были выбраны в соответствии с деревом характеристик качества программного изделия [3]. Для учета значимости отдельных параметров применяется балльно-индексный метод. Коэффициенты весомости каждого показателя в долях единицы были назначены Заказчиком в соответствии с потребностями организации.

Для анализа были проведены интервью с представителями Заказчика: специалистом по кадровым проектам дилерской сети и менеджером по обучению в области продаж и маркетинга. Сводные результаты анкетирования представлены в Таблице 5, в Таблице 6 – об аналоге.

Каждой характеристике дана оценка 1-5 (X_j), где оценки выставлены следующим образом:

5 баллов – идеальный продукт, полностью соответствующий запросам заказчика и не требующий доработки;

4 балла – продукт может быть допущен к использованию, полностью соответствует запросам заказчика, определены перспективы для доработки;

3 балла – продукт может быть допущен к использованию, в достаточной мере соответствует запросам заказчика, определены перспективы для доработки;

2 балла – продукт не может быть допущен к использованию без существенной доработки;

1 балл – продукт не может быть допущен к использованию заказчиком и/или совершенно не удовлетворяет запросам заказчика.

Таблица 5 — Сводные результаты анкетирования заказчика о проекте

Показатели качества	Средняя оценка	Комментарии
1. Удобство работы, удобство пользовательского интерфейса (UX)	4	Интерфейс ограничен лишь действительно необходимыми элементами.
2. Соответствие современным технологиям и стандартам организации	4	Реализован SDLC процесс, программный продукт является in-house разработкой
3. Соответствие запросам заказчика	4	В продукте реализованы все необходимые механики.
4. Ресурсная эффективность	3	Продукт достаточно сложно развернуть.
5. Надежность	3	Продукт не был протестирован в реальных условиях.
6. Производительность	4	При правильном развёртывании ожидается достаточная производительность системы.
7. Ожидаемая результативность	4	При внедрении системы ожидается значительное улучшение KPI Центра
8. Соотношение стоимость/возможности	4	Проект был разработан с минимальными вложениями со стороны Заказчика

Оценку с учётом коэффициента весомости обозначим V_j . Обобщённый показатель качества примем как сумму оценок. Расчёты представлены в Таблице 7.

Таблица 6 — Сводные результаты анкетирования заказчика об аналоге

Показатели качества	Средняя оценка	Комментарии
1. Удобство работы, удобство пользовательского интерфейса (UX)	2	Интерфейс не интуитивно понятен и сложен для учащегося.
2. Соответствие современным технологиям и стандартам организации	2	Продукт разработан на устаревшем стеке и не соответствует стандартам организации.
3. Соответствие запросам заказчика	2	Отсутствуют механики геймификации. Нет единого образовательного трека.
4. Ресурсная эффективность	4	Продукт легко развернуть и поддерживать.
5. Надежность	4	У продукта большая история исправления проблем и ошибок.
6. Производительность	4	Продукт разработан на php и достаточно производителен для ожидаемой аудитории из коробки.
7. Ожидаемая результативность	2	Продукт не решает проблем организации.
8. Соотношение стоимость/возможности	2	Несмотря на бесплатность продукта, возможности достаточно ограничены.

Таблица 7 — Расчет показателя качества балльно-индексным методом

Показатели качества	Коэфф. Весомости, B_j	Проект		Аналог	
		X_j	$B_j \times X_j$	X_j	$B_j \times X_j$
1. Удобство работы, удобство пользовательского интерфейса (UX)	0,14	4	0,56	2	0,28
2. Соответствие современным технологиям и стандартам организации	0,1	4	0,4	2	0,2
3. Соответствие запросам заказчика	0,2	4	0,8	2	0,4
4. Ресурсная эффективность	0,05	3	0,15	4	0,2
5. Надежность	0,13	3	0,39	4	0,52
6. Производительность	0,1	4	0,4	4	0,4
7. Ожидаемая результативность	0,13	4	0,52	2	0,26
8. Соотношение стоимость/возможности	0,09	4	0,36	2	0,18
Обобщенный показатель качества		3,58		2,44	

Отношение двух показателей качества составило 1,47, что указывает на более высокое качество разрабатываемого продукта по сравнению с аналогом. Поскольку коэффициент выше 1, то внедрение продукта с технической точки зрения оправдано.

3.1.2 Оценка затрат на разработку проекта

В проекте участвовали:

- руководитель проекта – менеджер по обучению в области продаж и маркетинга;
- исполнитель (инженер-программист, бизнес-аналитик);
- исполнитель (инженер-программист).

По данным Комсомольской Правды средняя заработная плата программиста в Москве на 2023 год составила 180 тыс. рублей в месяц. Или

8181 руб. в день. В целях упрощения расчётов экономической составляющей работы примем размер страховых взносов как 30% [2]. Проект выполняется в период с 05.02.2023 по 20.05.2023. В период реализации проекта входит 69 рабочих дней. Таким образом, затраты на заработные платы составят 1 128 978 руб. С учётом зарплатных взносов – 1 467 671 руб. Расчёт фонда оплаты труда для проекта представлен в Таблице 8.

По данным Jobfilter в 2023 средняя зарплата менеджера по обучению и развитию персонала в Москве составляет 55 000 руб. в месяц. С учётом международного и локального уровня компании-Заказчика примем зарплату руководителя проекта за 100 000 руб. в месяц или 4545 руб. в день. Можно предположить, что на управление проектом он будет тратить не более 2 часов в неделю. Таким образом, за время реализации проекта будет потрачено 28 рабочих дня. За время реализации проекта затраты на работу менеджера проекта составят 127 260 руб. С учётом зарплатных взносов — 165 438 руб.

Таблица 8 — Расчёт фонда оплаты труда проекта

Должность	Должностной оклад	Средняя дневная ставка, руб.	Затраты времени, человеко-дней	ФОТ, руб.
Программист	180 000	8181	69	564 489
Программист	180 000	8181	69	564 489
Руководитель проекта	100 000	4545	28	127 260
Итого				1 256 238
С учётом зарплатных взносов 30%				1 633 109

Примем стоимость офисного места (стоимость места в коворкинге, офисные расходы) за 1000 руб. в день. Тогда расходы на офисные места для двух нанятых программистов (руководитель проекта уже имеет офис) составят 138 000 руб.

Общие расходы на проект составят 1,8 млн. руб.

3.1.3 Оценка эксплуатационных затрат

Рассчитаем эксплуатационные (текущие) затраты для первого года эксплуатации.

