

Министерство науки и высшего образования  
Российской Федерации  
ФГБОУ ВО  
«Уральский государственный горный университет»  
Кафедра информатики

УТВЕРЖДАЮ

Зав. кафедрой информатики

\_\_\_\_\_ А.В. Дружинин

“10” июня 2024г.

**ИССЛЕДОВАНИЕ И РАЗРАБОТКА**  
**ПРОЦЕССА ФОРМИРОВАНИЯ**  
**ИНДИВИДУАЛЬНОЙ ТРАЕКТОРИИ**  
**ДОПОЛНИТЕЛЬНОГО ОБУЧЕНИЯ**

**Выпускная квалификационная работа магистра**

**Пояснительная записка**

Руководитель ВКР,  
кандидат технических наук

\_\_\_\_\_  
(подпись, дата)

А. В. Дружинин

\_\_\_\_\_  
(и. о. ф.)

Разработал студент

\_\_\_\_\_  
(подпись, дата)

А. В. Цапков

\_\_\_\_\_  
(и. о. ф.)

Группа

ИНФ.м-23

Екатеринбург, 2024

## СОДЕРЖАНИЕ

АННОТАЦИЯ .....	5
ABSTRACT .....	6
ВВЕДЕНИЕ.....	8
1 АНАЛИЗ ПРОБЛЕМАТИКИ.....	11
1.1. Характеристика проблемной ситуации/области .....	11
1.2. Аналитический обзор существующих решений .....	15
1.3 Основные положения, выносимые на защиту .....	19
ВЫВОДЫ ПО ГЛАВЕ 1 .....	19
2 ПРЕДЛАГАЕМОЕ РЕШЕНИЕ .....	21
2.1 Системный анализ проблемной ситуации/области .....	21
2.1.1 Математическая модель объекта управления .....	21
2.1.2 Концептуальное моделирование .....	28
2.2 Функциональное моделирование.....	38
2.3 Формирование требований к информационной системе .....	40
2.3.1 Требования к системе .....	40
2.3.2 Требования к надежности .....	47
2.3.3 Требования к квалификации персонала.....	49
2.3.4 Показатели назначения.....	50
2.3.5 Требования к приспособляемости системы к изменениям .....	52
2.3.6 Требования к пользовательскому интерфейсу и удобству использования .....	53
2.3.7 Требования к защите информации от несанкционированного доступа .....	55
ВЫВОДЫ ПО 2 ГЛАВЕ .....	58
3 ПРОВЕДЕНИЕ ЭКСПЕРИМЕНТОВ И АНАЛИЗ РЕЗУЛЬТАТОВ .....	60
3.1 Проектирование информационной системы .....	60
3.1.1 Выбор архитектуры информационной системы .....	60
3.2 Инфологическое проектирование .....	63
3.3 Проектирование интерфейса .....	66
3.4 Программная реализация проекта .....	69
3.4.1 Выбор средств реализации .....	69
3.4.2 Программная реализация .....	71
3.5 Проведение эксперимента .....	77
ВЫВОДЫ ПО 3 ГЛАВЕ .....	82
ЗАКЛЮЧЕНИЕ .....	84
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	87

					Д 09.04.01 2023 23262867 МД			
Должность	Фамилия	Подпись	Дата	Исследование и разработка процесса формирования индивидуальной траектории дополнительного обучения	Лит.	Масса	Масш	
Студент	Цапков А. В.		10.06.25					
Руководит.	Дружинин А. В.		10.06.25					
Консульт.					Лист	2	Листов	87
Н. Контроль	Дружинин А. В.		10.06.25		ИНФ.м-23 УГГУ, кафедра информатики			
Рецензент				Содержание				
Зав.кафедр	Дружинин А. В.		10.06.25					



## АННОТАЦИЯ

					Д 09.04.01 2023 23262867 МД	Лист
		Цапков А.В.		10.06.25		
Студ.	Фамилия	Подп.	Дата			4

## **АННОТАЦИЯ**

**Название ВКР:** Исследование и разработка процесса формирования индивидуальной траектории дополнительного обучения.

**ФИО автора:** Цапков Артём Вячеславович.

**Научный руководитель:** Кандидат технических наук Алексей Владимирович Дружинин.

**Ключевые слова:** траектория обучения, индивидуальное обучение, традиционное обучение, план обучения, компьютерный анализ успеваемости учащихся, преподавание.

**Цель исследования:** разработка автоматизированной системы формирования индивидуальной траектории обучения.

**Научная новизна** работы заключается в разработке системы анализа успеваемости и подбора на основе этой успеваемости своей траектории обучения для каждого из учащихся. Выпускная квалификационная работа посвящена исследованию, проектированию и разработке автоматизированной WEB-платформы для дополнительного обучения. Пояснительная записка выпускной квалификационной работы состоит из аннотации, введения, трёх разделов и заключения. Работа содержит 87 страницы, на которых размещены 32 рисунок, 6 таблицы, 6 листинга программного кода, 11 литературный источник. В работе выполнено исследование вопроса персонального подбора и способов анализа успеваемости, спроектирована база данных и интерфейс, сформулированы требования к системе. Спроектирован, разработан и представлен сервис для автоматизированного формирования индивидуальной траектории обучения. Внедрение данного продукта позволит изменить подход в дополнительном обучении.

Список трудов автора по тематике работы:

1. Цапков А. В., Нагаткин Е. Ю. ПРИМЕНЕНИЕ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ И АНАЛИЗА ДАННЫХ ДЛЯ ФОРМИРОВАНИЯ ИНДИВИДУАЛЬНОЙ ТРАЕКТОРИИ ОБУЧЕНИЯ// Международная научно-практическая конференция «Уральская горная школа – регионам» (сборник докладов). Екатеринбург: издательство УГГУ, 2025 г.

2. Цапков А. В., Нагаткин Е. Ю. СПОСОБЫ ФОРМИРОВАНИЯ ИНДИВИДУАЛЬНЫХ ОБРАЗОВАТЕЛЬНЫХ ТРАЕКТОРИЙ// Международная научно-практическая конференция «Уральская горная школа – регионам» (сборник докладов). Екатеринбург: издательство УГГУ, 2025 г. (готовится к печати)

## **ABSTRACT**

**Title of the final qualification work:** Research and development of the process of forming an individual trajectory of additional education.

**Full name of the author:** Tsapkov Artem Vyacheslavovich.

**Scientific adviser:** Candidate of technical sciences Alexey Vladimirovich Druzhinin

**Keywords:** learning trajectory, individual learning, traditional learning, learning plan, computer analysis of student performance, teaching.

**The objective of the research** is development of an automated system for forming an individual learning trajectory.

**The scientific novelty of the work** scientific novelty lies in the development of a system for analyzing academic performance and selecting an individual learning trajectory for each student based on their performance. The thesis is dedicated to the research, design, and development of an automated web platform for additional education. The explanatory note of the thesis consists of an abstract, introduction, three sections, and a conclusion. The work contains 87 pages, including 32 figures, 6 tables, 6 code listings, and 11 literary sources. The study investigates the personal selection and methods of analyzing academic performance, designs a database and interface, and formulates system requirements. A service for the automated formation of an individual learning trajectory has been designed, developed, and presented. The implementation of this product will change the approach to additional education.

List of the Author's Works on the Thesis Topic:

Tsapkov A. V., Nagatkin E. Yu. APPLICATION OF MACHINE LEARNING ALGORITHMS AND DATA ANALYSIS FOR FORMING AN INDIVIDUAL LEARNING TRAJECTORY // International Scientific and Practical Conference "Ural Mining School - to the Regions" (collection of reports). Ekaterinburg: UGGU Publishing House, 2025.

Tsapkov A. V., Nagatkin E. Yu. METHODS OF FORMING INDIVIDUAL EDUCATIONAL TRAJECTORIES // International Scientific and Practical Conference "Ural Mining School - to the Regions" (collection of reports). Ekaterinburg: UGGU Publishing House, 2025. (in press)

## ВВЕДЕНИЕ

					Д 09.04.01 2023 23262867 МД	Лист
		Цапков А.В.		10.06.25		
Студ.	Фамилия	Подп.	Дата			7

## **ВВЕДЕНИЕ**

### **Актуальность проблемы.**

В связи с развитием алгоритмов анализа данных и нейронных сетей появляется больше возможностей для изменений в сфере образования, а именно оптимизации и автоматизации формирования индивидуальной траектории обучения. Создание доступной платформы, которая имела необходимый функционал для этих целей является одним из этапов развития области индивидуального образования, эффективность которого подтверждается многими исследованиями.

**Объектом исследования** является автоматизированное формирование индивидуальной траектории обучения.

**Предметом исследования** является непрерывные анализ успеваемости учащихся и формирование наиболее подходящих материалов для обучения.

**Цель исследования:** разработка автоматизированной системы дополнительного обучения с использованием алгоритмов анализа и подбора индивидуального плана обучения.

### **Задачи исследования:**

- исследование существующих подходов и решений автоматизированного построения индивидуальной траектории обучения.
- исследование усовершенствования возможности анализа индивидуальных особенностей учащихся при подборе материала с минимальным участием человека.
- разработка сервиса для обучения с подбором рекомендаций учебных материалов.
- разработка алгоритмов анализа и рекомендаций, добавление базовых материалов на платформу.

**Научная новизна** работы заключается в разработке алгоритмов, которые позволяют автоматизировать систему рекомендаций по индивидуальному обучению, предоставляя возможность сформировать траекторию обучения под конкретного учащегося. Кроме этого, данный сервис, благодаря данным об успеваемости учащихся разделяет их на группы, предоставляя социальное взаимодействие между друг другом и обеспечивая комфортную среду при лекционных занятиях, благодаря схожим тенденциям, прогрессу и слабым сторонам участников одной группы. Так же на платформе доступна отчетность по успеваемости каждого студента в понятном и удобном формате, что повышает качество обратной связи для пользователя.



**Практические результаты** работы заключаются в разработке системы дополнительного образования с использованием анализа данных и машинных алгоритмов, которая собирает и анализирует результаты учебной деятельности, подбирает учебные материалы для каждого учащегося, учитывая его индивидуальные сильные и слабые стороны, формирует группы учащихся ориентируясь, на показатели обучения.

Также в рамках исследования был проведен эксперимент: эксперимент по сравнению эффективности обучения с использованием системы индивидуального формирования траектории обучения и традиционным способом.

Публикации автора по тематике исследования:

1. Цапков А. В., Нагаткин Е. Ю. ПРИМЕНЕНИЕ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ И АНАЛИЗА ДАННЫХ ДЛЯ ФОРМИРОВАНИЯ ИНДИВИДУАЛЬНОЙ ТРАЕКТОРИИ ОБУЧЕНИЯ// Международная научно-практическая конференция «Уральская горная школа – регионам» (сборник докладов). Екатеринбург: издательство УГГУ, 2025 г.

2. Цапков А. В., Нагаткин Е. Ю. СПОСОБЫ ФОРМИРОВАНИЯ ИНДИВИДУАЛЬНЫХ ОБРАЗОВАТЕЛЬНЫХ ТРАЕКТОРИЙ// Международная научно-практическая конференция «Уральская горная школа – регионам» (сборник докладов). Екатеринбург: издательство УГГУ, 2025 г. (готовится к печати)

## 1 АНАЛИЗ ПРОБЛЕМАТИКИ

					Д 09.04.01 2023 23262867 МД	Лист
		Цапков А.В.		10.06.25		
Студ.	Фамилия	Подп.	Дата			10

## **1 АНАЛИЗ ПРОБЛЕМАТИКИ**

### **1.1. Характеристика проблемной ситуации/области**

В современной системе образования наблюдается растущий интерес к индивидуальному подходу в обучении. Этот подход, также известный как персонализированное обучение, предполагает адаптацию учебного процесса к уникальным потребностям, интересам и способностям каждого учащегося. В отличие от традиционных методов, которые часто фокусируются на средних значениях и стандартизированных программах, индивидуальный подход стремится учитывать разнообразие студентов и создавать условия для их максимального развития. Так средний подход, в сравнении с индивидуальным критикуется американским ученым Тоддом Росом (Todd Rose). В целом, книга Роса предлагает убедительные аргументы в пользу индивидуального подхода к обучению, подчеркивая важность учета индивидуальных различий, использования технологий и данных, а также создания гибких и адаптивных образовательных моделей [1].

Индивидуальный подход в обучении имеет глубокие исторические корни, но его актуальность и значимость особенно возросли в последние десятилетия с развитием технологий и изменением образовательных парадигм. Современные технологии, такие как адаптивные образовательные платформы и аналитические инструменты, предоставляют новые возможности для персонализации обучения и мониторинга прогресса студентов [2][3]. Эти инновации позволяют учителям более эффективно адаптировать учебные материалы и методы к индивидуальным потребностям каждого студента, что способствует улучшению учебных достижений и мотивации [4]. Эффективность индивидуального подхода доказывается несколькими исследованиями. Одним из самых крупных является исследование Джон Хэтти [5]. В этой книге Джон проводит мета-анализ более 800 исследований, чтобы определить, какие факторы и методы обучения наиболее эффективны для достижения учебных результатов.

Одним из ключевых выводов книги является то, что индивидуальный подход к обучению, или персонализированное обучение, может быть очень эффективным. Хэтти использует концепцию "эффект размера" (effect size), чтобы количественно оценить влияние различных факторов на учебные достижения.

Исходя из исследований, выявлены следующие критерии, которые повышают эффективность обучения:

1. Персонализированное обучение
2. Обратная связь
3. Дифференцированное обучение

4. Индивидуальное наставничество
5. Адаптивное обучение

Однако, несмотря на многочисленные преимущества, индивидуальный подход сталкивается с рядом проблем и вызовов. Эти проблемы включают недостаток ресурсов и времени, необходимость дополнительной подготовки учителей, сложности в оценке и стандартизации, технологические барьеры, социальные и психологические аспекты. Кроме того, интеграция индивидуального подхода с традиционными методами обучения требует тщательного планирования и координации. Однако быстрый прогресс в технологиях искусственного интеллекта раскрывает новые возможности в формировании индивидуальной траектории обучения, позволяя часть необходимой подготовки и работы переложить на вычислительные способности компьютеров.

В функциональной модели IDEF0 (см. рисунок 1) основной процесс «как есть», включающий все действия, связанные с созданием персонализированного учебного плана для учащегося. Входами являются: данные о учащихся, включая их профили, предпочтения и текущие достижения. Образовательные стандарты, требования и стандарты, установленные образовательной системой. Учебные материалы, доступные учебные ресурсы, включая курсы, модули, задания и мультимедийные материалы. Обратная связь - комментарии и рекомендации от преподавателей, наставников и других учащихся.

Управляющие воздействия: Образовательные цели - цели и задачи, которые необходимо достичь в процессе обучения. Политики и процедуры - правила и процедуры, регулирующие процесс обучения. Ресурсы - доступные ресурсы, такие как время, финансы и технологии.

Механизмы: образовательная платформа - программное обеспечение и технологии, используемые для управления учебным процессом. Преподаватели и наставники - люди, участвующие в процессе обучения и предоставляющие поддержку учащимся.

Выходы: индивидуальные учебные планы - персонализированные планы обучения для каждого учащегося. Отчеты о прогрессе - отчеты о текущем прогрессе учащихся и достигнутых результатах. Обратная связь - комментарии и рекомендации, предоставляемые учащимся. Прогнозы и рекомендации - прогнозы будущих достижений и рекомендации по улучшению учебного процесса.

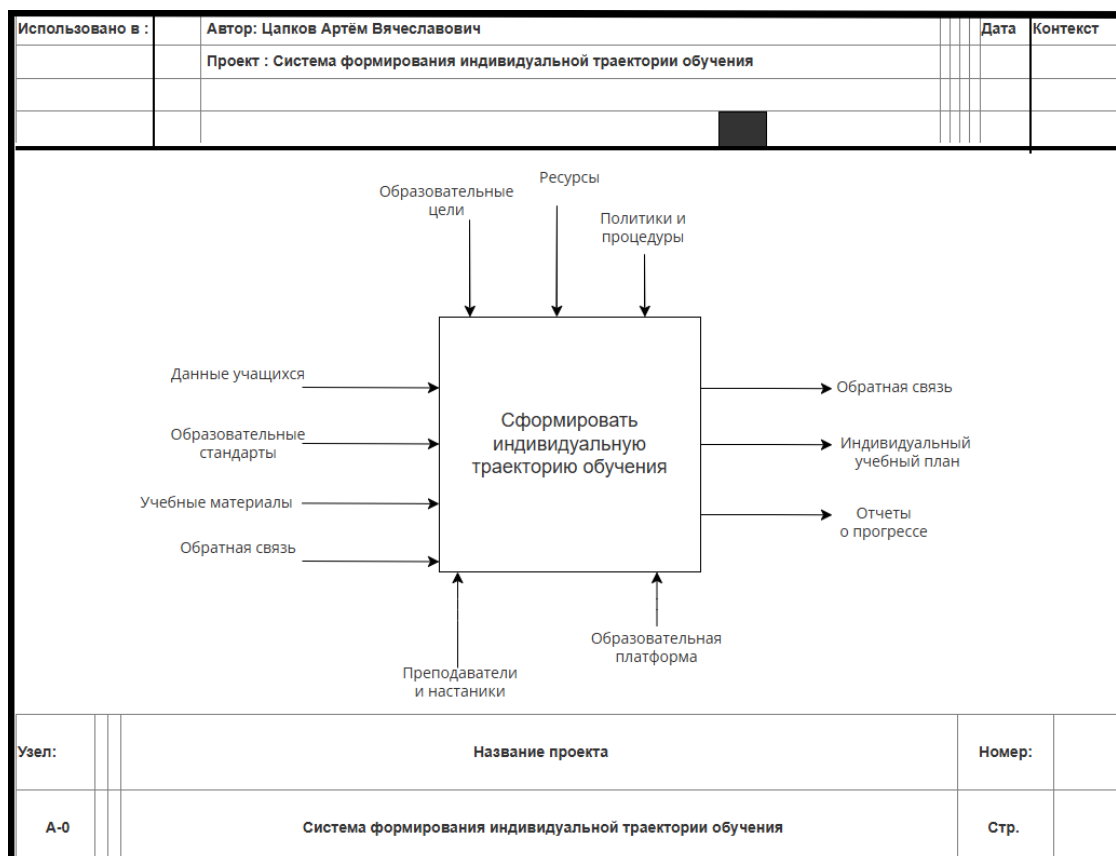


Рисунок 1 - IDEF0 модель уровня A0 “как есть”

Процесс формирования индивидуальной траектории обучения можно разделить на 4 основных, взаимосвязанных между собой, блока на рисунке 2.

- 1) Диагностировать и оценить учащихся
- 2) Разработать учебный план
- 3) Реализовать учебные планы
- 4) Оценить и скорректировать учебные планы

Выделим для каждого из блоков входные данные: Диагностика и оценка учащихся - данные о учащихся, образовательные стандарты. Разработка учебных планов - результаты диагностики, учебные материалы. Для реализация учебных планов ходами являются индивидуальные учебные планы и учебные материалы. Оценка и корректировка имеет входы: отчеты о прогрессе, обратная связь. “Политики и процедуры” как управляющие воздействия относятся к блокам A1 и A3. “Образовательные цели” для блока A1 и A2. “Ресурсы” управляющее воздействие для A2 и A4. Для всех блоков механизмами являются “Образовательная платформа” и “Преподаватели и наставники”. Для блока A4 есть отдельный механизм “Обратная связь”. Конечными выходными данными будут индивидуальные учебные планы, отчеты о прогрессе и обратная связь.

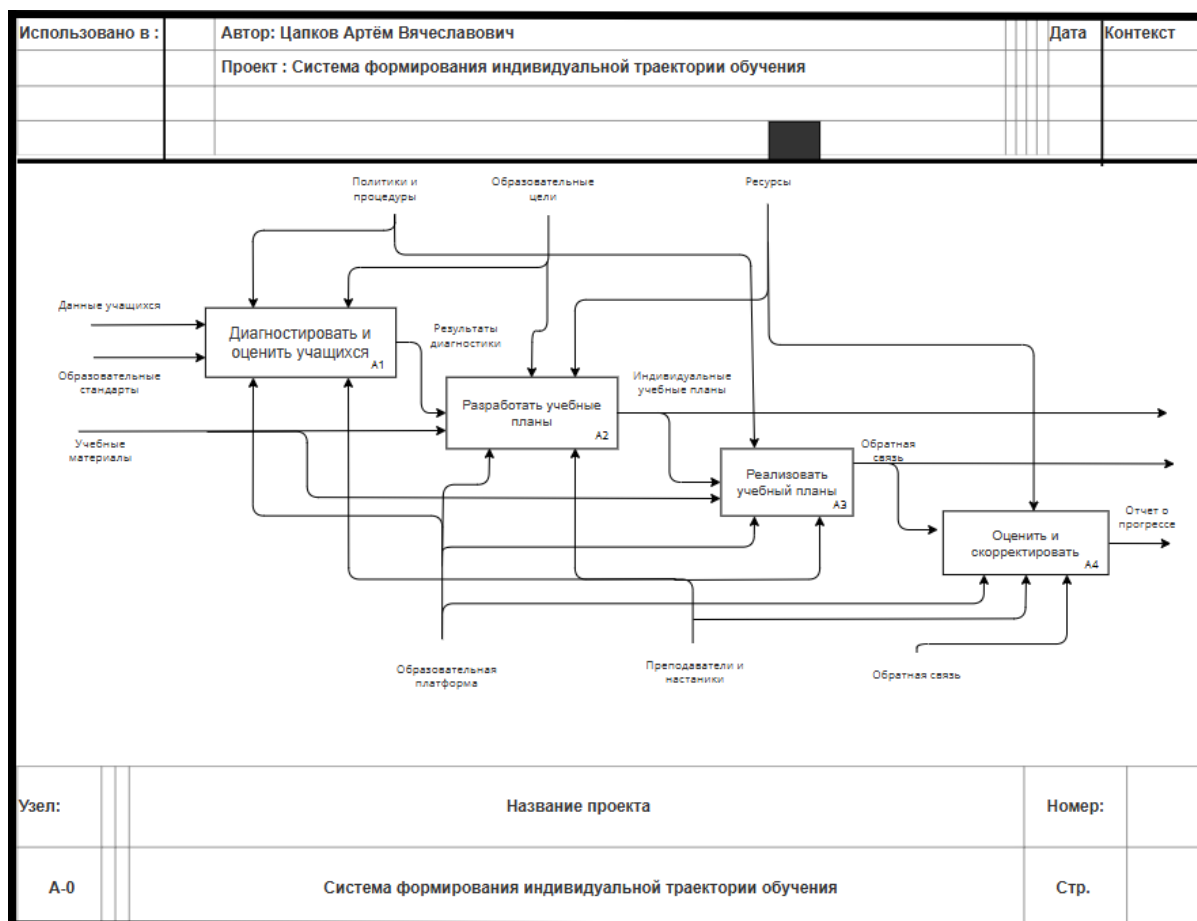


Рисунок 2 - IDEF0 модель уровня A1 “как есть”

Для всего процесса формирования индивидуальной траектории обучения требуется участие преподавателей и наставников, которые берут на себя задачи разработки плана, кроме и без того сложного процесса реализации учебного плана. Для оптимизации и автоматизации процесса формирования учебного плана, часть задач можно переложить на системы машинного обучения и воздействовать на процессы алгоритмами.

