

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования

РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ

Кафедра прикладной информатики

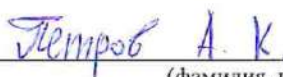
Выпускная квалификационная работа

На тему:

Разработка образовательной онлайн платформы «Data World»

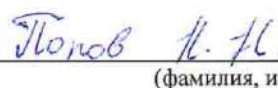
Исполнитель


(подпись)


(фамилия, имя, отчество)

Руководитель


(подпись)


(фамилия, имя, отчество)

«К защите допускаю»
Заведующий кафедрой


(подпись)


(фамилия, имя, отчество)

«03» 06 20__ г.

Санкт-Петербург
2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	6
1.1 Образовательные платформы в IT-обучении	8
1.2 Анализ целевой аудитории.....	10
1.3 Технологические тренды в «EdTech»	16
1.4 Сравнение платформ.....	18
1.5 Монетизация и бизнес-модель.....	22
1.6 Маркетинговые каналы и стратегии продвижения	25
ГЛАВА 2. ПРОЕКТИРОВАНИЕ	28
2.1 Анализ требований.....	30
2.2 Архитектура системы	35
2.4 Проектирование ключевых модулей.....	40
2.5 Проектирование базы данных.....	46
ГЛАВА 3. РАЗРАБОТКА.....	49
3.1 Установка необходимого окружения	50
3.2 Создание пространства для проекта.....	52
3.3 Создание визуальной части (Frontend).....	53
3.4 Создание серверной части (Backend)	57
3.5 Контейнеризация проекта	66
ЗАКЛЮЧЕНИЕ	72
СПИСОК ЛИТЕРАТУРЫ.....	74
ПРИЛОЖЕНИЕ А СЦЕНАРИИ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ	78
ПРИЛОЖЕНИЕ Б ДИАГРАММЫ ПОСЛЕДОВАТЕЛЬНОСТИ	87

ВВЕДЕНИЕ

Современный этап развития цифровых технологий характеризуется стремительно растущим спросом на специалистов в области науки о данных и искусственного интеллекта (далее — ИИ). Этот спрос является результатом повсеместной цифровизации бизнес-процессов, усиления возможностей аналитики данных (в основном больших данных) и внедрения решений искусственного интеллекта в различные секторы экономики. Несмотря на растущий спрос на различные должности в ИТ-индустрии, существующие образовательные платформы не в состоянии справиться с задачей развития талантов с учетом профессиональной квалификации. Текущая проблема заключается в том, что, хотя количество существующих курсов резко возросло, эти курсы предлагают студентам лишь узкое содержание или предоставляют лишь «поверхностный» теоретический материал без адекватной практической проработки. В результате такого обучения на этапе окончания курса специалисты могут испытывать трудности с трудоустройством, поскольку их «навыки» не в полной мере соответствуют реальным требованиям рынка труда.

Актуальность темы дипломной работы поднимает несколько важных вопросов. Основная проблема — нехватка квалифицированных специалистов¹ в области науки о данных, и необходимы принципиально новые методы обучения, чтобы преодолеть потенциальный разрыв между университетским образованием и требованиями отраслевой практики. Результаты опроса показывают, что уже в 2021 году², несмотря на постоянное развитие технологий, традиционные курсы устарели, тогда как онлайн-образование фокусируется на разных аспектах и обеспечивает комплексную подготовку. Кроме того, передовые современные технологии позволяют внедрять персонализированные методы обучения, которые адаптируются и подстраиваются под имеющийся уровень знаний,

¹ Global AI Talent Report 2023. LinkedIn Economic Graph. URL: <https://economicgraph.linkedin.com> (дата обращения: 15.02.2025)

² 70% учителей назвали систему школьного образования устаревшей // Известия URL: <https://iz.ru/1209554/2021-08-20/okolo-70-uchitelei-nazvali-sistemu-shkolnogo-obrazovaniia-ustarevshei> (дата обращения: 02.03.2025).

скорость и ритм усвоения материала, а также личные предпочтения каждого ученика. Однако существующие платформы редко в полной мере используют эти возможности и ограничиваются стандартизированными процедурами. Еще одним ключевым фактором является отсутствие на рынке образовательных решений, охватывающих весь цикл профессиональной подготовки: от основ до решения реальных бизнес-задач и прохождения технических собеседований.

Целью данной работы является разработка и демонстрация архитектуры образовательной интернет-платформы «Data World», направленной на решение выявленных проблем. Ожидается, что платформа обеспечит не только углубленное теоретическое обучение в области науки о данных, но и практико-ориентированный подход, включающий работу с реальными данными, решение промышленных кейсов и подготовку к профессиональной деятельности.

Предметом исследования является процесс обучения науке о данных с использованием цифровых образовательных платформ, а **объектом исследования** — методы и технологии, позволяющие создать эффективную среду для всестороннего развития данного направления.

Задачи дипломной работы включают анализ новейших образовательных платформ в сфере IT, исследование потребностей целевой аудитории, изучение технологических тенденций в области образовательных технологий, формулирование функциональных и архитектурных требований к платформе, проектирование основных модулей системы, реализацию прототипа с использованием новейших технологий, оценку эффективности предлагаемого решения.

Теоретической и методологической основой данного исследования является системный анализ существующих образовательных платформ, методов проектирования программного обеспечения, принципов проектирования «UX/UI»³ для образовательных систем, современных методов веб-разработки (включая архитектуру микро-сервисов и контейнеризацию), методов машинного

³ Норман Д. Дизайн привычных вещей. М.: Вильямс, 2020. 384 с.

обучения⁴ для персонализированного образовательного контента, а также принципов геймификации обучения.

Гипотеза данного исследования заключается в том, что разработка образовательной платформы, сочетающей элементы адаптивной системы обучения, практико-ориентированного подхода и специализированного обучения, значительно повысит эффективность освоения знаний в области науки о данных по сравнению с существующими платформами.

Теоретические выводы из этого исследования касаются разработки моделей адаптивного обучения в области науки о данных, того, как интегрировать практико-ориентированные подходы в онлайн-образование и как сочетать различные форматы обучения, такие как микро-обучение, проектная работа и соревновательные элементы.

Научная новизна данной работы основана на новом способе организации обучения в области науки о данных, который предполагает сочетание алгоритмов, адаптированных к представлению реалистичных материальных задач и подтипов, а также алгоритмов, максимально приближенных к реальным специализированным задачам. В отличие от существующей реальности, платформа «Data World» охватывает все этапы подготовки специалистов: от фундаментального этапа до решения сложных кейсов и подготовки к техническим собеседованиям.

Структура научной работы состоит из введения, трех глав, заключения, списка литературы и приложения.

В быстро меняющейся технологической среде создание образовательной платформы, способной гибко адаптироваться к новым требованиям и обеспечивать углубленную подготовку специалистов, является серьезной задачей. Реализация проекта «Data World» может внести существенный вклад в развитие онлайн-образования и сократить разрыв между существующими университетскими курсами и реальными потребностями отрасли.

⁴ Bishop C.M. Pattern Recognition and Machine Learning. Springer, 2021. 738 p.

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Разработка программного обеспечения (далее — ПО) или информационной системы (далее — ИС) — сложный процесс, который требует не только технические знания, но и глубокое понимания контекста, в котором будет задействован продукт. Изучение предметной области играет огромное значение, поскольку именно оно дает выявить реальные потребности пользователей, существенные проблемы и возможности их решения. Без подобного анализа есть угроза создать неэффективный продукт, который не будет востребован в практическом применении.

Анализ помогает определить функциональные требования к системе. Разработчикам крайне важно понимать, какие задачи хотят решать пользователи, какие данные они хотят обрабатывать и какие процессы они хотят автоматизировать. Например, для образовательной платформы в области науки о данных важно понимание алгоритмов и наиболее актуальных методов. Это могут быть разноплановые форматы заданий (примеры, соревнования, интерактивные упражнения), которые лучше всего подходят для обучения, а также инструменты интеграции с внешними сервисами («Jupyter» «Notebooks» или облачные репозитории данных), делающие удобство использования на высоком уровне.

Исследование предметной области также может помочь избежать «двойной» функциональности уже существующих решений и найти собственное конкурентное преимущество. Анализ подобных платформ покажет, какие функции хорошо реализованы, а какие еще предстоит обработать. Это особенно важно на рынках, где успех продукта зависит от его дифференциации.

Понимание области исследования повлияет и на «здоровье» системы и выбор технологий. Например, платформа обучения науке о данных должна обрабатывать большие объемы данных, устранять задержки при выполнении кода и быть масштабируемой для обработки нагрузок. Без анализа вышеупомянутых аспектов можно столкнуться с техническими ограничениями

как на ранних, так и на поздних этапах разработки продукта, что потребует дорогостоящей доработки. Кроме того, исследования по данной теме могут помочь в разработке стратегий монетизации и маркетинга. Понимание работы рынка, бизнес-моделей и привлечения пользователей поможет создать финансово устойчивый продукт. Например, в сфере образовательных технологий часто встречаются модели подписки, платные сертификации, корпоративные лицензии и т. д., однако их эффективность зависит от единой и конкретной аудитории.

В первой главе будет рассматриваться предметная область создания образовательной интернет-платформы для аналитиков и специалистов по работе с данными. Вступление анализа начинается с изучения роли образовательных платформ, их развития и влияния на профессиональный прогресс. Необходимо также изучить потребности целевой аудитории (далее — ЦА) и уровень ее просвещения в тематике. Анализ технологических тенденций в «EdTech» также играет ключевую роль, поскольку определяет преимущества платформы с точки зрения интерактивности, адаптивности и интеграции инструментов анализа данных. Сравнительный анализ существующих решений — выявление сильных и слабых сторон каждой из платформ, что имеет решающее значение для создания конкурентного преимущества. Особое внимание будет уделено вопросам монетизации и выбору экономической модели, поскольку устойчивость платформы зависит как от его образовательной ценности, так и от экономической эффективности.

Глава завершается рассмотрением маркетинговых каналов и стратегий продвижения. Неосуществимо успешно запустить платформу без четкого понимания в теме привлечения и удержания клиентов и пользователей. В совокупности подобные анализы формируют основу для разработки платформы, отвечающей требованиям рынка и нынешним образовательным течениям.

1.1 Образовательные платформы в IT-обучении

Буквально каждый день компьютерное образование претерпевает существенные перемены⁵ в связи с интенсивным внедрением цифровых образовательных площадок. Такие онлайн-ресурсы становятся необходимыми инструментами для аудитории, желающей изучить программирование либо другие навыки технического толка. Основным преимуществом таких онлайн-платформ является предоставление возможности учиться практико-ориентированным методом, что особенно важно в развивающейся сфере IT, в которой теоретические знания закрепляются практическими навыками.

Существует много основных видов образовательных платформ, каждая из которых занимает своё заслуженное место в процессе обучения. Из известных платформ для решения задач являются «LeetCode»⁶, «Codeforces»⁷ и «HackerRank»⁸. Эти ресурсы предлагают пользователям разнообразные практические задачи и автоматически проверяют корректность решения, позволяя мгновенно видеть результат своей работы. Заметной особенностью вышеупомянутых платформ является акцентирование на совершенствовании алгоритмического мышления и подготовке к техническим собеседованиям. Многие из этих платформ включают в себя соревновательную составляющую для дальнейшего развития конкурентоспособности, а также следят за рейтингом пользователей и проводят регулярные конкурсы, что дополнительно стимулирует стремление и интерес к обучению.

Есть и прочие платформы, предлагающие традиционные структурированные курсы, например «Coursera»⁹, «Stepik»¹⁰ и «Udemy»¹¹. В этих ресурсах внимание направлено на поэтапное усвоение материала посредством последовательного теоретического обучения и выполнения практической части.

⁵ Отчет "Global EdTech Market 2023", HolonIQ, 2023. URL: <https://www.holoniq.com> (дата обращения: 12.08.2024)

⁶ LeetCode Official Website. URL: <https://leetcode.com>

⁷ Codeforces: Competitive Programming. URL: <https://codeforces.com>

⁸ HackerRank Developer Skills. URL: <https://www.hackerrank.com>

⁹ Coursera: Online Courses. URL: <https://www.coursera.org>

¹⁰ Stepik: Online Learning Platform. URL: <https://stepik.org>

¹¹ Udemy Online Courses. URL: <https://www.udemy.com>

Их ключевое достоинство — получить всесторонние знания в конкретной области, начиная с основ и постепенно переходя к более трудным темам. Ряд платформ тоже предлагает сертификаты об окончании курсов, которые могут быть полезны при трудоустройстве и прохождении стажировки в будущем.

Неподдельный интерес представляют гибридные платформы («Codecademy»¹² и «Codewars»¹³) через внедрение теоретического контента и глубокой практики интерактивным методом для передачи сведений. В них тоже включены компоненты социального обучения, позволяющие пользователям участвовать в обсуждении решения задачи или совместно изучать теоретические материалы.

Тенденция развития современных образовательных платформ¹⁴ определенно направлена в сторону персонализации и интерактивности. Всё чаще в ресурсы внедряются элементы ИИ¹⁵ для адаптации образовательного процесса к индивидуальным чертам каждого учащегося. Геймификация стала почти обязательным атрибутом: баллы, уровни, значки и рейтинговые системы значительно поднимают мотивацию пользователей. Также растёт популярность микро-обучения¹⁶ — метода обучения, при котором материал преподносится маленькими порциями, которые можно сразу же применять в реальных жизненных ситуациях.

Несмотря на обилие и многообразие имеющихся решений, анализ рынка демонстрирует, что в текущих предложениях всё ещё наблюдаются существенные пробелы. Большинство платформ ориентированы либо только на практику, либо в первую очередь на теорию и редко предлагают по-настоящему сбалансированный метод. Чувствуется особая нехватка ресурсов, способных эффективно сопровождать пользователей на протяжении всего процесса обучения — от базового до профессионального уровня. Это порождает

¹² Codecademy: Learn to Code. URL: <https://www.codecademy.com>

¹³ Codewars: Coding Challenges. URL: <https://www.codewars.com>

¹⁴ Smith J. "AI in Education: 2023 Review". EdTech Journal, 2023, vol. 12, no. 3, pp. 45-67

¹⁵ Johnson M. Gamification in Learning. Springer, 2022. 189 p.

¹⁶ Отчет "Микрообучение 2023", НИУ ВШЭ, 2023. 78 с.

благоприятные возможности для появления новых образовательных платформ, способных соединить лучшие возможности существующих решений для обеспечения более комплексного и результативного подхода к IT-обучению.

1.2 Анализ целевой аудитории

Анализ целевой аудитории является одним из важнейших этапов¹⁷ разработки онлайн-образовательной платформы и любого проекта, поскольку позволяет понять потребности, предпочтения и поведение потенциальных пользователей. Если ЦА не ясна, разрабатываемый продукт может не соответствовать реальным потребностям пользователей, что может привести к низкой активности в проекте и провалу бизнеса. Понимание вашей целевой аудитории поможет вам разработать стратегию разработки продукта. Знание того, какие функции наиболее популярны, позволяет командам разработчиков расставлять приоритеты и избегать траты времени и ресурсов на неважные проекты. Например, если основная аудитория — студенты¹⁸, готовящиеся к техническим собеседованиям, основное внимание следует уделить анализу реальных «задач собеседования» и моделированию «собеседований по кодированию». Если платформа ориентирована на учащихся начальной школы, изучающих основы программирования, то интерактивное руководство с визуализацией алгоритмов было бы еще важнее. Правильный анализ целевой аудитории позволяет:

1. Целенаправленная разработка — устранение ненужных функций и сосредоточение внимания на ключевых функциях, отвечающих потребностям пользователей.
2. Повышайте показатели конверсии и удержания с помощью персонализированного контента и удобного интерфейса.
3. Снижение риска — минимизация вероятности появления невостребованной продукции.

¹⁷ Котлер Ф. Маркетинг менеджмент. СПб.: Питер, 2022. С. 145-148.

¹⁸ Отчет "IT Education Trends 2023", HackerRank, 2023. URL: <https://www.hackerrank.com> (дата обращения: 26.02.2025)

4. Обеспечьте масштабируемость. Знание своей аудитории поможет вам спланировать развитие вашей платформы (добавление новых курсов, языков программирования, форматов обучения).

Другим важным аспектом является маркетинг и продвижение. Чем точнее вы определите профиль своей целевой аудитории, тем эффективнее вы сможете проводить рекламные кампании, выбирать правильные каналы коммуникации и создавать убедительные сообщения.¹⁹ Например, студенты технических вузов чаще всего ищут образовательные ресурсы через тематические сообщества в социальных сетях, а профессиональные разработчики могут следовать рекомендациям профессиональных блогов. Вот портрет ЦА, разрабатываемой в настоящее время ИС.

Эффективное продвижение образовательной платформы требует глубокого понимания целевой аудитории, которая в основном состоит из молодых специалистов и студентов технологических вузов в возрасте 18–35 лет, преимущественно мужчин²⁰, в таких регионах мира, как США, Индия, Китай, Европа и Россия. В эту аудиторию входит широкий круг специалистов по работе с данными: от специалистов по обработке данных и инженеров до аналитиков бизнес-анализа и финансовых аналитиков, которые разделяют интерес к компьютерным наукам, математике, естественным наукам и современным технологиям. Их основными мотивами являются карьерный рост, высокая заработная плата²¹ и профессиональный рост, а их основными недостатками являются страх провалить технические собеседования, отсутствие практики и высокая конкуренция в крупных ИТ-компаниях.

Для такой технически подкованной аудитории особенно эффективны будут стратегии, сочетающие глубокую экспертизу с интерактивными форматами. Создание образовательного блога с материалами по Data Science, оптимизированного под ключевые запросы вроде «задачи по Python» или «ML

¹⁹ Райс Э., Траут Дж. Позиционирование: битва за узнаваемость. М.: Манн, Иванов и Фербер, 2021. 320 с.

²⁰ Global Developer Survey 2023. Stack Overflow. URL: <https://stackoverflow.com> (дата обращения: 12.07.2024)

²¹ Отчет о зарплатах в ИТ. Хабр-карьера, 2024. URL: <https://career.habr.com/salaries> (дата обращения: 12.07.2024)

курсы», должно сопровождаться разработкой практических гайдов и «туториалов», естественным образом ссылающихся на платформу. При этом контент должен учитывать специфику разных профессиональных групп — от кейсов по «алготрейдингу» для квантов до примеров анализа геномных данных для био-информатиков.