Для наполнения системы контентом и дальнейшего его обновления необходимо привлечь оператора. Можно предположить, что на работу с ИС он будет тратить не более 10 часов в неделю. Таким образом, за год будет потрачено 62,5 рабочих дня. Пусть оклад оператора составит 60 000 руб. в месяц или 2727 руб. в день. За год затраты на работу оператора составят 170 454 руб. С учётом зарплатных взносов — 221 591 руб.

Для установки, настройки и поддержания работы системы необходимы услуги системного администратора. Можно предположить, что на работу с ИС он будет тратить не более 4 часов в неделю. Таким образом, за год будет потрачено 25 рабочих дней. Пусть оклад системного администратора составит 60 000 руб. в месяц или 2727 руб. в день. Тогда, за год затраты на работу системного администратора составят 68 175 руб. С учётом зарплатных взносов — 88 628 руб.

Расчёты фонда оплаты труда представлены в Таблице 9.

Таблица 9 — Расчёт фонда оплаты для года эксплуатации

Должность	Должностной оклад	Средняя дневная ставка, руб.	Затраты времени, человеко-дней	ФОТ, руб.
Оператор	60 000	2727	62,5	170 454
Системный администратор	60 000	2727	25	68 175
Итого				238 629
С учётом зарплатных взносов 30%				310 218

Данные заказчика дают основания предположить, что приложение будет использоваться 2500 пользователями, 60% из которых будет заходить в

приложение ежедневно на период не более часа. Таким образом, в среднем системой одновременно будет пользоваться 63 пользователя.

Логично предположить, что даже при пиковом использовании системы в течение рабочего дня, нагрузка не превысит 1000 запросов в секунду.

Нагрузочное тестирование с использованием Postman показало, что единый процесс Бэкенда с ограничением в 1024 МБ RAM и 1 ядро внутри локальной сети способен обработать 1000 запросов в секунду без существенного падения времени ответа. Такой VDS сервер можно приобрести за 350 руб. в месяц. За год эксплуатации аренда VDS обойдется в 4250 руб. Регистрация доменного имени обойдётся в пределах 1000 руб. в год.

Таким образом, можно оценить расходы за год эксплуатации системы без дополнительной разработки в 315 468 руб.

3.1.4 Оценка эффективности разработанного проекта

Зарубежные исследования показали, что при увеличении вложений в обучение на 10% можно получить прирост производительности труда на 8,5%. Капиталовложения того же размера дают увеличение производительности труда лишь на 3,8% [13].

При текущей выручке холдинга в России в 3,5 млрд рублей в год, дообучение сотрудников дилерских центров и соответствующий рост в качестве обслуживания могут принести до 304 млн рублей дополнительной выручки. Эта сумма в 145 раз превышает стоимость разработки и эксплуатации системы до момента появления первых результатов.

Таким образом, можно говорить об оправданности высокорисковых инвестиций в проект как часть R&D компании.

3.2 Источники контента

Контент для системы предоставляется Заказчиком.

Контент по автомобилям BMW будет перенесён из внешнего проприетарного решения, используемого организацией по подписочной модели – BMW Start.

В рамках Проектной Деятельности ранее было разработано содержание курса для бренда MINI в составе BMW Group. Курс включил в себя материалы по темам:

- модельный ряд: MINI Clubman, MINI 3 door, MINI 5 door, MINI Countryman;
- история бренда MINI: основатели бренда, первые авто, как развивался бренд, кому принадлежал, что с брендом происходит сейчас;
- технологии MINI: двигатель, трансмиссия, технологии изготовления, комплектация, и т.д.;
- интересные факты и новости про MINI: мемы, упоминания в СМИ, книгах/фильмах/мультфильмах, курьезные ситуации в компании;
- дилерская сеть в РФ, представительства в мире.

Помимо статей в курс вошли тесты для самопроверки: один из многих, многие из многих, задания с произвольным ответом.

Часть контента по бренду MINI была включена в систему в демонстрационных целях.

3.3 Аспекты тестирования программного продукта

С первых работ, посвященных тестированию программного обеспечения, выпущенных в 1970ых, тестирование ПО становится лишь сложнее. Причиной тому – появление новых языков программирования, операционных систем и инфраструктуры. Программное обеспечение, выпускаемое сегодня, воздействует на жизни миллионов людей, позволяя им работать продуктивно и эффективно. Тестирование сегодня важнее чем когда-либо.

Важная часть тестирования ПО – обеспечение отслеживаемости действий программы или системы. Следы (traces) могут помочь в поиске и устранении ошибок, которые возникают в процессе работы системы. С помощью следов можно проанализировать последовательность действий, приведших к ошибке, и определить ее причину.

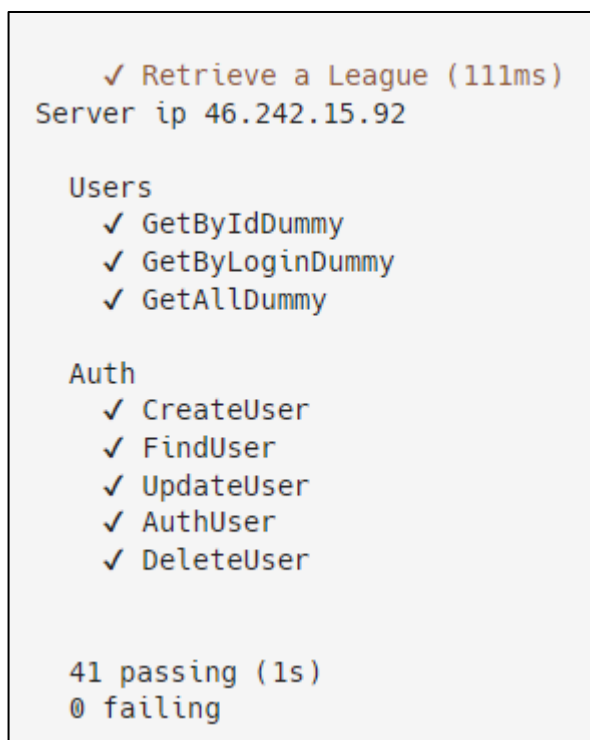
Использование трассировок охватывает сертификацию и тестирование программного обеспечения, определение местоположения функций, дебаггинг, генерацию кода и многое другое [31].

3.3.1 Unit тестирование

Модульное тестирование (юнит тестирование) — это вид тестирования ПО, при котором на наличие ошибок проверяется работа небольших изолированных блоков кода. Это один из традиционных шагов в процессе разработки программного обеспечения. Unit тестирование позволяет автоматически и в короткий срок обнаруживать ошибки, допускаемые разработчиками в процессе написания кода.