Цель данного исследования — проанализировать эффективность использования алгоритмов машинного обучения и анализа данных для формирования индивидуального плана и формирования индивидуального подхода в обучении, рассмотреть преимущества и вызовы их использования, а также предложить рекомендации по успешному внедрению в образовательную практику данных алгоритмов.

## 1.2. Аналитический обзор существующих решений

Искусственный интеллект (ИИ) активно используется в различных системах для поддержки индивидуального обучения. Большинство из них коммерческие. Рассмотрим несколько ключевых систем и платформ, которые применяют ИИ для персонализации образовательного процесса, а также их плюсы и минусы.

### 1. Khan Academy

Khan Academy — это бесплатная образовательная платформа, предлагающая видеоуроки, интерактивные упражнения и аналитические инструменты по широкому спектру предметов, включая математику, науки, искусство, экономику и многое другое. Интерфейс на рисунке 3.

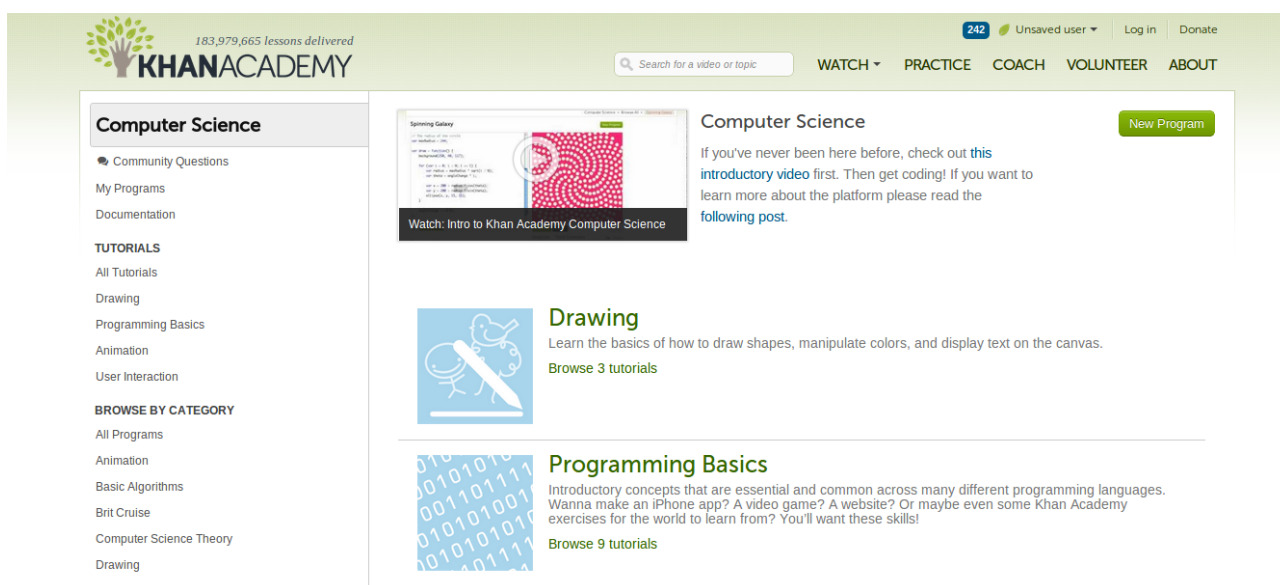


Рисунок 3 - Khan Academy

Использование ИИ:

- **Адаптивное обучение:** ИИ адаптирует учебные материалы и задания к уровню знаний и темпу обучения каждого студента. Платформа использует алгоритмы машинного обучения для анализа данных о прогрессе студентов и предоставления персонализированных рекомендаций.

- **Аналитика и отчеты:** Платформа предоставляет подробные отчеты о прогрессе студентов, что помогает учителям и родителям отслеживать достижения и выявлять проблемные области.

Плюсы:

- **Бесплатный доступ:** Все учебные материалы и ресурсы доступны бесплатно, что делает образование доступным для всех.

- Широкий спектр предметов: Платформа охватывает множество предметов и уровней сложности, от начальной школы до университетского уровня.
- Интерактивные упражнения: Интерактивные упражнения и задания позволяют студентам практиковаться и получать мгновенную обратную связь.
- Гибкость и доступность: Учебные материалы доступны в любое время и в любом месте, что позволяет студентам учиться в удобном для них темпе и графике.

Минусы:

- Ограниченное взаимодействие: Платформа в основном фокусируется на самостоятельное обучение, что может ограничивать возможности для взаимодействия с учителями и сверстниками.
- Качество контента: Некоторые видеоуроки и упражнения могут быть недостаточно глубокими или детализированными для некоторых студентов.

## 2. Duolingo

Duolingo на рисунке 4 — это платформа для изучения языков, предлагающая интерактивные уроки и упражнения.

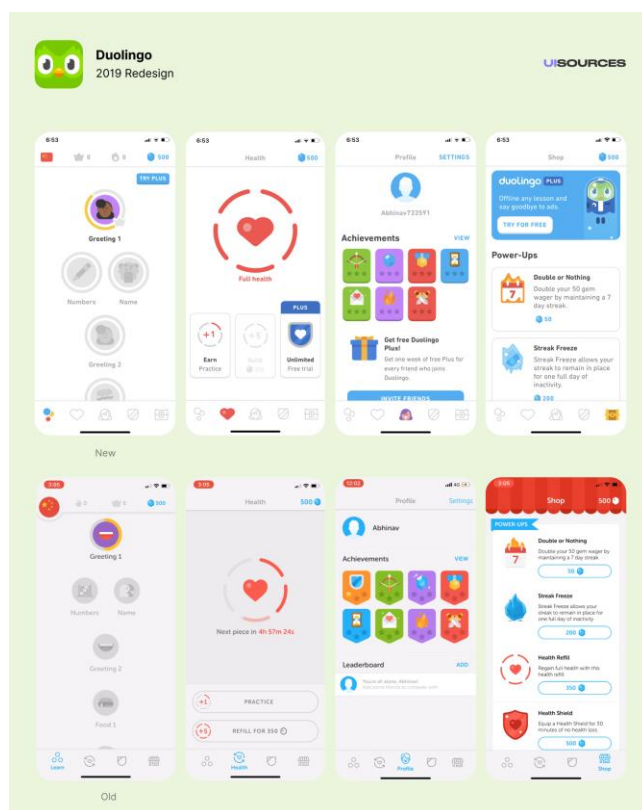


Рисунок 4 - Duolingo

Использование ИИ:

- Персонализация учебного плана: ИИ адаптирует уроки и упражнения к уровню владения языком и прогрессу каждого пользователя. Платформа использует алгоритмы



машинного обучения для анализа данных о прогрессе пользователей и предоставления персонализированных рекомендаций.

- **Аналитика и прогнозирование:** Платформа использует данные о прогрессе пользователей для предоставления персонализированных рекомендаций и прогнозов.

Плюсы:

- **Бесплатный доступ:** Основные функции платформы доступны бесплатно, что делает изучение языков доступным для всех.
- **Интерактивные и увлекательные уроки:** Уроки и упражнения на Duolingo интерактивны и увлекательны, что помогает удерживать внимание пользователей.
- **Поддержка множества языков:** Платформа поддерживает множество языков, что делает ее полезной для пользователей по всему миру.

Минусы:

- **Ограниченные возможности для глубокого изучения языка:** Хотя Duolingo отлично подходит для начального изучения языка, он может быть недостаточно глубоким для продвинутых пользователей.
- **Отсутствие индивидуальной обратной связи:** Платформа не предоставляет индивидуальной обратной связи от преподавателей, что может быть ограничением для некоторых пользователей.

### 3. Coursera

Coursera на рисунке 5 — это платформа для онлайн-курсов, предлагающая курсы от ведущих университетов и компаний.

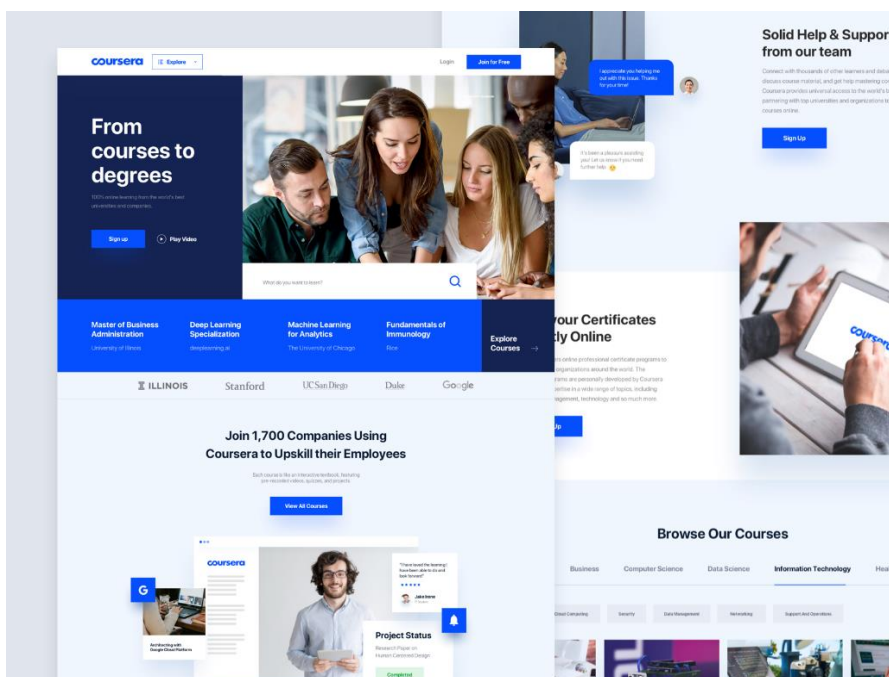


Рисунок 5 - Coursera

#### Использование ИИ:

- Рекомендательные системы: ИИ анализирует данные о прогрессе студентов и предлагает персонализированные рекомендации по курсам и учебным материалам. Платформа использует алгоритмы машинного обучения для анализа данных о прогрессе студентов и предоставления персонализированных рекомендаций.
- Аналитика и отчеты: Платформа предоставляет подробные отчеты о прогрессе студентов и их достижениях.

#### Плюсы:

- Широкий выбор курсов: Coursera предлагает широкий выбор курсов от ведущих университетов и компаний, что делает платформу полезной для студентов всех возрастов и уровней подготовки.
- Интерактивные уроки и задания: Курсы на Coursera включают интерактивные уроки и задания, что помогает улучшить понимание материала и закрепить знания.

#### Минусы:

- Некоторые курсы платные. Большинство из курсов требуют оплаты, что может быть ограничением для некоторых студентов.
- Ограниченные возможности для взаимодействия: Платформа может ограничивать возможности для взаимодействия с преподавателями и сверстниками, что может быть важным аспектом образовательного процесса для некоторых студентов.

Искусственный интеллект играет важную роль в персонализации образовательного процесса, предлагая адаптивные учебные планы, интерактивные уроки и аналитические инструменты. Однако, несмотря на многочисленные преимущества, существуют и ограничения, такие как ограниченное взаимодействие с преподавателями и сверстниками. В то время как важность двусторонней связи преподавателя и студента доказывается в нескольких научных исследованиях. Дональд Д. Фенкель рассматривает роль преподавателя как фасилитатора, который помогает студентам открывать знания самостоятельно. Он подчеркивает важность создания условий для активного и интерактивного обучения, где преподаватель играет ключевую роль в поддержке и направлении студентов [6]. Пауло Фрейре исследует роль преподавателя в освобождающем образовании, где студенты становятся активными участниками своего обучения. Он подчеркивает важность диалога и критического мышления в образовательном процессе и роли преподавателя [7].

### **1.3 Основные положения, выносимые на защиту**

1. Анализ с помощью интеллектуальных алгоритмов успеваемости учащихся, и подбор рекомендационных учебных материалов, на основе индивидуальных особенностей, позволит добиться ускоренного усвоения знаний, по сравнению с традиционным методом обучения.
2. Автоматическая корректировка образовательной траектории на основе анализа текущего прогресса учащегося способна повысить общую эффективность усвоения учебного материала в сравнении с традиционным методом обучения.
3. Разнообразные методы обучения способствуют развитию различных аспектов интеллекта и формируют специфические учебные навыки в зависимости от основного акцента в образовательном процессе.

### **ВЫВОДЫ ПО ГЛАВЕ 1**

Цель исследовательской работы заключается в создании системы, которая использует искусственный интеллект для формирования индивидуальных траекторий обучения. Эта система должна быть адаптивной, персонализированной и интерактивной. В результате, система должна способствовать улучшению учебных достижений, повышению мотивации и вовлеченности студентов, а также обеспечить двустороннюю связь между преподавателем и студентом. Создать симбиоз между действиями преподавателей, учеников и алгоритмами машинного обучения. Все это сопровождается постоянным анализом достижений и знаний учеников, что способствует обратной связи и возможности корректировать траекторию обучения, укреплять усвоенные знания и получать новые.

Структурный анализ конкурентов позволил выявить основные проблемы и недоработки, связанные с взаимодействием человека учителя и ученика. Важность учителя человека, в первую очередь, объясняется важностью психологического взаимодействия людей. Человек лучше поддержит и подскажет, чем компьютер. В обучении всё еще важную часть составляет преподаватель.

2 ПРЕДЛАГАЕМОЕ РЕШЕНИЕ

					Д 09.04.01 2023 23262867 МД	Лист
		Цапков А.В.		10.06.25		
Студ.	Фамилия	Подп.	Дата			20

## **2 ПРЕДЛАГАЕМОЕ РЕШЕНИЕ**

### **2.1 Системный анализ проблемной ситуации/области**

#### **2.1.1 Математическая модель объекта управления**

Математическая модель объекта управления в формировании индивидуальной траектории обучения представляет собой формализованное описание процесса обучения, которое позволяет количественно оценивать и прогнозировать результаты обучения, а также оптимизировать учебный процесс. Рассмотрим основные элементы такой модели:

##### **1. Определение переменных:**

$X(t)$  - Вектор состояния учащегося в момент времени  $t$ . Это может включать уровень знаний, навыки, мотивацию и другие характеристики.

$U(t)$  - Вектор управления, представляющий собой учебные воздействия, такие как учебные материалы, методы обучения, обратная связь и т.д.

$Y(t)$  - Вектор выходов, представляющий собой результаты обучения, такие как оценки, тесты, проекты и т.д.

$D(t)$  - Вектор внешних воздействий, таких как социальное окружение, семейные условия, здоровье и т.д.

##### **2. Уравнения состояния**

Уравнения состояния описывают, как состояние учащегося изменяется под воздействием учебных и внешних факторов по формуле [1]:

$$\frac{dX(t)}{dt} = f(X(t), U(t), D(t), t) \quad (1)$$

где  $f$  - функция, описывающая динамику изменения состояния учащегося.

##### **3. Уравнения выходов**

Уравнения выходов связывают состояние учащегося с наблюдаемыми результатами обучения по формуле [2]:

$$Y(t) = g(X(t), U(t), D(t), t) \quad (2)$$

где  $g$  - функция, описывающая зависимость результатов обучения от состояния, управления и внешних воздействий.

##### **4. Целевая функция**

Целевая функция  $J$  определяет критерий оптимальности учебного процесса. Она может быть сформулирована как интегральная функция, минимизирующая или максимизирующая определенные показатели по формуле [3]:

$$J = \int_{t_0}^{t_1} L(X(t), U(t), D(t), t) dt \quad (3)$$

где  $L$  - функция потерь или выигрыша,  $t_0, t_1$  - начальное и конечное время обучения.

## 5. Ограничения

Модель может включать различные ограничения, такие как:

Ограничения на управление:  $U(t) \in u$ , где  $u$  - допустимое множество учебных воздействий.

Ограничения на состояние:  $X(t) \in x$ , где  $x$  - допустимое множество состояний учащегося.

Ограничения на выходы:  $Y(t) \in y$ , где  $y$  - допустимое множество результатов обучения.

## 6. Пример модели

Рассмотрим упрощенный пример модели:

- Состояние:  $X(t) = [K(t), M(t)]$ , где  $K(t)$  - уровень знаний,  $M(t)$  – мотивация.
- Управление:  $U(t) = [T(t), F(t)]$ , где  $T(t)$  - время, затраченное на обучение,  $F(t)$

- обратная связь.

- Внешние воздействия:  $D(t) = [S(t), H(t)]$ , где  $S(t)$  – социальное окружение,

$H(t)$  – здоровье.

- Уравнения состояния [4] и [5]:

$$\frac{dK(t)}{dt} = a * T(t) * M(t) - b * (1 - F(t)), \quad (4)$$

$$\frac{dM(t)}{dt} = c * F(t) * S(t) - d * (1 - H(t)) \quad (5)$$

- Уравнения выходов [6]:

$$Y(t) = e * K(t) + f * M(t) \quad (6)$$

- Целевая функция [7]:

$$J = \int_{t_0}^{t_f} (K(t) - K_{target})^2 + (M(t) - M_{target})^2 dt \quad (7)$$

Где  $a, b, c, d, e, f$  - коэффициенты,  $K_{target}$  и  $M_{target}$  — целевые уровни знаний и мотивации.

### **Стратегия предлагаемая, как базовая на платформе:**

#### **1. Диагностика и оценка**

*Начальная диагностика:* Проведение начальной диагностики для оценки текущего уровня знаний, навыков и компетенций учащегося. Это может включать тестирование, интервью, анализ практических навыков и т.д.

*Математическая модель:* Пусть  $S_0$  - начальный уровень знаний учащегося,  $T_0$  - результаты тестирования,  $I_0$  - результаты интервью,  $P_0$  - анализ практических навыков. Тогда начальная диагностика может быть представлена как формула [8]:

$$S_0 = f(T_0, I_0, P_0) \quad (8)$$

где  $f$  - функция, агрегирующая результаты различных методов диагностики.

*Постоянная оценка:* Регулярное проведение оценочных мероприятий для отслеживания прогресса учащегося и корректировки учебного плана.

*Математическая модель:* Пусть  $S(t)$  - уровень знаний учащегося в момент времени  $t$ ,  $A(t)$  - результаты текущей оценки. Тогда постоянная оценка может быть представлена как формула:

$$S(t) = S(t - 1) + g(A(t)) \quad (9)$$

где  $g$  - функция, описывающая влияние текущей оценки на уровень знаний.

#### **2. Персонализированный учебный план**

*Индивидуальные цели:* Определение индивидуальных целей обучения на основе диагностики и потребностей учащегося.

*Математическая модель:* Пусть  $G$  - вектор целей обучения,  $N$  - потребности учащегося,  $S_0$  - начальный уровень знаний. Тогда цели представлены формулой [10]:

$$G = h(N, S_0) \quad (10)$$

где  $h$  - функция, определяющая цели на основе потребностей и начального уровня знаний.

*Модульная структура:* Разделение учебного материала на модули, которые можно комбинировать в зависимости от целей и интересов учащегося.

*Математическая модель:* Пусть  $M_i$  - модуль учебного материала,  $C_i$  — цель, соответствующая модулю. Тогда модульная структура может быть представлена как [11]:

$$M = \{M_i | C_i \in G\} \quad (11)$$

где  $M$  - множество модулей, соответствующих целям обучения.

*Адаптивные маршруты:* Создание адаптивных учебных маршрутов, которые могут изменяться в зависимости от прогресса и потребностей учащегося.

*Математическая модель:* Пусть  $R(t)$  - учебный маршрут в момент времени  $t$ ,  $P(t)$  - прогресс учащегося. Тогда адаптивный маршрут может быть представлен как формула [12]:

$$R(t) = R(t - 1) + k(P(t)) \quad (12)$$

где  $k$  - функция, описывающая изменение маршрута на основе прогресса.

### **3. Использование данных и аналитики**

*Сбор данных:* Сбор данных о прогрессе учащегося, его активности на платформе, результатах тестирования и т.д.

*Математическая модель:* Пусть  $D(t)$  - данные, собранные в момент времени  $t$ ,  $A(t)$  - активность учащегося,  $T(t)$  — результаты тестирования. Тогда сбор данных может быть представлен как функция [13]:

$$D(t) = \{A(t), T(t), \dots\} \quad (13)$$

*Анализ данных:* Использование аналитических инструментов для анализа собранных данных и выявления тенденций, проблем и возможностей.

*Математическая модель:* Пусть  $T(t)$  - тенденции,  $P(t)$  - проблемы,  $O(t)$  - возможности. Тогда анализ данных может быть представлен как [14]:



$$\{T(t), P(t), O(t)\} = l(D(t)) \quad (14)$$

где  $l$  - функция анализа данных.

*Персонализированные рекомендации:* Предоставление учащемуся персонализированных рекомендаций на основе анализа данных.

*Математическая модель:* Пусть  $Rec(t)$  - рекомендации в момент времени  $t$ . Тогда персонализированные рекомендации могут быть представлены как формула [15]:

$$Rec(t) = m(T(t), P(t), O(t)) \quad (15)$$

где  $m$  - функция, генерирующая рекомендации на основе анализа данных.

#### **4. Интерактивные и мультимедийные материалы**

*Разнообразие форматов:* Использование различных форматов учебных материалов, таких как видео, аудио, интерактивные симуляции, игры и т.д.

*Математическая модель:* Пусть  $F$  - множество форматов учебных материалов,  $V$  - видео,  $A$  - аудио,  $S$  - симуляции,  $G$  - игры. Тогда разнообразие форматов может быть представлено как формула [16]:

$$F = \{V, A, S, G, \dots\} \quad (16)$$

*Адаптивные задания:* Создание заданий, которые адаптируются под уровень знаний и навыков учащегося.

*Математическая модель:* Пусть  $Z(t)$  - задание в момент времени  $t$ ,  $L(t)$  - уровень знаний учащегося. Тогда адаптивное задание может быть представлено как формула [17]:

$$Z(t) = n(L(t)) \quad (17)$$

где  $n$  - функция, адаптирующая задание под уровень знаний.