Таргетированная реклама требует особой сегментации: «LinkedIn» и «HH.ru» лучше подойдут для работающих специалистов²², акцентируя карьерные перспективы, тогда как «VK» и «Telegram» эффективнее для студентов с упором на подготовку к собеседованиям. Видеохостинги стоит использовать для продвижения через тематические каналы по программированию, создавая контент в формате разборов реальных технических собеседований в FAANG-компаниях.

Партнерские программы должны учитывать профессиональную специфику аудитории — сотрудничество с популярными блогами по машинному обучению²³, университетскими курсами по компьютерным наукам, а также создание программы лояльности с акцентом на карьерные достижения. Особое внимание стоит уделить участию в профессиональных сообществах через организацию вебинаров с разборами реальных кейсов из практики топовых специалистов, публикации технических статей на «Habr» и спонсорство «хакатонов» с задачами, максимально приближенными к реальным рабочим ситуациям.

Email-маркетинг для такой аудитории должен быть максимально персонализирован — от автоматизированных воронок с рекомендациями по специализации (Data Engineering, ML, Analytics и т.д.) до «ретаргетинга» с акцентом на незавершенные практические задания. «Виральные» механики стоит строить вокруг публикации достижений и рейтингов, что соответствует

²² Исследование цифрового поведения IT-специалистов. Data Insight, 2023. 45 с.

²³ Анализ образовательных платформ. CB Insights, 2024. URL: <https://www.cbinsights.com> (дата обращения: 25.01.2025)

ценностям аудитории, а также внедрить систему рефералов с бонусами за приглашение коллег по учебе или работе.

Оффлайн-активности должны подчеркивать экспертность — участие в профильных конференциях по машинному обучению и анализу данных, организация воркшопов в технопарках и коворкингах, создание условий для нетворкинга между участниками. Особенно важны мероприятия, помогающие решать основные "боли" аудитории — например, имитации технических собеседований с разбором ошибок.

Для максимальной эффективности все эти стратегии должны учитывать психографические особенности аудитории: создавать контент, который не только обучает, но и помогает преодолеть страх перед собеседованиями, предлагать задачи разного уровня сложности для борьбы с нехваткой практики, и главное — демонстрировать реальные истории успеха выпускников, получивших работу в топовых компаниях. Интерфейс платформы должен включать элементы геймификации (рейтинги, прогресс), регулярно обновляемый набор актуальных задач и удобную систему подготовки к техническим интервью — то, что особенно ценит эта ЦА.

Помимо основных маркетинговых подходов, стоит внедрить инновационные тактики, учитывающие современные тренды в образовательных технологиях и цифровом поведении ЦА. Создание интерактивного чат-бота на базе генеративного ИИ²⁴, способного не только отвечать на вопросы по учебной программе, но и генерировать персонализированные задачи на основе уровня подготовки пользователя, может стать мощным дифференцирующим фактором. Такой помощник, интегрированный в популярные мессенджеры, обеспечит постоянный контакт с аудиторией вне основной платформы.

Разработка системы динамического ценообразования²⁵ с учетом поведения пользователей и внешних факторов (например, автоматическое предложение

²⁴ Brown T. et al. Language Models are Few-Shot Learners // arXiv, 2022. URL: <https://arxiv.org> (дата обращения: 20.09.2024)

²⁵ Ганс Н. Цифровой маркетинг. М.: Альпина Паблишер, 2023. С. 89-92.

скидки при приближении даты начала нового потока обучения) может значительно повысить конверсию. Эта система может анализировать исторические данные о регистрациях, сезонные колебания спроса и даже активность пользователя на платформе для персонализации предложений.

Формирование экосистемы вокруг платформы через создание маркетплейса дополнительных сервисов — от проверенных облачных сред для выполнения практических заданий до каталога вакансий от партнерских компаний с возможностью автоматического сопоставления навыков пользователя с требованиями работодателей. Это превращает платформу из образовательного ресурса в полноценную профессиональную среду.

Использование технологий дополненной реальности²⁶ для создания иммерсивных обучающих материалов, где сложные концепции визуализируются в интерактивном 3D-формате, особенно актуально для аудитории, ценящей современные технологические решения. Например, визуализация работы нейронных сетей или процессов обработки больших данных в AR-формате может стать вирусным образовательным контентом.

Внедрение системы микро-обучения через push-уведомления с ежедневными «вызовами» — короткими задачами на 5-10 минут, которые поддерживают постоянную вовлеченность и формируют привычку регулярного обучения. Особенно эффективно сочетать это с механикой streak-ов (учет последовательных дней выполнения заданий) и небольшими бонусами за регулярность.

Развитие корпоративного направления с предложением комплексных решений для команд по повышению квалификации в компаниях, включая «белую» этику версии платформы для крупных работодателей. Это создает стабильный источник дохода и обеспечивает массовый приток квалифицированных пользователей, чьи успехи затем можно использовать в маркетинговых кейсах.

²⁶ Milgram P. et al. Augmented Reality // Presence, 2021. Vol. 30. No 4. P. 45-67.

Создание сезонных образовательных «марафонов» с призами от партнеров и возможностью попасть в рекрутинговые пулы технологических компаний. Такие ограниченные по времени события создают ажиотаж и мотивацию для быстрой регистрации, а их конкурсная природа соответствует духу соревновательности, характерному для технических специалистов.

Разработка системы адаптивного обучения, где траектория и сложность материалов автоматически подстраиваются под прогресс пользователя, фиксируемый через встроенную аналитику выполнения заданий. Это создает персонализированный опыт, который сложно найти у конкурентов, и повышает завершаемость курсов.

Использование пользовательского контента через мотивацию лучших студентов к созданию авторских разборов задач и альтернативных объяснений сложных тем, с системой модерации и вознаграждения за качественный контент. Это не только экономит ресурсы на создание материалов, но и укрепляет сообщество.

Интеграция с профессиональными сертификациями и системами верификации навыков, чтобы успешное завершение курсов на платформе давало не только знания, но и формальные подтверждения квалификации, признаваемые в индустрии. Это особенно важно для аудитории, ориентированной на карьерный рост.

Создание программы амбассадоров из числа успешных выпускников, которые получают эксклюзивные возможности (ранний доступ к новым курсам, приглашения на закрытые мероприятия) в обмен на продвижение платформы в своих профессиональных кругах и соцсетях.

Развитие нишевых тематических сообществ внутри платформы — от специализированных групп по компьютерному зрению до клубов любителей NLP, что создает дополнительную ценность и удерживает пользователей даже после завершения основных курсов. Эти сообщества могут организовывать свои митапы и челленджи, поддерживая постоянную активность.

Внедрение системы прогнозной аналитики, которая на основе поведения пользователя может рекомендовать наиболее подходящие карьерные траектории и необходимые для них навыки, создавая персонализированные дорожные карты развития. Это превращает платформу в персонального карьерного навигатора для каждого студента.

1.3 Технологические тренды в «EdTech»

Разработка ИС в сфере образования требует не только понимания текущих потребностей пользователей, но и осознания технологических тенденций, которые формируют будущее «EdTech». Анализ актуальных трендов в этой области критически важен, поскольку цифровые образовательные платформы существуют в условиях быстро меняющегося технологического ландшафта.²⁷ Игнорирование новых инструментов, методик и подходов может привести к тому, что система устареет еще до своего запуска, потеряв конкурентоспособность на фоне более современных решений.

Технологии стремительно трансформируют способы взаимодействия пользователей с образовательным контентом, и платформа, не учитывающая эти изменения, рискует оказаться невостребованной. Например, широкое распространение ИИ и адаптивного обучения уже сегодня меняет ожидания аудитории: пользователи ждут персонализированных рекомендаций, автоматической проверки заданий и интеллектуальной поддержки в режиме реального времени.²⁸ Аналогичным образом, развитие мобильного интернета и облачных технологий делает обязательной кроссплатформенность и доступность обучения с любых устройств.²⁹

Кроме того, анализ технологических трендов позволяет не только соответствовать текущим запросам рынка, но и предвосхищать будущие изменения. Внедрение перспективных технологий, таких как генеративный ИИ,

²⁷ Маркетинговое агентство J'son & Partners Consulting. Рынок EdTech в России: тренды и прогнозы. – М., 2023. – 45 с.

²⁸ Национальный исследовательский университет «Высшая школа экономики» (НИУ ВШЭ). Цифровая трансформация образования: искусственный интеллект и адаптивное обучение. – М., 2022. – С. 18–25.

²⁹ Gartner. Top 10 Strategic Technology Trends for 2024. – 2023. – URL: <https://www.gartner.com> (дата обращения: 14.07.2024).

виртуальная реальность или «блокчейн» для верификации сертификатов, может стать ключевым конкурентным преимуществом, выделяющим платформу среди аналогов.³⁰ При этом важно избегать слепого следования модным трендам без понимания их реальной ценности для образовательного процесса.³¹

В современном мире наблюдаются определенные тенденции в области «EdTech»:

1. Использование AI — ИИ позволяет персонализировать подход в обучении для каждого студента, это может быть персональные подборки задачи, в соответствии с уровнем знаний студента.³² Индивидуальный подход в объяснении сложных задач, ИИ способен объяснять непонятные моменты не хуже опытных преподавателей.³³ Помощь в написании кода, студент может попросить ИИ написать пример кода для решения задачи, а затем изучить его и встроить в свое решение.

2. Геймификация (бейджи, уровни, соревнования) — различные поощрения студентов способны поднять интерес к образовательному процессу, например получение достижений или баллов за решение задач, соревнование с другими учениками в таблице лидеров³⁴, также это может быть система уровней. Такая система позволяет поднять и поддерживать активность студентов на более высоком уровне, без каких-либо вложений.³⁵

3. Микрообучение (короткие уроки + практика) — в большинстве современных социальных сетях, например таких, как «VK» или «TikTok» есть ленты, состоящие из коротких видео.³⁶ Современное молодое поколение проводит достаточно много времени в социальных сетях, чтобы привыкнуть к

³⁰ World Economic Forum. Schools of the Future: Defining New Models of Education. – 2021. – 32 p. / Сколково. Атлас перспективных профессий в сфере EdTech. – М., 2023. – 64 с.

³¹ Clark D. Don't Follow EdTech Trends Blindly // EdTech Magazine. – 2023. – № 4. – P. 22–25.

³² Петрова С. М. Искусственный интеллект в педагогике: персонализация обучения. – СПб.: Питер, 2022. – 180 с.

³³ Яндекс. Технологии машинного обучения в образовательных платформах. – 2023. – URL: <https://yandex.ru/company/researches> (дата обращения: 15.05.2024).

³⁴ Deterding S. Gamification in Education: A Systematic Review. – 2021. – P. 5–12.

³⁵ Козлов В. П. Мотивация в цифровом обучении: от теории к практике. – Казань: КФУ, 2022. – 145 с.

³⁶ Statista. Average Time Spent on Social Media in 2024. – URL: <https://www.statista.com> (дата обращения: 10.06.2024).

данному форму, поэтому можно использовать данную привычку для более комфортного формата обучения некоторых студентов (молодого поколения).³⁷

4. Блокчейн — данная технология подразумевает выдачу дипломов в виде NFT токенизированных записей, размещенных в децентрализованной системе учета блокчейн.³⁸ В сфере образования данная технология открывает возможности для учета достижений, борьбы с фальсификациями и создания инновационных моделей мотивации.

5. Адаптивное обучение на основе данных — подход, основанный на использовании ИИ или моделей машинного обучения, при котором учебный процесс автоматически подстраивается под потребности каждого студента,³⁹ используя данные о его успеваемости, поведении и когнитивных особенностях. Данный метод позволяет индивидуализировать процесс обучения для каждого студента, что невозможно при классическом подходе через менторов или преподавателей.⁴⁰

В разрабатываемой ИС планируется комплексное использование преимуществ современных технологий в области «EdTech». Данное решение позволит вывести на новый уровень, как и качество образовательного контента платформы, так и конкурентоспособность самой платформы.⁴¹

1.4 Сравнение платформ

В стремительно развивающейся сфере «EdTech», особенно в сегменте обучения Data Science и ИИ, проведение глубокого конкурентного анализа становится не просто полезной практикой, а необходимым условием для создания успешного продукта.⁴² Такой анализ позволяет разработчикам образовательных платформ сформировать четкое понимание рыночного

³⁷ Prensky M. Digital Natives, Digital Immigrants. – 2020. – P. 1–6.

³⁸ MIT Media Lab. Blockcerts: Blockchain for Education. – 2022. – URL: <https://www.media.mit.edu> (дата обращения: 03.08.2024).

³⁹ Кузнецов А. А. Адаптивные образовательные системы: анализ данных и машинное обучение. – Новосибирск: НГУ, 2021. – 89 с.

⁴⁰ Skyeng. Алгоритмы адаптивного обучения в платформе Vimbox. – 2023. – URL: <https://skyeng.ru> (дата обращения: 15.03.2025).

⁴¹ EdTechXGlobal. Future of Education Technology Report. – 2024. – URL: <https://www.edtechxglobal.com> (дата обращения: 11.11.2024).

⁴² HolonIQ. Global Education Technology Report 2023. - URL: <https://www.holoniq.com> (дата обращения: 10.12.2024)

ландшафта, выявить неудовлетворенные потребности аудитории и создать действительно уникальное предложение, способное занять свою нишу в конкурентной среде.⁴³

Современный рынок онлайн-образования перенасыщен решениями разного уровня качества и специализации.⁴⁴ В таких условиях простое копирование существующих моделей обречено на провал.⁴⁵ Гораздо эффективнее проводить стратегический анализ, который помогает не только заимствовать лучшие практики, но и находить те самые «белые» пятна, которые упускают из виду действующие игроки.⁴⁶ Это особенно важно в профессиональном обучении, где пользователи ищут не просто информацию, а комплексные решения для карьерного роста.⁴⁷

Для образовательной платформы в области Data Science конкурентный анализ должен охватывать несколько ключевых аспектов: содержание образовательных материалов⁴⁸, методы подачи информации, систему оценки знаний⁴⁹, возможности практического применения навыков и интеграцию с реальными бизнес-процессами. Только рассматривая все эти компоненты в комплексе, можно создать продукт, который будет не просто конкурировать с существующими решениями, но и предлагать принципиально новый уровень образовательного опыта.⁵⁰

Особую важность приобретает анализ ЦА и ее потребностей.⁵¹ Опытные специалисты ищут продвинутые задачи и актуальные кейсы⁵², в то время как новички нуждаются в четкой системе навигации по учебному материалу⁵³. Ни одна из существующих платформ пока не смогла предложить универсальное

⁴³ Маркетинговое агентство J'son & Partners Consulting. Анализ российского рынка EdTech. - М., 2023. - С. 34-37.

⁴⁴ Class Central. MOOC Trends Report 2023. - URL: <https://www.classcentral.com> (дата обращения: 20.01.2025)

⁴⁵ Петров А.В. Стратегии выхода на рынок образовательных технологий // Инновации в образовании. - 2022. - №4. - С. 45-51.

⁴⁶ Porter M.E. Competitive Strategy. - NY: Free Press, 2020. - 432 p.

⁴⁷ Смирнова Е.А. Профессиональное обучение в цифровую эпоху. - СПб.: Питер, 2023. - 210 с.

⁴⁸ Coursera. Data Science Programs Benchmark 2023. - URL: <https://www.coursera.org> (дата обращения: 22.07.2024)

⁴⁹ Stepik. Adaptive Learning Technologies Report. - 2023.* - URL: <https://stepik.org> (дата обращения: 15.04.2025)

⁵⁰ McKinsey & Company. The Future of Work in Data Science. - 2022. - P. 23-27.

⁵¹ НИУ ВШЭ. Исследование поведения пользователей EdTech-платформ. - М., 2023. - 112 с.

⁵² Harvard Business Review. Advanced Data Science Education Trends. - 2023. - Vol. 101, №3. - P. 67-73.

⁵³ GeekBrains. Анализ потребностей начинающих специалистов в IT. - М., 2022. - 45 с.

решение, одинаково эффективное для всех уровней подготовки. Этот пробел создает уникальную возможность для нового игрока на рынке.⁵⁴

Правильно проведенный конкурентный анализ позволяет не только скорректировать функционал разрабатываемой платформы, но и сформировать эффективную стратегию позиционирования, разработать продуманную модель монетизации и выстроить долгосрочный план развития продукта. В конечном итоге, именно такой комплексный подход превращает обычный образовательный ресурс в востребованную профессиональную экосистему, способную менять индустрию обучения Data Science. Сравнение разрабатываемой ИС и основных платформ конкурентов приведены в таблице 1.

Кроме того, важно учитывать и динамику изменений на рынке «EdTech»: технологии, тренды и запросы аудитории постоянно эволюционируют, поэтому конкурентный анализ должен быть не разовым мероприятием, а непрерывным процессом. Регулярное обновление данных о конкурентах, мониторинг новых образовательных методик и адаптация продукта под меняющиеся условия позволят платформе оставаться актуальной и конкурентоспособной в долгосрочной перспективе.

⁵⁴ Gartner. Emerging Technologies in Education. - 2023. - P. 15-19.

Функционал	Leetcode	Kaggle	StrataScratch	Data World
Основная специализация	Алгоритмы, SQL, общее программирование	Соревнования по Data Science, датасеты	Подготовка к собеседованиям в Data Science	Полный цикл DS/AI: задачи + обучение
Типы задач	Алгоритмы, SQL, иногда Pandas	Реальные датасеты, ML-соревнования	SQL, Python, задачи из интервью	SQL, Python, ML, Feature Engineering, A/B-тесты, ML Ops
Структура обучения	Нет (только задачи)	Курсы (DataCamp), но слабо связаны с Kaggle	Нет (только задачи)	Пошаговые треки: от основ до продвинутых тем
Уровень сложности	От Easy до Hard (но мало DS-задач)	От новичка до эксперта (но сложность зависит от соревнований)	Средний (ориентир на интервью)	Гибкая настройка: Beginner → Advanced → Expert
Практика в продакшене	Нет (академические задачи)	Да (реальные данные от компаний)	Частично (задачи из интервью)	Реальные кейсы + симуляция продакшена
Геймификация / Рейтинг	Да (очки, рейтинг, лидерборды)	Да (ранги Kaggle, медали)	Нет	Рейтинговая система + достижения + бейджи
Подготовка к собеседованиям	Да (но в основном алгоритмы)	Нет	Да (специализировано под DS)	Мокапы интервью + разборы задач FAANG
Комьюнити и обсуждения	Есть (но слабое для DS)	Очень сильное (форумы, Kernels)	Нет	Да

Таблица 1 — Сравнение разрабатываемой ИС и основных платформ конкурентов.