Unit тестирование также помогает предотвратить регрессию ПО в следствие изменений в коде [44].

В рамках данного проекта для обеспечения качества работы ПО и предотвращения регрессии было внедрено Unit тестирование. Для этого применяется библиотека mocha. С ней все тесты можно выполнить одной командой «npm run test» (Рис. 41).



```
✓ Retrieve a League (111ms)
Server ip 46.242.15.92

Users
  ✓ GetByIdDummy
  ✓ GetByLoginDummy
  ✓ GetAllDummy

Auth
  ✓ CreateUser
  ✓ FindUser
  ✓ UpdateUser
  ✓ AuthUser
  ✓ DeleteUser

41 passing (1s)
0 failing
```

Рисунок 41 — Окончание логов выполнения тестов

Покрывание unit тестами составляет порядка 50%. Остальные части кода сильно зависят от тестовых данных в БД, а потому были протестированы на этапах тестирования API и интеграционного тестирования.

3.3.2 Тестирование API

Тестирование API – это вид тестирования ПО, при котором проверяется соответствует ли API ожиданиям в отношении функциональности, надежности, производительности и безопасности [47].

Для тестирования API существуют готовые инструменты, такие как SoapUI, Postman, ReadyAPI, cURL, Parasoft SOAtest, Tricentis Tosca, Karate DSL, REST-Assured, HTTP Master и другие. Обычно они могут производить следующие действия:

- отправить запрос;
- проверить заголовки и тело ответа;
- проверить наличие кодов ошибок;
- десериализовать файлы JSON и XML;
- проверить определенные элементы или атрибуты страницы;
- проверить время ответа;
- имитировать рабочие процессы пользователей;
- сопоставить данные ответа с ожидаемыми значениями.

Наиболее продвинутые из программ для тестирования могут делать снимки экрана для отладки [49].

Для тестирования API применяется программа Postman. В поставляемой вместе с проектом коллекции Postman представлены предварительно созданные запросы к API (Рис. 42).

3.3.3 Интеграционное тестирование

Интеграционное тестирование — это вид тестирования ПО, при котором проверяется, насколько хорошо разные части программной системы работают вместе. Обычно это проводится после разработки программного

обеспечения, но также может выполняться в процессе разработки. Интеграционное тестирование проверяет, что все части системы взаимодействуют должным образом.

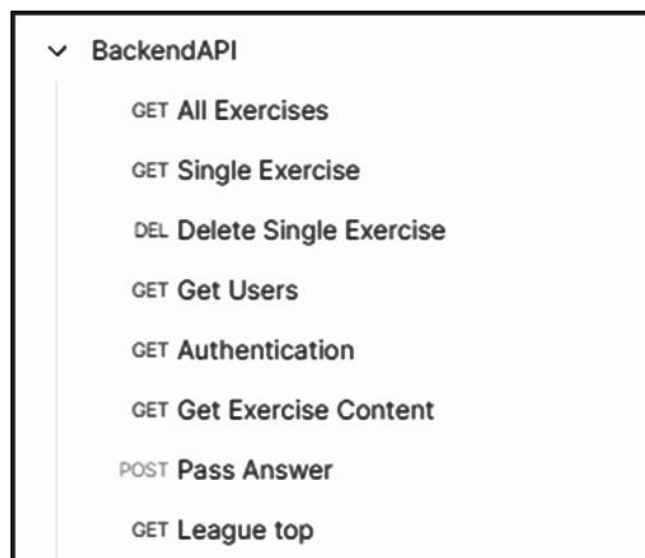


Рисунок 42 — Некоторые из запросов коллекции Postman

В Бэкенде применяется автоматизированное e2e тестирование нескольких ключевых сценариев использования на уровне сервисов. Интеграционное тестирование проводится по ручным сценариям тестирования с использованием Postman.

3.4 Аспекты информационной безопасности

Информационная безопасность — это практика защиты электронной информации путем снижения информационных рисков. Проблемы информационной безопасности могут включать защиту электронных систем и данных от случайного или несанкционированного доступа, уничтожения, модификации или раскрытия.

Организации, использующие электронную информацию, должны защищать свои системы и данные от различных рисков, включая несанкционированный доступ, перехват, повреждение и проникновение злоумышленников. Эффективная информационная безопасность требует понимания как физических, так и электронных угроз системам и данным, а

также надлежащего развертывания технологий и процедур смягчения последствий.

Предотвращение информационных рисков является сложной задачей и обычно требует комплексной работы различных сотрудников с разным опытом и знаниями [51].

Компании, согласно законодательству РФ, несут ответственность за сохранность личных данных своих сотрудников и клиентов [1]. Разрабатываемая система работает с персональными данными сотрудников организации.

Материалы курсов можно отнести к коммерческой тайне. Утечка содержимого курсов конкурентам (в результате промышленного шпионажа или внутренних ошибок) может лишить BMW конкурентного преимущества.

В программе имеются встроенные средства защиты, осуществляющие авторизацию и идентификацию пользователей после введения пользователем пароля. Доступ к материалам курсов выдаётся для каждой учебной группы инструкторами.

Для обеспечения дополнительной безопасности системы организации рекомендуется:

- ограничить доступ к системе рамками внутренней сети Интранет;
- разрешить доступ пользователям корпоративного VPN в дилерских центрах;
- заблокировать доступ к Интранет с иностранных IP;
- осуществлять мониторинг трафика в сети Интранет для обнаружения потенциальных угроз;
- перед внедрением системы в деятельность организации провести аттестацию системы;
- обеспечить режим конфиденциальности для лиц, имеющих доступ к материалам курсов: провести инструктажи среди сотрудников, составить официальные договоры о неразглашении;

- на квартальной основе проверять список пользователей, имеющих доступ к системе. Удалять устаревшие доступы, например, уволенных сотрудников;

- использовать DLP и SIEM системы для ограничения копирования информации на сторонние носители и отслеживания запросов к СУБД.

3.5 Установка и настройка программных компонентов

3.5.1 Необходимое программное обеспечение, которое должно быть установлено на рабочем месте

Для компиляции и запуска Бэкенда и Панели Администратора необходимо следующее программное обеспечение:

- Node.JS версии v18.14.1;
- TypeScript версии v4.9.5;
- NPM версии v9.5.0.

3.5.2 Установка и запуск Бэкенда

Для установки и запуска Бэкенда необходимо:

1. Клонировать исходные коды Бэкенда из git репозитория.
2. Из окружения командной строки установить пакеты и библиотеки Node.js командой `npm install`.