#### **5. Обратная связь и поддержка**

*Регулярная обратная связь:* Предоставление учащемуся регулярной обратной связи о его прогрессе и достижениях.

*Математическая модель:* Пусть  $FB(t)$  - обратная связь в момент времени  $t$ ,  $P(t)$  - прогресс учащегося. Тогда обратная связь может быть представлена как формула [18]:

$$FB(t) = o(P(t)) \quad (18)$$

где  $o$  - функция, генерирующая обратную связь на основе прогресса.

*Поддержка наставников:* Включение в процесс обучения наставников или менторов, которые могут помогать учащемуся в достижении его целей.

*Математическая модель:* Пусть  $Mentor(t)$  поддержка наставника в момент времени  $t$ ,  $G$  - цели учащегося. Тогда поддержка наставника может быть представлена как формула [19]:

$$Mentor(t) = p(G) \quad (19)$$

где  $t$  - функция, определяющая поддержку наставника на основе целей.

*Сообщество учащихся:* Создание сообщества учащихся для обмена опытом, совместного решения задач и поддержки.

*Математическая модель:* Пусть  $C(t)$  - сообщество учащихся в момент времени  $t$ ,  $E(t)$  - опыт учащихся,  $S(t)$  - совместные задачи. Тогда сообщество представлено на формуле (20):

$$C(t) = q(E(t), S(t)) \quad (20)$$

где  $q$  - функция, формирующая сообщество на основе опыта и задач.

## 6. Геймификация и мотивация

*Элементы геймификации:* Внедрение элементов геймификации, таких как баллы, значки, уровни и т.д., для повышения мотивации учащегося.

*Математическая модель:* Пусть  $Gam(t)$  - элементы геймификации в момент времени  $t$ ,  $Points(t)$  - баллы,  $Badges(t)$  - значки,  $Levels(t)$  - уровни. Тогда геймификация может быть представлена как формула (21):

$$Gam(t) = \{Points(t), Badges(t), Levels(t), \dots\} \quad (21)$$

*Индивидуальные мотиваторы:* Определение и использование индивидуальных мотиваторов учащегося для поддержания его интереса к обучению.

*Математическая модель:* Пусть  $Mot(t)$  - мотиваторы в момент времени  $t$ ,  $I$  - интересы учащегося. Тогда мотиваторы могут быть представлены как формула [22]:

$$Mot(t) = r(I) \quad (22)$$

где  $r$  - функция, определяющая мотиваторы на основе интересов.

## 7. Интеграция с внешними ресурсами

*Доступ к библиотекам и базам данных:* Интеграция платформы с внешними библиотеками, базами данных и образовательными ресурсами.

*Математическая модель:* Пусть  $R$  - множество внешних ресурсов,  $L$  - библиотеки,  $DB$  - базы данных,  $ER$  - образовательные ресурсы. Тогда интеграция может быть представлена как формула [23]:

$$R = \{L, DB, ER, \dots\} \quad (23)$$

## 8. Безопасность и конфиденциальность

*Защита данных:* Обеспечение защиты данных учащегося и конфиденциальности его личной информации.

*Математическая модель:* Пусть  $Sec(t)$  - уровень защиты данных в момент времени  $t$ ,  $D(t)$  - данные учащегося. Тогда защита данных может быть представлена как формула [24]:

$$Sec(t) = s(D(t)) \quad (24)$$

где  $s$  - функция, обеспечивающая защиту данных.

*Этические нормы:* Соблюдение этических норм при сборе и использовании данных учащегося.

*Математическая модель:* Пусть  $Eth(t)$  - этические нормы в момент времени  $t$ ,  $D(t)$  - данные учащегося. Тогда этические нормы могут быть представлены как формула [25]:

$$Eth(t) = e(D(t)) \quad (25)$$

где  $e$  - функция, обеспечивающая соблюдение этических норм.

## 9. Постоянное улучшение

*Обратная связь от пользователей:* Сбор обратной связи от учащихся и преподавателей для улучшения платформы.

*Математическая модель:* Пусть  $UFB(t)$  - обратная связь от пользователей в момент времени  $t$ ,  $U$  - пользователи. Тогда обратная связь может быть представлена как формула [26]:

$$UFB(t) = u(U) \quad (26)$$

где  $u$  — функция, собирающая обратную связь от пользователей.

*Обновление контента:* Регулярное обновление учебных материалов и заданий для соответствия современным требованиям и стандартам.

*Математическая модель:* Пусть  $Cont(t)$  - контент в момент времени  $t$ ,  $Std(t)$  - стандарты. Тогда обновление контента может быть представлено как формула [27]:

$$Cont(t) = v(Std(t)) \quad (27)$$

где  $v$  - функция, обновляющая контент на основе стандартов.

*Исследования и инновации:* Проведение исследований и внедрение инноваций для улучшения качества обучения.

*Математическая модель:* Пусть  $Inn(t)$  - инновации в момент времени  $t$ ,  $Res(t)$  - исследования. Тогда инновации могут быть представлены как формула [28]:

$$Inn(t) = w(Res(t)) \quad (28)$$

где  $w$  функция, внедряющая инновации на основе исследований.

### 2.1.2 Концептуальное моделирование

Современные образовательные системы сталкиваются с необходимостью адаптации к индивидуальным потребностям учащихся. В условиях цифровизации и автоматизации образования особую значимость приобретает разработка гибких систем, способных к постоянной корректировке образовательного процесса. Особое внимание в современных исследованиях уделяется механизмам постоянной корректировки образовательного процесса. В работах Эльконин, Д. Б.<sup>[8]</sup> подчеркивается важность анализа пробелов в знаниях для дальнейшего развития учащегося. В исследованиях Нечаевой Т. А.<sup>[9]</sup> рассматриваются психологические особенности обучения школьников, что также требует постоянной адаптации образовательного процесса. Представленный ориентированный граф бизнес-процесса формирования индивидуального плана образования на рисунок 6 демонстрирует ключевые этапы, с особым акцентом на механизмы постоянной оценки и внесения коррективов. Данный подход основан на принципах адаптивного обучения и формативной оценки, что обеспечивает динамичность и эффективность образовательного процесса.

Граф представляет собой направленный граф (digraph), который визуализирует последовательность этапов формирования индивидуального плана образования. Каждый узел графа соответствует определенному этапу процесса, а связи между узлами отражают переходы между этапами. Особое внимание уделяется этапам оценки и корректировки, что подчеркивает их значимость в образовательном процессе.

**Начало(А):** Начальный узел графа, обозначающий старт процесса формирования индивидуального плана образования. Этот этап включает подготовку к диагностике и сбору начальных данных об учащемся.

**Определение начального уровня знаний (В):** На этом этапе проводится диагностика для определения текущего уровня знаний учащегося. Используются различные методы оценки, такие как тесты, задания, оценка развернутых ответов.

**Формирование индивидуального плана обучения (С):** На основе данных, полученных на предыдущем этапе, разрабатывается индивидуальный план обучения. Этот план учитывает начальный уровень знаний, образовательные цели, индивидуальные особенности, интересы и стили обучения учащегося.

**Реализация плана обучения (D):** Проведение учебных занятий и мероприятий в соответствии с разработанным планом. Используются различные образовательные технологии и методы, включая интерактивные и визуальные подходы.

**Оценка текущих знаний (Е):** Проведение регулярных оценок для мониторинга прогресса учащегося. Используются формативные и суммативные методы оценки. Формативная оценка фокусируется на процессе обучения и предоставляет постоянную обратную связь, что способствует улучшению образовательных результатов. Суммативная для понимания работы системы, дает ли она необходимые результаты.

**Проверка достижения целей (F):** Сравнение текущих знаний учащегося с установленными образовательными целями. Этот этап включает определение, достигнуты ли цели или требуется корректировка плана.

**Завершение (G):** Завершающий узел графа, обозначающий успешное достижение образовательных целей. Этот этап включает подведение итогов и завершение процесса формирования индивидуального плана образования.

**Анализ пробелов в знаниях (H):** Идентификация областей, в которых учащийся испытывает трудности. Анализ причин пробелов и разработка стратегий для их устранения. В работах Нечаевой подчеркивается важность анализа пробелов в знаниях для корректировки образовательного процесса.

**Корректировка плана обучения (I):** Внесение изменений в план обучения на основе анализа пробелов. Адаптация методов и материалов для улучшения результатов обучения.

Постоянное внесение корректировок в образовательный процесс играет ключевую роль в обеспечении его эффективности. В работах Блума<sup>[10]</sup> подчеркивается, что индивидуальный подход в обучении значительно превосходит групповое обучение по эффективности. Это связано с тем, что постоянная корректировка позволяет адаптировать образовательный процесс к изменяющимся потребностям учащегося. В исследованиях Томлинсон<sup>[11]</sup> рассматривается важность дифференциации обучения для учета разнообразных способностей учеников. Постоянная корректировка плана обучения позволяет учитывать эти различия и обеспечивать более эффективное обучение. В работах Эльконина Д. Б.<sup>[8]</sup> подчеркивается важность анализа пробелов в знаниях для дальнейшего развития учащегося. Постоянная корректировка плана обучения на основе этого анализа позволяет устранять пробелы и улучшать образовательные результаты.

Представленный граф бизнес-процесса формирования индивидуального плана образования с акцентом на постоянное внесение корректировок отражает современные тенденции в педагогике и автоматизации образования. Этот граф основан на принципах адаптивного обучения и формативной оценки, что позволяет обеспечить динамичность и эффективность образовательного процесса. Внедрение таких систем способствует улучшению образовательных результатов и повышению мотивации учащихся.

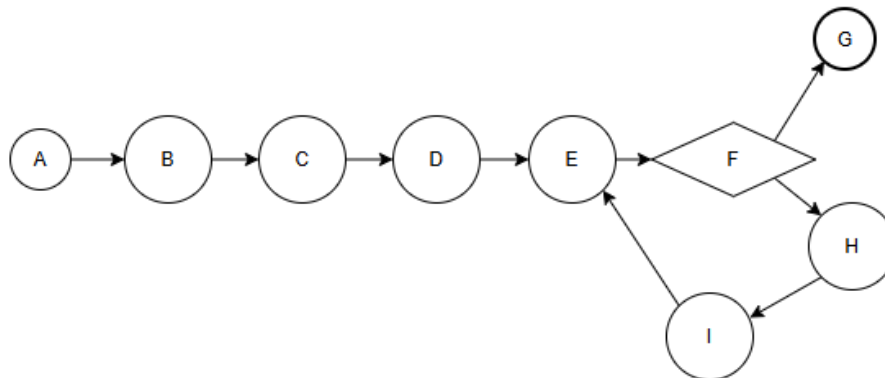


Рисунок 6 - Пример представления бизнес-процесса

Рассмотрим сущности, которые участвуют в процессе. Объектом автоматизации является составление траектории обучения с применением алгоритмов машинного обучения и анализа данных. В рамках диссертации создана концептуальная ER-диаграмма Питера Чена, содержащая в себе сущности: Учащийся, Курс, Наставник и слабые сущности: Модуль, Учебный материал, Задание, Результат, Обратная связь, Геймификация, Аналитика, Зачисление, Наставничество. Сущности по отдельности представлены на рисунках 7-18 и отношения между ними на рисунке 19.

На рисунке 7 представлена сильная сущность – Student(Учащийся). Эта сущность играет одну из ключевых ролей. Атрибуты этой сущности позволяют хранить различные аспекты профиля пользователя и обеспечивать его идентификацию в системе. Включает в себя: StudentId (primary key), FirstName – имя учащегося, LastName – фамилия Учащегося, DateOfBirth – дата рождения учащегося, которая может использоваться для определения возрастных групп и адаптации учебного материала. Email – электронная почта учащегося, используемая для аутентификации. Password – пароль для доступа к системе (храниться в зашифрованном виде), EnrollmentDate – дата зачисления учащегося на платформу или курс.

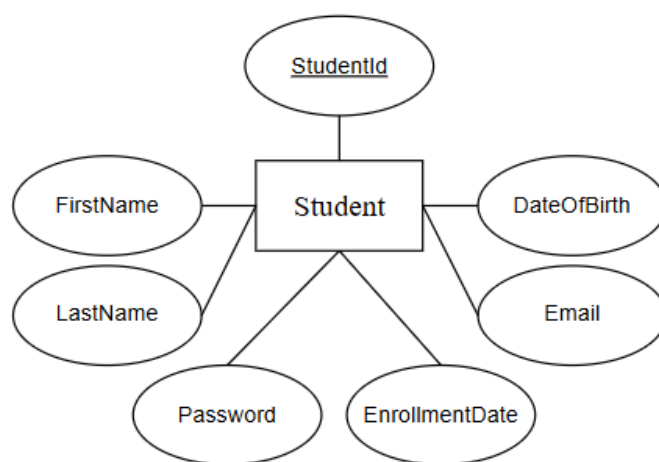


Рисунок 7 - сущность “Student”

Курс (Course) – представляет собой образовательную программу, которую учащийся проходит в рамках своей учебной траектории. Курс является важным компонентом, поскольку является самым высоким уровнем в иерархии частей траектории обучения и объединяет в себе следующие по иерархии модули. Включает в себя: CourseId(primary key), CourseName – название курса, Description – описание курса, включающее цели, содержание и ожидаемые результаты. Duration – продолжительность курса. Credits – количество кредитов, которые учащийся получает за успешное завершение курса.

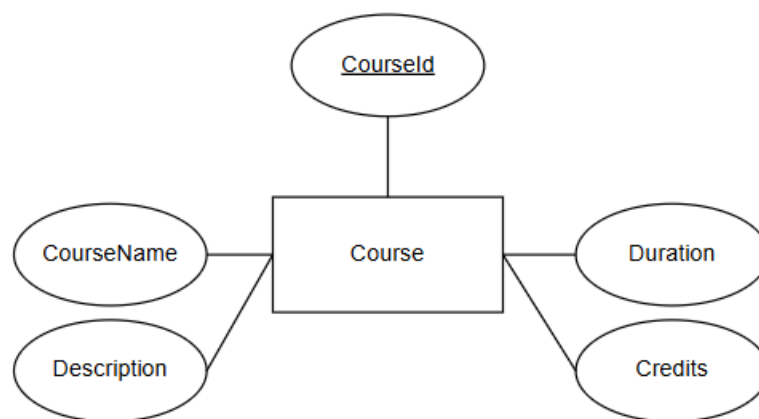


Рисунок 8 - сущность “Course”

Модуль (Module) — это часть курса, которая охватывает определенную тему или набор тем. Курс может состоять из нескольких модулей. Включает в себя: ModuleID(primary key). ModuleName - название модуля. Description - описание модуля, включающее цели и содержание. CourseID - идентификатор курса, к которому относится модуль (внешний ключ).

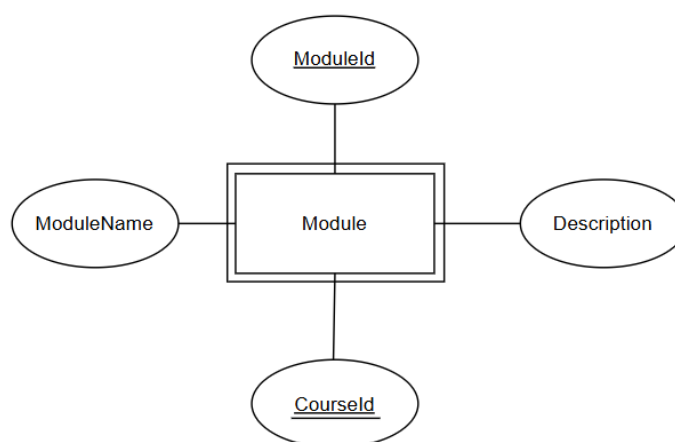


Рисунок 9 - сущность “Module”

Учебный материал (LearningMaterial) включает в себя все ресурсы, используемые для обучения, такие как видео, тексты, аудиозаписи и интерактивные элементы. Включает в себя: MaterialID - уникальный идентификатор учебного материала. MaterialName - название материала. MaterialType - тип материала (например, видео, текст, аудио). Content - Содержание материала (может быть ссылкой на файл или текстом). ModuleID - идентификатор модуля, к которому относится материал (внешний ключ).



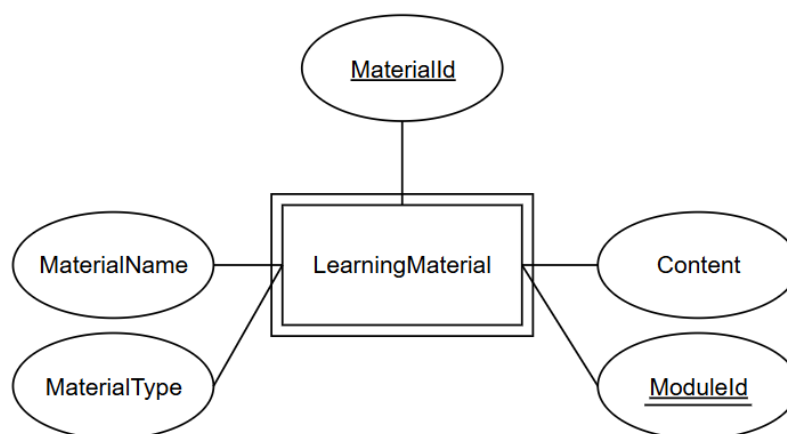


Рисунок 10 - сущность "LearningMaterial"

Задание (Assignment) - это задача или проект, который учащийся должен выполнить в рамках модуля. Включает в себя: AssignmentID - уникальный идентификатор задания. AssignmentName - название задания. Description - описание задания, включающее требования и критерии оценки. Deadline - срок выполнения задания. ModuleID - идентификатор модуля, к которому относится задание (внешний ключ).

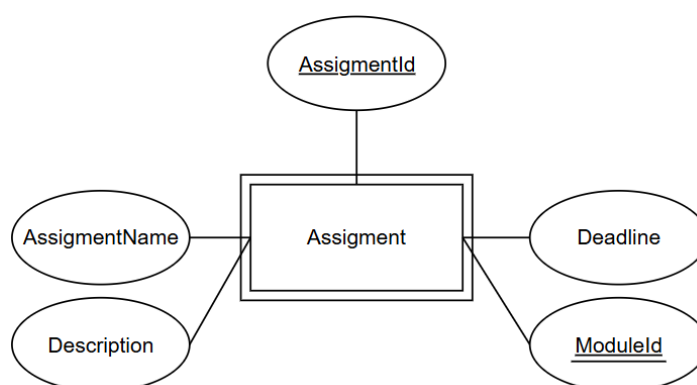


Рисунок 11 - сущность "Assignment"

Результат (Result) представляет собой оценку выполнения задания учащимся. Включает в себя: ResultID - уникальный идентификатор результата. StudentID - идентификатор учащегося, который выполнил задание (внешний ключ). AssignmentID - идентификатор задания, которое было выполнено (внешний ключ). Score - оценка за выполнение задания. SubmissionDate - дата сдачи задания.

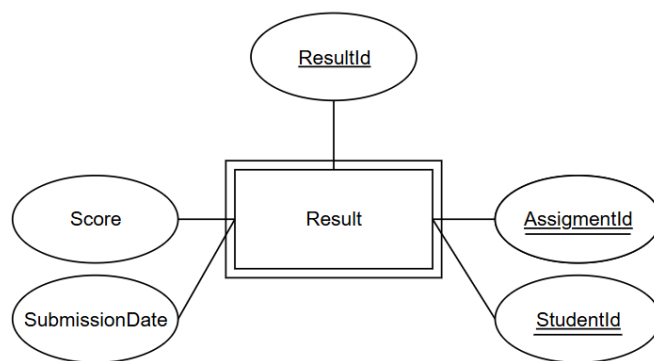


Рисунок 12 - сущность “Result”

Наставник (Mentor) — это преподаватель или эксперт, который помогает учащемуся в процессе обучения. Включает в себя: MentorID – уникальный идентификатор наставника. FirstName - имя наставника. LastName - фамилия наставника. Email - электронная почта наставника для связи. Specialization - специализация или область знаний наставника.

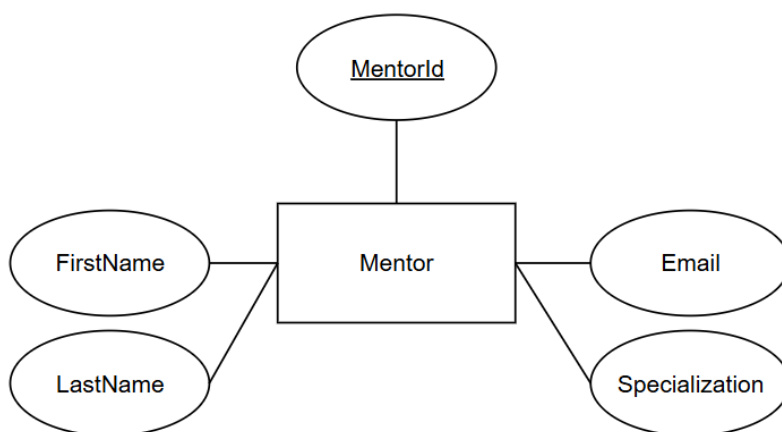


Рисунок 13 - сущность “Mentor”

Обратная связь (Feedback) представляет собой комментарии и рекомендации, предоставляемые наставником учащемуся. Включает в себя: FeedbackID - уникальный идентификатор обратной связи. StudentID - идентификатор учащегося, которому предоставляется обратная связь (внешний ключ). MentorID - идентификатор наставника, который предоставляет обратную связь (внешний ключ). FeedbackText - текст обратной связи. FeedbackDate - дата предоставления обратной связи.

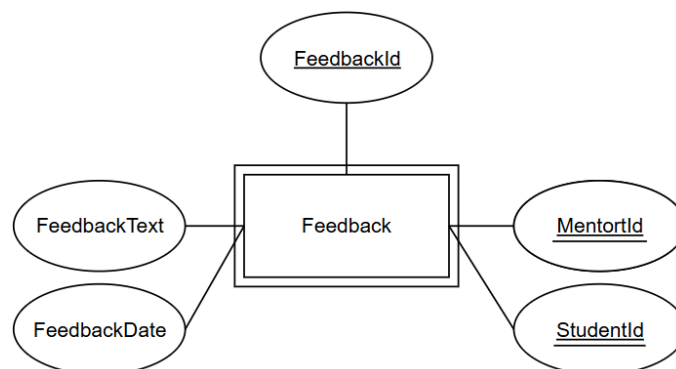


Рисунок 14 - сущность "Feedback"

Геймификация (Gamification) включает в себя элементы, такие как баллы, значки и уровни, которые используются для мотивации учащегося. GamificationID - уникальный идентификатор элемента геймификации. StudentID - идентификатор учащегося, к которому относится элемент геймификации (внешний ключ). Points - Количество баллов, набранных учащимся. Badges - значки, полученные учащимся. Levels - уровни, достигнутые учащимся.