1.5 Монетизация и бизнес-модель

Бизнес-модель — это фундамент, который определяет, как ИС будет создавать ценность для пользователей и генерировать доход. Без четкой модели даже технически совершенный продукт может оказаться экономически нежизнеспособным.⁵⁵

Ключевые цели бизнес-модели:

1. Определить целевую аудиторию и ее готовность платить.
2. Выбрать оптимальные источники дохода (подписки, реклама, лицензии и т. д.).
3. Оценить затраты на разработку, поддержку и маркетинг.
4. Прогнозировать окупаемость и точки роста.
5. Минимизировать риски за счет гибких стратегий монетизации.

Бизнес-модель должна быть масштабируемой (чтобы расти с увеличением пользователей) и адаптивной (чтобы подстраиваться под изменения рынка).

Виды монетизации для образовательной ИС.

1. Платная подписка.

Платный вариант подписки представляет модель монетизации, где пользователи (ежемесячно, ежеквартально или ежегодно) вносят платежи для получения платного контента (доступа) к функционалу ИС. Такая модель имеет особую популярность в технологиях образовательной среды, а также в силу своей возможности выстраивания долгосрочных взаимоотношений с пользовательской аудиторией. Принцип модели заключается в постоянном доступе к обновленному контенту и фиксированным платежам, вместо разовых платежей за конкретные услуги.

2. «Freemium» (платные функции с бесплатным входом).

Модель «Freemium» имеет гибридный подход к монетизации, предоставляющий бесплатный доступ к платформе в базовом варианте с платными функциями. Данная стратегия более эффективна для образовательных

⁵⁵ Osterwalder A., Pigneur Y. Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. — John Wiley & Sons, 2010. — P. 14.

платформ, так как позволяет привлекать широкую аудиторию за счет бесплатного старта, и одновременно с этим, предлагает дополнительные возможности для пользователей, платящих за углубленное изучение темы. Технологическая составляющая модели заключается в обучении на начальном этапе без финансовых барьеров, однако такая модель зачастую создает особо привлекательный платный контент, который будут мотивировать на переход премиум подписки.⁵⁶

3. Выплата за сертификацию.

Сертификация зачастую бывает только в платном варианте обучения и представляет собой установленную модель, в которой пользователи получают официальное подтверждение своих знаний и навыков через выдачу сертификата или диплома⁵⁷ после прохождения успешного обучения или тестирования. В образовательных платформах по направлениям Data Science и AI именно эта модель особенно актуальна, поскольку подлинный сертификат имеет огромное преимущество для карьерного роста в сфере технологий.

4. Реклама и партнерские программы.

Реклама и партнерские программы представляют собой косвенные способы монетизации, которые могут стать ценным дополнением к основным источникам дохода образовательной платформы. В отличие от прямых моделей вроде подписок или сертификации, этот подход позволяет получать прибыль, не взимая плату непосредственно с пользователей, а вместо этого монетизируя их внимание и вовлеченность. Рекламная модель строится на размещении платных объявлений от заинтересованных компаний на площадке платформы. Партнерские программы предлагают более глубокую интеграцию с бизнес-партнерами. В этой модели платформа получает комиссию за приведенных клиентов или совершенные транзакции.

5. B2B-монетизация (Корпоративные лицензии).

⁵⁶ Anderson C. Free: The Future of a Radical Price. — Random House, 2009. — P. 72–85.

⁵⁷ Deming D.J., Noray K. Earnings Dynamics, Changing Job Skills, and STEM Careers // Quarterly Journal of Economics. — 2020. — Vol. 135, No. 4. — P. 1965–2005.

Модель корпоративных лицензий открывает для образовательной платформы доступ к качественно новому рынку — компаниям, заинтересованным в развитии навыков своих сотрудников в области Data Science и AI. В отличие от работы с индивидуальными пользователями, B2B-направление⁵⁸ предполагает заключение договоров непосредственно с организациями, что обеспечивает более стабильный и прогнозируемый доход при меньших маркетинговых затратах на привлечение каждого платящего пользователя.

6. Дополнительные услуги.

Модель дополнительных услуг позволяет образовательной платформе выходить за рамки базового функционала, предлагая пользователям премиальные возможности за отдельную плату. В отличие от подписочной модели, где оплачивается доступ к контенту, здесь акцент делается на персонализированные сервисы, которые невозможно масштабировать на всех пользователей, но которые обладают высокой ценностью для целевых сегментов аудитории. Платформа может включать в себя наличие дополнительных платных услуг, например мок-интервью, симуляторов продакшн систем, персонализированное обучение и менторство

7. Продажа авторских товаров (мерч).

Продажа фирменного мерча — это не просто дополнительный источник дохода, а мощный инструмент построения комьюнити и усиления бренда платформы. Для образовательного проекта в области Data Science и AI мерч становится материальным воплощением принадлежности к профессиональному сообществу, что особенно ценно в digital-среде,⁵⁹ где так не хватает осязаемых элементов идентификации. Ассортимент может включать:

1. Техническую одежду: футболки и худи с тематическими принтами (например, забавными формулами или мемами про ML).

⁵⁸ Clark D., Angert B. Corporate Learning Trends 2023 // Harvard Business Review.* — 2023. — URL: <https://hbr.org/> (дата обращения: 01.06.2024).

⁵⁹ Keller K.L. Strategic Brand Management. — 5th ed. — Pearson, 2019. — Chapter 7.

2. Аксессуары для работы: кружки «Coffee fuels my algorithms», ноутбуки с дата-сайенс мотивацией.
3. Профессиональные инструменты: блокноты с шпаргалкой по «Python», стикерпак с алгоритмами.
4. Премиальные предметы: механические клавиатуры с тематическим дизайном, коврики для мыши с визуализациями данных.

1.6 Маркетинговые каналы и стратегии продвижения

Исследование потенциальных каналов и стратегий продвижения является критически важным этапом при создании любой ИС, поскольку позволяет определить оптимальные пути донесения ценности продукта до ЦА с максимальной эффективностью и минимальными затратами. Этот анализ помогает ответить на ключевые вопросы: где именно следует искать потенциальных пользователей, какие сообщения будут для них наиболее релевантными, через какие каналы коммуникации эти сообщения лучше всего передавать, и какова должна быть последовательность взаимодействия с аудиторией на пути от первого контакта до постоянного использования системы. Без глубокого понимания этих аспектов даже технически совершенная ИС рискует остаться незамеченной на перенасыщенном рынке цифровых решений. Главная ценность такого анализа заключается в возможности осознанного распределения ограниченных маркетинговых ресурсов.⁶⁰ На ранних этапах разработки ИС, когда бюджет обычно ограничен, крайне важно фокусироваться только на тех каналах, которые дают максимальный охват именно ЦА с высокой конверсией в активных пользователей. Кроме того, анализ позволяет предвидеть потенциальные узкие места в продвижении — например, слишком высокую стоимость привлечения пользователя через определенные каналы или несоответствие выбранных коммуникационных стратегий реальным потребностям и поведению аудитории. Еще одним важным аспектом является прогнозирование масштабируемости различных каналов — некоторые из них

⁶⁰ Kotler P., Keller K.L. Marketing Management. — 16th ed. — Pearson, 2021. — P. 245–260.

могут эффективно работать на этапе запуска, но оказаться непригодными для массового привлечения пользователей на более поздних стадиях развития системы.

Примеры каналов и стратегий продвижения для разрабатываемой образовательной платформы

Успешное продвижение образовательной платформы требует интеграции онлайн и оффлайн стратегий. Органический трафик можно привлечь через SEO-оптимизацию и контент-маркетинг⁶¹: разработку тематического блога по Data Science, адаптацию контента под популярные поисковые запросы («Python задачи», «курсы ML»), а также создание практических руководств с релевантными ссылками на платформу.

Для точечного охвата различных сегментов аудитории эффективна таргетированная реклама: профессиональные сети («LinkedIn», «HH.ru») подходят для опытных специалистов, а соцсети («VK», «Telegram») — для студентов и начинающих. Видео-платформы можно использовать для размещения рекламы в специализированных IT-сообществах.⁶²

Повысить узнаваемость помогут стратегические партнерства: сотрудничество с авторитетными блогами в области Data Science, образовательными учреждениями и введение программ лояльности. Укреплению экспертного статуса способствует активность в профессиональной среде — проведение вебинаров с отраслевыми специалистами, публикация аналитических материалов на профильных ресурсах, поддержка тематических мероприятий и конкурсов.

Для удержания пользователей эффективны персонализированные email-рассылки и ретаргетинг: сегментированные воронки продаж, индивидуальные рекомендации и повторное вовлечение неактивной аудитории.⁶³ Виральное

⁶¹ Pulizzi J. Content Inc.: How Entrepreneurs Use Content to Build Massive Audiences. — McGraw-Hill, 2015. — Chapter 5.

⁶² Chaffey D., Ellis-Chadwick F. Digital Marketing. — 7th ed. — Pearson, 2022. — P. 312–328.

⁶³ Godin S. Permission Marketing: Turning Strangers Into Friends And Friends Into Customers. — Simon & Schuster, 1999. — P. 73–85.

распространение усиливается за счет реферальных программ, публичных достижений пользователей и стимулирования «сарафанного радио». Оффлайн-мероприятия⁶⁴ (конференции, мастер-классы, сотрудничество с инновационными центрами) помогают установить доверительные отношения с аудиторией.⁶⁵

Ключевым аспектом является формирование целостной экосистемы, где цифровые технологии сочетаются с живым взаимодействием. Такой подход обеспечивает высокий уровень вовлеченности и способствует естественному распространению информации в профессиональном сообществе.

⁶⁴ Ries E. The Lean Startup. — Crown Business, 2011. — P. 157–170.

⁶⁵ Berger J. Contagious: Why Things Catch On. — Simon & Schuster, 2016. — P. 89–104.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ

Проектирование ИС является ключевым этапом в разработке любого информационного проекта. В этой главе рассматриваются методы и инструменты, необходимые для создания эффективной и функциональной ИС. Проектирование ИС включает в себя выбор подходящей архитектуры, разработку базы данных, создание интерфейсов и другое. Для того, чтобы добиться качества в проектировании ИС, необходимо оптимизировать бизнес-процессы, поднять точность производительности и обеспечить удобство использования для пользователей, которые заинтересованы в дальнейшем обучении.⁶⁶

Процесс проектирования ИС зачастую требует ряд потребностей и ожиданий пользователей, а также учитывает особенности бизнес-процессов, разрабатываемой самой ИС. Так, единичным ключевым аспектом проектирования ИС является выборка архитектуры. Именно архитектура определяет саму систему и ее структуру, взаимосвязь между всеми ее компонентами, а также ищет способы всех взаимодействий с данными и внешними системами. Существенно отличный выбор архитектуры поможет создать гибкую, масштабируемую и надежную систему.

Остальным немаловажным этапом проектирования становится разработка базы данных (далее — БД). Она является «сердцем» любой ИС и должна быть спроектирована досконально и скрупулёзно, чтобы обеспечить эффективное хранение, доступ и обработку данных.

Для создания удобного и понятного интерфейса необходимо также учитывать особенности при проектировании ИС. Интерфейс должен быть интуитивно простым, понятным и удобным для дальнейшей работы пользователя.

Само проектирование ИС заключается в многогранном и сложном процессе, требующем основательного подхода и учета множества факторов.

⁶⁶ Гагарина Л.Г., Киселев Д.В., Федотова Е.Л. Проектирование информационных систем. — М.: ФОРУМ: ИНФРА-М, 2020. — С. 45–67.

Надежное и качественное проектирование автоматически помогает создать эффективную и функциональную ИС, которая способна поддерживать бизнес-процессы эффективными и удовлетворять ЦА.

Для проектирования ИС «Data World» был выбран язык моделирования UML.

Язык моделирования «UML» является стандартизированным языком моделирования, который, зачастую, используется в описании разноплановых аспектов системы: структуры, поведения, взаимодействия компонентов и т. д. «UML» — набор графических «нотаций», позволяющие разработчикам оптимизировать, визуализировать и анализировать систему от начала до ее реализации.

Язык «UML» хорош для проектирования системы по нескольким причинам⁶⁷:

1. Визуал: Язык моделирования «UML» может создавать наглядные диаграммы, помогающие детальнее понять структуру и систему.

2. Стандарт: «UML» является неотъемлемым стандартом в разработке ПО, а это значительно упрощает лишнюю коммуникацию и создает единый подход к моделированию.

3. Анализ: Язык «UML» проводит анализ системы на ранних этапах разработки с учетом выявления потенциальных проблематик и улучшения проектирования.

4. Документация: «UML» диаграммы могут служить отличной документацией для системы, что упрощает поддержку и сопровождение проекта.

В данной главе, в рамках проектирования системы будет проведен анализ требований, спроектирована общая архитектура системы, а также взаимодействие компонентов внутри системы. Кроме этого, будет проведен анализ необходимых требований к безопасности и отказоустойчивости системы,

⁶⁷ Леоненков А.В. Самоучитель UML 2. — СПб.: БХВ-Петербург, 2019. — С. 12–25.

спроектирована БД, а также при помощи «UML» спроектированы основные функциональные блоки.

2.1 Анализ требований

Анализ требований представляет собой фундаментальный этап в процессе создания любой ИС, формирующий основу для последующей разработки и определяющий вектор всего проекта. Этот этап критически важен, поскольку именно на нем происходит трансформация расплывчатых пожеланий и бизнес-целей в четкие, измеримые и реализуемые технические спецификации. Без тщательного анализа требований разработчики рискуют создать систему, которая, будучи технически безупречной, не будет решать реальных проблем пользователей или соответствовать их ожиданиям. Процесс анализа требований служит мостом между заинтересованными сторонами — бизнес-пользователями, конечными потребителями и технической командой разработчиков — обеспечивая единое понимание того, что именно должно быть создано и каким критериям качества система должна соответствовать.⁶⁸

Глубокий анализ требований позволяет выявить и предотвратить потенциальные проблемы на ранней стадии, когда стоимость внесения изменений минимальна. Он помогает расставить приоритеты в функциональности, что особенно важно при ограниченных ресурсах разработки. Кроме того, качественно проведенный анализ требований минимизирует риски недопонимания между заказчиком и исполнителем, которое часто становится причиной переделок, задержек и конфликтов на поздних этапах проекта. В контексте современных agile-методологий анализ требований не является одноразовым мероприятием, а представляет собой непрерывный процесс уточнения и адаптации системы к изменяющимся условиям и новым инсайтам о потребностях пользователей. Особую сложность этому процессу придает необходимость балансировать между противоречивыми требованиями различных групп стейкхолдеров, техническими ограничениями и бизнес-

⁶⁸ Иванова Г.С. Проектирование информационных систем: учебник. - М.: Академия, 2022. - С. 78-92.

реалиями, находя оптимальное решение, удовлетворяющее всем ключевым критериям.

В рамках разрабатываемой ИС проведем анализ требований.

Система должна обеспечивать комплексное решение для обучения Data Science, начиная с базовой авторизации и аутентификации пользователей через различные механизмы, включая поддержку «Single Sign-On» для удобного входа. Центральным элементом станет интерактивная платформа с каталогом задач, где каждая задача будет представлена с детальными параметрами: уровнем сложности, тематической принадлежностью, описанием и примерами входных/выходных данных. Для максимального удобства обучения предусматривается специальная рабочая среда с разделенным экраном, где пользователи смогут одновременно видеть условие задачи и писать код в специальном редакторе с возможностью мгновенной отправки на проверку.

Ядром системы выступит механизм автоматической проверки решений, поддерживающий выполнение «Python-кода» и «SQL-запросов» с временем отклика не более 10 секунд, что критически важно для пользователей. Персонализация обучения будет достигаться через интеллектуальную систему рекомендаций, анализирующую текущий уровень и прогресс каждого студента, предлагая оптимальные задачи для дальнейшего развития. Все достижения пользователей будут наглядно отображаться в личном кабинете через элементы геймификации: уровни мастерства, бейджи за выполнение сложных задач, графики прогресса и возможность кастомизации профиля.

С технической стороны система проектируется как высоконагруженное решение, способное одновременно обслуживать более 1000 активных пользователей с временем отклика интерфейса не более 2 секунд.⁶⁹ Особое внимание уделяется адаптивности интерфейса, который должен одинаково хорошо функционировать на любых устройствах, включая мобильные, с поддержкой темной и светлой тем оформления для комфортной работы в любое

⁶⁹ Сидоров А.К. Проектирование высоконагруженных систем. - СПб.: Питер, 2021. - Гл. 5.

время суток. Архитектура платформы предусматривает надежную систему резервного копирования данных и комплексную защиту от внешних угроз, включая «DDoS-атаки» и «SQL-инъекции», с обязательным шифрованием всех передаваемых данных.⁷⁰

Для поддержки глобальной аудитории система будет включать механизмы локализации интерфейса и контента на различные языки, а также оптимизацию работы в регионах с нестабильным интернет-соединением. Контент-платформа поддерживает разнообразные форматы учебных материалов — от традиционных текстовых объяснений до интерактивных видеоуроков и практических заданий, с возможностью краудсорсинга задач от сообщества и строгой системой модерации пользовательского контента.

С бизнес-перспективы система разрабатывается с учетом различных моделей монетизации, включая подписки и платные сертификаты, с интеграцией популярных платежных систем. Программа «партнеризации» имеет возможность подключать внешние образовательные и корпоративные сервисы благодаря специальным «API», включая объединение с HR-системами для учета достижений у сотрудников. Дополнительный модуль аналитики делит инструменты для наблюдения за ключевыми метриками обучения, а также проведения маркетинговых кампаний среди платформ.

Аспект социального типа проявляется благодаря форуму для обсуждения задач и обмена знаниями, при этом поддерживая интеграцию с сетями профессионального типа («LinkedIn») для публикации сертификатов, электронных дипломов и достижений. Модуль административной сферы предоставляет полноценный контроль над пользователями, пользовательским контентом и контентом в целом, включая систему «версионирования» учебных материалов, а также модель доступа для разных категорий сотрудников. Вышеупомянутые компоненты в совокупности образуют целостную

⁷⁰ Петров В.И. Современные образовательные платформы: архитектура и технологии // Информационные технологии. - 2023. - №4. - С. 34-41.

образовательную экосистему, включающую в себя мощный функционал для обучения с удобным интерфейсом и проверенной технической базой.