3. NPM автоматически скачает и установит все библиотеки, перечисленные в зависимостях в файле `package.json` (разделы `dependencies` и `devDependencies`).

4. Для запуска программы с использованием SQLite (без дополнительной настройки подключения к СУБД) необходимо скопировать БД (файл `db.db`) в папку `data`.

5. Запустить программу командой `npm run dev`.

Программа будет скомпилирована и сразу же запущена (Рис. 43).

Для разворачивания Бэкенда в реальном окружении необходимо:

- заменить подключение БД в файле `source/Database.ts` на MySQL или PostgreSQL, а затем перекомпилировать программу;

```
C:\Users\mi\Documents\Projects2023\bmweduction>npm run dev

> couponsbot@1.0.0 dev
> nodemon --exec "ts-node src/index.ts --files src" -V -w src -e ts

[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] or send SIGHUP to 40124 to restart
[nodemon] watching path(s): src\**\*
[nodemon] watching extensions: ts
[nodemon] starting `ts-node src/index.ts --files src`
[nodemon] spawning
[nodemon] child pid: 43140
[nodemon] watching 71 files
Bot started
Server listening at http://localhost:3000
Server ip 149.202.225.165
OPTIONS to /api/exercises?filter=%7B%7D&range=%5B0%2C9%5D&sort=%5B%22id%22%2C%22ASC%22%5D
GET to /api/exercises?filter=%7B%7D&range=%5B0%2C9%5D&sort=%5B%22id%22%2C%22ASC%22%5D
█
```

Рисунок 43 — Запуск Бэкенда в консоли

- предварительно скомпилировать программу командой `npm build`, скопировать на сервер папку `bin` и запустить командой `npm start`;
- использовать `reverse проху` для открытия доступа к Бэкенду извне, по умолчанию система доступна с портом 3000.

3.5.3 Установка и запуск Панели Администратора

Для установки и запуска Панели Администратора необходимо:

1. Клонировать исходные коды панели-администратора из `git` репозитория.
2. Из окружения командной строки установить пакеты и библиотеки Node.js командой `npm install`. NPM автоматически скачает и установит все библиотеки, перечисленные в зависимостях в файле `package.json` (разделы `dependencies` и `devDependencies`).
3. Запустить программу командой `npm run dev`. Программа будет скомпилирована и сразу же запущена (Рис. 44).
4. По умолчанию панель администратора будет пытаться подключиться к Бэкенду по адресу `http://localhost:3000`. Поменять адрес можно в файле `package.json`.

```
C:\Users\mi\Documents\Projects2023\bmweduction>cd my-admin

C:\Users\mi\Documents\Projects2023\bmweduction\my-admin>npm run dev

> my-admin@0.0.0 dev
> vite

VITE v4.1.3  ready in 3124 ms

  → Local:   http://localhost:5173/
  → Network: use --host to expose
  → press h to show help
█
```

Рисунок 44 — Запуск панели администратора в консоли

Для разворачивания панели администратора в реальном серверном окружении необходимо:

- предварительно скомпилировать программу командой `npm build`, скопировать на сервер папку `bin` вместо папки `src` и запустить командой `npm start`;
- использовать reverse proxy (например, `nginx`) для открытия доступа к панели извне.

3.6 Аспекты надежности, отказоустойчивости и производительности программного продукта

3.6.1 Надёжность программного продукта

Надёжность программного обеспечения – способность программного обеспечения выполнять свои функции без сбоев в конкретных условиях в течение периода времени. Надёжность программного обеспечения показывает насколько можно положиться и доверять работе программного обеспечения. Надёжность достигается с помощью тестирования, отладка, обеспечения отказоустойчивости, обработки ошибок и др. [53].

3.6.2 Отказоустойчивость программного продукта

Отказоустойчивость ПО — это способность программной системы продолжать работать при наличии сбоев или ошибок. Для обеспечения

отказоустойчивости системы необходимо разрабатывать программное обеспечение таким образом, чтобы оно могло обнаруживать и устранять сбои и не позволять им влиять на общую производительность или функциональность системы. Отказоустойчивые программные системы устойчивы к аппаратным и программным сбоям и могут продолжать работать, даже если один или несколько компонентов выходят из строя.

Разрабатывая программные системы с учетом отказоустойчивости, программисты могут повысить общую надежность программного обеспечения, что приводит к повышению удовлетворенности пользователей и снижению затрат из-за простоев или сбоев.

Кроме того, отказоустойчивость программного обеспечения также может повысить производительность ПО за счет уменьшения влияния ошибок или сбоев на систему. Благодаря быстрому обнаружению и устранению сбоев система может избежать длительных простоев или сбоев системы, которые могут негативно сказаться на производительности. Это может привести к повышению доступности системы, сокращению времени отклика и увеличению пропускной способности.

Отказоустойчивость помогает обеспечить надежность и производительность программных систем в условиях непредвиденных событий или сбоев [38]. Отказоустойчивость является важнейшим компонентом разработки программного обеспечения.

Поскольку деятельность системы не является критичной для операционной деятельности компании, возможные ошибки и перебои в работе системы несут низкий уровень угрозы.

Панель администратора выполняется полностью в браузере пользователя. В случае возникновения ошибки страницу можно обновить и продолжить работу.

Бэкенд построен по транзакционной модели и не хранит собственного состояния в памяти. Программа продолжит работу в прежнем режиме в случае возникновения ошибок с тем или иным входящим запросом.

Для обеспечения отказоустойчивости работы системы рекомендуется развернуть несколько экземпляров Бэкенда, подключенных к единой СУБД, использующей кластеризацию.

Доступ к API Бэкенда может затем осуществляться как с единым адресом (с применением обратного прокси), так и напрямую к каждому экземпляру.

Для обеспечения фоновой работы Бэкенда в виде сервиса, автоматического перезапуска в случае критической ошибки программы или перезапуска сервера, удалённого управления и мониторинга рекомендуется использовать менеджер процессов pm2.

Менеджер процессов pm2 предназначен для работы именно с Node.js приложениями [43].

Вместе с исходным кодом программы поставляется готовая конфигурация для pm2 (файл ecosystem.config.js). Конфигурация настроена для 1 копии Бэкенда, автоматического перезапуска, и вывода логов в файлы. Также pm2 перезапустит Бэкенд автоматически при замене файлов в папке bin (для релиза новой версии). Для запуска приложения в pm2 достаточно одной команды – «pm2 start».

3.6.3 Производительность программного продукта

Производительность программного обеспечения – мера того, насколько хорошо программа или система выполняет поставленные перед ней задачи и насколько быстро она может обрабатывать данные, реагировать на действия пользователя и выполнять операции.