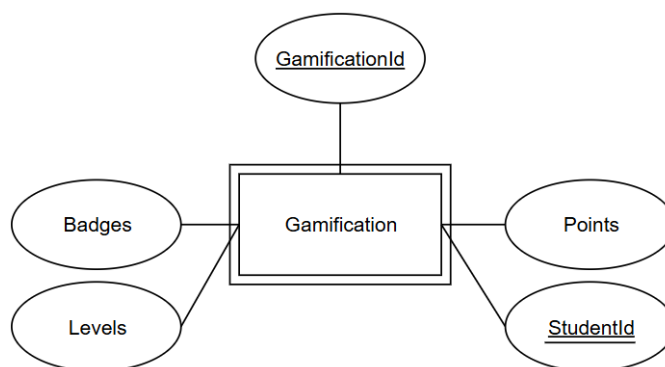


Рисунок 15 - сущность "Gamification"

Аналитика (Analytics) включает в себя данные о прогрессе, тенденциях, проблемах и возможностях учащегося. AnalyticsID - уникальный идентификатор аналитики. StudentID - идентификатор учащегося, к которому относится аналитика (внешний ключ). Progress - прогресс учащегося в обучении. Trends - тенденции в обучении учащегося. Issues - Проблемы, с которыми сталкивается учащийся. Opportunities - возможности для улучшения обучения.

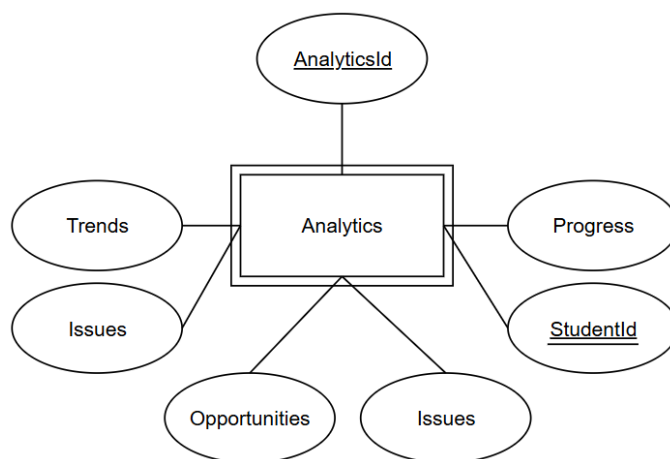


Рисунок 16 - сущность “Analytics”

Промежуточная таблица (Enrollment), связывающая учащихся и курсы, на которые они зачислены. Включает в себя: EnrollmentID - уникальный идентификатор зачисления. StudentID - идентификатор учащегося (внешний ключ). CourseID - идентификатор курса (внешний ключ). EnrollmentDate - дата зачисления на курс.

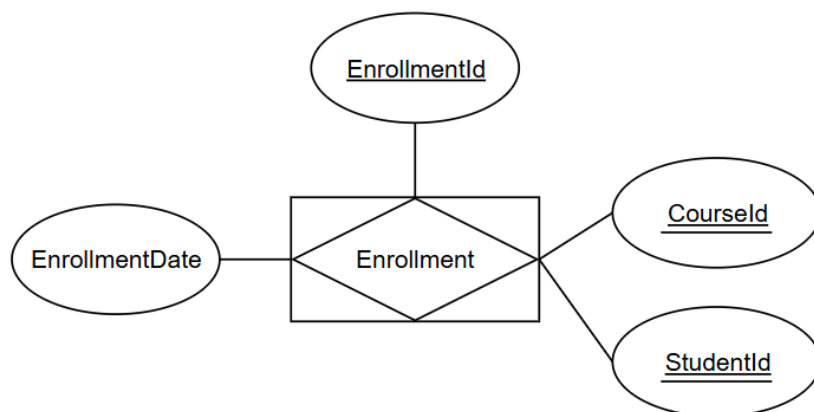


Рисунок 17 - сущность “Enrollment”

Mentorship - промежуточная таблица, связывающая учащихся и наставников. Включает в себя: MentorshipID - уникальный идентификатор наставничества. StudentID - идентификатор учащегося (внешний ключ). MentorID - идентификатор наставника (внешний ключ). StartDate - дата начала наставничества. EndDate - дата окончания наставничества.

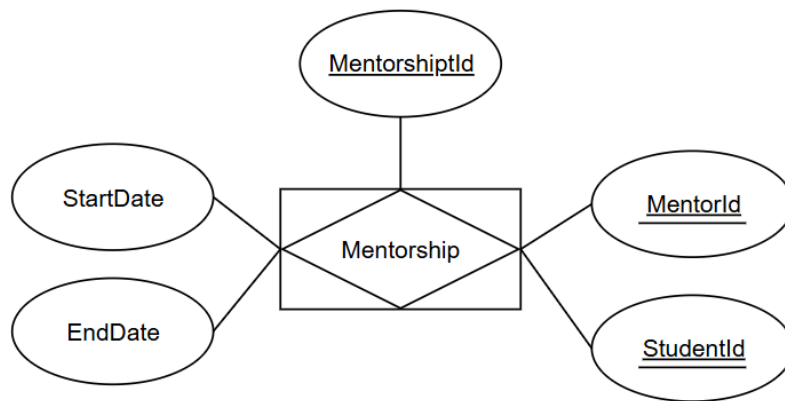


Рисунок 18 - сущность “Mentorship”

На рисунке 19 представлена диаграмма Питера Чена со всеми связями, сущностями, первичными ключами. Данная диаграмма показывает отношения между описанными сущностями. Видим, что основными сущностями являются Учащийся (Student) и Курс (Course), которые имеют много связей между друг другом.

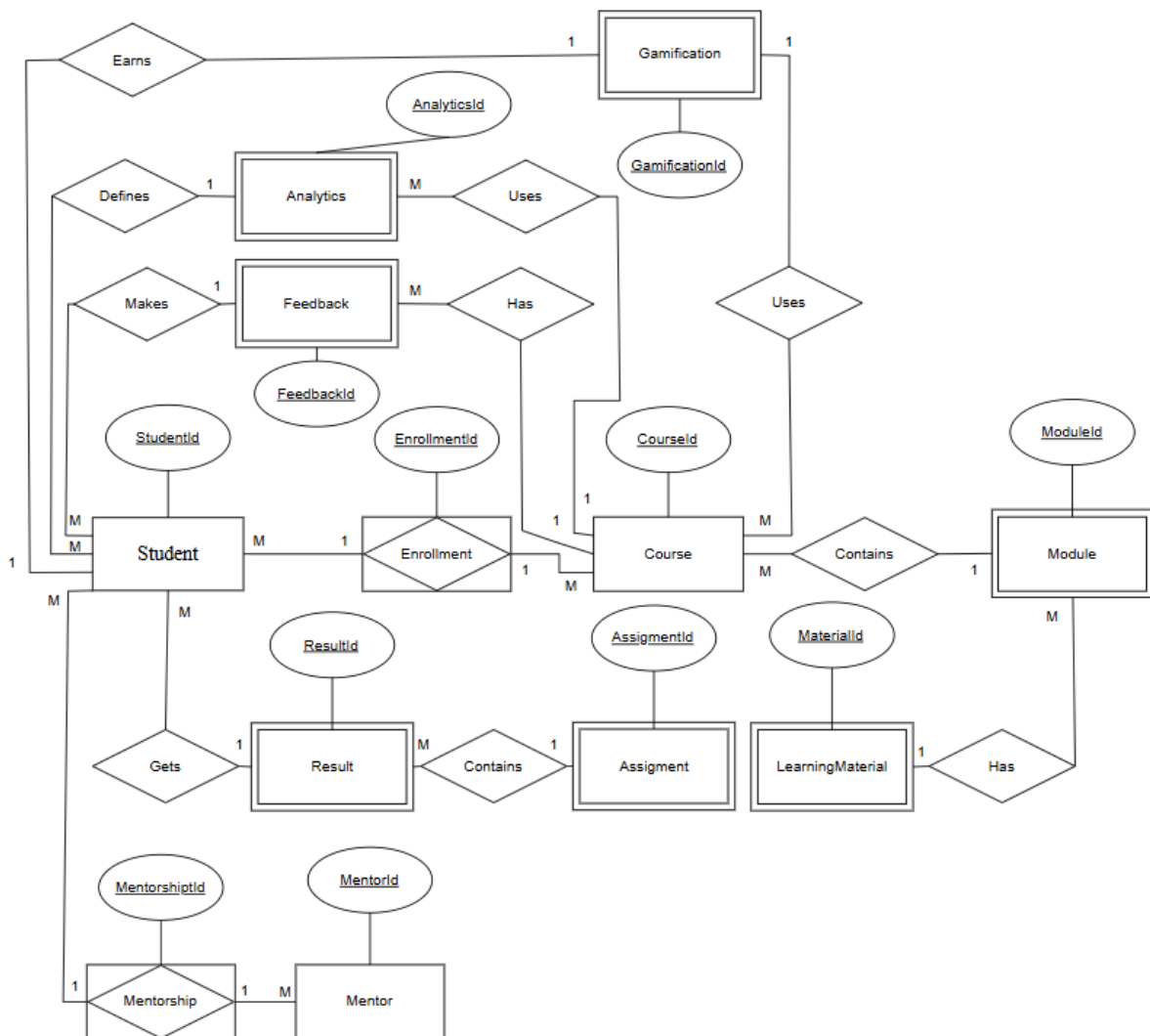


Рисунок 19 - Концептуальная ER диаграмма Питера Чена

## 2.2 Функциональное моделирование

Рассмотрев модель IDEF0 «как есть» можно выделить основные компоненты для автоматизации, кроме этого на рисунке 20 в функциональной модели IDEF0 «как будет» представлен недостающий компонент вводных данных, такой как исторические данные. Благодаря которым добиться циклическое развития системы и получить дополнительный выход в виде прогнозов и рекомендаций, по составлению будущих учебных планов.

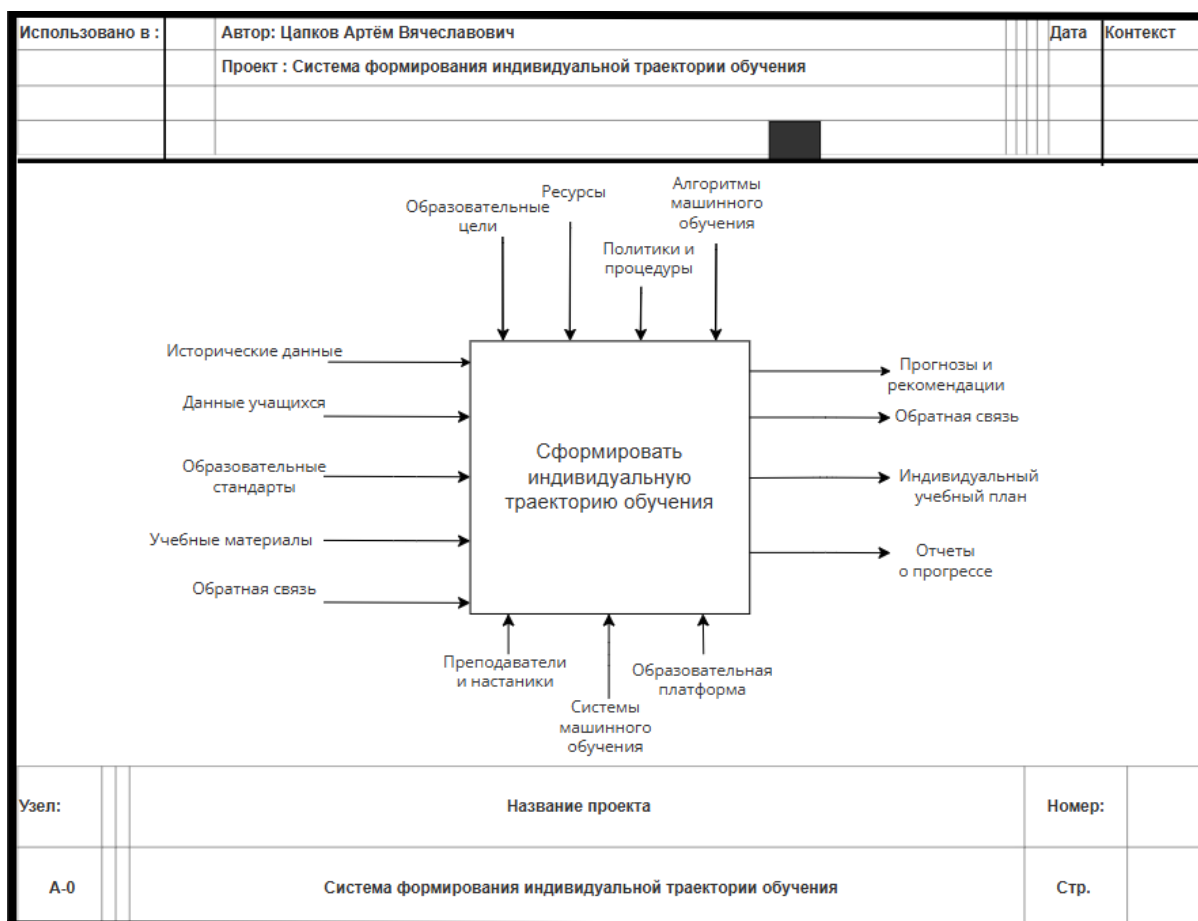


Рисунок 20 - IDEF0 модель уровня A0 “как будет”

Декомпозиция диаграммы “Как будет” на рисунке 21 имеет основное изменение в механизмах и управляющих воздействиях, а именно основное участие системы и алгоритмов машинного обучения. Диагностику и оценку учащихся на себя полностью берет система с использованием ИИ. Теперь процессы блоков A2 и A3 включают в себя совместную работу преподавателей/наставников и системы. По декомпозиции видим, что часть работы с людей снята. Особенно это важно для высокоинтеллектуальных задач, требуемых анализа и обработки большого количества данных, в том числе исторических данных.

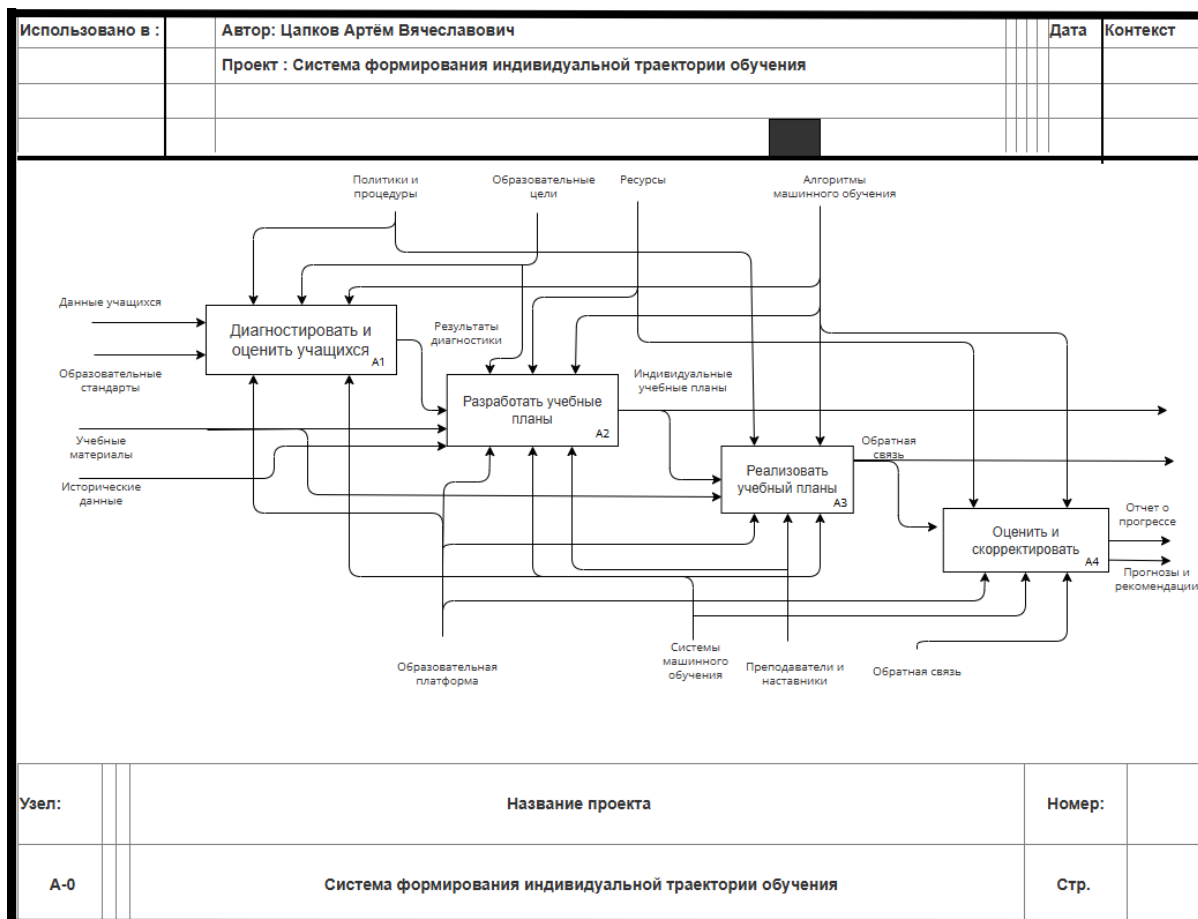


Рисунок 21 - IDEF0 модель уровня A1 “как будет”

Декомпозиция блока “Разработать учебные планы” показанная на рисунке 22 включает в себя следующие процессы:

1. Анализировать результаты диагностики;
2. Определить индивидуальные образовательные цели и задачи;
3. Разработать персональный учебный план;
4. Оптимизировать учебный план;

Входными данными служат: результаты диагностики из блока A1, учебные материалы, исторические данные. Ресурсы управления, в свою очередь, наследовались из диаграммы IDEF0 модели уровня A1 “как будет”. Выходными данными всего процесса являются индивидуальные учебные планы.

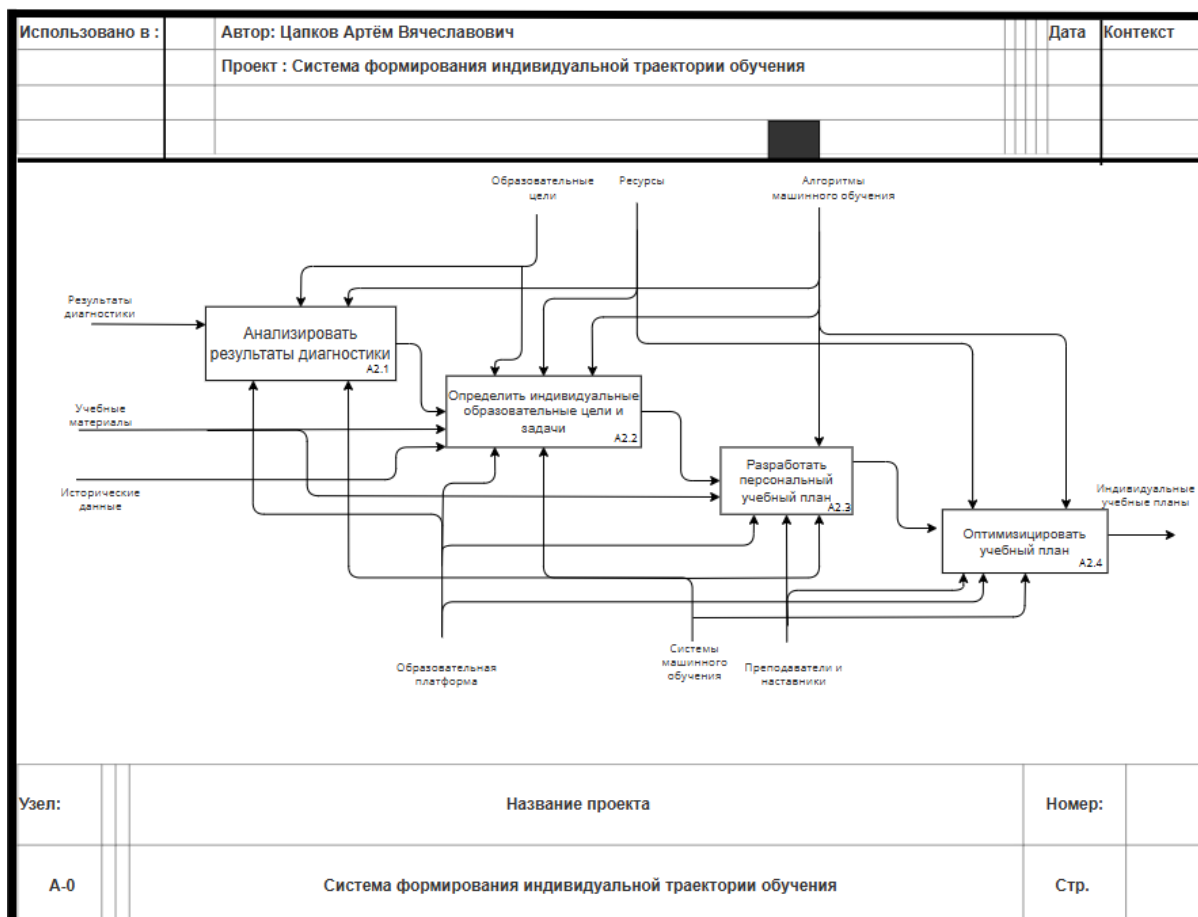


Рисунок 22 - IDEF0 модель уровня A2 “как будет”

Таким образом, с помощью функционального моделирования, показаны процессы того, как на данный момент решается задача, а также как данные решения будут оптимизированы в будущем. Диаграммы “как есть” дают информацию о том, как на данный момент реализуется решение проблемной области, а именно “сформировать индивидуальную траекторию обучения”. Диаграммы “как будет”, то, как будет выглядеть итоговый процесс после реализации задуманной web-платформы.

## 2.3 Формирование требований к информационной системе

### 2.3.1 Требования к системе

Перечень подсистем для системы формирования индивидуальных траекторий обучения можно выделить следующие подсистемы, их функции, технические требования, а также требования к уровням иерархии и степени централизации системы:

1. Подсистема управления данными учащихся – управляет данными о учащихся, включая их профили, достижения, предпочтения и историю обучения.