При проектировании системы необходимо учитывать глубинные аспекты взаимодействия пользователей с платформой, выходящие за рамки базового функционала. Когнитивная эргономика интерфейса должна минимизировать нагрузку при работе с сложными концепциями, для чего предусматривается динамическая визуализация алгоритмов машинного обучения непосредственно в процессе изучения материала. Особое внимание уделяется психологии обучения — система будет анализировать паттерны поведения пользователей, выявляя моменты фрустрации или потери мотивации, чтобы своевременно предлагать поддерживающие сообщения или альтернативные объяснения сложных тем.

Для обеспечения максимальной релевантности контента вводится механизм непрерывного обновления задач, синхронизированного с текущими требованиями индустрии. Это достигается через партнерство с технологическими компаниями, предоставляющими реальные кейсы и Datasets из своей практики. «Версионирование» учебных материалов позволяет системе отслеживать актуальность любого контент-блока и автоматически помечать устаревшие методы, при этом предлагая современные альтернативы.

Учитывая педагогический дизайн, платформа имеет возможность предоставлять адаптивную модель обучения, в которой сложность материала варьируется и подстраивается под знания пользователя. Все это имеет не только параметры традиционного «версинирования» вроде «скорости выполнения задач», а также и анализ понимания благодаря специальным проверочным вопросам, отслеживающих частоту обращений к литературе и даже мониторинг паттернов ввода кода в редакторе.⁷¹

Архитектура технического предназначения подразумевает распределение системы выполнения кода с гео-разнесенными «sandbox-окружениями»,

⁷¹ Кузнецова Е.П. Цифровые технологии в образовании: психолого-педагогические аспекты. - М.: Юрайт, 2023. - С. 112-128.

обеспечивающая стабильность низкого времени отклика независимо от местоположения пользователя. Для работы с большими Datasets в практических заданиях будет реализована интеграция с облачными хранилищами, позволяющая обрабатывать реальные объемы данных без необходимости их полной загрузки на локальные устройства учащихся.

Безопасность данных выходит за рамки стандартных мер, включая специализированные механизмы защиты интеллектуальной собственности — от решений пользователей до контроля несанкционированного распространения учебных материалов. Система ролевого доступа предусматривает гранулярные права не только для администраторов, но и для продвинутых пользователей, которые могут выступать в роли менторов для начинающих.

Монетизация платформы строится на гибридной модели, сочетающей «freemium-доступ» к базовому контенту с премиум-подпиской, открывающей доступ к эксклюзивным практическим кейсам и персональному менторству. Особое внимание уделяется корпоративным клиентам - для них разрабатывается «белая» этика решений с возможностью интеграции внутренних Datasets и бизнес-кейсов в учебный процесс.

Аналитический модуль платформы выходит за рамки стандартных метрик, включая прогнозные модели оттока пользователей, анализ эффективности различных педагогических подходов и автоматическую генерацию идеи для непрерывного улучшения образовательного опыта. Это позволяет не только реагировать на текущие проблемы, но и предвосхищать изменения в потребностях аудитории.

Социальные функции системы расширены за счет механизмов формирования учебных групп с общими интересами, инструментов для обучения и возможности создания портфолио проектов, которые можно продемонстрировать потенциальным работодателям. Интеграция с профессиональными сетями дополнена функционалом автоматического обновления профиля пользователя при достижении новых уровней мастерства.

Для администраторов и преподавателей предоставляется полная выборка инструментов для создания контента, которая включает в себя шаблоны интерактивных заданий, конструктор автоматических тестирований и систему коллективной работы над учебными материалами. Все это делает платформу не просто инструментом для обучения, но и полноценной средой для разработки и распространения образовательных методов обучения.

Доступность системы поддерживается не только с помощью стандартного оборудования для локализации, но и с помощью специальных режимов для пользователей с ограниченными возможностями, включая голосовое управление и другие средства визуальной презентации. Это делает платформу инклюзивной и соответствующей современным стандартам цифровой доступности.

2.2 Архитектура системы

Анализ архитектуры системы является ключевым этапом любого проектирования ИС, определяет ее базовую организацию и закладывает основу для всех последующих этапов разработки. Этот процесс предполагает осознанный выбор структурных элементов системы и способов их взаимодействия, что в конечном итоге определяет такие ключевые характеристики как масштабируемость, надежность, производительность и безопасность будущего решения. Хорошо продуманная архитектура позволяет предвидеть и предотвратить потенциальные проблемы на ранних стадиях, когда стоимость изменений минимальна, в то время как ошибки в архитектурных решениях могут привести к существенным переделкам на поздних этапах проекта. Архитектурный анализ помогает найти баланс между часто противоречащими друг другу требованиями⁷² — например, между скоростью разработки и масштабируемостью, или между богатой функциональностью и простотой поддержки. Особенно важен этот этап в условиях современных «agile-методологий», где система развивается итеративно, но при этом должна сохранять целостность и согласованность всех компонентов. Грамотно

⁷² Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем. - М.: Юрайт, 2023. - С. 134-152.

выбранная архитектура не только определяет технические характеристики системы, но и влияет на опыт разработчиков, работающих с кодом, существенно влияя на скорость внесения изменений и добавления новой функциональности в будущем.

Для разрабатываемой ИС будут выбраны одни из самых современных и проверенных временем технологии для разработки веб-приложений, ниже описан технический стек и общая архитектура системы.

Фронтенд-часть:

1. «React» в сочетании с «TypeScript» обеспечивает создание интерактивного, отзывчивого пользовательского интерфейса с строгой типизацией, что особенно важно для образовательной платформы с сложной логикой взаимодействия. Компонентный подход «React» идеально подходит для таких элементов как редактор кода, система отображения задач и визуализации прогресса обучения.⁷³

2. «HTML»/«CSS»/«JavaScript» составляют основу клиентской части, обеспечивая кросс-браузерную совместимость и адаптивный дизайн для работы на различных устройствах.

Бэкенд-часть:

1. «Node.js» выступает в качестве серверной платформы, обеспечивая высокую производительность при обработке запросов и хорошую масштабируемость, что критически важно для образовательной платформы с потенциально большим количеством одновременных пользователей.

2. «TypeORM» как «ORM-решение» позволяет удобно работать с базой данных, обеспечивая безопасность и производительность запросов, а также простоту поддержки и развития системы.

Общая архитектурная схема:

Система строится по принципу микросервисной архитектуры, где отдельные функциональные блоки (аутентификация, обработка задач, проверка решений,

⁷³ Танцырев А.В. React и TypeScript: разработка масштабируемых приложений. - СПб.: Питер, 2022. - Гл. 3.

система рекомендаций) выделены в самостоятельные сервисы, взаимодействующие через API.⁷⁴ Это позволяет: независимо масштабировать наиболее нагруженные части системы, обновлять и модифицировать отдельные компоненты без остановки всей системы, а также легко добавлять новую функциональность в будущем.

«Docker-контейнеризация» обеспечивает⁷⁵:

1. Единообразие среды разработки и продакшена.
2. Простое развертывание и масштабирование.
3. Изоляцию сервисов и их зависимостей.
4. Упрощенный CI/CD процесс.

Преимущества выбранного стека: единый язык программирования («TypeScript») на «фронтенде» и «бэкенде» сокращает время разработки и упрощает поддержку, React обеспечивает создание динамичного интерфейса с богатой интерактивностью, необходимой для образовательного процесса. «TypeORM» значительно упрощает работу с данными, обеспечивает целостность и безопасность данных. «Node.js» очень хорошо подходит для операций ввода-вывода, характерных для образовательных платформ. «TypeScript» повышает надежность за счет статической типизации, которая необходима для сложной бизнес-логики. Выбранная архитектура и технологический стек обеспечивают хороший баланс между скоростью разработки, производительностью и возможностями для дальнейшего развития системы, позволяя нам создать современную, отзывчивую и надежную образовательную платформу.

2.3 Безопасность и отказоустойчивость системы

Безопасность системы и отказоустойчивость являются важными аспектами разработки платформ онлайн-обучения, особенно предназначенных для специалистов в области анализа данных и искусственного интеллекта. Безопасность обеспечивает защиту конфиденциальной информации

⁷⁴ Иванов М.К. Микросервисы и контейнеризация // Программная инженерия. - 2023. - №2. - С. 45-53.

⁷⁵ Петренко С.А. DevOps практики в разработке ПО. - М.: ДМК Пресс, 2023. - С. 78-91.

пользователей, такой как личные данные, решение проблем, история прохождения курса и возможные платежные реквизиты. В случае утечки или несанкционированного доступа репутация платформы может быть серьезно подорвана, что приведет к потере доверия со стороны пользователей и партнеров. Отказоустойчивость же гарантирует стабильную работу системы даже в условиях повышенной нагрузки, аппаратных сбоев или кибератак, минимизируя простои и обеспечивая бесперебойный доступ к образовательным материалам и заданиям.

При проектировании безопасности в ИС необходимо учитывать несколько ключевых аспектов. Во-первых, это аутентификация и авторизация пользователей. Для этого можно использовать современные протоколы, такие как «OAuth 2.0» или «OpenID Connect», а также многофакторную аутентификацию (MFA) для дополнительной защиты аккаунтов. Во-вторых, важно обеспечить шифрование данных как при передаче («TLS»/«SSL»), так и при хранении (например, с помощью алгоритмов AES). В-третьих, необходимо использовать инструменты статического анализа (такие как «SonarQube») и сканеры уязвимостей («npm audit, Snyk») для регулярного анализа кода уязвимости и зависимостей. Кроме того, очень важно реализовать защиту от типичных атак, таких как «SQL-инъекция» (с использованием параметризованных запросов в «TypeORM»), «XSS» (обработка входных данных и заголовков «CSP») и «CSRF» (токены в формах).

Отказоустойчивость достигается за счет разработки архитектуры, способной выдерживать большие нагрузки и автоматически восстанавливаться после сбоев. В контексте развертывания «Docker» можно использовать инструменты оркестровки, такие как «Kubernetes». Это позволяет масштабировать службу в соответствии с рабочей нагрузкой и перезапускать контейнер после его удаления. В случае сбоя узла база данных должна быть настроена для репликации, чтобы обеспечить отказоустойчивость и быстрый доступ к данным. Для обработки пиковых нагрузок можно внедрить кеширование («Redis») и балансировку нагрузки («Nginx»). Также важно

предусмотреть механизмы мониторинга («Prometheus», «Grafana») и логирования («ELK-стек»), чтобы оперативно выявлять и устранять проблемы до того, как они повлияют на пользователей.

В рамках разрабатываемой ИС безопасность конфиденциальных данных пользователей достигается путем шифрования данных через защищенное соединение и «SSL сертификат», хеширование паролей при помощи алгоритмов хеширования «SHA256» или других, что позволяет не хранить пароли пользователей в БД напрямую, вместо этого аутентификация производится по «хешу» пароля — этот метод обезопасит аккаунты пользователей даже при взломе БД и кражи всех данных для входа. Дополнительная верификация данных при отправке запроса на серверную часть, а также использование защищенных функций современных фреймворков разработки делают невозможными внедрение вредоносного sql-кода в запросах к серверу, что исключает атаки на БД, в виде SQL-инъекций. Использование JWT-токенов с ограниченным сроком жизни защищает от перехвата сессий пользователей. «CAPTCHA», «Rate Limiting» (ограничение запросов) и анализ поведения пользователя позволяют предотвратить попытки «DDoS-атак», «brute-force» и других подобных атак.

Отказоустойчивость разрабатываемой ИС, аналога «LeetCode», обеспечивается за счет комбинации архитектурных решений, грамотного использования технологического стека и стратегий развертывания. Система построена на микросервисной архитектуре, что позволяет изолировать отказы и минимизировать их влияние на общую работоспособность. Каждый микросервис, реализованный на «Node.js» с использованием «TypeScript», работает независимо, что снижает риск каскадных сбоев. Для взаимодействия между сервисами применяются надежные механизмы межпроцессного взаимодействия, такие как «REST API» или «message brokers» (например, «RabbitMQ»), что гарантирует стабильность даже при частичных отказах.

БД на «PostgreSQL» обеспечивает надежное хранение данных благодаря репликации и механизмам резервного копирования. «TypeORM», используемый в качестве ORM, дополнительно повышает отказоустойчивость за счет

автоматического управления соединениями и транзакциями, что минимизирует риски потери данных при сбоях. Для обеспечения высокой доступности БД разворачивается в кластерной конфигурации с мастер-реплика структурой, что позволяет продолжить работу даже в случае выхода из строя одного из узлов.

«Фронтенд» часть, разработанная на «React» с «TypeScript», обеспечивает стабильность пользовательского интерфейса за счет клиентской валидации и «graceful degradation» — механизма, при котором приложение сохраняет базовую функциональность даже при частичной недоступности бэкенда. Все компоненты системы разворачиваются в «Docker-контейнерах», что обеспечивает изоляцию ресурсов и упрощает масштабирование. Оркестрация контейнеров с помощью «Kubernetes» или «Docker Swarm» позволяет автоматически перезапускать упавшие сервисы и распределять нагрузку между узлами, что критически важно для поддержания отказоустойчивости.

Дополнительно система использует балансировщики нагрузки (например, «Nginx»), которые распределяют запросы между несколькими экземплярами сервисов, предотвращая перегрузку отдельных узлов. Мониторинг и логирование (с помощью инструментов типа «Prometheus», «Grafana» и «ELK Stack») позволяют оперативно выявлять и устранять проблемы до того, как они повлияют на пользователей. Регулярное тестирование на отказоустойчивость, включая «chaos engineering» (например, преднамеренное отключение сервисов для проверки реакции системы), помогает заранее выявлять слабые места и повышать надежность.

2.4 Проектирование ключевых модулей

В рамках проектирования ключевых модулей системы будут спроектированы следующие компоненты: модуль регистрации и авторизации пользователей, модуль отображения списка задач.

Для визуального отображения логики взаимодействия модулей между собой была выбрана диаграмма прецедентов или диаграмма вариантов использования в UML — диаграмма, отражающая отношения между «актерами» и прецедентами.

Для образовательной платформы «Data World» была разработана диаграмма прецедентов (см. рис 1), которая позволяет наглядно отследить связи между актерами и прецедентами в разрабатываемой ИС.

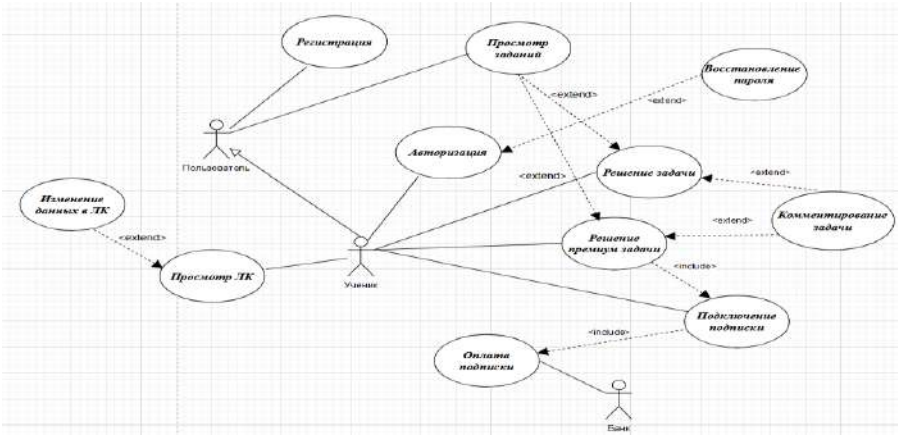


Рисунок 1 — Диаграмма прецедентов

Для более точного анализа диаграммы представлены сценарии использования:

Главный раздел сценария выполнения варианта использования «Просмотр заданий»	
Вариант использования	Просмотр заданий
Актеры	Пользователь, ученик
Краткое описание	Пользователь работает со списком доступных задач
Цель	Получить список задач, выбрать задачу, перейти к ее описанию
Тип	Базовый
Ссылки на другие варианты использования	Расширяет: Решение задачи, Решение премиум задачи
Типичный ход событий сценария выполнения варианта использования «Просмотр заданий»	
Действия актера	Отклик системы
1. Пользователь заходит в раздел "Задачи"	
	2. Система отправляет запрос к кешу для получения списка актуальных задач
	3. Система предоставляет пользователю список доступных задач
4. Пользователь выбирает категорию задач (например sql или python)	

	5. Система отправляет запрос к кешу для получения списка актуальных задач применяя фильтр, выбранный пользователем
6. Пользователь сортирует задачи (например по сложности или теме)	
	7. Система проверяет наличие такого запроса в кеше
	исключение 1: запрос отсутствует в кэш памяти
	8. Система выдает отсортированный и отфильтрованный список задач
Исключения сценария выполнения варианта использования "Просмотр заданий"	
Действия актеров	Отклик системы
Исключение 1: запрос отсутствует в кэш памяти	
	7.1 Система видит, что запрашиваемые данные отсутствуют в кэш-памяти
	7.2 Система увеличивает каунтер данного запроса, чтобы при многократном увеличении добавить этот запрос в кэш
	7.3 Система отправляет запрос в базу данных и получает ответ

Таблица 2 — Вариант использования «Просмотр заданий»

С остальными вариантами использования можно ознакомиться в Приложении 1.

Диаграмма последовательности — это тип «UML-диаграммы», представляющий последовательность сообщений между объектами или компонентами в определенном сценарии использования. Это позволяет визуализировать последовательность операций и взаимодействий между различными элементами системы.

Диаграмма последовательности имеет следующие основные элементы:

1. Объект (или компонент), которому передается сообщение.
2. Представляет линию жизни каждого объекта как часть выполнения схемы.
3. Сообщение, которое отправляется между объектами и отображается в виде стрелки.

Диаграммы последовательности играют важную роль в проектировании ИС:

1. Визуализируйте и проанализируйте набор операций и взаимодействий между компонентами системы.
2. Определите потенциальные проблемы и узкие места в процессе обмена сообщениями.
3. ИС оптимизирует архитектуру системы и повышает производительность и эффективность системы.
4. ИС служит документом, который помогает разработчикам и другим участникам проекта лучше понять логику системы.
5. ИС оптимизирует процесс разработки и тестирования системы, позволяя более эффективно выявлять ошибки на ранней стадии и исправлять их.
6. Диаграммы последовательности — это ясный и простой для понимания способ выразить логику системы и облегчить общение между участниками проекта.
7. Это помогает регистрировать системные требования и функциональные возможности, а также помогает поддерживать целостность и ясность проекта на протяжении всего жизненного цикла разработки.