Высокая производительность программного обеспечения гарантирует, что пользователи смогут выполнять свои задачи быстро и эффективно, не сталкиваясь с задержками или ошибками. Производительность также помогает обеспечить комфортную работу пользователя с программой, что приводит к повышению продуктивности и удовлетворенности пользователей сервисом [39].

Бэкенд с использованием Node.JS является однопоточным, т.е. использует ресурсы только одного ядра. Для увеличения производительности применяется горизонтальное масштабирование путём запуска копий приложения на одном или разных серверах.

Вместе с исходным кодом программы поставляется готовая конфигурация для Nginx (файл `nginx.conf`). Конфигурация настроена для 4 копий Бэкенда на портах 3000-3003 того же сервера.

Для дальнейшей доработки системы необходимо использовать отдельное тестовое окружение.

3.7 Выводы

В третьей главе работы по разработке серверной части веб-приложения для геймификации процесса обучения сотрудников дилерских центров BMW Group решались задачи подтверждения технико-экономической целесообразности проекта, тестирования компонентов системы, обеспечения безопасной и стабильной работы системы.

В процессе решения поставленных задач использовались анализ академических источников, теория тестирования ПО как часть SDLC, методы бизнес-анализа.

В процессе решения поставленных задач было сделано следующее:

- подтверждена технико-экономическая целесообразность разработки системы, определена стоимость разработки в размере 1.8 млн рублей;
- предложена схема разворачивания системы на основании горизонтального масштабирования каждого из компонентов системы для обеспечения надежности, отказоустойчивости и производительности системы;
- предложен ряд рекомендаций по обеспечению информационной безопасности: использование Интранета, режим корпоративной тайны, контроль доступов, и т.д. (полный список представлен в пункте 3.4);

Все поставленные задачи были выполнены.

Для дальнейшего усовершенствования работы системы организации предлагается:

- вынести инструкцию по установке и настройке, руководство по использованию панели администратора отдельными документами корпоративную базу знаний;
- переформулировать документацию API из коллекции Postman в формат wiki или словаря.

ЗАКЛЮЧЕНИЕ

В ходе работы по разработке серверной части веб-приложения для геймификации процесса обучения сотрудников дилерских центров BMW Group был реализован актуальный и технически и экономически целесообразный проект разработки системы корпоративной системы обучения для BMW Group. В рамках проекта были решены задачи:

1. Анализ предметной области, аналогов и конкурентов.
2. Разработка проекта программного продукта системы корпоративного обучения, демонстрирующего особенности реализации и функционирования ИС.
3. Оценка экономической целесообразности проекта.
4. Разработка БД системы, серверного модуля системы, панели администрирования.
5. Тестирование компонентов системы.
6. Рекомендация методов анализа данных, собираемых системой.
7. Обеспечение информационной безопасности, надежности, отказоустойчивости и производительности системы.

Все поставленные задачи были выполнены.

Цель выпускной квалификационной работы по разработке серверной части информационной системы для корпоративного обучения сотрудников дилерских центров была достигнута.

При разработке использовались: язык программирования TypeScript, JavaScript, платформа Node.JS, фреймворк React, СУБД MySQL, SQLite.

В процессе проектирования системы использовались анализ академических источников, методы бизнес-анализа, контекстные диаграммы, IDEF0 диаграммы, BPMN диаграммы, теория SDLC.

Запросы заказчика к системе были:

- перевод материалов тренингов и упражнений в онлайн формат, с соблюдением при этом политики и процедуры организации;

- установка чётких сроков прохождения материалов и упражнений: сколько обучающийся в состоянии слушать одну тему, как часто ему необходимо менять вид активности в онлайн. В очном режиме это контролируется тренером, онлайн сложно отслеживать состояние аудитории;

- обработка предварительно созданного контента для построения единого образовательного трека.

Разработанная система удовлетворяет всем предъявленным требованиям. Для увеличения мотивации и вовлеченности учащихся были применены методы геймификации. Система объединяет процессы как тренеров, так и учащихся.

Для усовершенствования системы в дальнейшем предлагается:

- провести анализ процессов AS IS и TO BE не только Центра Обучения, но и дилерских центров для обнаружения процессов, которые можно также автоматизировать системой;

- внедрить уведомления пользователей через Email и бота Telegram;

- ограничить возможность пропускать текстовые и видео-шаги без ознакомления с контентом;

- добавить более сложные типы вопросов, например, на сопоставление элементов;

- внедрить более точные и сложные инструменты аналитики;

- вынести инструкцию по установке и настройке, руководство по использованию панели администратора отдельными документами корпоративную базу знаний;

- доработать функционал уведомлений, внедрив, например, Email уведомления пользователей, достижений;

- переформулировать документацию API из коллекции Postman в формат wiki или словаря.

Этот проект был представлен в докладах на следующих конференциях:

1. Международной научно-практической конференции памяти заслуженного деятеля науки РФ В.И. Кравцовой [18].

2. Управление устойчивым инновационным развитием России в условиях цифровой трансформации [19].

3. Цифровая трансформация: информатика, экономика и образование (DTIEE2023) [20].

СПИСОК ИСТОЧНИКОВ

1. Федеральный закон «О персональных данных» от 27.07.2006 N 152-ФЗ (с изменениями и дополнениями от 6 февраля 2023 г.).
2. Постановление Правительства Российской Федерации от 25.11.2022 № 2143 «О единой предельной величине базы для исчисления страховых взносов с 1 января 2023 г.»
3. ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению.
4. Алексеева А.З., Соломонова Г.С., Аетдинова Р.Р. Геймификация в образовании // Педагогика. Психология. Философия. 2021. №4 (24).
5. Ананченкова П.И., Бураков В.И., Спасенникова М.Г. Корпоративное обучение как инструмент управления человеческими ресурсами // Baikal Research Journal. 2019. №3.
6. Андриянова М.В. Внутрифирменное обучение персонала в России: тенденции и перспективы // АНИ: экономика и управление. 2018. №2 (23).
7. Баева О.Н. Оценка участия работников организаций в дополнительном образовании (обучении) / О.Н. Баева, Н.П. Шерстянкина // Труд и социальные отношения. – 2018. – No 3. – С. 16-26.
8. Бердникова Л.Ф., Шнайдер В.В. Финансовая устойчивость: понятие, сущность и типы // Научен вектор на Балканите. 2019. Т. 3. No 2 (4). С. 94-96.
9. Ганаева Е.А. Технология оценки результативности корпоративного обучения / Е.А. Ганаева, А.Ф. Гузаирова, Н.В. Липаткина // Вестник Оренбургского государственного университета. 2019. N 5(223). С. 100–107.
10. Дилеры Mini. Mini. [Электрон. ресурс] URL: https://www.mini.ru/ru_RU/home/mini-centres/dealer-locator.html (дата обращения: 06.05.2023) – Загл. с экрана.
11. Дороговцева А.А., Томилина Э.И., Ерыгина А.В. Современные HR-тренды, как важный аспект конкурентоспособности компаний // УПИРР. 2018. №4.