*Функциональные возможности:*

- Сбор и хранение данных о учащиххся.
- Обеспечение безопасности и конфиденциальности данных.
- Обновление и корректировка данных.

*Технические требования:*

- Высокая надежность и доступность.
- Поддержка шифрования данных.
- Интеграция с другими подсистемами.

*Уровень иерархии:* Высокий. Эта подсистема является центральной и взаимодействует со всеми другими подсистемами.

*Степень централизации:* Высокая. Все данные о учащиххся хранятся и управляются централизованно для обеспечения безопасности и конфиденциальности.

*ГОСТ:* ГОСТ Р 34.201-89 "Автоматизированные системы. Термины и определения".

2. Подсистема диагностики и оценки - проводит диагностику и оценку знаний, навыков и компетенций учащиххся.

*Функциональные возможности:*

- Проведение начальной диагностики.
- Регулярная оценка прогресса учащиххся.
- Анализ результатов оценки с использованием алгоритмов машинного обучения.

*Технические требования:*

- Точность и надежность алгоритмов оценки.
- Поддержка различных методов диагностики.
- Интеграция с подсистемой управления данными учащиххся.

*Уровень иерархии:* Средний. Эта подсистема взаимодействует с подсистемой управления данными учащиххся и подсистемой разработки учебных планов.

*Степень централизации:* Средняя. Диагностика и оценка могут проводиться как централизованно, так и децентрализованно, в зависимости от конкретных задач.

*ГОСТ:* ГОСТ 34.602-89 "Автоматизированные системы. Требования к содержанию документов".

3. Подсистема разработки учебных планов - разрабатывает персонализированные учебные планы на основе данных об учащиххся и об образовательных стандартах.

*Функциональные возможности:*

- Определение образовательных целей и задач.
- Создание учебных планов с использованием алгоритмов машинного обучения.

- Оптимизация учебных планов для повышения эффективности обучения.

*Технические требования:*

- Поддержка различных форматов учебных планов.
- Интеграция с подсистемой диагностики и оценки.
- Возможность адаптации планов под индивидуальные потребности.

*Уровень иерархии:* Средний. Эта подсистема взаимодействует с подсистемой диагностики и оценки, а также с подсистемой управления учебными материалами.

*Степень централизации:* Средняя. Разработка учебных планов может быть централизованной, но с возможностью адаптации на уровне отдельных учащихся.

*ГОСТ:* ГОСТ 34.201-89 "Автоматизированные системы. Термины и определения".

4. Подсистема управления учебными материалами - управляет учебными материалами, включая курсы, модули, задания и мультимедийные ресурсы.

*Функциональные возможности:*

- Хранение и организация учебных материалов.
- Обеспечение доступа к учебным материалам.
- Обновление и добавление новых материалов.

*Технические требования:*

- Поддержка различных форматов материалов (видео, текст, аудио и т.д.).
- Интеграция с подсистемой разработки учебных планов.
- Высокая доступность и надежность хранения.

*Уровень иерархии:* Средний. Эта подсистема взаимодействует с подсистемой разработки учебных планов и подсистемой реализации учебных планов.

*Степень централизации:* Средняя. Учебные материалы могут управляться централизованно, но с возможностью добавления и изменения материалов на уровне отдельных курсов или модулей.

*ГОСТ:* ГОСТ 34.201-89 "Автоматизированные системы. Термины и определения".

5. Подсистема реализации учебных планов - обеспечивает реализацию учебных планов, включая предоставление учебных материалов и заданий.

*Функциональные возможности:*

- Предоставление учебных материалов учащимся.
- Контроль выполнения заданий.
- Обратная связь и поддержка учащихся.

*Технические требования:*

- Поддержка интерактивных элементов (видео, аудио, симуляции).

- Интеграция с подсистемой управления учебными материалами.
- Возможность отслеживания прогресса в реальном времени.

*Уровень иерархии:* Низкий. Эта подсистема взаимодействует с подсистемой управления учебными материалами и подсистемой оценки и корректировки.

*Степень централизации:* Низкая. Реализация учебных планов происходит на уровне отдельных учащихся и может быть децентрализована.

*ГОСТ:* ГОСТ 34.602-89 "Автоматизированные системы. Требования к содержанию документов".

6. Подсистема оценки и корректировки - оценивает прогресс учащихся и корректирует учебные планы на основе анализа данных.

*Функциональные возможности:*

- Сбор данных о прогрессе учащихся.
- Анализ данных с использованием алгоритмов машинного обучения.
- Корректировка учебных планов для улучшения результатов обучения.

*Технические требования:*

- Точность и надежность алгоритмов анализа.
- Интеграция с подсистемой реализации учебных планов.
- Возможность автоматической корректировки планов.

*Уровень иерархии:* Средний. Эта подсистема взаимодействует с подсистемой реализации учебных планов и подсистемой аналитики и отчетности.

*Степень централизации:* Средняя. Оценка и корректировка могут проводиться как централизованно, так и децентрализованно, в зависимости от конкретных задач.

*ГОСТ:* ГОСТ 34.201-89 "Автоматизированные системы. Термины и определения".

7. Подсистема машинного обучения - включает алгоритмы и модели машинного обучения, используемые для анализа данных и оптимизации учебного процесса.

*Функциональные возможности:*

- Обучение моделей машинного обучения на исторических данных.
- Прогнозирование будущих достижений учащихся.
- Рекомендации по улучшению учебных планов.

*Технические требования:*

- Высокая производительность и масштабируемость.
- Поддержка различных алгоритмов и моделей машинного обучения.
- Интеграция с другими подсистемами для предоставления аналитических данных.

*Уровень иерархии:* Высокий. Эта подсистема взаимодействует со всеми другими подсистемами, предоставляя аналитические данные и рекомендации.

*Степень централизации:* Высокая. Алгоритмы машинного обучения управляются централизованно для обеспечения точности и надежности.

*ГОСТ:* ГОСТ 34.602-89 "Автоматизированные системы. Требования к содержанию документов".

8. Подсистема управления пользователями - управляет пользователями системы, включая учащихся, преподавателей, наставников и администраторов.

*Функциональные возможности:*

- Регистрация и аутентификация пользователей.
- Управление правами доступа.
- Обеспечение безопасности и конфиденциальности данных пользователей.

*Технические требования:*

- Поддержка многофакторной аутентификации.
- Высокая надежность и доступность.
- Интеграция с другими подсистемами для управления доступом.

*Уровень иерархии:* Высокий. Эта подсистема взаимодействует со всеми другими подсистемами, обеспечивая управление доступом и безопасностью.

*Степень централизации:* Высокая. Управление пользователями и их правами доступа происходит централизованно.

*ГОСТ:* ГОСТ Р 34.201-89 "Автоматизированные системы. Термины и определения".

9. Подсистема интеграции с внешними ресурсами - обеспечивает интеграцию системы с внешними образовательными платформами, базами данных и другими ресурсами.

*Функциональные возможности:*

- Подключение к внешним базам данных и образовательным платформам.
- Обмен данными с внешними системами.
- Обеспечение совместимости и безопасности интеграции.

*Технические требования:*

- Поддержка различных протоколов и стандартов интеграции.
- Высокая надежность и безопасность обмена данными.
- Возможность масштабирования интеграции.

*Уровень иерархии:* Средний. Эта подсистема взаимодействует с подсистемой управления учебными материалами и подсистемой аналитики и отчетности.

*Степень централизации:* Средняя. Интеграция может быть централизованной, но с возможностью децентрализованного подключения к различным внешним ресурсам.

*ГОСТ:* ГОСТ 34.201-89 "Автоматизированные системы. Термины и определения".

10. Подсистема аналитики и отчетности - сбор и анализ данных о прогрессе учащихся, эффективности учебных планов и работе системы.

*Функциональные возможности:*

- Сбор данных о прогрессе учащихся.
- Анализ данных с использованием алгоритмов машинного обучения.
- Генерация отчетов и визуализаций.

*Технические требования:*

- Поддержка различных форматов отчетов и визуализаций.
- Интеграция с подсистемой оценки и корректировки.
- Высокая точность и надежность аналитических данных.

*Уровень иерархии:* Средний. Эта подсистема взаимодействует с подсистемой оценки и корректировки и подсистемой машинного обучения.

*Степень централизации:* Средняя. Аналитика и отчетность могут быть централизованными, но с возможностью генерации отчетов на уровне отдельных учащихся или курсов.

*ГОСТ:* ГОСТ 34.602-89 "Автоматизированные системы. Требования к содержанию документов".

11. Подсистема обратной связи и поддержки - обеспечивает предоставление обратной связи и поддержки учащимся.

*Функциональные возможности:*

- Сбор и анализ обратной связи от учащихся и преподавателей.
- Предоставление рекомендаций и поддержки.
- Обеспечение взаимодействия между учащимися, преподавателями и наставниками.

*Технические требования:*

- Поддержка различных каналов обратной связи (электронная почта, чаты, форумы).
- Интеграция с подсистемой реализации учебных планов.
- Высокая доступность и надежность.

*Уровень иерархии:* Низкий. Эта подсистема взаимодействует с подсистемой реализации учебных планов и подсистемой управления пользователями.

*Степень централизации:* Низкая. Обратная связь и поддержка предоставляются на уровне отдельных учащихся и могут быть децентрализованы.

*ГОСТ:* ГОСТ 34.201-89 "Автоматизированные системы. Термины и определения".

12. Подсистема геймификации и мотивации - включает элементы геймификации для повышения мотивации учащихся.

*Функциональные возможности:*

- Начисление баллов, значков и уровней.
- Организация соревнований и челленджей.
- Предоставление наград и поощрений.

*Технические требования:*

- Поддержка различных элементов геймификации.
- Интеграция с подсистемой реализации учебных планов.
- Возможность персонализации элементов геймификации.

*Уровень иерархии:* Низкий. Эта подсистема взаимодействует с подсистемой реализации учебных планов и подсистемой аналитики и отчетности..

*Степень централизации:* Низкая. Элементы геймификации и мотивации могут быть децентрализованы и адаптированы под каждого учащегося.

*ГОСТ:* ГОСТ 34.201-89 "Автоматизированные системы. Термины и определения".

13. Подсистема управления контентом - управляет созданием, редактированием и публикацией учебного контента.

*Функциональные возможности:*

- Создание и редактирование учебных материалов.
- Публикация и обновление контента.
- Обеспечение доступа к контенту для учащихся и преподавателей.

*Технические требования:*

- Поддержка различных форматов контента.
- Интеграция с подсистемой управления учебными материалами.
- Высокая доступность и надежность хранения.

*Уровень иерархии:* Средний. Эта подсистема взаимодействует с подсистемой управления учебными материалами и подсистемой реализации учебных планов.

*Степень централизации:* Средняя. Управление контентом может быть централизованным, но с возможностью децентрализованного создания и редактирования материалов.

*ГОСТ:* ГОСТ 34.201-89 "Автоматизированные системы. Термины и определения".

14. Подсистема безопасности и защиты данных - обеспечивает безопасность и защиту данных в системе.

*Функциональные возможности:*

- Шифрование данных.
- Контроль доступа и аутентификация пользователей.
- Мониторинг и предотвращение угроз безопасности.

*Технические требования:*

- Высокая надежность и безопасность.
- Поддержка современных стандартов шифрования и защиты данных.
- Интеграция с другими подсистемами для обеспечения безопасности.

*Уровень иерархии:* Высокий. Эта подсистема взаимодействует со всеми другими подсистемами, обеспечивая безопасность и защиту данных.

*Степень централизации:* Высокая. Безопасность и защита данных управляются централизованно для обеспечения надежности и конфиденциальности.

*ГОСТ:* ГОСТ Р 34.201-89 "Автоматизированные системы. Термины и определения".

Перечень подсистем системы формирования индивидуальной траектории обучения с использованием алгоритмов машинного обучения включает подсистемы управления данными учащихся, диагностики и оценки, разработки учебных планов, управления учебными материалами, реализации учебных планов, оценки и корректировки, машинного обучения, управления пользователями, интеграции с внешними ресурсами, аналитики и отчетности, обратной связи и поддержки, геймификации и мотивации, управления контентом, а также безопасности и защиты данных. Каждая подсистема выполняет специфические функции, обеспечивая комплексное и эффективное функционирование всей системы.

### **2.3.2 Требования к надежности**

Надежность системы формирования индивидуальной траектории обучения с использованием алгоритмов машинного обучения определяется рядом показателей, которые характеризуют способность системы выполнять свои функции в течение заданного времени без отказов. Эти показатели могут быть описаны в соответствии с *ГОСТ 34.201-89 "Автоматизированные системы. Термины и определения"*. Ниже приведены основные показатели надежности для системы в целом.

### 1. Высокая доступность

Описание: Система должна быть доступна для использования в течение всех рабочих часов, учитывая потенциальную нагрузку и поступление аудиоинформации для обработки.

Механизмы: Необходимы механизмы обеспечения непрерывной доступности, такие как резервирование серверов и сетевых соединений.

### 2. Стабильная работа

Описание: Система должна гарантировать низкую вероятность возникновения проблем при функционировании основных компонентов.

Меры: Проведение регулярных тестов на стойкость к нагрузкам и неожиданным ситуациям помогает обеспечить стабильную работу.

### 3. Защита от атак

Описание: Система должна быть защищена от вредоносных атак, включая DDoS-атаки, взломы и вирусы.

Меры: Применение современных методов защиты, таких как брандмауэры и антивирусное ПО, обеспечивает безопасность системы.

### 4. Безопасность данных

Описание: Все данные, обрабатываемые и хранимые системой, должны быть защищены от несанкционированного доступа и утечек.

Меры: Использование методов шифрования и строгих политик доступа помогает обеспечить безопасность данных.

### 5. Резервное копирование и восстановление

Описание: Система должна регулярно создавать резервные копии данных для возможности быстрого восстановления в случае сбоев или аварий.

Меры: Резервные копии данных должны храниться в защищенных местах с ограниченным доступом.

### 6. Мониторинг и управление

Описание: Система должна быть оборудована средствами мониторинга ее состояния и автоматического реагирования на возникающие проблемы.

Меры: Настройка систем оповещения и автоматических мер по восстановлению помогает поддерживать надежную работу системы.

### 7. Регулярные обновления и аудит безопасности

Описание: Система должна регулярно обновляться для исправления уязвимостей и обеспечения безопасности.

Меры: Проведение регулярных аудитов безопасности помогает выявить потенциальные угрозы и уязвимости.



### 2.3.3 Требования к квалификации персонала

Для обеспечения эффективного функционирования системы необходимо четко определить роли и обязанности пользователей и администраторов. Ниже приведены требования к пользователям и администраторам, которые помогут поддерживать безопасность, надежность и качество работы системы.

#### *Пользователь сервиса*

- Доступ к сервису: Пользователь должен быть включен в белый список для работы с системой, что обеспечивает контроль доступа и безопасность данных.
- Соблюдение правил: Использование платформы должно соответствовать ее целям и не нарушать законодательство и правила пользования системой.
- Конфиденциальность информации: Пользователь обязан бережно обращаться с конфиденциальной информацией и не передавать ее третьим лицам или лицам, не находящимся в белом списке сервиса.
- Компетенция: Пользователь должен обладать достаточной компетенцией для анализа информации, предоставляемой системой после обработки данных.
- Поддержка: Пользователь имеет возможность получить оперативную и качественную поддержку от администрации системы при возникновении вопросов или проблем с использованием платформы.

#### *Администратор платформы*

- Компетенция: Администратор системы должен обладать достаточным уровнем компетенции в области обработки голосовых данных и нейронных сетей.
- Контроль работоспособности: Администратор следит за работоспособностью всех модулей и компонентов платформы, проводя периодические проверки и контроль их функциональности.
- Обновления и актуализация: Администратор обязан следить за обновлениями и актуализацией информации о методах обработки голоса и современных технологиях в этой области.
- Оперативная поддержка: Администратор должен оперативно реагировать на запросы и обращения пользователей, предоставляя качественную и своевременную поддержку.
- Улучшение функционала: Администратор регулярно обновляет и улучшает функционал системы в соответствии с потребностями пользователей и последними достижениями в области распознавания и разметки голосовых записей.

- **Контроль за действиями пользователей:** Администратор контролирует действия пользователей на платформе, включая выявление нарушений правил использования системы.
- **Обучение и консультирование:** Администратор обеспечивает адекватную поддержку пользователям, консультируя и обучая их по всем вопросам, связанным с использованием системы.
- **Взаимодействие с разработчиками:** Администратор содействует эффективному взаимодействию с разработчиками и технической поддержкой для решения любых проблем, связанных с функционированием платформы.
- **Этика и профессиональная ответственность:** Администратор соблюдает принципы этики и профессиональной ответственности в общении с пользователями и обработке их данных.

Требования к пользователям и администраторам системы включают доступ к сервису, соблюдение правил, конфиденциальность информации, компетенцию, поддержку, контроль работоспособности, обновления и актуализацию, оперативную поддержку, улучшение функционала, контроль за действиями пользователей, обучение и консультирование, взаимодействие с разработчиками, а также соблюдение принципов этики и профессиональной ответственности. Эти требования обеспечивают высокую надежность и качество работы системы, способствуя эффективному и бесперебойному учебному процессу. Соблюдение этих требований в соответствии с ГОСТ обеспечивает высокое качество и надежность системы в целом.

#### **2.3.4 Показатели назначения**

1. Система должна обладать высокой приспособляемостью к изменениям в структуре и характере данных, а также к изменениям параметров записей:

**Приспособляемость к изменениям:** Способность системы адаптироваться к изменениям в структуре данных. В контексте образовательной системы это может включать адаптацию к новым образовательным стандартам или методикам.

**Гибкость настройки алгоритмов:** Возможность гибкой настройки алгоритмов машинного обучения для различных учебных программ и стилей обучения.

2. Система должна быть разработана с учетом потенциала для будущих модернизаций и расширений функционала:

**Масштабируемость архитектуры:** Архитектура системы должна быть гибкой и масштабируемой, позволяя легко внедрять изменения и добавлять новые функции в процессе ее развития, интеграции новых образовательных модулей или курсов.

**Поддержка новых методов:** Возможность интеграции новых методов обучения нейронных сетей без кардинальных изменений в программном обеспечении системы. В

образовательной системе это может означать интеграцию новых педагогических подходов или технологий.

3. Система должна обеспечивать высокую доступность и надежность, чтобы сохранить свою функциональность в различных ситуациях:

Время безотказной работы: Среднее время, в течение которого система работает без сбоев во время учебного процесса.

Доступность системы: Уровень доступности системы для пользователей в различных условиях. Это включает доступность учебных материалов и ресурсов для учащихся и преподавателей.

Скорость реакции на изменения: Скорость обновления учебных планов или материалов в ответ на изменения в образовательных стандартах.

4. Оценка скорости, с которой система обрабатывает данные:

Скорость обработки: Среднее время, обработки учебных заданий или тестов.

Количество обрабатываемых записей в единицу времени: Производительность системы в обработке данных.

5. Оценка надежности и стабильности работы системы:

Надежность работы: Процент времени, в течение которого система работает без сбоев или ошибок.

Стабильность работы: Устойчивость системы к внешним воздействиям и изменениям в условиях работы.

6. Оценка возможности системы хранить исторические данные для последующего анализа и использования в обучении нейронных сетей:

Глубина хранения данных: Время, в течение которого система хранит исторические данные (не менее 1-2 лет). Данные, которые необходимо хранить в течении этого времени - данные о прогрессе учащихся для анализа и улучшения учебных планов.

Объем хранимых данных: Объем данных, который система может хранить для анализа и обучения.

7. Оценка возможности системы поддерживать одновременную работу нескольких пользователей:

Количество одновременно работающих пользователей: Число пользователей, которые могут одновременно работать с системой (не менее 50 пользователей для операционной деятельности и не менее 10 пользователей для остальных подсистем).

8. Оценка возможности системы формировать отчеты в установленные сроки:

Время формирования отчета: Время, необходимое для формирования отчетов о прогрессе учащихся или результатах тестов (не более 10 минут).

Сроки предоставления отчетности: Время, в течение которого система предоставляет отчетность в готовом виде (не более 24 часов).

Показатели назначения включают гибкость и адаптивность, возможности модернизации, временные показатели, точность распознавания и разметки данных, скорость обработки данных, надежность и стабильность работы, хранение исторических данных, поддержку одновременной работы пользователей и формирование отчетов. Эти показатели помогают оценить, насколько эффективно система выполняет свои функции и соответствует ли она заявленным целям, обеспечивая высокое качество и надежность обучения.

### **2.3.5 Требования к приспособляемости системы к изменениям**

Приспособляемость системы к изменениям является ключевым аспектом, обеспечивающим её способность адаптироваться к новым требованиям, технологиям и условиям. Ниже приведены основные требования к приспособляемости системы.

#### **1. Гибкость архитектуры**

Описание: Система должна иметь модульную и гибкую архитектуру, позволяющую легко добавлять, изменять или удалять компоненты без нарушения функциональности.

- **Модульность:** Архитектура системы должна быть разделена на независимые модули, каждый из которых выполняет определенную функцию.
- **Интерфейсы взаимодействия:** Модули должны взаимодействовать друг с другом через четко определенные интерфейсы, что обеспечивает легкость замены или модернизации отдельных компонентов.

**2. Поддержка новых методов и технологий.** Система должна поддерживать интеграцию новых методов обучения, алгоритмов машинного обучения и технологий без необходимости кардинальных изменений в программном обеспечении.

- **Интеграция алгоритмов:** Система должна поддерживать интеграцию новых алгоритмов машинного обучения и методов анализа данных.
- **Обновление моделей:** Возможность обновления и переобучения моделей машинного обучения на основе новых данных и методов.

**3. Масштабируемость.** Система должна быть способна масштабироваться для увеличения производительности и объема обрабатываемой информации.

- **Горизонтальное масштабирование:** Возможность добавления новых серверов или узлов для распределения нагрузки.
- **Вертикальное масштабирование:** Возможность увеличения ресурсов (CPU, RAM) на существующих серверах для повышения производительности.

**4. Адаптивность к изменениям в данных.** Система должна быть способна адаптироваться к изменениям в структуре и характере данных.

- Автоматическое обновление моделей: Система должна автоматически обновлять модели машинного обучения на основе новых данных и изменений в структуре данных.

5. Поддержка различных образовательных стандартов. Система должна поддерживать различные образовательные стандарты и методики, адаптируя учебные планы и материалы под требования различных образовательных учреждений.

- Конфигурируемость учебных планов: Возможность настройки учебных планов под различные образовательные стандарты и требования.

- Интеграция с внешними ресурсами: Возможность интеграции с внешними образовательными платформами и базами данных для получения актуальных учебных материалов.