Таким образом, диаграммы последовательности являются мощным инструментом для анализа, проектирования и регистрации информационных систем, помогая создавать эффективные и высококачественные решения, отвечающие потребностям клиентов и пользователей.

Ниже продемонстрирована диаграмма последовательности для функционально блока «Просмотр заданий».

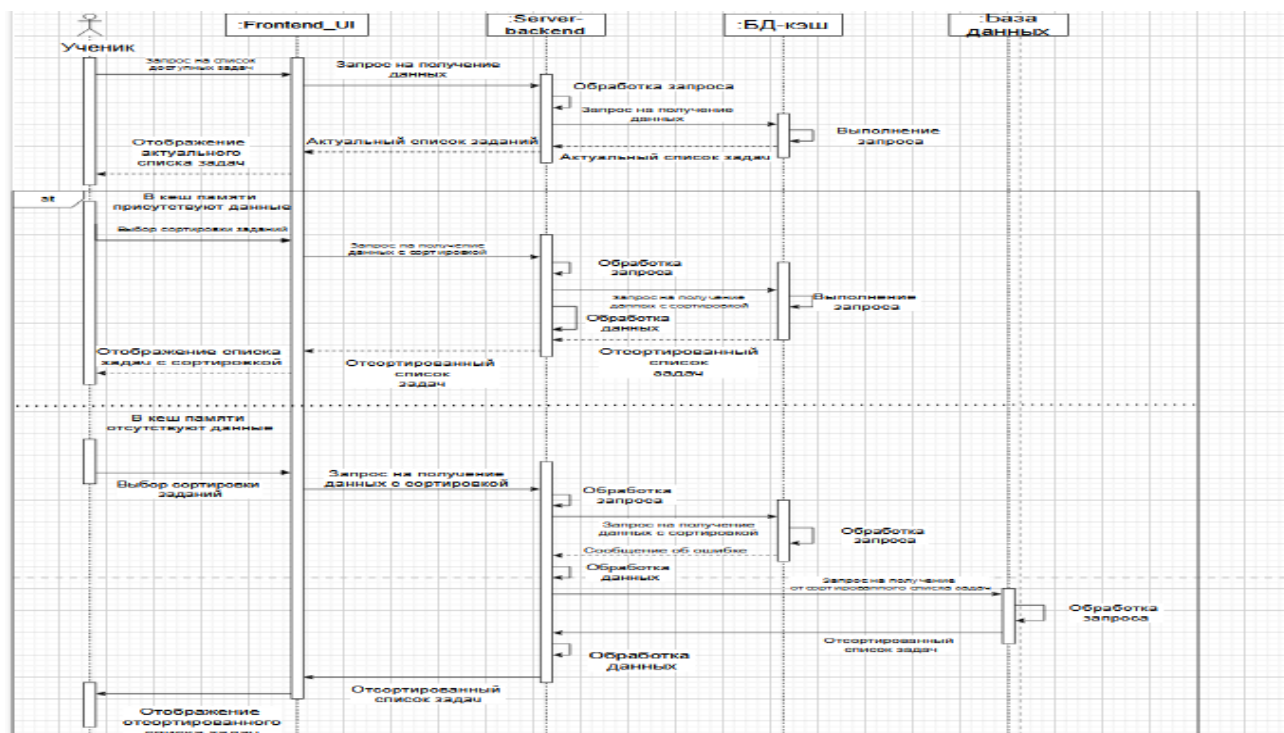


Рисунок 2 — Диаграмма последовательности для блока «Просмотр заданий»

Диаграмма компонентов — элемент языка моделирования UML, статическая структурная диаграмма, которая показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п. Более подробно с диаграммами последовательности можно ознакомиться в приложении Б

Для образовательной платформы «Data World» была спроектирована диаграмма компонентов.

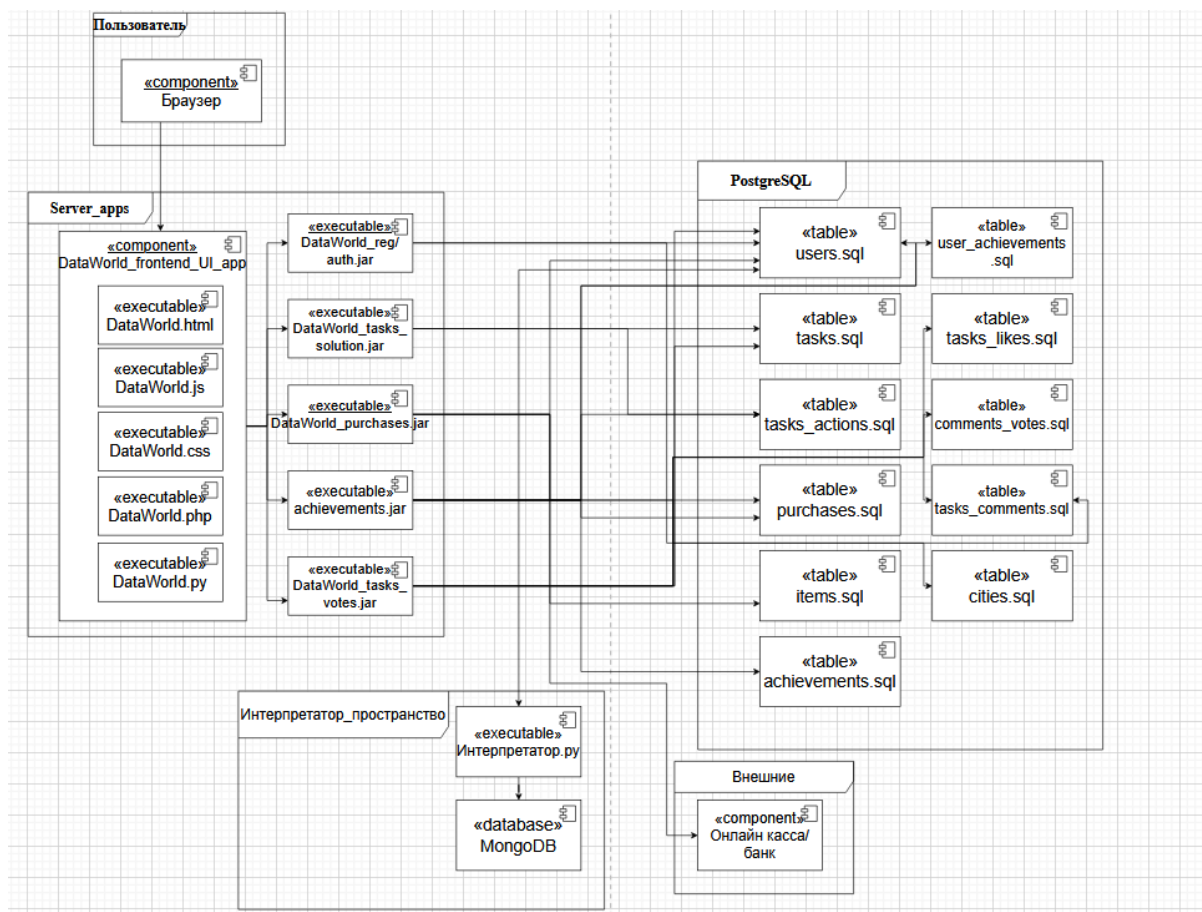


Рисунок 3 — Диаграмма компонентов

Диаграмма развертывания UML (Deployment Diagram) — это тип диаграммы, который используется для моделирования физической структуры системы, включая распределение компонентов, узлов и связей между ними. Диаграмма развертывания позволяет описать, как компоненты системы размещаются на аппаратном обеспечении (узлах) и как они взаимодействуют друг с другом.

Важность диаграммы развертывания при проектировании информационных систем заключается в следующем:

1. Визуализация архитектуры: схемы развертывания позволяют разработчикам и архитекторам визуализировать физическую структуру системы, чтобы лучше понять ее архитектуру и конфигурацию компонентов.
2. Оптимизация ресурсов: можно использовать схемы развертывания для оптимизации распределения компонентов системы между узлами на основе

требований к ресурсам и производительности узлов. Это позволяет эффективно использовать вычислительные мощности и другие ресурсы.

3. Выявление «узких» мест: анализируя схемы развертывания, возможно выявить потенциальные «узкие» места и проблемы с распределением компонентов, чтобы избежать потенциальных проблем с производительностью и недоступностью системы.

4. Планы расширения: схема развертывания поможет спланировать расширение системы, указав, как добавлять новые узлы или компоненты для поддержки роста бизнеса.

В целом, схемы развертывания являются важными инструментами проектирования ИТ-систем, которые помогают разработчикам лучше понимать физическую структуру систем, оптимизировать ресурсы и предотвращать проблемы с производительностью.

Сама схема разработки была разработана как часть дизайна образовательной платформы «Data World», как показано ниже.

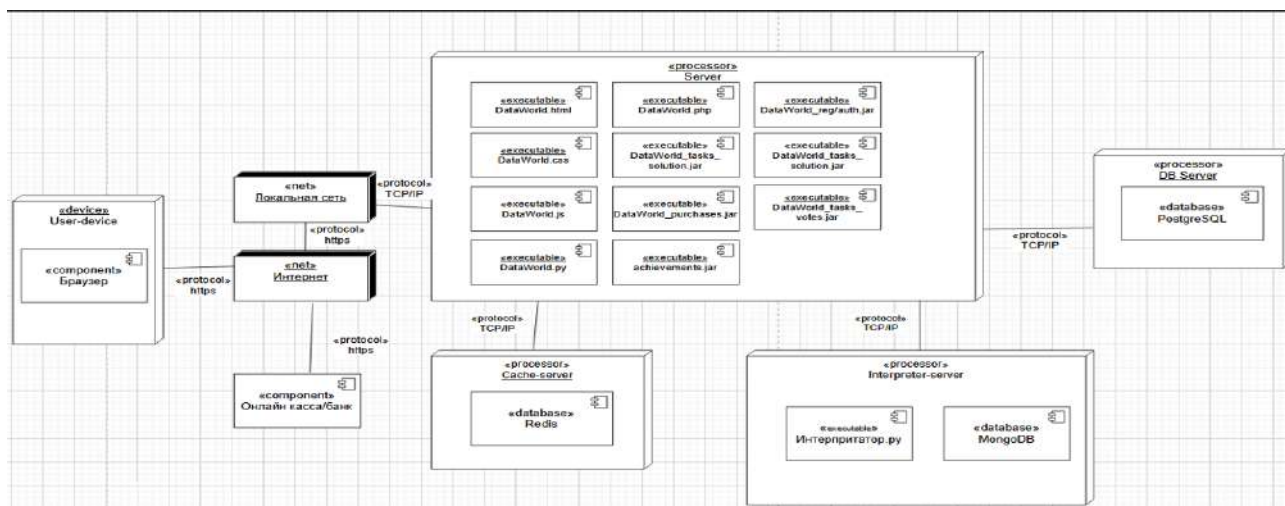


Рисунок 4 — Диаграмма развертывания

2.5 Проектирование базы данных

Разработка архитектуры БД является фундаментальным шагом, который во многом определяет успех всего проекта. Хорошо продуманная структура БД

обеспечивает надежность, производительность и масштабируемость системы, облегчая разработку и обслуживание.

Во-первых, дизайн БД позволяет четко определить, какие данные будут храниться и как они будут связаны. Без этого разработчики столкнулись бы с хаотичным хранением информации, дублированием данных и несогласованностью операционной логики. Например, если между Пользователем и решением его задачи нет связи, система не может правильно отображать прогресс или историю пробных версий.

Кроме того, при правильной архитектуре БД риск аномалий данных, таких как потеря целостности или избыточности, может быть сведен к минимуму. Например, если таблица не нормализована, одни и те же данные могут повторяться в разных местах, что затрудняет обновление и приводит к ошибкам. С другой стороны, чрезмерная нормализация может замедлить выполнение запроса, поэтому важно найти баланс между структурой и производительностью.

Еще одним важным аспектом является масштабируемость. На старте проекта может казаться, что простой структуры достаточно, но по мере роста системы (новые функции, больше пользователей) плохо спроектированная БД станет узким местом. Например, если изначально не заложить индексы для часто используемых запросов, с ростом данных скорость работы системы резко упадет.

Проектирование БД также помогает разделить ответственность между разработчиками. Когда схема данных четко определена, фронтенд- и бэкенд-разработчики могут работать параллельно, зная, какие данные доступны и как их запрашивать. Это ускоряет процесс разработки и снижает количество ошибок из-за недопонимания.

Начинать проектирование БД необходимо после сбора и анализа основных функциональных требований для системы, для этого необходимо обратиться к пункту 2.1 Анализ требований, после этого можно приступать к проектированию БД.

В рамках разрабатываемой ИС спроектируем БД для базового функционала системы, а именно: регистрация и авторизация пользователей,

решение задач и получения списка доступных задач, а также покупки premium подписки.

Для реализации функционала регистрации и входа нам понадобится таблица Users, содержащая информацию об зарегистрированных пользователях, таблица Problems, содержащая информацию о задачах, например их сложность, названия, тему и т.д. Таблица Submissions, содержащая информацию о попытках решения задач пользователями, таблица «SolutionsHistory», в которой будет храниться информация только об успешных попытках решения задач, а также таблицы предназначенные для хранения информации об оплате платных услуг и транзакций клиентов.

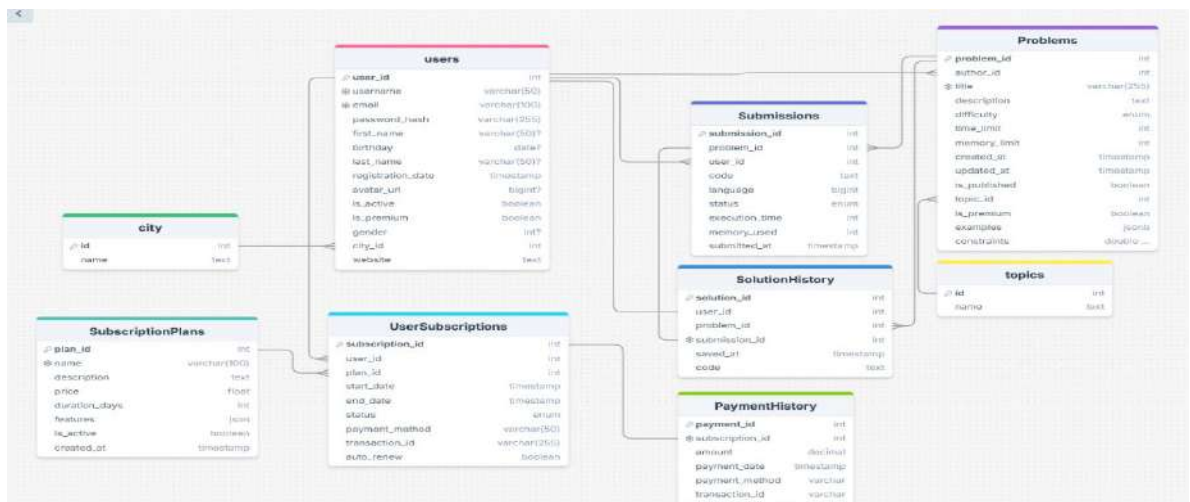


Рисунок 5 — Архитектура таблиц БД

ГЛАВА 3. РАЗРАБОТКА

ИС будет разрабатываться в микро сервисной архитектуре, что позволит достичь гибкости в разработке, а также обеспечит отказоустойчивость отдельных сервисов при критической ошибке других, путем изолированности работы друг от друга. Выбранный подход минимизирует влияние сбоев на общую работоспособность сервиса, поскольку каждый блок-системы функционирует автономно и взаимодействует через четко определенные программные интерфейсы приложения (далее — API). Также ИС будет развернута в Docker контейнере, что упростит развертывание и деплой. Контейнеризация также позволяет легко масштабировать сервисы, оркестрировать их с помощью «Kubernetes» или других подобных инструментов и оперативно восстанавливать работу в случае сбоев.

В качестве языков разработки выбраны: «HTML», «CSS», «JavaScript» (далее — «JS»), «React.js» и «TypeScript». Выбранный стек позволяет создать современное, производительное и удобное в поддержке веб-приложение. «HTML» и «CSS» формируют базовую структуру и стилизацию интерфейса, обеспечивая кросс-браузерную совместимость и адаптивность под разные устройства. «JavaScript», как основной язык веб-разработки, добавляет интерактивность и динамическое поведение страниц, а его расширение, «TypeScript», вносит строгую типизацию, повышая надежность кода, упрощая его рефакторинг и снижая вероятность ошибок на этапе разработки. Использование «React.js» в качестве frontend-фреймворки позволяет строить высокопроизводительные пользовательские интерфейсы с компонентным подходом, что значительно ускоряет разработку и облегчает поддержку кода. Виртуальный интерфейс программирования приложений (далее — DOM) в «React» оптимизирует рендеринг, минимизируя нагрузку на браузер и обеспечивая плавную работу даже в сложных интерфейсах. Кроме того, экосистема «React» включает множество готовых решений и библиотек (таких как «Redux» для управления состоянием или «React Router» для навигации), что

сокращает время реализации типовых функций. В качестве баз данных будут использоваться «PostgreSQL».

Дополнительно будет использоваться распределённая система управления версиями «Git». Это необходимо для организации удобной работы на разных устройствах и между различными командами разработки. В качестве редактора кода будет использоваться один из самых популярных для Web-разработки, а именно «Visual Studio Code» от компании Microsoft.

3.1 Установка необходимого окружения

Для начала необходимо скачать и установить на локальный компьютер программу «Node.js» для запуска локального сервера. Сделать это можно с официального сайта. Дополнительно необходимо установить некоторые зависимости и библиотеки для более продвинутой разработки, например: `npm i react-scripts`.



Рисунок 6 — Установка программы «Node.js» на ПК

Далее, для удобного редактирования кода и файлов установим редактор исходного кода «Visual Studio Code» от компании Microsoft, сделать это можно также с официального сайта⁷⁶. После установки компонентов, необходимых для непосредственной разработки ИС, необходимо установить компоненты, отвечающие за развертывание и деплой системы. Установим «GitHub Desktop» и «Docker Desktop».

Далее для удобного редактирования кода и файлов установим редактор исходного кода «Visual Studio Code» от компании Microsoft, сделать это можно также с официального сайта⁷⁷.