12. Карьера в компании BMW Group Россия. BMW. [Электрон. ресурс] URL: <https://www.bmw.ru/ru/topics/fascination-bmw/recruitment/recruitment.html> (дата обращения: 12.05.2023) – Загл. с экрана.

13. Кондратенко А.С., Миронова Н.Н. Российский авторынок в 2018 г. Заметки по повод менеджменты и маркетинга отечественных автодилеров. // Вестник НИБ. 2018. №34.

14. Лидерство в премиальном сегменте и новые рекорды: BMW Group объявляет результаты продаж в 2021 году. [Электронный ресурс]. URL: <https://www.press.bmwgroup.com/russia/article/detail/T0364335RU/лидерство-в-премиальном-сегменте-и-новые-рекорды:-bmw-group-объявляет-результаты-продаж-в-2021-году?language=ru> (дата обращения: 12.05.2023) – Загл. с экрана.

15. Матюнин Л.В., Чекан А.А. Основные тенденции в сфере корпоративного обучения персонала. // Мировые цивилизации. 2021. №6.1 С. 101-108.

16. Миньков С.Л. Техничко-экономическое обоснование выполнения проекта: методическое пособие / С.Л. Миньков. – Томск: ТУСУР, 2014. – 30 с.

17. Мыльникова С.А., Погосян В.А. Корпоративное обучение // Universum: Вестник Герценовского университета. 2014. №3-4.

18. Наконечный П.А., Бородачева Л.В. Перспективы внедрения систем онлайн-обучения в деятельность автоконцернов РФ // Материалы Междунар. науч.-практ. конф. памяти заслуженного деятеля науки РФ В.И. Кравцовой. Москва: Московский политехнический университет, 2023.

19. Наконечный П.А., Змазнева О.А. Разработка веб-приложения для геймификации процесса обучения сотрудников дилерских центров BMW Group // Междунар. конф. «Язык, сознание, коммуникация: проблемы информационного общества». Москва: Московский политехнический университет, 2023.

20. Наконечный П.А., Змазнева О.А. Разработка веб-приложения для геймификации процесса обучения сотрудников дилерских центров BMW

Group // Proceedings of SPIE – The International Society for Optical Engineering. Междуна. науч.-практ. конф. «Цифровая трансформация: информатика, экономика и образование» (DTIEE2023). Москва: Ферганский государственный университет и Московский политехнический университет, 2023.

21. Обучение как драйвер изменения поведения: предпосылки и практические подходы Аналитический отчет. // АНО ДПО «Корпоративный университет Сбербанка», 2020. С. 64.

22. Официальные дилеры BMW. BMW [Электрон. ресурс]. URL: <https://www.bmw.ru/ru/fastlane/dealer-locator.html> (дата обращения: 06.05.2023) – Загл. с экрана.

23. Официальные дилеры Rolls-Royce. Bibika.ru [Электрон. ресурс]. URL: http://www.bibika.ru/official_dealers/rolls-royce (дата обращения: 06.05.2023) – Загл. с экрана.

24. Полякова А.И., Котова Е.В. Разработка системы корпоративного обучения в условиях цифровизации. // Современное педагогическое образование. 2022. №12.

25. Суходолов А.П. Научно-образовательный потенциал и стратегия развития байкальского университета (к 85-летию вуза) / А.П. Суходолов // Известия Иркутской государственной экономической академии. – 2015. – Т. 25, No 2. – С. 187-195.

26. Татаринов Константин Анатольевич Геймификация в обучении студентов // БГЖ. 2019. №1 (26).

27. Толстякова О.В. Формирование системы развития человеческого капитала путем внедрения непрерывного обучения // Инновации и инвестиции, 2021. №6.

28. Шальнев О.Г. Современные форматы организации корпоративного обучения в условиях диджитализации промышленности // Организатор производства. 2020. №3.

29. Юстус Г.В. Корпоративное обучение специалистов: актуальность, стратегии и методы // The Scientific Heritage. 2020. №47-3 (47).
30. 2022 Duolingo Language Report [Электрон. ресурс]. URL: <https://blog.duolingo.com/2022-duolingo-language-report/> (дата обращения: 28.02.2023).
31. Beyer, Dirk. Advances in Automatic Software Testing: Test-Comp 2022. // FASE, 2022.
32. Brands & Services. BMW Group [Электрон. ресурс]. URL: <https://www.bmwgroup.com/en/brands-and-services.html> (дата обращения: 06.05.2023).
33. Chapman, Chris, Wai Fong Chua, and Tanya Fiedler. Seduction as control: Gamification at Foursquare. Management Accounting Research. 2021. №53.
34. Does gamification inspire intrinsic motivation? [Электрон. ресурс] Growth Engineering. 2019. URL: <https://www.growthengineering.co.uk/intrinsicmotivation-gamification/>
35. Introduction to Material Design [Электрон. ресурс]. URL: <https://m2.material.io/design/introduction> (дата обращения 14.04.2023).
36. Jahn K., et al. Individualized gamification elements: The impact of avatar and feedback design on reuse intention. // Computers in Human Behavior. 2021. No. 119.
37. Kok A. How to Manage the Inclusion of E-Learning in Learning Strategy Insights From a Turkish Banking Institution // International Journal of Advanced Corporate Learning. 2013. N 6 (1). P. 20-27.
38. Koren, Israel, and C. Mani Krishna. Fault-tolerant systems. // Morgan Kaufmann, 2020.
39. Laplante, Phillip A., and Mohamad Kassab. What every engineer should know about software engineering. // CRC Press, 2022.
40. Li, Xiayi. The Gamification of Dating: Tinder and Game-playing on Reddit communities. // The School of Culture and Communication, 2017.