6. Поддержка различных типов пользователей. должна поддерживать различные типы пользователей, включая учащихся, преподавателей, администраторов и наставников, обеспечивая каждому из них необходимые функции и права доступа.

- Ролевая модель доступа: Настройка прав доступа и функций для различных типов пользователей.

- Персонализация интерфейса: Возможность настройки интерфейса под потребности различных типов пользователей.

7. Обратная совместимость. Система должна поддерживать обратную совместимость с предыдущими версиями и компонентами, обеспечивая плавный переход при обновлениях.

- Совместимость с предыдущими версиями: Возможность работы с данными и компонентами предыдущих версий системы.

- Поддержка устаревших форматов: Возможность работы с устаревшими форматами данных и учебных материалов.

### **2.3.6 Требования к пользовательскому интерфейсу и удобству использования**

1. Интерфейс системы должен быть интуитивно понятным и легким в использовании для всех категорий пользователей, включая учащихся, преподавателей и администраторов.

- Простота навигации: Навигация по системе должна быть логичной и простой, с минимальным количеством шагов для выполнения основных задач.

- Понятные метки и подсказки: Все элементы интерфейса должны иметь понятные метки и подсказки, объясняющие их назначение.

2. Интерфейс должен поддерживать персонализацию под потребности различных пользователей.

- Настраиваемые профили: Возможность настройки профилей пользователей с учетом их предпочтений и потребностей.

- Адаптивный дизайн: Интерфейс должен адаптироваться под различные устройства и размеры экранов.

3. Интерфейс должен обеспечивать быструю и надежную работу, минимизируя время загрузки и отклика.

- Оптимизация производительности: Интерфейс должен быть оптимизирован для быстрой загрузки и работы на различных устройствах.

- Минимальное время отклика: Время отклика системы на действия пользователя должно быть минимальным.

4. Система должна предоставлять пользователям возможность получения обратной связи и поддержки.

- Система уведомлений: Наличие системы уведомлений для информирования пользователей о важных событиях и изменениях.

- Чат и поддержка: Встроенные инструменты для общения с поддержкой и получения помощи в реальном времени.

5. Интерфейс должен обеспечивать безопасность и конфиденциальность данных пользователей.

- Защита данных: Использование методов шифрования и аутентификации для защиты данных пользователей.

- Контроль доступа: Настройка прав доступа для различных категорий пользователей.

6. Интерфейс должен поддерживать интерактивные и мультимедийные элементы для повышения вовлеченности пользователей.

- Интерактивные задания: Возможность выполнения интерактивных заданий и тестов.

- Мультимедийные материалы: Поддержка видео, аудио и других мультимедийных форматов.

7. Интерфейс должен предоставлять пользователям доступ к аналитическим данным и отчетам.

- Дашборды и визуализации: Наличие дашбордов и визуализаций для отображения ключевых показателей и отчетов.

### **2.3.7 Требования к защите информации от несанкционированного доступа**

1. Гарантировать строгую аутентификацию всех пользователей с использованием сильных паролей, механизмов двухфакторной аутентификации и других методов подтверждения личности.

- Сильные пароли: Использование сложных паролей, включающих комбинации букв, цифр и специальных символов.
- Двухфакторная аутентификация (2FA): Внедрение механизмов 2FA для дополнительной защиты учетных записей.

2. Обеспечить каждому пользователю строго определенные права доступа к различным функциям и данным системы. Администраторам предоставить возможность гибкого управления правами доступа.

- Ролевая модель доступа: Настройка прав доступа на основе ролей пользователей (учащиеся, преподаватели, администраторы).
- Гибкое управление правами: Возможность администраторов назначать и изменять права доступа для различных пользователей.
- Аудит прав доступа: Регулярный аудит прав доступа для выявления и устранения излишних или некорректных настроек.

3. Зашифровать всю передаваемую и хранимую информацию для защиты от несанкционированного доступа. Использовать современные алгоритмы шифрования с длинными ключами для обеспечения безопасности данных.

- Шифрование данных в покое: Использование алгоритмов шифрования (например, AES-256) для защиты данных, хранящихся на серверах.
- Шифрование данных в транзите: Использование протоколов шифрования (например, TLS) для защиты данных при передаче между клиентами и серверами.
- Управление ключами: Безопасное хранение и управление ключами шифрования.

4. Применить меры защиты сетевого периметра, включая брандмауэры, VPN-соединения и другие средства. Реализовать политики безопасности для ограничения доступа к сетевым ресурсам.

- Брандмауэры: Установка и настройка брандмауэров для защиты от несанкционированного доступа.
- VPN-соединения: Использование VPN для защиты данных при передаче через небезопасные сети.

- Политики безопасности: Реализация политик безопасности для ограничения доступа к сетевым ресурсам.

5. Обеспечить мониторинг сетевой активности, входящих и исходящих запросов, а также поведения пользователей. Реализовать системы обнаружения нештатных ситуаций для быстрого реагирования на потенциальные угрозы.

- Системы мониторинга: Внедрение систем мониторинга для отслеживания сетевой активности и поведения пользователей.

- Системы обнаружения вторжений (IDS): Использование IDS для выявления и предотвращения потенциальных угроз.

- Журналирование событий: Ведение журналов событий для анализа и расследования инцидентов.

6. Регулярно обновлять систему для обеспечения актуальной защиты и обучать персонал правилам безопасности информации и соблюдению политик защиты данных.

- Регулярные обновления: Регулярное обновление программного обеспечения и систем безопасности для защиты от новых угроз.

- Обучение персонала: Проведение регулярных тренингов для персонала по вопросам безопасности информации и соблюдению политик защиты данных.

- Аудит безопасности: Регулярный аудит системы безопасности для выявления и устранения уязвимостей.

7. Система должна периодически создавать резервные копии всех важных данных, включая голосовые записи, результаты распознавания, метаданные и технические настройки. Резервные копии должны храниться в отдельном безопасном хранилище, обеспечивающем защиту от потери данных в случае аварии.

- Автоматическое резервное копирование: Настройка автоматического создания резервных копий данных.

- Хранение резервных копий: Хранение резервных копий в отдельном безопасном хранилище.

- Проверка целостности данных: Регулярная проверка целостности резервных копий для обеспечения их доступности и корректности.

8. При возникновении аварийных ситуаций система должна автоматически восстанавливать данные из резервных копий. Это позволит минимизировать простой системы и предотвратить потерю важной информации.

- Автоматическое восстановление: Настройка автоматического восстановления данных из резервных копий.



- Тестирование восстановления: Регулярное тестирование процедур восстановления для проверки их эффективности.

- Мониторинг восстановления: Мониторинг процесса восстановления для обеспечения его корректности и полноты.

9. Система должна обладать методами защиты данных от случайного или умышленного повреждения, такими как механизмы, предотвращающие возможность записи или удаления файлов без соответствующих разрешений доступа. Также важно внедрить инструменты для мониторинга целостности данных с целью обнаружения повреждений и их автоматического восстановления.

- Контроль доступа: Настройка прав доступа для предотвращения несанкционированных изменений данных.

- Мониторинг целостности данных: Использование инструментов для мониторинга целостности данных.

- Автоматическое восстановление данных: Настройка автоматического восстановления данных при обнаружении повреждений.

10. Если восстановление работы системы на текущем месте невозможно, она должна быть способна к аварийному отключению и переносу работы на резервные серверы или инфраструктуру. Это позволит сократить время простоя и обеспечить сохранность данных.

- Аварийное отключение: Настройка процедур аварийного отключения системы.

- Перенос работы: Настройка процедур переноса работы на резервные серверы или инфраструктуру.

- Тестирование аварийного отключения: Регулярное тестирование процедур аварийного отключения и переноса работы.

11. Необходимо регулярно тестировать процедуры восстановления после аварийных ситуаций, чтобы проверить их эффективность и сократить время восстановления. Это включает проверку восстановления из резервных копий и переход на резервные ресурсы для обеспечения непрерывности работы системы.

- Регулярное проведение тестирования процедур восстановления.

- Документирование результатов: Документирование результатов тестирования для анализа и улучшения процедур восстановления.

- Обучение персонала: Обучение персонала процедурам восстановления и аварийного отключения.

## **ВЫВОДЫ ПО 2 ГЛАВЕ**

В заключение хотелось бы отметить, что разработанная модель индивидуальной траектории обучения представляет собой важный инструмент для персонализированного образования. Она обеспечивает возможность постоянного развития за счет одновременного использования различных образовательных стратегий, что делает её ценным ресурсом для студентов и преподавателей. Представленные сущности и их атрибуты, описанные в концептуальном моделировании, предоставляют основу для разработки базы данных для образовательной платформы. Эти данные позволяют пользователям управлять своими учебными планами, осуществлять образовательные операции и анализировать свою учебную деятельность.

Функциональное моделирование подтвердило вышеизложенную информацию, а также декомпозиция процесса разбила сложные процессы на мелкие этапы, которые позволят небольшими шагами дойти до оптимального решения поставленной задачи. Имитационная модель для тестирования образовательных стратегий представляет собой неотъемлемый инструмент. Она обеспечивает возможность анализа и оптимизации стратегий на исторических данных, что позволяет предсказать и оценить их эффективность в реальных условиях обучения.

В заключение, были сформированы требования к системе. Эти требования учитывают не только технические аспекты, но и потребности пользователей. Платформа должна быть не просто функциональной, но и удобной, надежной и производительной. Важно, чтобы система была простой в использовании, минимизировала ошибки и обеспечивала удовлетворение потребностей студентов и преподавателей. Надежность системы должна быть обеспечена за счет автоматического восстановления данных и регулярного создания резервных копий. Производительность системы должна гарантировать оперативное функционирование при минимальных затратах ресурсов. Поддержка системы должна обеспечивать её дальнейшее развитие и актуальность функций анализа образовательных данных. Все требования должны быть сформулированы с учетом стандартов и методологий разработки программного обеспечения, чтобы обеспечить полноценное функционирование и успешное использование платформы.

### 3 ПРОВЕДЕНИЕ ЭКСПЕРИМЕНТОВ И АНАЛИЗ РЕЗУЛЬТАТОВ

					Д 09.04.01 2023 23262867 МД	Лист
		Цапков А.В.		10.06.25		
Студ.	Фамилия	Подп.	Дата			59

### 3 ПРОВЕДЕНИЕ ЭКСПЕРИМЕНТОВ И АНАЛИЗ РЕЗУЛЬТАТОВ

#### 3.1 Проектирование информационной системы

##### 3.1.1 Выбор архитектуры информационной системы

Микросервисная архитектура представляет собой подход к разработке программного обеспечения, при котором приложение строится как набор небольших независимых сервисов, каждый из которых выполняет конкретную бизнес-функцию. В отличие от монолитных приложений, где все компоненты тесно связаны, микросервисы предлагают модульный подход с четким разделением ответственности. На рисунке 23 показано схематическое представление архитектуры приложения.

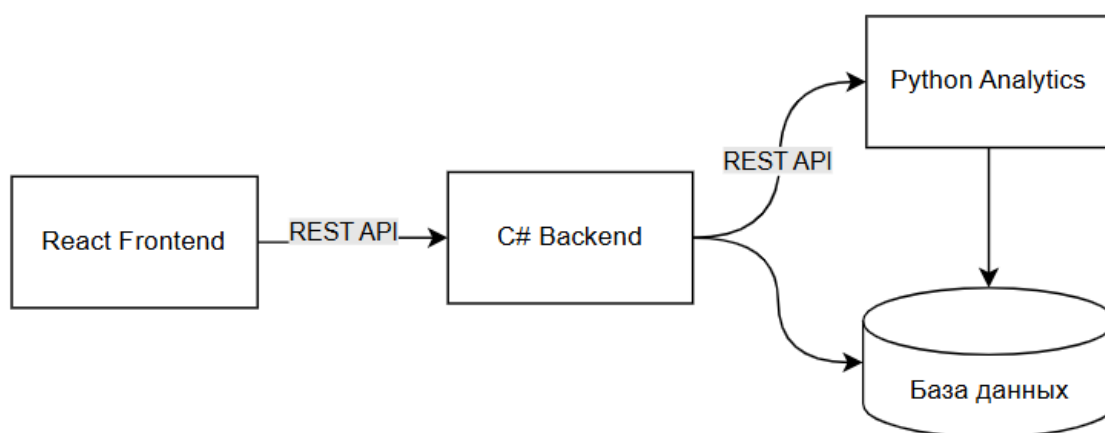


Рисунок 23 - Архитектура системы

Основные принципы микросервисной архитектуры:

- Декомпозиция по бизнес-функциям. Каждый микросервис должен быть ориентирован на выполнение конкретной бизнес-задачи. Это означает, что сервисы выделяются не по техническим критериям, а по бизнес-возможностям.
- Независимое развертывание. Одним из ключевых преимуществ микросервисов является возможность независимого развертывания. Каждый сервис может обновляться, масштабироваться и развертываться отдельно от других.
- Технологическая независимость. Микросервисы позволяют использовать различные технологии для разных сервисов. Это означает, что для каждого сервиса можно выбрать наиболее подходящие инструменты и языки программирования.
- Взаимодействие между микросервисами осуществляется через четко определенные интерфейсы.

- Децентрализованное управление данными. В микросервисной архитектуре каждый сервис отвечает за свои собственные данные. Такой подход позволяет выбирать наиболее подходящие решения для хранения данных для каждого сервиса.

- Устойчивость к сбоям. Микросервисы должны быть спроектированы с учетом возможных сбоев. Это означает, что система должна продолжать работать даже при отказе отдельных сервисов.

- Масштабируемость. Каждый микросервис может масштабироваться независимо в зависимости от нагрузки. Это позволяет эффективно использовать ресурсы, масштабируя только те сервисы, которые в этом нуждаются, а не всю систему целиком.

Преимущества микросервисной архитектуры:

1. Упрощение разработки и поддержки

Разделение приложения на небольшие сервисы значительно упрощает процесс разработки. Разработчики могут сосредоточиться на конкретных функциональных областях, не отвлекаясь на всю систему. Это также упрощает тестирование и отладку, так как проблемы можно изолировать в пределах одного сервиса.

2. Гибкость в выборе технологий

Микросервисная архитектура позволяет использовать различные технологии для разных сервисов. Это дает возможность выбирать наиболее подходящие инструменты для каждой конкретной задачи, а не использовать один стек технологий для всего приложения.

3. Улучшенная масштабируемость

Возможность независимого масштабирования отдельных сервисов позволяет эффективно использовать ресурсы. Можно масштабировать только те сервисы, которые испытывают высокую нагрузку, а не всю систему целиком. Это приводит к более эффективному использованию ресурсов и снижению затрат.

4. Упрощение развертывания

Независимое развертывание микросервисов значительно упрощает процесс внедрения изменений. Можно обновлять отдельные сервисы без необходимости развертывания всей системы. Это уменьшает риски и позволяет быстрее внедрять новые функции.

5. Повышенная отказоустойчивость

В микросервисной архитектуре отказ одного сервиса не приводит к отказу всей системы. Это значительно повышает общую надежность приложения. Кроме того, можно использовать различные механизмы для обработки сбоев, такие как повторные попытки и схемы отката.

## 6. Улучшенная производительность

Разделение на микросервисы позволяет оптимизировать производительность для каждого сервиса отдельно. Можно использовать различные подходы к кэшированию, балансировке нагрузки и другим механизмам оптимизации для каждого сервиса в зависимости от его конкретных требований.

## 7. Упрощение командной работы

Микросервисная архитектура позволяет разным командам работать над разными сервисами независимо. Это упрощает организацию работы и позволяет командам специализироваться на конкретных областях функциональности.

## 8. Улучшенная безопасность

Разделение на микросервисы позволяет применять различные механизмы безопасности для разных сервисов в зависимости от их требований. Это дает возможность более гибко настраивать политики безопасности и уменьшать потенциальные уязвимости.

## 9. Упрощение тестирования

Небольшой размер микросервисов значительно упрощает их тестирование. Можно создавать более фокусированные и эффективные тесты для каждого сервиса отдельно. Это приводит к более надежному коду и уменьшению количества ошибок.

## 10. Гибкость в развитии

Микросервисная архитектура позволяет легче вносить изменения в систему. Можно обновлять или заменять отдельные сервисы без необходимости переписывания всей системы. Это дает возможность постепенно улучшать систему и адаптироваться к изменяющимся требованиям.

Микросервисная архитектура предлагает множество преимуществ по сравнению с традиционными монолитными приложениями. Она обеспечивает гибкость, масштабируемость и устойчивость, которые критически важны для современных сложных приложений. Микросервисная архитектура улучшает процесс разработки, повышает надежность системы и обеспечивает лучшее соответствие бизнес-требованиям. Она позволяет создавать более гибкие и адаптируемые системы, которые могут эффективно развиваться вместе с изменяющимися потребностями бизнеса.

На рисунке 24 показана диаграмма развертывания системы. Система разворачивается внутри Docker. Docker содержит контейнеры «Web server Ngnix», «Web Server IIS», «Web server Gunicorn», «PostgreSQL». «Web server Ngnix», отвечает за внешнее API взаимодействия с клиентами, запросы и в случае надобности взаимодействует с «Web Server IIS» по REST API. Asp.Net core приложение на сервере IIS взаимодействует с базой данных «PostgreSQL» получения и записи данных. В случае необходимости сложного анализа данных «Web Server

IIS» обращается к «Web server Gunicorn» по REST API который выполняет сложные аналитические операции. «Web server Gunicorn» обеспечивает работу Python приложения. Python app отвечает за кластеризацию, анализ ответов, анализ данных учащихся и рекомендацию учебного материала.

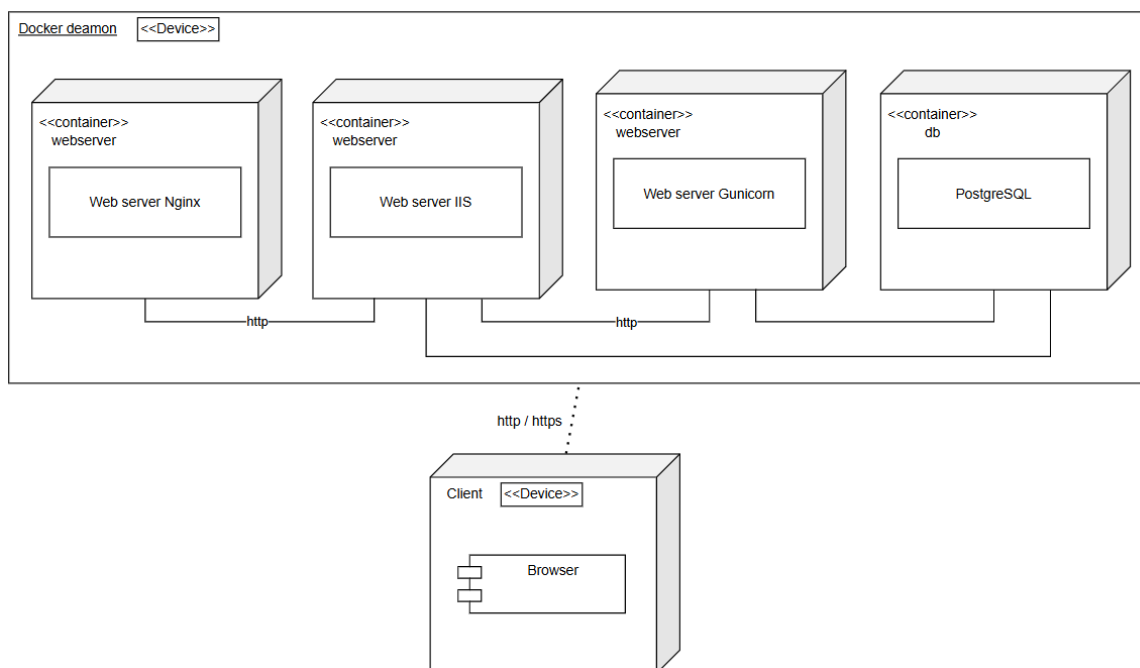


Рисунок 24 – Диаграмма развертывания

### 3.2 Инфологическое проектирование

Инфологическая модель системы формирования индивидуальной траектории обучения представляет собой диаграмму "сущность-связь", которая отображает объекты и их взаимосвязи в системе. Эта модель, показанная на рисунке 25 включает различные сущности, такие как учащиеся, курсы, модули, учебные материалы, задания, результаты, наставники, обратная связь, геймификация, аналитика и конкретная аналитика по тенденциям, слабым/сильным сторонам.

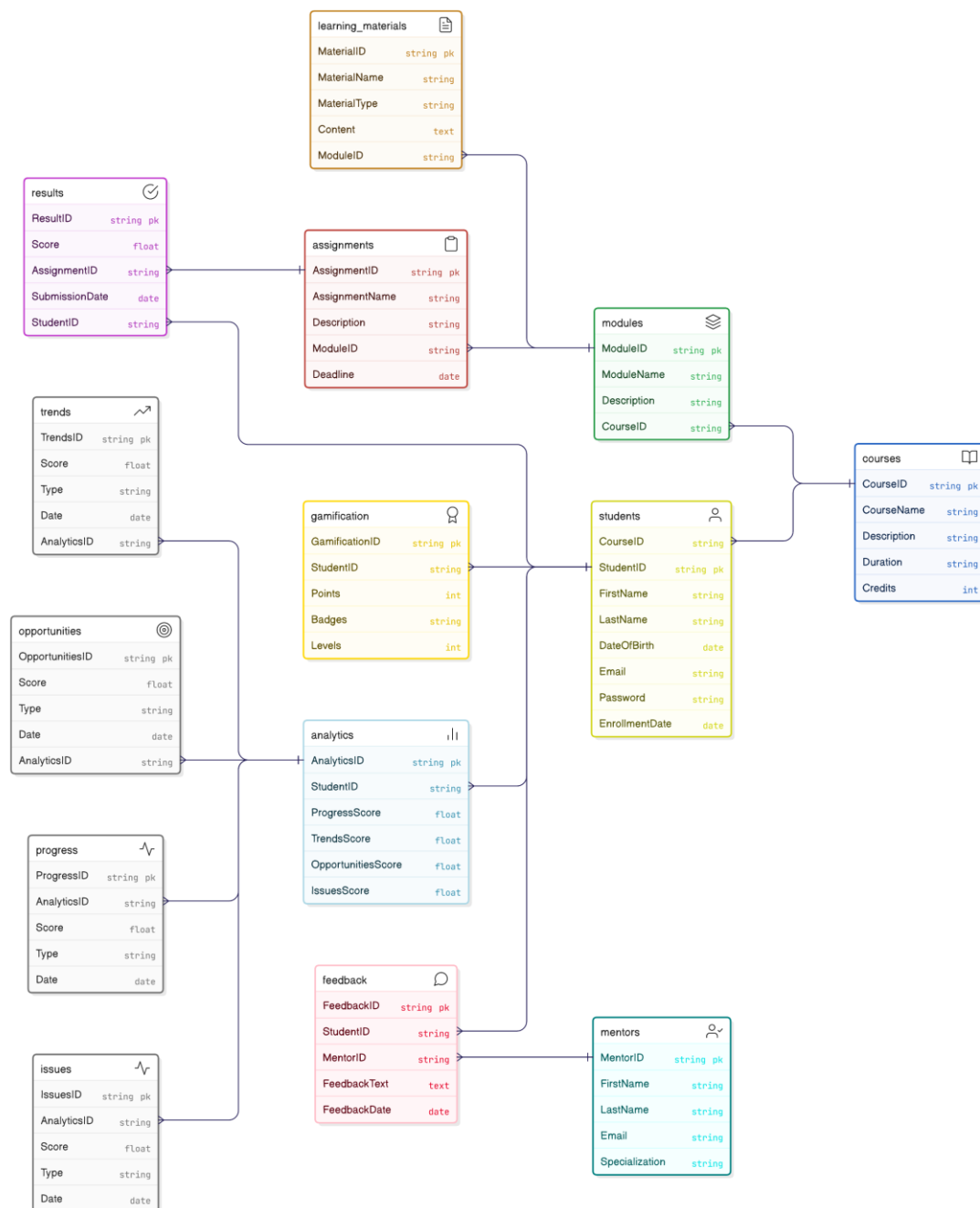


Рисунок 25 - Инфологическая модель

- Учащиеся характеризуются уникальным идентификатором, именем, фамилией, датой рождения, электронной почтой, паролем и датой зачисления.
- Курсы включают уникальный идентификатор, название курса, его описание, продолжительность и количество кредитов – очков геймификации.
- Модули содержат уникальный идентификатор, название модуля, его описание и идентификатор курса.