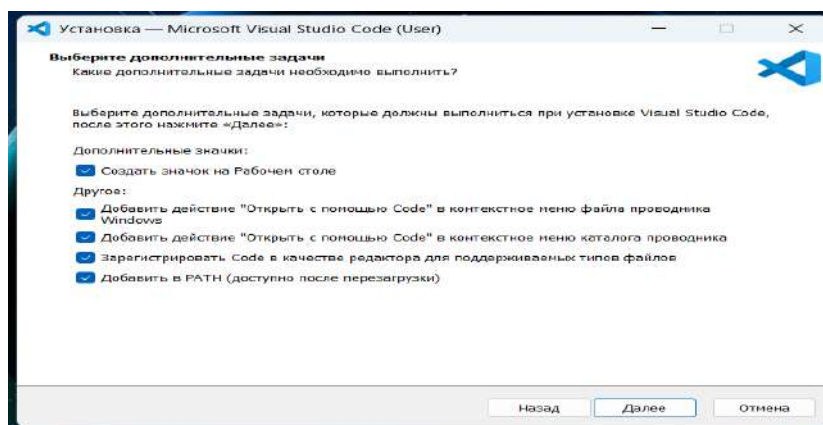


Рисунок 7 — Установка программы «Visual Studio Code» на ПК

После установки компонентов, необходимых для непосредственной разработки ИС, необходимо установить компоненты, отвечающие за развертывание и деплой системы. Установим «GitHub Desktop» и «Docker Desktop».

⁷⁶ <https://code.visualstudio.com/>

⁷⁷ <https://code.visualstudio.com/>

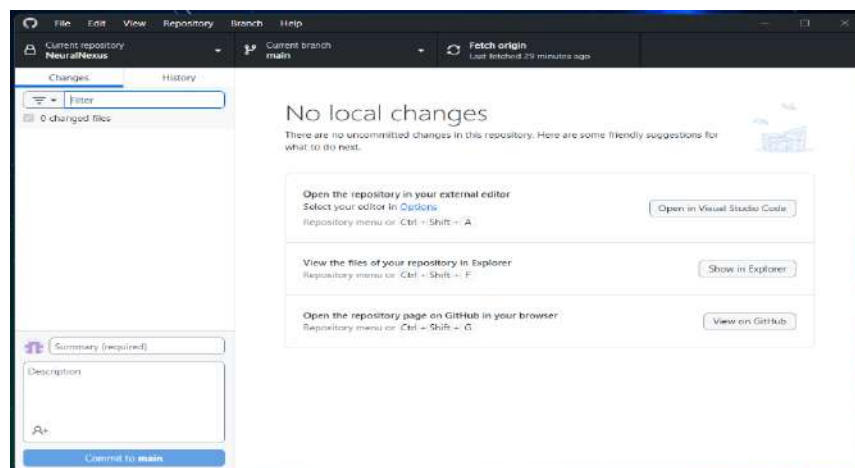


Рисунок 8 — Установка программ «GitHub» и «Docker Desktop» на ПК

После установки всех необходимых компонентов создадим рабочую директорию и объявим в ней «git» репозиторий, а также для удобства разработки можем связать репозиторий с «GitHub». Это даст возможность работать над проектом с разных устройств, не перемещая данные на флешки или посторонние носители, а также упростит контроль версий проекта.

3.2 Создание пространства для проекта

Для начала необходимо создать скелет «React-проекта» в нашей рабочей директории. Ниже приведено начальное древо директорий проекта с подписями о их значении. Стоит отметить, что в древе отсутствуют сами файлы, а также некоторые директории, например: «node_modules» и «.git» т.к. они содержат системную информацию не важную для самой разработки, это решение позволяет отобразить древо директории в читаемом для человека виде.

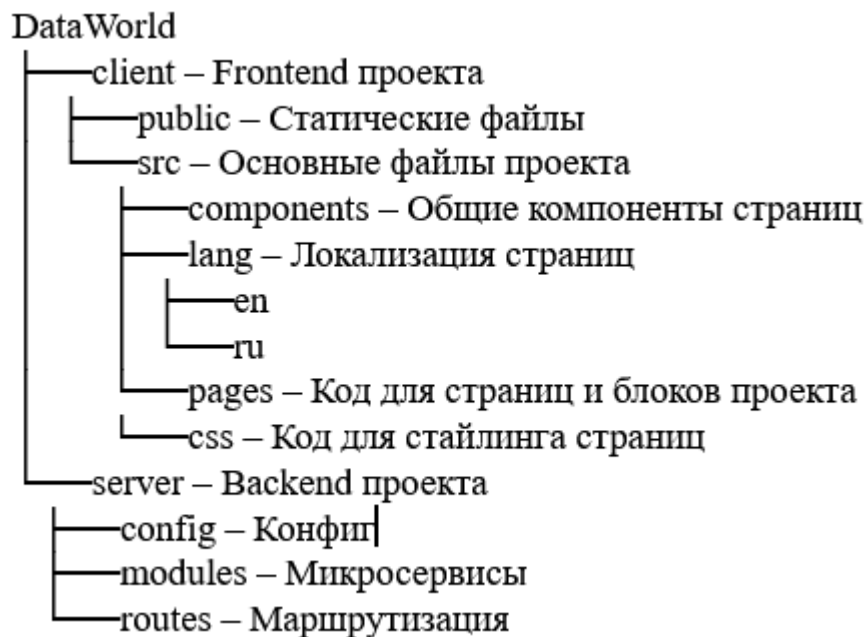


Рисунок 9 — Древо директории проекта

3.3 Создание визуальной части (Frontend)

Для создания визуальной части будем использовать классическую комбинацию «HTML» и «CSS», с дополнением «JavaScript» и «Typescript», в следующем формате: пишем классический HTML-код, в котором создаем скелет страницы со всеми функциональными блоками. Данный «HTML» код оборачиваем в функцию, которую в дальнейшем «JS» будет использовать для вывода страницы сайта на экран, к этому же файлу импортируем файл с аналогичным названием, содержащий CSS-стили.

Ниже представлен пример реализации вышеописанного метода для страницы со списком задач.

```
1  import React from 'react';
2  import './css/ProblemsList.css';
3  function ProblemsList() {
4    return (
5      <div className='ProblemsList'>
6        <h1>Problems List</h1>
7      </div>
8    );
9  }
10 export default ProblemsList;
```

Рисунок 10 — Шаблон создания страницы

Данный пример демонстрирует шаблон для создания страницы. Логика заключается в создании функции (желательно одноименную со страницей), в которой описывается работа страницы, создание переменных, а также сама структура страницы на классическом «HTML». Для добавления нового функционала необходимо редактировать код внутри «return()», добавляя туда, например, новые блоки «div», которые отвечают за контейнеры на странице.

Если страница требует более сложной логики работы, то в самой функции можно создавать дополнительные подфункции, которые будут отвечать за дополнительный функционал страницы, например: кнопки, рекламные баннеры и т.д. В самом конце файла экспортируем целевую функцию для того, чтобы в главном файле была возможность импортировать ее и использовать. По такому шаблону создадим страницы сайта. Данные файлы будут лежать в директории проекта: «DataWorld\client\src\pages».

Далее, создадим компоненты, которые будут располагаться на каждой из страниц: так называемые, «шапку» и «подвал» сайта, местонахождение файлов: «DataWorld\client\src\components». Также, в данной директории создадим файл, отвечающий за навигацию по сайту.

```
1  import { Link } from 'react-router-dom';
2  import React from 'react';
3
4  function Navigation() {
5    return (
6      <nav>
7        <ul>
8          <li><Link to="/">Home</Link></li>
9          <li><Link to="/problems">Problems</Link></li>
10         <li><Link to="/register">Register</Link></li>
11         <li><Link to="/login">Login</Link></li>
12       </ul>
13     </nav>
14   );
15 }
16
17 export default Navigation;
```

Рисунок 11 — Функция навигации по сайту

Далее создадим функцию «App.js», которая будет объединять в себе весь функционал сайта и оркестровать его работу.

```
1 import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
2 import HomePage from './pages/HomePage.tsx';
3 import ProblemsList from './pages/ProblemsList.tsx';
4 import Problem from './pages/Problem.tsx';
5 import RegistrationPage from './pages/RegistrationPage.tsx';
6 import LoginPage from './pages/LoginPage.tsx';
7 import Navigation from './components/Navigation.tsx';
8 import Footer from './components/Footer.tsx';
9 import './App.css';
10
11 function App() {
12   return (
13     <Router>
14       <div className="app-container">
15         <Navigation />
16         <div className="content-wrap">
17           <Routes>
18             <Route path="/" element={<HomePage />} />
19             <Route path="/problems" element={<ProblemsList />} />
20             <Route path="/problems/:id" element={<Problem />} />
21             <Route path="/register" element={<RegistrationPage />} />
22             <Route path="/login" element={<LoginPage />} />
23           </Routes>
24         </div>
25         <Footer />
26       </div>
27     </Router>
28   );
29 }
30 export default App;
```

Рисунок 12 — Основное приложение «App.js»

В данном приложении импортируем все страницы, компоненты и функцию навигации, также указывая элементы «Route» между страницами, тем самым создаем логику работы сайта.

Далее, при помощи командной строки «cmd» зайдём в директорию нашего проекта и введём команду «npm start», которая запустит локальный сервер (по умолчанию: <http://localhost:3000>) для демонстрации работы разработанного сайта.



Рисунок 13 — Пример работы страницы регистрации

На данном этапе страницы выглядят очень просто из-за ненастроенной стилизации «CSS», однако они довольно функциональны: все ссылки кликабельны, при помощи «шапки» мы можем перемещаться по страницам, в форме «Ввод» можем вводить данные, которые пока никуда не попадут.

Теперь попробуем реализовать различные динамические эффекты для страницы «Список задач» при помощи «JS». Создадим рекламный баннер, также добавим дополнительных блоков, например: таблицу, которая будет выводить список доступных задач, пока он будет фиксированным, а также немного стилизуем страницу при помощи «CSS» файла.

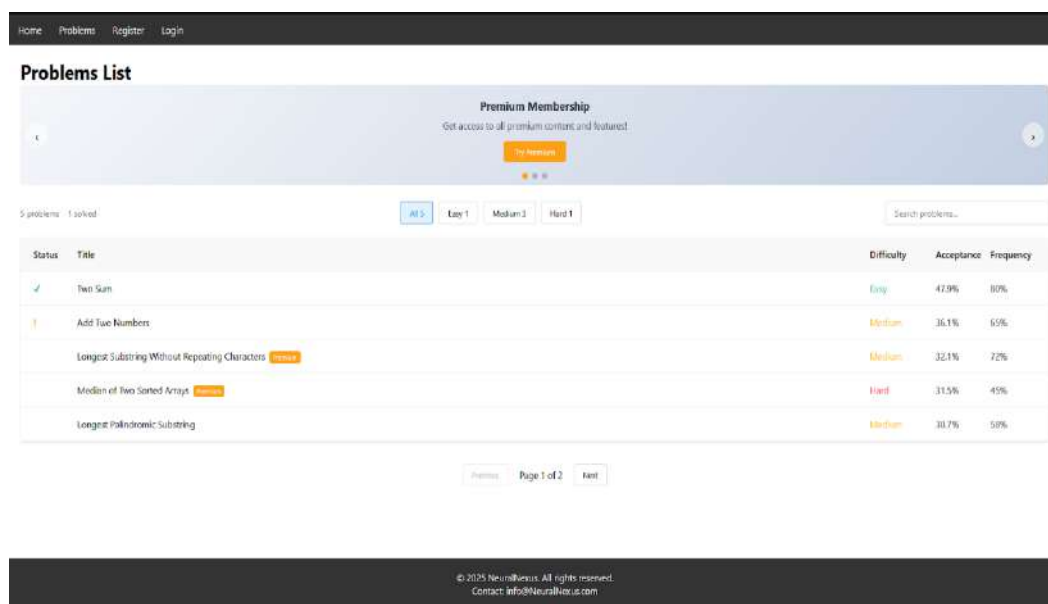


Рисунок 14 — Страница со списком задач

На рис. 9 представлены результаты работ по улучшению страницы «ProblemsList», на которую были добавлены динамические эффекты, а также кнопки, которые функционируют на «JS» и позволяют сортировать список задач.

Таким образом были созданы все функциональные страницы образовательной платформы «DataWorld». На этапе создания визуальной части (Frontend) все страницы имеют урезанный функционал, т.к. не имеют серверной части и подключенной БД.

3.4 Создание серверной части (Backend)

Для того, чтобы Web-приложение функционировало, необходимо разработать серверную часть приложения, которая будет отвечать за «backend» составляющую. Для этого в корневой директории проекта создадим директорию «Server», которая будет являться отдельным приложением, развернутым на «порте 5000».

Первым шагом необходимо создать основное приложение «[app.js](#)», которое будет содержать в себе основную логику работы приложения, для создания будет использовать фреймворк «js express».

```

1  const express = require('express');
2  const cors = require('cors');
3  const sequelize = require('./config/database');
4  const authRoutesRegistration = require('./routes/registration');
5  const authRoutesProblems = require('./routes/problems');
6  const authRoutesAuth = require('./routes/auth');
7  const authRoutesVerify = require('./routes/verify');
8  require('dotenv').config();
9
10 const app = express();
11
12 // Middleware
13 app.use(cors());
14 app.use(express.json());
15
16 // Routes
17 app.use('/api/registration', authRoutesRegistration);
18 app.use('/api/login', authRoutesAuth);
19 app.use('/api/problems_get', authRoutesProblems);
20 app.use('/api/verify', authRoutesVerify);
21
22 // Database connection
23 sequelize.sync()
24   .then(() => {
25     console.log('Database connected');
26     const PORT = process.env.PORT || 5000;
27     app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
28   })
29   .catch(err => console.error('Database connection error:', err));

```

Рисунок 15— Код «app.js»

В коде нам необходимо импортировать все зависимости, отвечающие за функционирование системы, а также настроить маршруты между компонентами. При помощи «app.use» перехватываем запросы с «frontend» части и перенаправляем их в соответствующие файлы, например запрос со страницы регистрации «/api/registration» перенаправляем в «endpoint /routes/registration», в котором запрос обрабатывается и создается новый пользователь в БД. Также в основном приложении создаем подключение к БД.

Далее создадим модель «user», которая будет инициироваться при создании БД. Для простоты и наглядности сделаем это без дополнительных параметров, которые будут использоваться на продакшене.

```

const { DataTypes } = require('sequelize');
const sequelize = require('../config/database');
const bcrypt = require('bcryptjs');

const User = sequelize.define('User', {
  username: {
    type: DataTypes.STRING,
    allowNull: false,
    unique: true,
  },
  email: {
    type: DataTypes.STRING,
    allowNull: false,
    unique: true,
    validate: {
      isEmail: true,
    },
  },
  password: {
    type: DataTypes.STRING,
    allowNull: false,
  },
}, {
  hooks: {
    afterSync: () => console.log('Таблица Users синхронизирована'),
    afterCreate: (user) => console.log('Пользователь создан:', user.username),
    beforeCreate: async (user) => {
      user.password = await bcrypt.hash(user.password, 10);
    },
  },
});

module.exports = User;

```

Рисунок 16 — Модель «User»

Создадим «endpoint» регистрации клиентов, в него будут попадать все запросы на регистрацию нового пользователя, которые будут поступать с «frontend» части.

Логика работы структуры такова: клиент заполняет данные на странице регистрации и нажимает кнопку регистрации, после этого «frontend» отправляет POST запрос на сервер «localhost:5000», серверное приложение «[App.js](#)» принимает эти запросы через «app.use('/api/registration', authRoutesRegistration)»; и отправляет их по маршруту «authRoutesRegistration», который ведет в «endpoint» регистрации, созданный выше. Затем «endpoint» подключается к модели «user» (таблица в БД) и создает в ней нового пользователя с данными, которые лежат в теле POST запроса «frontend» указанные пользователем при заполнении формы регистрации.

```

1  const express = require('express');
2  const router = express.Router();
3  const { User } = require('../models/User.js');
4  const bcrypt = require('bcryptjs');
5
6  // Registration endpoint
7  router.post('/registration', async (req, res) => {
8    try {
9      const { username, email, password } = req.body;
10
11      const User = require('../models/User.js');
12
13      // Check if user already exists
14      const existingUser = await User.findOne({ where: { email } });
15      if (existingUser) {
16        return res.status(400).json({ message: 'User already exists' });
17      }
18
19      // Create new user
20      const user = await User.create({ username, email, password });
21
22      res.status(201).json({
23        message: 'User registered successfully',
24        user: {
25          id: user.id,
26          username: user.username,
27          email: user.email,
28        }
29      });
30    } catch (error) {
31      console.error('Registration error:', error);
32      res.status(500).json({ message: 'Registration failed' });
33    }
34  });
35
36  module.exports = router;

```

Рисунок 17 — Endpoint регистрации пользователя

Нам осталось только добавить функцию в код страницы регистрации, которая будет собирать данные и отправлять их на серверную часть.

```
function RegistrationPage() {
  const [formData, setFormData] = useState({
    username: '',
    email: '',
    password: '',
    agreedToTerms: false
  });
  const [error, setError] = useState('');
  const [isLoading, setIsLoading] = useState(false);
  const navigate = useNavigate();

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData(prev => ({ ...prev, [name]: value }));
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    setIsLoading(true);
    setError('');

    try {
      const response = await fetch('http://localhost:5000/api/registration/registration', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify(formData),
      });

      const data = await response.json();

      if (!response.ok) {
        throw new Error(data.message || 'Registration failed');
      }

      navigate('/login'); // Перенаправляем после успешной регистрации
    } catch (err) {
      setError(err.message);
    } finally {
      setIsLoading(false);
    }
  };
}
```

Рисунок 18 — Функция отправки POST запроса на регистрацию нового пользователя

При помощи инструментов разработчика посмотрим, что происходит при отправке POST-запроса со страницы регистрации.