41. Okesola, O. J., Adebisi, A. A., Owoade, A. A., Adeaga, O., Adeyemi, O., & Odun-Ayo, I.. Software Requirement in Iterative SDLC Model. In Intelligent Algorithms in Software Engineering: Proceedings of the 9th Computer Science On-line Conference 2020. // Springer International Publishing, 2020. №1.9. P. 26-34.
42. Jagoda, Patrick. Experimental Games: Critique, Play, and Design in the Age of Gamification. // Chicago: University of Chicago Press. 2020. 386 p.
43. Patrou, Maria, Kenneth B. Kent, and Michael Dawson. Scaling Parallelism under CPU-intensive Loads in Node. js. 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). // IEEE, 2019.
44. Qasim, Muhammad, et al. Test case prioritization techniques in software regression testing: An overview. // International Journal of advanced and applied sciences, 2021. №8.5. P. 107-121.
45. Rodrigues, Ivo, et al. How can Gamified applications drive engagement and brand attitude? The case of Nike run club application. Administrative Sciences. 2021. №11.3 (92).
46. Securities Industry Essentials (SIE) Exam [Электрон. ресурс]. URL: <https://www.stcusa.com/sie/> (дата обращения: 28.02.2023).
47. Soni, Anshu, and Virender Ranga. API features individualizing of web services: REST and SOAP. // International Journal of Innovative Technology and Exploring Engineering, 2019. №8.9. P. 664-671.
48. The past, present, and future of gamification. 2020 report. [Электрон. ресурс]. // Growth Engineering. 2022. URL: <https://www.growthengineering.co.uk/resources/future-of-gamification-white-paper/> (дата обращения 14.04.2023).
49. Thooriqoh, Hazna At, et al. A Recommendation Model of REST API Testing Framework based on Resource Utilization of ISO/IEC 25010. // 2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI). IEEE, 2022.
50. Tiuraniemi, Aleksi. Applying gamification in student coaching industry: A competitive environment analysis. 2021.

51. Usmonov, Maxsud. Basic Concepts of Information Security. // Scienceweb academic papers collection, 2021.

52. Yue, Cen April, Linjuan Rita Men, and Mary Ann Ferguson. Bridging transformational leadership, transparent communication, and employee openness to change: The mediating role of trust. // Public relations review. 2019. №45.3.

53. Zhu, Mengmeng, and Hoang Pham. Software reliability modeling and methods: A state of the art review. Optimization Models in Software Reliability. 2022. P. 1-29.

ПРИЛОЖЕНИЕ А

Техническое задание на ИС

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИСПОЛНИТЕЛЬ

Наконечный П.А.

КУРАТОР

Змазнева О.А.

Дата

Дата

**СЕРВЕРНАЯ ЧАСТЬ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ГЕЙМИФИКАЦИИ
ПРОЦЕССА ОБУЧЕНИЯ СОТРУДНИКОВ ДИЛЕРСКИХ ЦЕНТРОВ
BMW GROUP**

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

На 8 листах

Москва, 2023

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Наименование программы

Полное наименование программы: «Информационная система корпоративного обучения сотрудников аккредитованных дилерских центров BMW. Серверный модуль».

Краткое наименование проекта: «Бэкенд системы корпоративного обучения BMW Group».

Далее – программный продукт.

1.2 Заказчики

Надзор за выполнением проекта осуществляют:

- Безрукова Анна Александровна – заказчик проекта, специалист по кадровым проектам дилерской сети.

- Змазнева Олеся Анатольевна – руководитель ВКР, к.ф.н, доцент кафедры инфокогнитивных технологий, Московский политехнический университет;

- Даньшина Марина Владимировна – РОП, заместитель декана факультета информационных технологий, Московский политехнический университет.

1.3 Разработчики

Команда разработки состоит из следующих специалистов:

- Наконечный Павел – студент группы 191-361.

1.4 Плановые сроки начала и окончания работ

Начало выполнения работ – 5 февраля 2023.

Плановый срок сдачи работ – 14 июня 2023.

1.5 Основание для разработки

Учебный план факультета информационных технологий Московского политехнического университета.

Порядок оформления и предъявления результатов работ определен учебным планом факультета информационных технологий Московского политехнического университета.

2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

2.1 Назначение системы

Осуществление автоматизации прохождения учебных курсов и тренингов учебного центра BMW.

2.2 Цели создания системы

1. Перевод материалов тренингов и упражнений в онлайн формат, соблюдая при этом политики и процедуры организации.

2. Установка чётких сроков прохождения материалов и упражнений. Сколько обучающийся в состоянии слушать одну тему, как часто ему нужно менять активность в онлайн. В очном режиме это контролируется тренером, онлайн сложно отслеживать состояние аудитории.

3. Обработка предварительно созданного контента для построения единого образовательного трека.

2.3 Критерии достижения целей

Для реализации поставленных целей система должна решать следующие задачи:

1. Создание программного продукта (портала), позволяющего проходить теоретические материалы и выполнять упражнения онлайн, агрегировать результаты этих упражнений, успеваемость учащихся в единой системе.

2. Создание программного продукта (панели администратора), позволяющего управлять сущностями системы, просматривать успеваемость учащихся.

3. Создание серверного модуля (далее – программный продукт), обеспечивающего реализацию бизнес-процессов системы и интеграцию портала и панели администратора с единым источником правды.

2.4 Условия применения системы

Клиентский модуль системы применяется в условиях веб-сервера. Сервер отвечает на HTTP запросы, возвращая HTTP ответ. Ответ – страница сайта в виде HTML документа, которая содержит ссылки на прикрепленные к документу CSS и JS файлы. Эти файлы будут загружены браузером автоматически.

Модуль панели администратора системы также применяется в условиях веб-сервера.

Серверный модуль системы разворачивается в серверном окружении. Другие модули системы могут обращаться к серверному модулю посредством RESTful API (HTTP запросов). БД модуля может использоваться непосредственно для построения аналитики по системе.

3 ТРЕБОВАНИЯ К СИСТЕМЕ

3.1 Требования к программному обеспечению

1. Программа должна быть совместима с серверным окружением ОС Linux (Debian 10) или Windows Server (2016 и старше).

2. Программа должна корректно работать без доступа к сети Интернет (работа в сети Интранет).

3.2 Требования к численности и квалификации персонала

1. Системный администратор. Требуется знание Linux, Windows, СУБД MySQL, PostgreSQL, SQLite, языка SQL, навыки обслуживания БД. Навыки работы с Postman и чтения логов.

2. Администратор. Требуются базовые навыки работы с ПК, оформлением материалов, знание системы.

3.3 Требования к функционалу

Программный продукт поставляется как единая и неделимая единица программного обеспечения. Подключение или отключение отдельных компонентов программного продукта без дополнительной разработки не предусмотрено.

Всю информацию, необходимую для функционирования системы хранить в реляционной СУБД. Предлагается использовать СУБД MySQL для развёртывания на production серверах и СУБД SQLite для тестового развёртывания системы.