- Учебные материалы обладают уникальным идентификатором, названием материала, типом материала, содержанием и идентификатором модуля.

- Задания включают уникальный идентификатор, название задания, его описание, срок выполнения и идентификатор модуля.

- Результаты содержат уникальный идентификатор, идентификаторы учащегося и задания, оценку и дату сдачи.

- Наставники характеризуются уникальным идентификатором, именем, фамилией, электронной почтой и специализацией.

- Обратная связь включает уникальный идентификатор, идентификаторы учащегося и наставника, текст обратной связи и дату обратной связи.

- Геймификация содержит уникальный идентификатор, идентификатор учащегося, баллы, значки и уровни.

- Аналитика включает уникальный идентификатор, идентификатор учащегося, оценку общего прогресса, оценку общих тенденций, оценку общих проблем и оценку общих возможностей.

- Тенденции включают уникальный идентификатор, идентификатор аналитики, оценку тенденций в определенный промежуток времени, тип оценки, дату.

- Проблемы включают уникальный идентификатор, идентификатор аналитики, оценку проблем в определенный промежуток времени, тип оценки, дату.

- Прогресс включают уникальный идентификатор, идентификатор аналитики, оценку прогресса в определенный промежуток времени, тип оценки, дату.

- Возможности включают уникальный идентификатор, идентификатор возможностей, оценку тенденций в определенный промежуток времени, тип оценки, дату.

Связи между сущностями в модели также играют важную роль:

- Учащиеся и курсы связаны отношением многие ко многим через промежуточную таблицу зачисления, которая содержит идентификатор зачисления, идентификаторы учащегося и курса, а также дату зачисления.

- Курсы и модули связаны отношением один ко многим.

- Модули и учебные материалы, а также модули и задания также связаны отношением один ко многим.

- Учащиеся и результаты, а также задания и результаты связаны отношением один ко многим.

- Учащиеся и наставники связаны отношением многие ко многим через промежуточную таблицу наставничества, включающую идентификатор наставничества, идентификаторы учащегося и наставника, дату начала и окончания наставничества.

- Учащиеся и обратная связь, а также наставники и обратная связь связаны отношением один ко многим.

- Учащиеся и геймификация, а также учащиеся и аналитика связаны отношением один к одному.

- Аналитика связана с тенденциями, проблемами, прогрессом, возможностями связью один ко многим.

Эти сущности и их взаимосвязи позволяют эффективно организовать данные о учащихся, курсах, модулях, учебных материалах, заданиях, результатах, наставниках, обратной связи, геймификации, общей и конкретной аналитике, обеспечивая структуру для хранения и обработки информации в системе формирования индивидуальной траектории обучения.

### 3.3 Проектирование интерфейса

На рисунке 26 изображена вкладка обучения. В левом меню разделение курса на логические единицы – модули, модули в свою очередь делятся на отдельные группы заданий. Задания делятся на три типа: теория, практика и ответы на теорию, переход между типами происходит через кнопки.

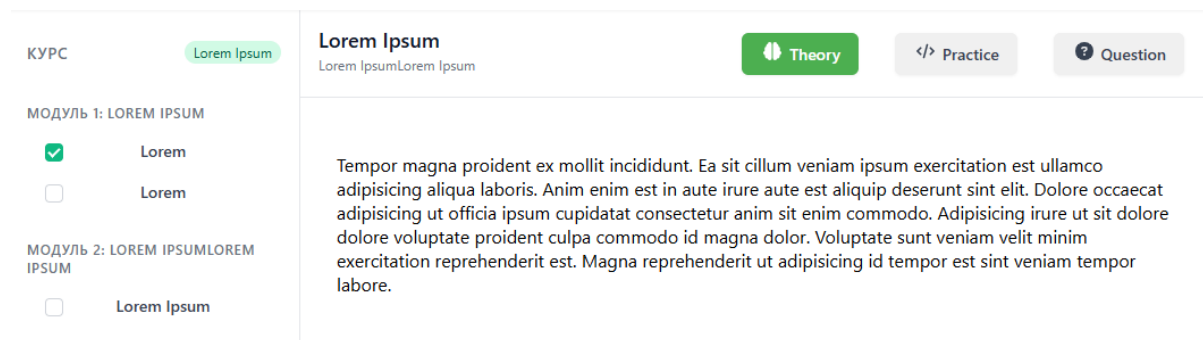


Рисунок 26 - Область обучения

Важную часть приложения играет аналитика успеваемости. Поэтому важно визуализировать результаты и прогресс обучения. Вся аналитика находится на одной вкладке. На рисунках 27 - 30 показаны примеры графиков для аналитики.

## Выполнение заданий

Неделя Месяц Год

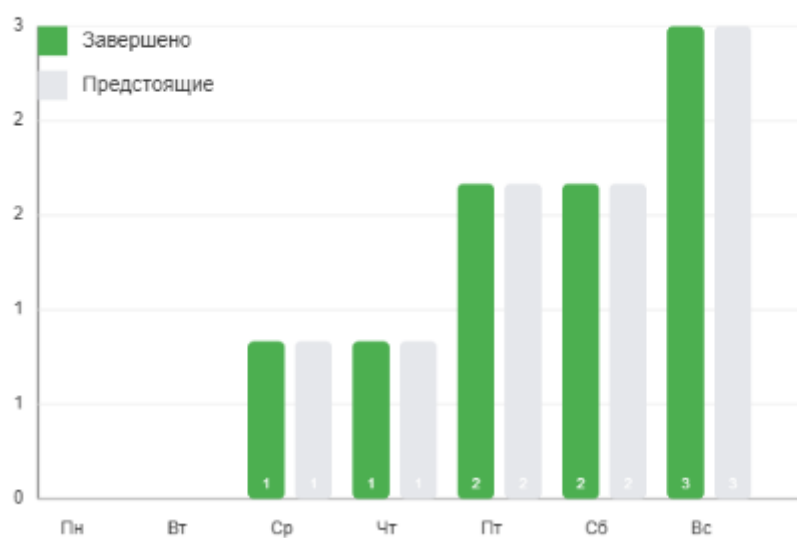


Рисунок 27 - график выполнения заданий.

## Прогресс курсов

[Показать все](#)

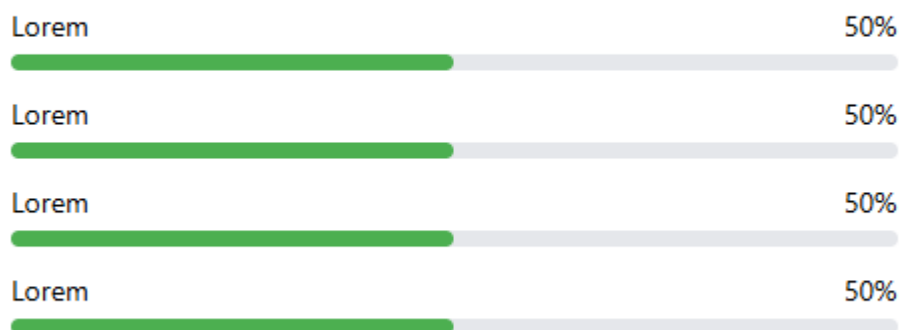


Рисунок 28 - Прогресс прохождения курсов

## Прогресс

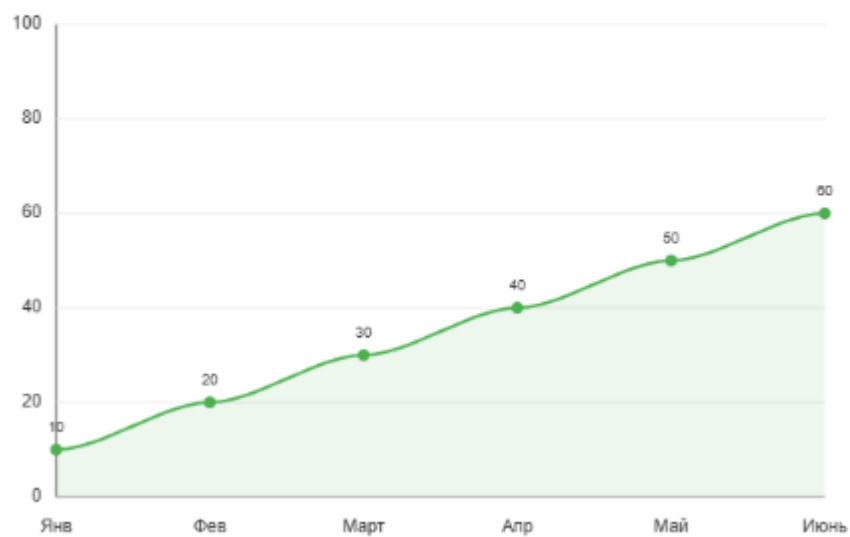
[Подробнее](#)

Рисунок 29 - Прогресс в динамике

## Аналитика по модулям

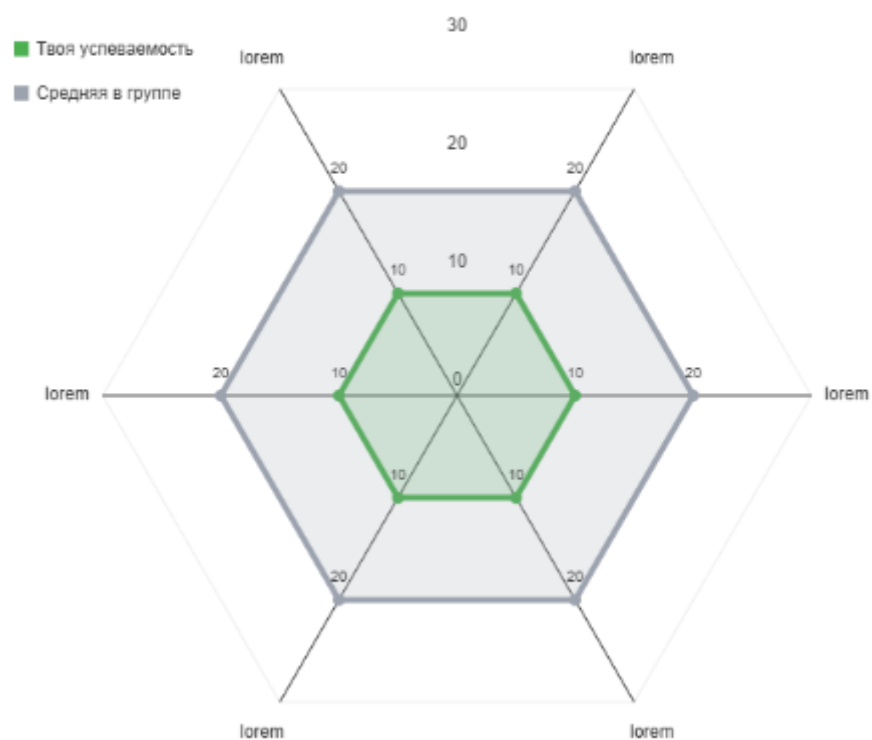
[Сравнить](#)

Рисунок 30 - Аналитика по модулям

Исходя из аналитических данных, пользователю рекомендуются следующие задания для прохождения представленные на рисунке 31.

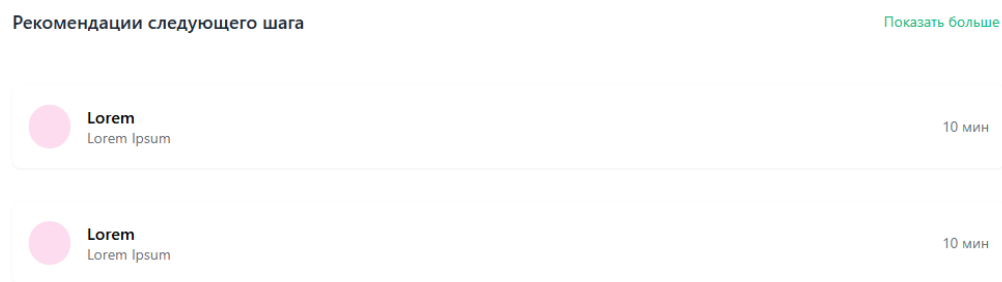


Рисунок 31 - Рекомендации

Для обратной связи используется текстовый чат пользователей, представленный на рисунке 32.

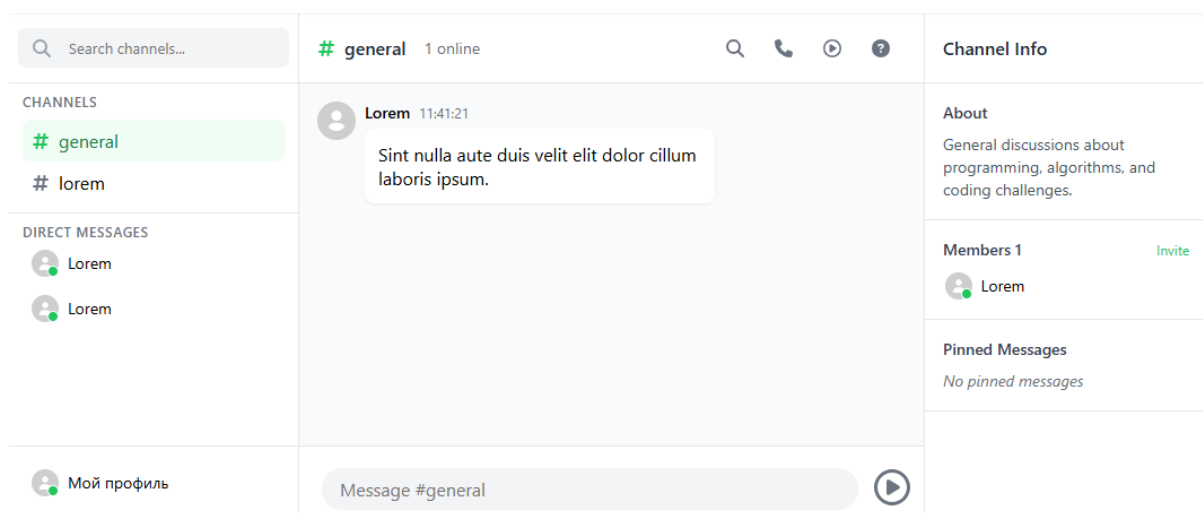


Рисунок 32 - Вкладка чата

### 3.4 Программная реализация проекта

#### 3.4.1 Выбор средств реализации

При разработке системы формирования индивидуальной траектории обучения с использованием алгоритмов машинного обучения, выбор технологий играет ключевую роль в обеспечении эффективности, надежности и масштабируемости. Рассмотрим преимущества каждой из выбранных технологий и их применимость к данной системе.

C# является мощным языком программирования, который обеспечивает высокую производительность и безопасность. Он идеально подходит для разработки серверной части системы благодаря своей тесной интеграции с платформой .NET. Это позволяет использовать мощные библиотеки и инструменты для создания веб-приложений и сервисов. Поддержка асинхронного программирования в C# позволяет разрабатывать высокопроизводительные и

отзывчивые приложения, что особенно важно для обработки и хранения данных учащихся и учебных материалов.

**PostgreSQL** — это надежная и масштабируемая система управления базами данных (СУБД), которая отлично подходит для хранения и управления большими объемами данных. Она поддерживает сложные SQL-запросы и предоставляет мощные инструменты для анализа данных учащихся по их успеваемости. Расширяемость PostgreSQL позволяет создавать пользовательские типы данных, функции и операторы, что делает её очень гибкой и адаптируемой под различные задачи. Это особенно важно для обеспечения безопасного хранения и управления данными учащихся, курсов и учебных материалов.

**React** позволяет разрабатывать пользовательские интерфейсы с использованием компонентного подхода, что делает код более модульным и легко поддерживаемым. Высокая производительность React обеспечивается за счет использования виртуального DOM, что гарантирует быструю и плавную работу интерфейса. Богатая экосистема библиотек и инструментов React упрощает разработку и расширение функциональности, что критически важно для создания интуитивно понятных и адаптивных пользовательских интерфейсов.

**TypeScript** добавляет статическую типизацию к JavaScript, что позволяет выявлять ошибки на этапе компиляции и улучшает качество кода. Полная совместимость TypeScript с JavaScript позволяет постепенно внедрять его в существующие проекты. Поддержка современных возможностей JavaScript и дополнительные инструменты для разработки делают TypeScript идеальным выбором для улучшения качества и надежности кода.

**RESTful API** предоставляет стандартизированный способ взаимодействия между клиентом и сервером, а так же между микросервисами, что упрощает интеграцию и поддержку. Легкость масштабирования RESTful API позволяет обрабатывать большое количество запросов. Поддержка различных форматов данных, таких как JSON и XML, делает его очень гибким и адаптируемым под различные задачи, что особенно важно для обеспечения взаимодействия между клиентом и сервером.

**Python** — это язык программирования с простым и читаемым синтаксисом, что делает его идеальным выбором для быстрой разработки и прототипирования. Богатая экосистема библиотек Python для работы с данными, машинного обучения и анализа данных обеспечивает доступ к мощным инструментам для реализации алгоритмов машинного обучения. Поддержка множества библиотек для машинного обучения, таких как scikit-learn, делает Python идеальным выбором для разработки и тестирования алгоритмов машинного обучения, что критически важно для адаптации и оптимизации учебных планов.

**Docker** предоставляет мощные инструменты для контейнеризации приложений, что значительно упрощает их развертывание и масштабирование. Использование Docker

позволяет создавать изолированные среды для различных компонентов системы, обеспечивая согласованность и переносимость между различными средами разработки, тестирования и продакшена. Это особенно важно для обеспечения быстрого развертывания и масштабирования системы, а также для упрощения управления зависимостями и конфигурациями.

Выбранный стек технологий, включающий C#, PostgreSQL, React, TypeScript, RESTful API, Python и Docker, обеспечивает надежность, производительность и масштабируемость системы формирования индивидуальной траектории обучения. Каждая из этих технологий вносит свой вклад в создание эффективной и удобной системы, способной адаптироваться к потребностям учащихся и предоставлять персонализированные учебные планы.

### 3.4.2 Программная реализация

Для оценки успеваемости, учащиеся на каждом этапе проходят практическую и теоретическую проверку знаний. Оценка практики происходит в зависимости от направления обучения. Теоретическая часть проверяется двумя способами: тестовые задания и развернутый ответ на вопрос. Для оценки развернутого ответа используются алгоритмы анализа текста.

Проверяется наличие ключевых концепций в текстовом ответе в функции показанной на листинге 1.

Листинг 1 - Функция проверки концепций

```
def check_concepts(self, required_concepts, text):
    text = transform_text(text)
    found_concepts = [concept for concept in required_concepts
                      if concept in text.lower()]
    return len(found_concepts) / len(required_concepts)
```

Проверяет наличие ключевых концепций ООП в текстовом ответе студента.

#### Параметры:

- text (string): Текст ответа студента, который нужно проверить на наличие ключевых концепций.
- required\_concepts (string[]) список концепций по теме вопроса, которые в идеале должны присутствовать в ответе.

#### Возвращает:

- float: Доля найденных ключевых концепций в диапазоне от 0 до 1. 1.0 означает, что все ключевые концепции найдены, 0.0 означает, что ни одна концепция не найдена.

#### Описание:

Функция проверяет текст на наличие ключевых терминов, связанных с темой вопроса. Текст ответа преобразуется в нормальную форму, чтобы форма слов была в нужном числе, времени и т.д. Возвращает долю найденных концепций от общего количества концепций в списке.

В одной из функций происходит оценка развернутости ответа.

Листинг 2 - функция оценки развернутости ответа

```
def check_completeness(self, text):
    doc = nlp(text)
    sentences = [sent.text for sent in doc.sents]
    if len(sentences) < 2:
        return 0.3
    avg_sentence_length = sum(len(sent.split()) for sent in sentences) /
len(sentences)
    completeness = min(1.0, len(sentences) * 0.2 + avg_sentence_length *
0.01)
    return completeness
```

#### Параметры:

- text (string): Текст ответа студента, который нужно оценить на развернутость.

#### Возвращает:

- float: Оценка развернутости ответа в диапазоне от 0 до 1. 1.0 означает очень развернутый ответ, 0.0 означает очень краткий ответ.

#### Описание:

Функция оценивает развернутость ответа на основе:

1. Количества предложений в ответе
2. Средней длины предложений

Минимальное значение возвращается, если в ответе меньше 2 предложений. Оценка рассчитывается как комбинация количества предложений и их средней длины.

В одной из функций оценивается грамматическую правильность текстового ответа студента ошибки листинг 3.