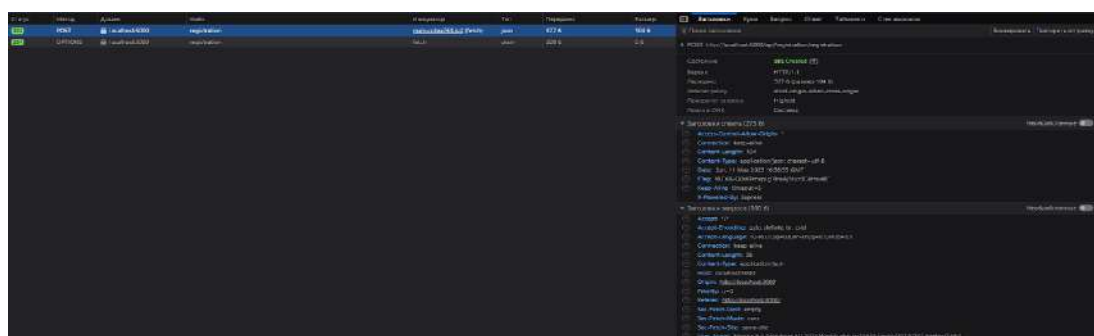


Рисунок 19 — Отправка POST-запроса

Отправка запроса успешна, справа на скриншоте можно увидеть тело запроса, в ответ сервер отправляет «код 201», что свидетельствует об успешном

завершении процедуры по созданию нового пользователя, отправку кода 201 можно проследить в файле «endpoint» регистрации пользователя.

Следующим шагом, после регистрации, является аутентификация пользователя на сайте, для этого аналогично создадим «endpoint» аутентификации пользователя. В качестве архитектуры будет использоваться «JWT-ключи». Данное решение выигрывает перед подходом с классическими сессиями в масштабируемости, т.к. «JWT» не требует хранения состояния на сервере, его легко использовать в микросервисной архитектуре и распределённых системах. Также, в «JWT» можно хранить любые данные (например, «user_id», «role», «permissions»), что уменьшает количество запросов к БД.

```
1 // server/routes/auth.js
2 const express = require('express');
3 const router = express.Router();
4 const bcrypt = require('bcryptjs');
5 const jwt = require('jsonwebtoken');
6 const { User } = require('../models/User.js');
7 //const authMiddleware = require('../middleware/authMiddleware.js');
8
9 router.post('/login', async (req, res) => {
10   const { email, password } = req.body;
11
12   const User = require('../models/User.js');
13
14   try {
15     // 1. Проверяем существование пользователя
16     const user = await User.findOne({ where: { email } });
17
18     if (!user) {
19       return res.status(401).json('Неверный email или пароль');
20     }
21
22     // 2. Проверяем пароль
23     const validPassword = await bcrypt.compare(password, user.password);
24
25     if (!validPassword) {
26       return res.status(401).json('Неверный email или пароль');
27     }
28
29     // 3. Генерируем JWT токен
30     const token = jwt.sign(
31       { userId: user.id },
32       process.env.JWT_SECRET,
33       { expiresIn: '1h' }
34     );
35
36     res.json({ token, userId: user.id });
37
38   } catch (err) {
39     console.error(err.message);
40     res.status(500).send('Ошибка сервера');
41   }
42 });
43
44 module.exports = router;
```

Рисунок 20 — «Endpoint» регистрации пользователя

По аналогии с логикой регистрации необходимо добавить функцию, которая будет отправлять запрос на серверную часть в «[App.js](#)», которое будет перенаправлять запрос в «endpoint», где будет происходить проверка существования пользователя, а также выдавать «JWT-токен» юзеру.

Следующим шагом станет разработка основного функционала, а именно функции решения задач. Для этого нам необходимо создать новый компонент «ProblemsList», который будет создавать таблицу с задачами в БД при ее инициализации. В коде компонента нам необходимо отразить все поля, а также их атрибуты, необходимые для корректной работы сервиса.

```
1  const { DataTypes } = require('sequelize');
2  const sequelize = require('../config/database');
3  const problemsSeedData = require('../config/problemsSeedData');
4
5  const ProblemsList = sequelize.define('ProblemsList', {
6    title: {
7      type: DataTypes.STRING,
8      allowNull: false,
9      unique: true,
10   },
11   difficulty: {
12     type: DataTypes.ENUM('Easy', 'Medium', 'Hard'),
13     allowNull: false,
14   },
15   description: {
16     type: DataTypes.TEXT,
17     allowNull: false,
18   },
19   topic: {
20     type: DataTypes.STRING,
21     allowNull: false,
22   },
23   is_premium: {
24     type: DataTypes.BOOLEAN,
25     allowNull: false,
26     defaultValue: false,
27   },
28   examples: {
29     type: DataTypes.JSONB,
30     allowNull: false,
31     defaultValue: [],
32   },
33   constraints: {
34     type: DataTypes.ARRAY(DataTypes.STRING),
35     allowNull: false,
36     defaultValue: [],
37   },
38   tags: {
39     type: DataTypes.ARRAY(DataTypes.STRING),
40     allowNull: false,
41     defaultValue: [],
42   },
43 }, {
44   hooks: {
45     afterSync: async (options) => {
46       console.log('Таблица ProblemsList синхронизирована');
47       await seedProblemsList();
48     }
49   }
50 });
```

Рисунок 21 — Код компонента «ProblemsList»

Далее, нам необходимо инициализировать начальные задачи, которые будут создаваться вместе с таблицей. Сделаем это в файле «problemsSeedData.js», который будет подключаться к созданной выше компоненте. В данном файле создадим 10 произвольных задач на практику программирования, добавив к ним все необходимые поля. Нужно это для автоматизации процесса создания таблиц при разворачивании «Docker-контейнера».

После того, как таблица с необходимыми полями создана в БД, в эту таблицы внесены несколько строк с заданиями и их описанием, мы можем приступить к созданию серверных компонентов для запроса к БД и получению списка задач, а также информации по каждой конкретной задаче для отображения ее страницы. Создадим файл «endpoint-Problems», в котором пропишем логику обработки запросов, которые будут идти с клиентской части сервиса. В данном файле необходимо создать две функции (два «endpoint»), одна из которых будет обрабатывать запрос на получение всего списка задач (включая отфильтрованный список), а другая будет обрабатывать запрос на получение информации по одной конкретной задаче.

```
1 const express = require('express');
2 const router = express.Router();
3 const { ProblemList } = require('../models/ProblemList.js');
4 const { Op } = require('sequelize');
5
6 // Основной endpoint
7 router.get('/problems_get', async (req, res) => {
8   try {
9
10     const ProblemList = require('../models/ProblemList.js');
11
12     const { difficulty, search, page = 1 } = req.query;
13     const itemsPerPage = 10;
14     const offset = (page - 1) * itemsPerPage;
15
16     // Создаем условия для фильтрации
17     const where = {};
18
19     if (difficulty && difficulty !== 'All') {
20       where.difficulty = difficulty;
21       console.log('Применен фильтр сложности: ${difficulty}');
22     }
23
24     if (search) {
25       where.title = {
26         [Op.iLike]: `%${search}%`
27       };
28     }
29
30     // Запрос в базу данных
31     const { count, rows } = await ProblemList.findAndCountAll({
32       where,
33       limit: itemsPerPage,
34       offset,
35       order: [['id', 'ASC']]
36     });
37
38     const responseData = {
39       problems: rows,
40       totalPages: Math.ceil(count / itemsPerPage)
41     };
42     res.json(responseData);
43   } catch (error) {
44     console.error(error);
45   }
46 }
```

Рисунок 22 — Код компонента «endpoint-Problems»

Теперь, когда серверная часть обработки запросов задач готова, нам необходимо на клиентской стороне создать функцию отправки запроса на получение списка задач, логика работы будет аналогична системе регистрации и входа, клиент будет посылать запрос в файл «App.js», который в свою очередь перенаправит запрос на соответствующий «Endpoint-file».

После выполнения вышеперечисленных действий можем наблюдать работу данной функции. При попадании на страницу со списком задач, клиентская часть сразу посылает запрос на сервер для получения списка доступных задач, после этого отображает их на странице.

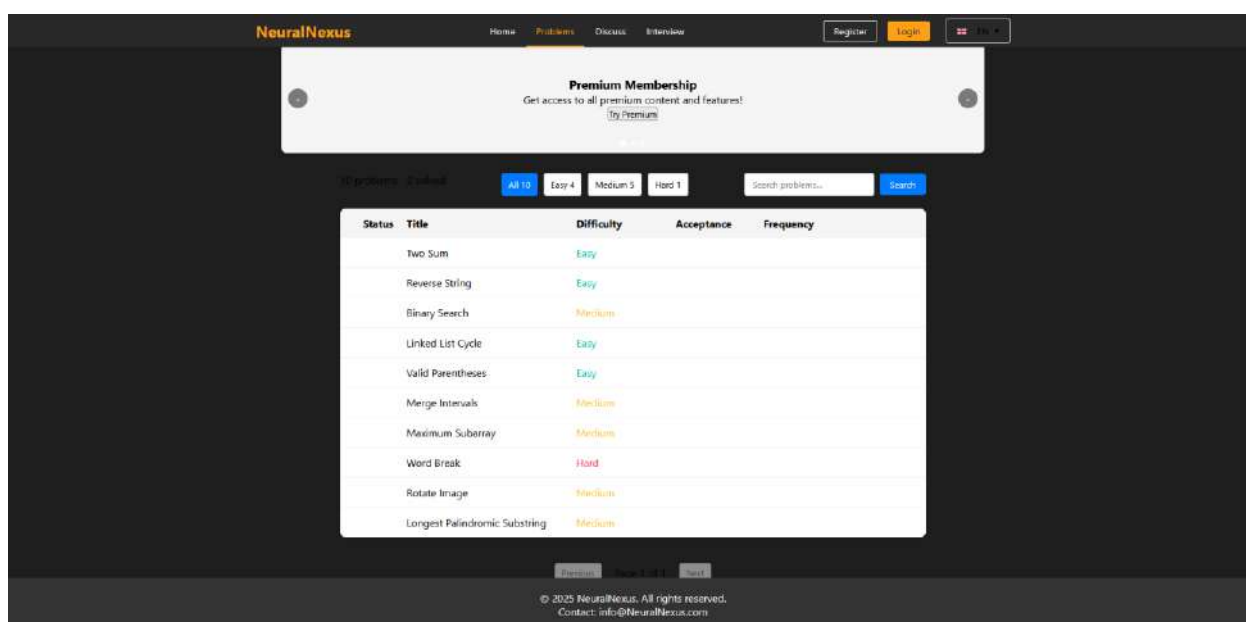


Рисунок 23 — Макет страницы “Список задач”

Аналогичный процесс будет выполнен и для карточки задачи, клиентская часть будет отправлять «GET-запрос» на получение данных о задаче, номер задачи функция получает из адресной строки, где помечается id задачи из общего списка, который мы получили ранее, поэтому информация будет всегда подтягиваться именно по выбранной задаче. Данный запрос попадет во второй «endpoint», который был специально подготовлен для обработки «GET-запросов» на получение данных об одной задаче, результат работы виден на приведённом ниже скриншоте.

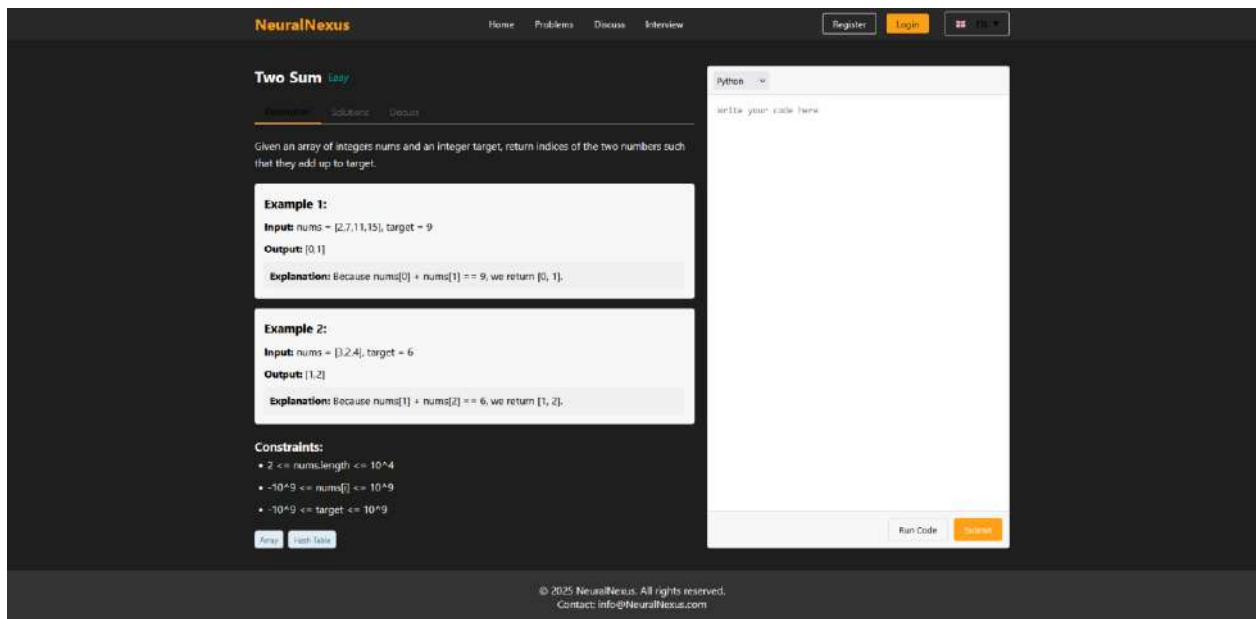


Рисунок 24 — Макет страницы “Задач”

3.5 Контейнеризация проекта

Весь созданный проект будет размещаться в виде «Docker-контейнера». Это делается для того, чтобы приложение можно было развернуть в любой системе и на любом компьютере. «Docker» — это платформа для разработки, развертывания и управления приложениями в контейнерах. Контейнеры позволяют изолировать приложение и его зависимости от основной системы, обеспечивая переносимость и согласованность работы в разных средах.

Основные концепции «Docker»:

1. Образ (Image);
2. Это шаблон для создания контейнеров (как «класс» в ООП);
3. Содержит ОС, зависимости и код приложения;
4. Примеры: «nginx», «postgres», «python:3.9»;
5. Образы хранятся в реестрах («Docker Hub», «GitLab Registry» и др.);
6. Контейнер («Container»);
7. Это запущенный экземпляр образа (как «объект» класса);
8. Изолированная среда с собственными процессами, сетью и файловой системой;
9. Может быть запущен, остановлен, удален;

10. «Dockerfile»;
11. Текстовый файл с инструкциями для сборки образа.

В предыдущих главах, разработали клиентскую (client) и серверную (server) части приложения, в рамках микро сервисной архитектуры они будут являться отдельными «Docker-образами». Это значит, что для них будут нужны разные «Dockerfile» и собираться они будут отдельно друг от друга. Ниже приведен код для основного «Dockerfile», размещенного в корне проекта, «Dockerfile» для «client» и «server» частей являются аналогичными на этапе сборки.

```
1  # Этап сборки
2  FROM node:16 AS builder
3  WORKDIR /app
4  COPY package*.json ./
5  RUN npm install
6  COPY . .
7  RUN npm run build
8
9  # Этап запуска
10 FROM nginx:alpine
11 COPY --from=builder /app/build /usr/share/nginx/html
12 EXPOSE 3000
13 CMD ["nginx", "-g", "daemon off;"]
```

Рисунок 25 — Основной «Dockerfile»

Каждый из «Dockerfile» требует зависимости в виде файла «package.json», который должен лежать в той же директории, что и сам «Dockerfile» (опционально).

«package.json» — это сердце любого проекта на «Node.js» и современного «JavaScript-приложения». Он играет роль манифеста, в котором хранится вся ключевая информация о проекте, его зависимостях и сценариях работы.

Одна из главных задач «package.json» — управление зависимостями. В этом файле есть два основных раздела: «dependencies» и «devDependencies». Первый содержит библиотеки, без которых приложение не сможет работать в продакшене (например, «express» для сервера или «react» для «frontend»). Второй

включает инструменты, нужные только во время разработки: тестовые фреймворки (jest), линтеры (eslint), сборщики (webpack) и другие утилиты вроде «nodemon» для автоматической перезагрузки сервера.

Ещё одна важная функция — определение сценариев в разделе «scripts». Здесь можно прописать команды для запуска приложения (start), тестирования (test), сборки (build) или любых других задач. Ниже представлен пример «package.json» для разрабатываемой системы.

```
1  {
2    "name": "neuralnexus",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/dom": "^10.4.0",
7      "@testing-library/jest-dom": "^6.6.3",
8      "@testing-library/react": "^13.4.0",
9      "@testing-library/user-event": "^13.5.0",
10     "jsonwebtoken": "^9.0.2",
11     "axios": "^1.9.0",
12     "bcryptjs": "^2.4.3",
13     "bootstrap": "^5.3.5",
14     "pg": "^8.15.6",
15     "react": "^18.2.0",
16     "react-bootstrap": "^2.10.9",
17     "react-dom": "^18.2.0",
18     "react-router": "^7.5.1",
19     "react-router-dom": "^6.22.1",
20     "react-scripts": "^5.0.1",
21     "reflect-metadata": "^0.2.2",
22     "sequelize": "^6.0.0",
23     "sequelize-cli": "^6.6.2",
24     "typeorm": "^0.3.23",
25     "web-vitals": "^2.1.4"
26   },
27   "scripts": {
28     "start": "react-scripts start",
29     "build": "react-scripts build",
30     "test": "react-scripts test",
31     "eject": "react-scripts eject"
32   },
33   "eslintConfig": {
34     "extends": [
35       "react-app",
36       "react-app/jest"
37     ]
38   },
39   "browserslist": {
40     "production": [
41       ">0.2%",
42       "not dead",
43       "not op_mini all"
44     ],
45     "development": [
46       "last 1 chrome version",
47       "last 1 firefox version",
48       "last 1 safari version"
49     ]
50   }
51 }
```

Рисунок 26 — Код для «package.json»

Завершающим пунктом станет создание «docker-compose.yaml» файла.