Программный продукт должен обеспечивать прохождение всех шагов учебного процесса системы. Пример организации учебного процесса в системе:

1. Подготовительные этапы. Администратор вносит упражнения и задания в систему. Группы распределяются между тренерами. Тренеры устанавливают сроки прохождения заданий для групп.

2. Обучающимся сотрудникам дилерских центров, одновременно поступающих на обучение по единой программе, организовать в учебную группу (Group).

3. Группе объяснить принципы обучения, подход асинхронного обучения, назначить еженедельный звонок группы с Тренерами для обсуждения возникающих вопросов и т.д.

4. Тренер определяет учебный план группы, собранный из существующих блоков-упражнений. При необходимости – администратор добавляет новые блоки.

5. Тренер вносит расписание блоков в систему: какое задание можно выполнять в какие даты. При этом последовательность и зависимость блоков друг от друга определена администратором. Тренер может лишь блокировать/разблокировать блоки в определенные временные периоды.

6. Начало учебного периода.

7. Тренер открывает лигу для учащихся на весь период обучения или его модуль. Конкуренция за баллы в рамках лиги поощряет учащихся меньше ошибаться, проходить задания быстрее и выполнять дополнительные упражнения, которые не являются обязательными для завершения курса.

8. Учебная группа приступает к прохождению блоков. В случае возникновения вопросов – они поднимаются на общем звонке.

9. Тренер параллельно проверяет упражнения с загрузкой ответа и выставляет баллы.

10. Завершение обучения.

11. Автоматически завершается лига, подсчитывается рейтинг учащихся, победитель получает награду-достижение в профиль.

12. Определяется список учеников, выполнивших все блоки для успешного завершения обучения. Они считаются успешно завершившими обучение.

Программный продукт должен реализовывать все бизнес-процессы системы:

- процессы авторизации;
- процессы, связанные с выполнением образовательных блоков и оцениванием их выполнения;
- процессы, связанные с функционированием лиг и достижений;
- процессы, связанные с администрированием системы, созданием, изменением и удалением её сущностей.
- процессы, связанные с выводом различного рода информации о состоянии системы в других компонентах системы.

Доступ к этим процессам должен осуществляться другими компонентами через RESTful API.

Программный продукт должен поддерживать зависимость образовательных блоков многие к одному (Рис. 1). Так, для прохождения следующего блока, учащемуся необходимо выполнить несколько

предыдущих. Аналогичным образом, при выполнении одного блока, пользователю может стать доступным сразу несколько новых блоков.

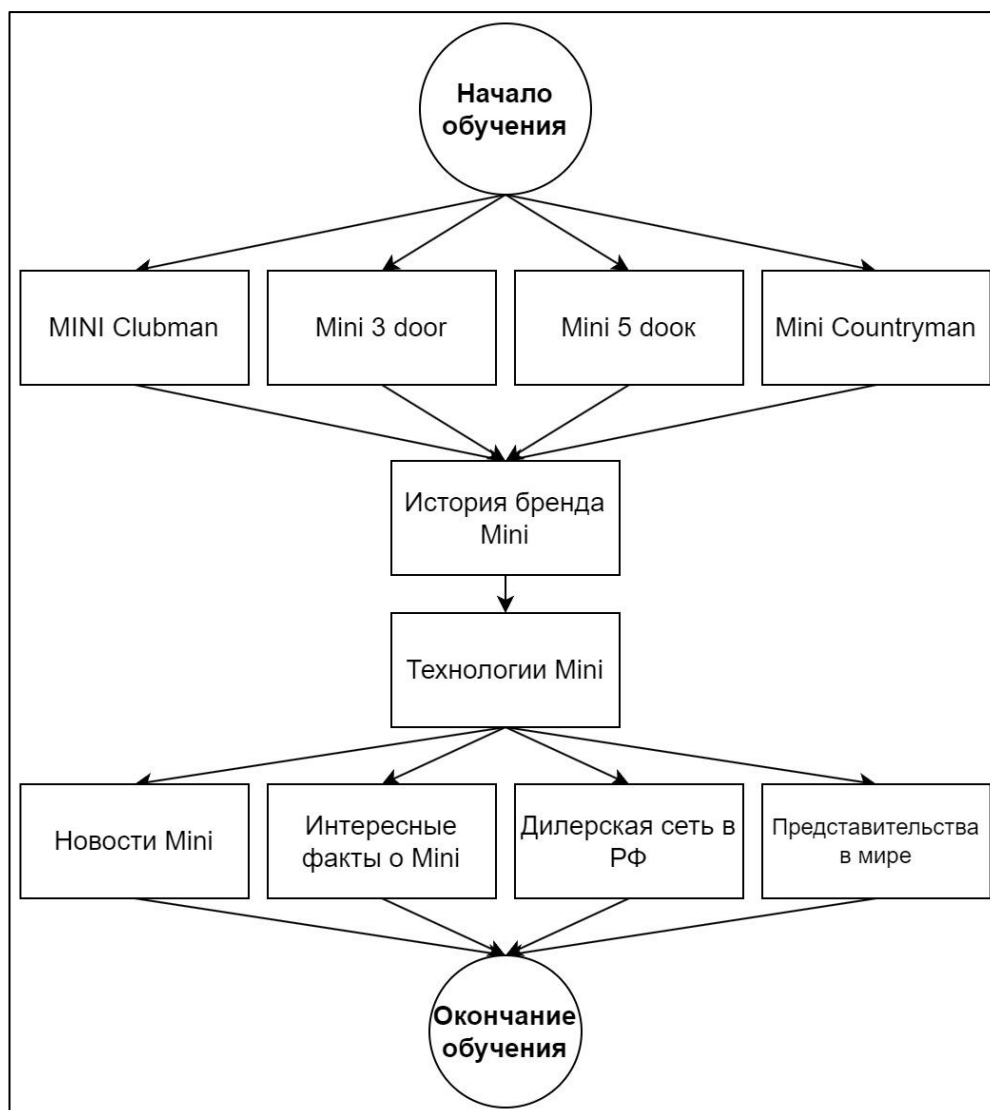


Рисунок 1 – Пример структуры зависимости блоков

3.4 Группы пользователей

Учащиеся. Проходят образовательные блоки, конкурируют за места в лигах, получают достижения и т.д.

Тренеры. Устанавливают сроки прохождения блоков, устраивают лиги, просматривают успеваемость учащихся.

Администраторы. Управляют содержанием образовательных блоков. Управляют пользователями системы.