### Листинг 3 - Функция проверки грамматики

```
def evaluate_grammar(self, text):
    try:
        corrected = str(TextBlob(text).correct())
        original_words = len(text.split())
        corrected_words = len(corrected.split())
        grammar_score = 1.0 - min(1.0, abs(original_words - corrected_words)
/ max(1, original_words))
        return grammar_score
    except Exception as e:
        print(f"Error in grammar evaluation: {e}")
        return 0.5
```

#### Параметры:

- text (string): Текст ответа студента, который нужно проверить на грамматическую правильность.

#### Возвращает:

- float: Оценка грамматической правильности в диапазоне от 0 до 1. 1.0 означает грамматически правильный текст, 0.0 означает текст с множеством грамматических ошибок.

#### Описание:

Функция использует TextBlob для исправления грамматических ошибок в тексте. Сравнивает количество слов в оригинальном и исправленном тексте. Чем меньше изменений, тем выше оценка грамматики. В случае ошибки возвращает среднее значение 0.5.

Следующая функция на листинге 4 оценивает релевантность текстового ответа студента по сравнению с эталонными ответами.

### Листинг 4 - Функция оценки релевантности

```
def evaluate_relevance(self, definitions, text):
    scores = []
    for definitions in self.correct_definitions:
        embeddings = self.model.encode([text, definitions])
        similarity = util.cos_sim(embeddings[0], embeddings[1]).item()
        scores.append(similarity)
    return max(scores) if scores else 0.0
```

### Параметры:

- `definitions (string[])`: эталонные ответы на вопрос.
- `text (string)`: Текст ответа студента, который нужно оценить на релевантность.

### Возвращает:

- `float`: Оценка релевантности в диапазоне от 0 до 1. 1.0 означает высокую релевантность эталонным определениям, 0.0 означает низкую релевантность.

### Описание:

Функция использует `sentence-transformers` для оценки сходства текста студента с эталонными ответами. Рассчитывает косинусное сходство между эмбедингами текста студента и каждого эталонного ответа. Возвращает максимальное значение сходства.

Получив все эти значения, получаем итоговую оценку, учитывая несколько критериев по следующему правилу:

1. Релевантность ответа (50% веса)
2. Наличие ключевых концепций (30% веса)
3. Развернутость ответа (10% веса)
4. Грамматическая правильность (10% веса)

В итоге получаем взвешенную сумму всех оценок, ограниченную диапазоном [0, 1].

На листинге 5 представлена функция для разделения студентов на группы.

Листинг 5 - функция кластеризации студентов

```
def full_cluster_analysis(students):
    data = []
    for student in students:
        course_avg = sum(student.course_scores.values()) /
len(student.course_scores) if student.course_scores else 0
        unit_avg = sum(student.unit_scores.values()) /
len(student.unit_scores) if student.unit_scores else 0
        topic_avg = sum(student.topic_scores.values()) /
len(student.topic_scores) if student.topic_scores else 0
        test_avg = sum(student.test_scores) / len(student.test_scores) if
student.test_scores else 0
        practical_avg = sum(student.practical_scores) /
len(student.practical_scores) if student.practical_scores else 0
        comprehensive_avg = sum(student.comprehensive_scores) /
len(student.comprehensive_scores) if student.comprehensive_scores else 0
        progress = calculate_progress(student)
```

#### Окончание листинга 5

```
data.append([
    course_avg, unit_avg, topic_avg,
    test_avg, practical_avg, comprehensive_avg,
    progress
])
if len(data) < 2:
    return []
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)
pca = PCA(n_components=2)
reduced_data = pca.fit_transform(scaled_data)
n_clusters = find_optimal_clusters(students)
if n_clusters < 1:
    return []
kmeans = KMeans(n_clusters=n_clusters, random_state=42)
clusters = kmeans.fit_predict(reduced_data)
return list(zip(students, clusters))
```

#### Параметры:

- students – список студентов

#### Возвращает:

- список студентов с кластером

#### Описание:

Функция разделяет студентов на кластеры с помощью K-means алгоритма, основываясь на средних оценках учащихся.

Основываясь на информации о кластере и студенте для него подбираются конкретные задания и учебные материалы. Функция на листинге 6 выполняет операции выборки для каждого из типов заданий практика, тест и теория.

Листинг 6 - функция выборки заданий.

```
def select_tasks_with_progress(student, task_db, cluster_info):
    student_score = calculate_student_score(student)
    difficulty_level = determine_difficulty_level(student_score,
cluster_info)
    current_topics = get_current_topics(student)
    topic_progress = get_topic_progress(student)
```

#### Окончание листинга 6

```
selected_tasks = []
test_tasks = task_db.get_tasks_by_type('test')
test_tasks = [t for t in test_tasks if t.topic in current_topics]
test_tasks.sort(key=lambda t: (topic_progress.get(t.topic, 0),
t.difficulty))
for task in test_tasks:
    if task.difficulty <= difficulty_level + 1 and task.difficulty >=
difficulty_level - 1:
        selected_tasks.append(task)
        if len(selected_tasks) >= 3:
            break
practical_tasks = task_db.get_tasks_by_type('practical')
practical_tasks = [t for t in practical_tasks if t.topic in
current_topics]
practical_tasks.sort(key=lambda t: (topic_progress.get(t.topic, 0),
t.difficulty))
for task in practical_tasks:
    if task.difficulty <= difficulty_level + 1 and task.difficulty >=
difficulty_level - 1:
        selected_tasks.append(task)
        if len(selected_tasks) >= 5:
            break
theory_tasks = task_db.get_tasks_by_type('theory')
theory_tasks = [t for t in theory_tasks if t.topic in current_topics]
theory_tasks.sort(key=lambda t: (topic_progress.get(t.topic, 0),
t.difficulty))
for task in theory_tasks:
    if task.difficulty <= difficulty_level + 1 and task.difficulty >=
difficulty_level - 1:
        selected_tasks.append(task)
        if len(selected_tasks) >= 7:
            break

return selected_tasks
```

Основываясь на аналитических данных формируются индивидуальная траектория. Задания подстраиваются конкретно под успеваемость каждого студента, чем больше score студента, тем сложнее задания для него по определенной теме. Кроме этого объединение по кластерам студентов – позволяет сформировать группы, в которых все студенты имеют примерно схожую успеваемость, что позволяет так же подбирать преподавателей с определенным темпом и стилем обучения. При помощи листингов описаны основные технические аспекты автоматизированного анализа студентов и подбора для них своего плана обучения.

### **3.5 Проведение эксперимента**

В данной работе изучается эффективность внедрения системы формирования индивидуальных образовательных траекторий на основе ИИ в современном образовании.

#### **Цель исследования**

Сравнить эффективность адаптивной системы обучения с традиционным методом преподавания на примере курса "Основы JavaScript".

#### **Основные положения**

- 1 Анализ с помощью интеллектуальных алгоритмов успеваемости учащихся, и подбор рекомендационных учебных материалов, на основе индивидуальных особенностей, позволит добиться ускоренного усвоения знаний, по сравнению с традиционным методом обучения.
- 2 Автоматическая корректировка образовательной траектории на основе анализа текущего прогресса учащегося способна повысить общую эффективность усвоения учебного материала в сравнении с традиционным методом обучения.
- 3 Разнообразные методы обучения способствуют развитию различных аспектов интеллекта и формируют специфические учебные навыки в зависимости от основного акцента в образовательном процессе.

#### **Методология**

В данном исследовании приняли участие индивидуумы, перечисленные в таблице 1. Участники были разделены на две группы, каждая из которых имела своего куратора в лице преподавателя. Группа А, именуемая адаптивной, обучалась посредством специализированной платформы, которая обеспечивала автоматический подбор сложности заданий в зависимости от уровня подготовки учащихся. Кроме того, платформа предоставляла рекомендации по повторению тем, основанные на анализе допущенных ошибок. Это позволяло индивидуализировать процесс обучения и сосредоточиться на слабых местах каждого участника.

Группа Б, традиционная, обучалась по классической методике, включающей лекции и фиксированные домашние задания. Для всех участников этой группы был установлен единый порядок изучения тем, а проверка выполненных заданий осуществлялась преподавателем без использования автоматической адаптации. Такой подход обеспечивал единообразие учебного процесса для всех участников группы.

Таблица 1 - участники эксперимента

Группа	Кол-во	Опыт в программировании	Опыт в JS
А	3	Нет	Нет
А	2	Да (1-3 года)	Нет
В	2	Нет	Нет
В	2	Да (1-3 года)	Нет

JavaScript». Важно отметить, что между группами не было пересечений в лекционных, практических и тестовых заданиях, что позволяло избежать взаимного влияния и обеспечивало независимость результатов. Продолжительность обучения составляла шесть недель. Формат курса включал теоретические лекции и учебные материалы, практические задания, а также проверочные работы, направленные на закрепление полученных знаний и навыков.

Для оценки эффективности обучения использовались различные метрики, которые позволяли сравнить результаты выполнения трех типов заданий. Практические задания оценивались по критериям корректности и эффективности написанного кода с использованием автоматических тестов. Тестовые задания измеряли точность выбора правильных ответов, выражаемую в процентах правильных ответов. Развернутые ответы оценивались на точность и глубину объяснения концепций, при этом оценка проводилась независимым преподавателем, что обеспечивало объективность и непредвзятость.:

Таблица 2 - типы заданий

Тип задания	Критерии оценки	Как измеряется
Практические	Корректность и эффективность кода	Автоматические тесты
Тестовые	Точность выбора правильного ответа	% правильных ответов
Развернутые ответы	Точность и глубина объяснения концепций	Оценка независимого преподавателя.

Таким образом, данная методология исследования обеспечивала комплексный подход к оценке эффективности различных методов обучения, позволяя выявить их преимущества и недостатки в контексте изучения основ JavaScript.

## Результаты

### 1. Практические задания (кодирование):

В ходе исследования было выявлено, что участники адаптивной группы, обладающие опытом в программировании, демонстрировали значительный прогресс, достигая уровня выполнения заданий на 90% и более к пятой неделе обучения. В то же время, новички, входящие в группу А, показали результаты на 15-20% выше по сравнению с их коллегами из группы традиционного метода обучения. Это свидетельствует о высокой эффективности адаптивной системы обучения как для новичков в программировании, так и для учащихся имеющих такой опыт.

Таблица 3 - Оценка практических заданий

Неделя	Группа А (новички)	Группа А (программисты)	Группа В (новички)	Группа В (программисты)
1	38,20	57,10	33,50	52,80
2	46,70	68,30	42,90	63,40
3	59,10	78,60	53,20	72,80
4	72,40	87,20	65,70	81,50
5	81,90	93,40	74,20	88,10
6	88,30	96,70	82,60	91,30

Особенно показательным является тот факт, что максимальный разрыв в результатах между новичками группы А и группы В наблюдался на четвертой неделе обучения, составив 16.7% в пользу адаптивной группы. Это подчеркивает важность индивидуализированного подхода к обучению, который позволяет учащимся значительно быстрее усваивать материал.

Итоговая дельта изменения для группы А +50,1 у новичков +39,6 учащихся с опытом программирования против +49,1 и +38,5 соответственно.

### 2. Тестовые вопросы (множественный выбор):

Анализ результатов тестовых заданий показал, что разрыв в успеваемости между программистами, обучающимися по разным методикам на этом сроке, был минимален и составлял всего 2-3%. Это говорит о том, что обе методики обучения примерно одинаково эффективны для лиц с опытом в программировании, когда речь идет о тестовых знаниях. Однако, адаптивная система обучения давала новичкам преимущество в 4-7% по точности выполнения тестовых заданий. Это еще раз подтверждает, что индивидуализированный подход более эффективен для новичков.

Таблица 4 - Оценка тестовых заданий

Неделя	Группа А (новички)	Группа А (программисты)	Группа В (новички)	Группа В (программисты)
1	47,3	62,8	45,1	60,3
2	58,6	73,4	55,2	68,9
3	67,2	82,1	63,8	77,5
4	76,5	88,3	72,1	84,7
5	83,7	92,6	79,3	89,2
6	87,4	94,8	84,1	91,5

Вывод:

Критической точкой роста для всех подгрупп стала третья неделя обучения. На этом этапе наблюдался значительный скачок в результатах, что может быть связано с адаптацией учащихся к учебному процессу и усвоением базовых концепций.

### 3. Развернутые ответы

При оценке развернутых ответов было отмечено, что программисты, обучающиеся по традиционной методике, стабильно получали высокие оценки, превышающие 9 баллов. Это указывает на то, что традиционная методика обучения способствует глубокому пониманию материала и умению четко и развернуто излагать свои мысли.

Таблица 5 - Оценки развернутых ответов

Неделя	Группа А (новички)	Группа А (программисты)	Группа В (новички)	Группа В (программисты)
1	3,7	5,0	4,2	5,2
2	5,3	6,7	5,9	7,1
3	6,1	7,5	6,8	8,0
4	6,7	8,2	7,5	8,6
5	7,2	8,7	8,1	9,0
6	7,6	9,1	8,5	9,3

Интересно отметить, что новички из группы А так и не смогли догнать своих коллег из традиционной группы к шестой неделе обучения. Максимальный разрыв в оценках между новичками двух групп наблюдался на пятой и шестой неделе, составив 0.9 балла в пользу группы В. Это может свидетельствовать о том, что традиционная методика обучения, несмотря на свою менее выраженную эффективность на начальных этапах, позволяет новичкам достичь сопоставимых результатов к концу курса.



## Интерпретация результатов

Проведенное исследование позволило выявить ключевые аспекты эффективности различных методик обучения. Адаптивная система обучения продемонстрировала высокую эффективность, особенно для новичков, не имеющих опыта в программировании. Индивидуализированный подход способствует более быстрому усвоению материала, особенно для практических результатов, дельта изменения score за 6 недель у группы А во всех случаях имела более быстрый рост знаний по оценкам заданий. Разница дельт первой и шестой недели для группы А в сравнении с группой Б составили +1,5% для практических заданий, +0,95% для тестовых заданий, +0,2 баллов оценки для развернутых ответов. Это подтверждает первое положение о том, что подбор рекомендационных учебных материалов, на основе индивидуальных особенностей, позволит добиться ускоренного усвоения знаний. Таким образом положение 1 - доказано.

Начальный этап обучения является ключевым для новичков, так как непонимание материала на этом этапе в традиционной системе обучения может препятствовать дальнейшему обучению. Благодаря чему максимальная эффективность с использованием анализа успеваемости и корректировкой траектории обучения помогла достичь более высоких результатов по практическим и тестовым заданиям. Для практики разница итогового показателя составила у новичков +5,7%, у программистов +5,4% в пользу группы А. Для тестового задания обе подгруппы группы А опережают подгруппы группы Б на +3,3%. Однако результаты в оценке развернутых ответов оказались выше у группы В. +0.9 балла для у новичков, +0,2 у группы программистов. Таким образом автоматическая корректировка образовательной траектории на основе анализа текущего прогресса учащегося не превышает итоговую эффективность традиционного метода по всем показателям, что опровергает положение 2.

Индивидуальный подход позволяет учащимся повторять материал, что способствует лучшему усвоению знаний. Традиционная методика обучения показала свою эффективность в формировании глубокого понимания материала и умения развернуто излагать свои мысли. Это особенно проявилось в результатах программистов группы Б, которые стабильно получали высокие оценки за развернутые ответы. Традиционный метод обучения обеспечивает глубину понимания материала, что особенно важно для программистов. Несмотря на то, что в адаптивном обучении также присутствуют лекции с преподавателем, им уделяется значительно меньше времени по сравнению с традиционным подходом. Из этого следует что положение 3, а именно - наличие различности развития учебных навыков при разных подходах обучения доказано.

### **ВЫВОДЫ ПО 3 ГЛАВЕ**

Для разработки архитектуры системы формирования индивидуальной траектории обучения была выбрана микросервисная архитектура. Данный подход позволил создать масштабируемое и отказоустойчивое приложение, способное эффективно справляться с высокими нагрузками и обеспечивать непрерывность учебного процесса.

В процессе разработки платформы использовался язык программирования C# для создания серверной части приложения. Для разработки пользовательского интерфейса были выбраны React и TypeScript, что позволило создать динамичный и интерактивный интерфейс. Python применялся для реализации алгоритмов машинного обучения и анализа данных, что способствовало более точной адаптации учебного процесса под индивидуальные потребности учащихся. Кроме того, использовался Docker для контейнеризации приложений, что упростило развертывание и управление сервисами. В качестве системы управления базами данных была выбрана PostgreSQL, что обеспечило надежное и эффективное хранение данных.

Инфологическое проектирование сыграло ключевую роль в тщательной проработке структуры будущей базы данных. Оно позволило установить связи и зависимости между её элементами, обеспечивая тем самым целостность и согласованность данных.

Проектирование интерфейса также являлось важной частью процесса разработки. Оно позволило ускорить процесс создания дизайна платформы, сделав его более удобным и интуитивно понятным для пользователей. В ходе проектирования интерфейса были учтены современные тенденции в области пользовательского опыта и юзабилити, что способствовало созданию привлекательного и функционального интерфейса.

В разделе программной реализации описаны ключевые компоненты системы: анализ ответов, кластеризация учащихся, подбор заданий на основании успеваемости.

Экспериментальная часть работы продемонстрировала практическую работоспособность и эффективность разработанного приложения. В ходе экспериментов были проведены многочисленные тесты, подтвердившие стабильность и надежность системы. Результаты экспериментов позволили подвести итоги всего комплекса выполненных работ, подтвердив, что разработанная система формирования индивидуальной траектории обучения соответствует поставленным требованиям и способна улучшить процесс обучения.

## ЗАКЛЮЧЕНИЕ

					Д 09.04.01 2023 23262867 МД	Лист
		Цапков А.В.		10.06.25		
Студ.	Фамилия	Подп.	Дата			83

## ЗАКЛЮЧЕНИЕ

В ходе работы над выпускной квалификационной работой была выявлена проблема: несмотря на научные обоснования эффективности индивидуального образования и возможность использования машинных алгоритмов в обучении, существует крайне мало автоматизированных решений, которые на основе анализа знаний учащихся систематизируют и формируют траекторию обучения. В связи с этим было принято решение о создании сервиса, способного решить эту проблему.

В первой главе проекта представлены конкуренты – платформы, которые внедряют в свой функционал передовые технологии анализа и рекомендаций. Также описаны преимущества и недостатки каждого из них, что подчеркивает моменты, которые необходимо реализовать в новом сервисе. Важной частью главы является формирование принципов, которые служат основой для всей дальнейшей работы.

В рамках проекта была исследована математическая модель, описывающая процесс формирования индивидуальных образовательных траекторий. Были выполнены концептуальное моделирование, раскрывающее сущности системы, их роли и функциональное моделирование, описывающее процесс автоматизации формирования индивидуальной траектории обучения так, как он осуществляется в данный момент и каким он будет после реализации сервиса. Имитационная модель выступила инструментом для тестирования стратегий, применимых к алгоритмам машинного обучения в рамках формирования образовательных траекторий. Требования, сформированные с учетом стандартов, позволят системе качественно функционировать.

В третьей главе диссертационной работы описана архитектура информационной системы, в качестве которой используется микросервисная архитектура. Инфологическое проектирование системы позволило продумать все связи внутри базы данных и сущности, необходимые для эффективной работы платформы. На следующем этапе были выбраны средства реализации проекта и спроектирован интерфейс будущей платформы. Программная реализация представлена в виде листингов и описания функций, использующихся для реализации данного функционала. По итогам каждой главы сделаны выводы, выступающие промежуточными этапами с описанием и подведением локальных результатов.

По завершении разработки был проведен эксперимент, на основе которого можно подвести итоги, касающиеся принципов, представленных в работе:

1. Алгоритмы рекомендаций помогают быстрее закрепить практические навыки, и освоить базовые знания, но показатели формулирования при развернутом растут медленнее чем у группы с традиционным подходом в обучении.
2. Группа с корректировкой траектории обучения получает более высокие показатели в тестовой и практической оценке усвоенности учебного материала, но не дает таких же показателей как традиционное обучение в оценке развернутого ответа.
3. Разные методы обучения по-разному влияют на результаты разных типов проверки знаний. Традиционное обучение, за счет более тесного социального взаимодействия и общения дает лучшие показатели в оценке развернутых ответов. Обучение с корректировками и адаптациям к успеваемости учащихся дает более высокие показатели в оценке выполнения тестовых заданий и практических навыков.

В связи с вышеизложенным, видно, что результаты подтвердились частично, платформа дала определенные результаты, но показала недостаточные результаты из-за возможного недостаточного взаимодействия учащихся и ментора/преподавателя.

Результатом работы стала реализация платформы, обладающей необходимым функционалом для решения поставленных задач. Таким образом, предлагаемая платформа позволят развить навыки практических навыков в доступных дисциплинах дополнительного образования. Визуализация прогресса изучения позволит оценить свои навыки и посмотреть на рост в разрезе времени, понять свои слабые и сильные стороны. Использование алгоритмов машинного обучения позволит пользователям обучаться по оптимальным планам, исходя из личных показателей успеваемости.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

					Д 09.04.01 2023 23262867 МД	Лист
		Цапков А.В.		10.06.25		
Студ.	Фамилия	Подп.	Дата			86

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Rose, T. The End of Average: How We Succeed in a World That Values Sameness. [Place of publication not specified]: [Publisher not specified], [Year not specified]. (In English)
2. Zmuda, H. H. Jacobs, and A. Zmuda. Personalized Learning: A Guide for Engaging Students with Technology. [Place of publication not specified]: [Publisher not specified], [Year not specified]. (In English)
3. Khan, S. The One World Schoolhouse: Education Reimagined. [Place of publication not specified]: [Publisher not specified], [Year not specified]. (In English)
4. Tomlinson, C. A. Differentiated Instruction: Strategies for the Mixed-Ability Classroom. [Place of publication not specified]: [Publisher not specified], [Year not specified]. (In English)
5. Hattie, J. Visible Learning: A Synthesis of Over 800 Meta-Analyses Relating to Achievement. [Place of publication not specified]: [Publisher not specified], [Year not specified]. (In English)
6. Finkel, D. L. Teaching with Your Mouth Shut. [Place of publication not specified]: [Publisher not specified], [Year not specified]. (In English)
7. Freire, P. Pedagogy of the Oppressed. [Place of publication not specified]: [Publisher not specified], [Year not specified]. (In English)
8. Эльконин, Д. Б. Детская психология. Москва: Просвещение, 1978.
9. Нечаева, Т. А. Психологические особенности обучения школьников. Москва: Педагогика, 1985.
10. Bloom, B. S. "The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring." Educational Researcher, vol. 13, no. 6, 1984, pp. 4-16. (In English)
11. Tomlinson, C. A. How to Differentiate Instruction in Mixed-Ability Classrooms. Alexandria, VA: ASCD, 2001. (In English)