«docker-compose.yaml» (или «docker-compose.yml») — это декларативное описание «multi-container» приложения в «Docker». Он заменяет набор CLI-команд вида `docker run --link -v -p` единой конфигурацией в «YAML-формате», автоматизируя развертывание связанных сервисов. Основными

функциями являются: оркестрация сервисов, абстракция сетевого стека, управление состояниями, унификация окружения, для разрабатываемой системы «docker-compose» будет следующим.

```
1  services:
2    > Run Service
3    postgres:
4      image: postgres
5      environment:
6        POSTGRES_USER: ${DB_USER}
7        POSTGRES_PASSWORD: ${DB_PASSWORD}
8        POSTGRES_DB: ${DB_NAME}
9      ports:
10       - "5432:5432"
11      volumes:
12       - postgres_data:/var/lib/postgresql/data
13      healthcheck:
14        test: ["CMD-SHELL", "pg_isready -U postgres"]
15        interval: 5s
16        timeout: 5s
17        retries: 5
18
19    > Run Service
20    server:
21      build:
22        context: ./server
23      environment:
24        DB_HOST: postgres
25        DB_PORT: ${DB_PORT}
26        DB_NAME: ${DB_NAME}
27        DB_USER: ${DB_USER}
28        DB_PASSWORD: ${DB_PASSWORD}
29        NODE_ENV: development
30      ports:
31       - "5000:5000"
32      depends_on:
33        postgres:
34          condition: service_healthy
35      restart: unless-stopped
36
37    > Run Service
38    client:
39      build:
40        context: ./client
41      ports:
42       - "3000:80"
43      depends_on:
44        - server
45      restart: unless-stopped
46
47  volumes:
48    postgres_data:
```

Рисунок 27 — Код для package.json

Кроме трех вышеперечисленных файлов необходимы и другие, например «.gitignore» — файл, позволяющий игнорировать файлы или директории при работе с git, «.dockerignore» — файл, позволяющий игнорировать файлы или директории при работе с «Docker», «.env» — файл для хранения переменных окружения, «[readme.md](#)» — файл описания проекта.

После создания всех необходимых файлов микросервисная приложение готово к контейнеризации. Для данного процесса необходимо зайти в командную строку (cmd) и перейти в директорию проекта при помощи команды «cd», далее необходимо ввести команду `docker-compose up -build`, которая соберет все компоненты в единое целое, создаст контейнер из образов и запустит его. В приложении «Docker desktop», которое установили ранее, можем наблюдать работу нашего запущенного контейнера, а также его логи, то есть системные уведомления.

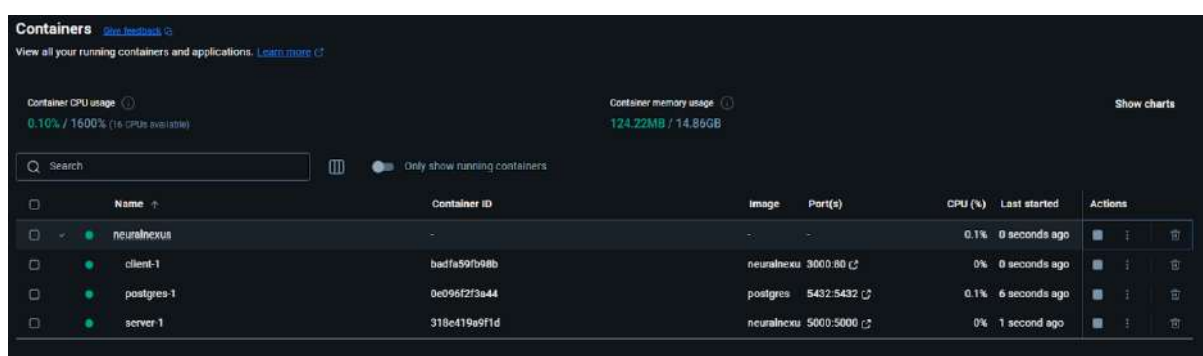


Рисунок 28 — Запущенный контейнер

На изображении видно, что контейнер состоит из трех отдельных, а именно из «client», «server» и «postgres». Образ «postgres» был скачан с официального репозитория «Docker-hub», остальные образы были описаны созданы в этой главе. Также напротив каждого контейнера можно увидеть порт, на которых он работает, для проверки корректности работы перейдем на адрес «<http://localhost:3000/login>».

В адресной строке виден адрес локального порта, который мы указывали при работе с «Docker», а сама страница работает корректно, значит развертывание сервиса прошло успешно, в логах «Docker» также отсутствуют какие-либо ошибки.

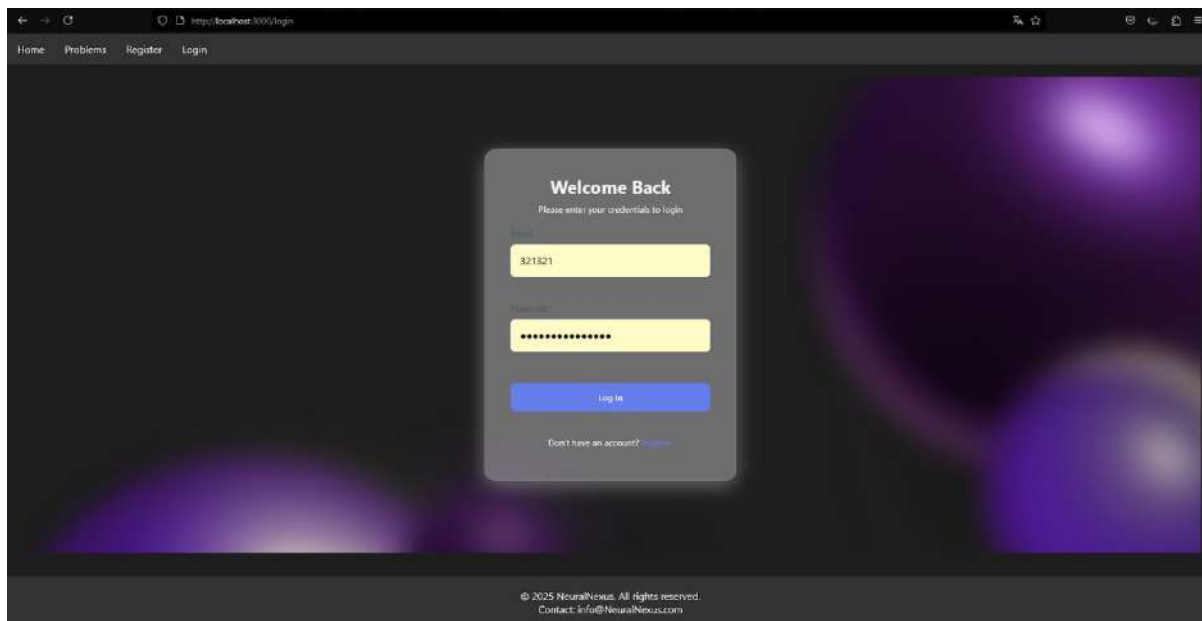


Рисунок 29 — Макет страницы “Вход” в docker

ЗАКЛЮЧЕНИЕ

В ходе проведенного исследования была подтверждена актуальность разработки специализированной образовательной платформы для подготовки специалистов в области Data Science и ИИ. Современный рынок труда испытывает острый дефицит квалифицированных кадров, в то время как существующие образовательные программы зачастую не обеспечивают необходимого уровня практической подготовки. Решение этой проблемы требует комплексного подхода, сочетающего глубокую теоретическую базу, практико-ориентированные задания и адаптивные методы обучения.

В рамках дипломной работы была достигнута поставленная цель — разработана и обоснована архитектура образовательной платформы «Data World», направленной на устранение ключевых недостатков современных «EdTech-решений». В ходе исследования был проведен анализ существующих образовательных платформ, выявлены их слабые стороны, такие как отсутствие системного подхода, недостаток практики и персонализации, определены ключевые потребности ЦА, включая необходимость работы с реальными кейсами, адаптивного обучения и подготовки к техническим собеседованиям, разработаны требования к функционалу платформы, включая модули для теоретического обучения, практических заданий и геймификации, спроектирована архитектура системы с использованием современных технологий, таких как микро-сервисная структура и контейнеризация, что обеспечивает гибкость и масштабируемость решения а также реализован прототип платформы, демонстрирующий ее основные возможности, такие как регистрация и авторизация пользователей,

Теоретическая значимость работы заключается в разработке модели обучения, объединяющей различные форматы (микро-обучение, проектная работа, геймификация) и методы персонализации образовательного контента. **Практическая ценность** исследования подтверждается созданием прототипа

платформы, который может быть использован для дальнейшего развития и внедрения в образовательный процесс.

Перспективы дальнейших исследований включают: расширение функционала платформы за счет интеграции новых технологий, таких как генеративный ИИ для автоматизации проверки заданий, углубленное тестирование системы с привлечением большего числа пользователей для оценки эффективности обучения, а также разработку мобильной версии для повышения доступности обучения. В условиях стремительного развития цифровых технологий создание гибкой и эффективной образовательной платформы, такой как «Data World», способно внести значимый вклад в подготовку высококвалифицированных специалистов, сократив разрыв между академическим образованием и требованиями индустрии. Реализация данного проекта открывает новые возможности для онлайн-обучения, делая его более персонализированным, практико-ориентированным и соответствующим актуальным запросам рынка труда.

СПИСОК ЛИТЕРАТУРЫ

1. Гагарина Л.Г., Киселев Д.В., Федотова Е.Л. Проектирование информационных систем. — М.: ФОРУМ: ИНФРА-М, 2020. — 320 с.
2. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем. — М.: Юрайт, 2023. — 456 с.
3. Иванова Г.С. Проектирование информационных систем: учебник. — М.: Академия, 2022. — 278 с.
4. Кузнецова Е.П. Цифровые технологии в образовании: психолого-педагогические аспекты. — М.: Юрайт, 2023. — 215 с.
5. Леоненков А.В. Самоучитель UML 2. — СПб.: БХВ-Петербург, 2019. — 480 с.
6. Норман Д. Дизайн привычных вещей. — М.: Вильямс, 2020. — 384 с.
7. Петров В.И. Современные образовательные платформы: архитектура и технологии // Информационные технологии. — 2023. — №4. — С. 34–41.
8. Райс Э., Траут Дж. Позиционирование: битва за узнаваемость. — М.: Манн, Иванов и Фербер, 2021. — 320 с.
9. Сидоров А.К. Проектирование высоконагруженных систем. — СПб.: Питер, 2021. — 412 с.
10. Смирнова Е.А. Профессиональное обучение в цифровую эпоху. — СПб.: Питер, 2023. — 210 с.
11. Anderson C. Free: The Future of a Radical Price. — Random House, 2009. — 288 p.
12. Bishop C.M. Pattern Recognition and Machine Learning. — Springer, 2021. — 738 p.
13. Brown T. et al. Language Models are Few-Shot Learners // arXiv, 2022. — URL: <https://arxiv.org>

14. Chaffey D., Ellis-Chadwick F. Digital Marketing. — 7th ed. — Pearson, 2022. — 800 p.
15. Clark D., Angert B. Corporate Learning Trends 2023 // Harvard Business Review. — 2023.
16. Deming D.J., Noray K. Earnings Dynamics, Changing Job Skills, and STEM Careers // Quarterly Journal of Economics. — 2020. — Vol. 135, No. 4. — P. 1965–2005.
17. Deterding S. Gamification in Education: A Systematic Review. — 2021.
18. Godin S. Permission Marketing: Turning Strangers Into Friends And Friends Into Customers. — Simon & Schuster, 1999. — 256 p.
19. Johnson M. Gamification in Learning. — Springer, 2022. — 189 p.
20. Keller K.L. Strategic Brand Management. — 5th ed. — Pearson, 2019.
21. Kotler P., Keller K.L. Marketing Management. — 16th ed. — Pearson, 2021.
22. Milgram P. et al. Augmented Reality // Presence, 2021. — Vol. 30, No. 4. — P. 45–67.
23. Osterwalder A., Pigneur Y. Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. — John Wiley & Sons, 2010.
24. Porter M.E. Competitive Strategy. — NY: Free Press, 2020. — 432 p.
25. Prensky M. Digital Natives, Digital Immigrants. — 2020.
26. Pulizzi J. Content Inc.: How Entrepreneurs Use Content to Build Massive Audiences. — McGraw-Hill, 2015.
27. Ries E. The Lean Startup. — Crown Business, 2011.
28. Smith J. AI in Education: 2023 Review // EdTech Journal, 2023. — Vol. 12, No. 3. — P. 45–67.
29. Coursera. Data Science Programs Benchmark 2023. — URL: <https://www.coursera.org>
30. EdTechXGlobal. Future of Education Technology Report. — 2024. — URL: <https://www.edtechxglobal.com>
31. Gartner. Emerging Technologies in Education. — 2023.

32. Gartner. Top 10 Strategic Technology Trends for 2024. — URL: <https://www.gartner.com>
33. GeekBrains. Анализ потребностей начинающих специалистов в IT. — М., 2022.
34. Global AI Talent Report 2023. LinkedIn Economic Graph. — URL: <https://economicgraph.linkedin.com>
35. Harvard Business Review. Advanced Data Science Education Trends. — 2023. — Vol. 101, №3.
36. HolonIQ. Global Education Technology Report 2023. — URL: <https://www.holoniq.com>
37. McKinsey & Company. The Future of Work in Data Science. — 2022.
38. MIT Media Lab. Blockcerts: Blockchain for Education. — 2022. — URL: <https://www.media.mit.edu>
39. НИУ ВШЭ. Исследование поведения пользователей EdTech-платформ. — М., 2023.
40. Отчет "Global EdTech Market 2023", HolonIQ. — URL: <https://www.holoniq.com>
41. Отчет о зарплатах в IT. Хабр-карьера, 2024. — URL: <https://career.habr.com/salaries>
42. Skyeng. Алгоритмы адаптивного обучения в платформе Vimbox. — 2023. — URL: <https://skyeng.ru>
43. Stack Overflow. Global Developer Survey 2023. — URL: <https://stackoverflow.com>
44. Statista. Average Time Spent on Social Media in 2024. — URL: <https://www.statista.com>
45. Stepik. Adaptive Learning Technologies Report. — 2023. — URL: <https://stepik.org>
46. World Economic Forum. Schools of the Future: Defining New Models of Education. — 2021.

47. Яндекс. Технологии машинного обучения в образовательных платформах. — 2023. — URL: <https://yandex.ru/company/researches>
48. Codeforces: Competitive Programming. — URL: <https://codeforces.com>
49. HackerRank Developer Skills. — URL: <https://www.hackerrank.com>
50. LeetCode Official Website. — URL: <https://leetcode.com>

ПРИЛОЖЕНИЕ А СЦЕНАРИИ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

Главный раздел сценария выполнения варианта использования "Решение задачи"	
Вариант использования	Решение задачи
Актеры	Ученик
Краткое описание	Ученик находится в карточке задания, видит описание, может начать решение, отправить решение и узнать корректность решения
Цель	Решить задачу
Тип	Базовый
Ссылки на другие варианты использования	Расширяется: Выбор задания, Комментирование задачи
Типичный ход событий сценария выполнения варианта использования "Решение задачи"	
Действия актера	Отклик системы
1. Пользователь вписывает ответ на задачу в специальный виджет	
2. Пользователь решает проверить вывод написанного им кода и нажимает кнопку запустить	
	3. Система принимает ответ пользователя
	4. Система отправляет код пользователя в интерпретатор
	5. Система получает ответ от интерпретатора
	6. Система возвращает пользователю ответ интерпретатора
7. Пользователь видит, что ответ на задачу не совпадает с его ответом и корректирует код	
8. Пользователь вновь отправляет код на обработку	
	9. Система принимает ответ пользователя
	10. Система отправляет код пользователя в интерпретатор
	11. Система получает ответ от интерпретатора
	12. Система возвращает пользователю ответ интерпретатора
13. Пользователь видит, что ответ на задание совпадает с его ответом	

14. Пользователь нажимает кнопку отправить решение	
	15. Система принимает ответ пользователя
	16. Система отправляет код пользователя в интерпретатор
	17. Интерпретатор прогоняет через код пользователя несколько тестов
	исключение 1: код не прошел все тесты
	18. Система получает ответ от интерпретатора о том, что код успешно прошел все тесты
	19. Система уведомляет пользователя о том, что решение верно
	20. Система отправляет запись в бд о решенном задании
	исключение 2: нет ответа от БД
Исключения сценария выполнения варианта использования "Выбор задания"	
Действия актеров	Отклик системы
Исключение 1: код не прошел все тесты	
	17.1 Система получает ответ от интерпретатора о том, что код не прошел некоторые из тестов
	17.2 Система уведомляет пользователя о том, что решение неверно
Исключение 2: нет ответа от БД	
	20.1 Система не получает ответа от БД, значит транзакция записи не завершена
	20.2 Система заносит запись в специальную очередь, из которой эта запись позже попадет в БД автоматически
	20.3 Система уведомляет пользователя о том, что при отправке запроса в БД возникла ошибка и ответ пользователя запишется в БД чуть позже

Таблица А.1 — Вариант использования «Решение задачи»

Главный раздел сценария выполнения варианта использования "Решение премиум задачи"	
Вариант использования	Решение задачи
Актеры	Ученик
Краткое описание	Ученик находится в карточке задания, видит описание, может начать решение, отправить решение и узнать корректность решения
Цель	Решить задачу
Тип	Базовый
Ссылки на другие варианты использования	Расширяется: Выбор задания, Комментирование задачи
Типичный ход событий сценария выполнения варианта использования "Решение премиум задачи"	
Действия актера	Отклик системы
	1. Система проверяет наличие платной подписки у пользователя
	исключение 1: у пользователя отсутствует платная подписка
2. Пользователь вписывает ответ на задачу в специальный виджет	
3. Пользователь решает проверить вывод написанного им кода и нажимает кнопку запустить	
	4. Система принимает ответ пользователя
	5. Система отправляет код пользователя в интерпретатор
	6. Система получает ответ от интерпретатора
	7. Система возвращает пользователю ответ интерпретатора
8. Пользователь видит, что ответ на задачу не совпадает с его ответом и корректирует код	
9. Пользователь вновь отправляет код на обработку	
	10. Система принимает ответ пользователя
	11. Система отправляет код пользователя в интерпретатор

	12. Система получает ответ от интерпретатора
	13. Система возвращает пользователю ответ интерпретатора
14. Пользователь видит, что ответ на задание совпадает с его ответом	
15. Пользователь нажимает кнопку отправить решение	
	16. Система принимает ответ пользователя
	17. Система отправляет код пользователя в интерпретатор
	18. Интерпретатор прогоняет через код пользователя несколько тестов
	исключение 2: код не прошел все тесты
	19. Система получает ответ от интерпретатора о том, что код успешно прошел все тесты
	20. Система уведомляет пользователя о том, что решение верно
	21. Система отправляет запись в бд о решенном задании
	исключение 3: нет ответа от БД
Исключения сценария выполнения варианта использования "Выбор задания"	
Действия актеров	Отклик системы
Исключение 1: у пользователя отсутствует платная подписка	
	1.1 Система не дает доступ пользователю к решению задачи и предлагает оформить подписку
1.2 Пользователь нажимает на кнопку "Купить подписку"	
исключение 1.1 пользователь нажимает кнопку "отмена"	
	1.3 Система перенаправляет пользователя на страницу оплаты
Исключение 1.1: пользователь нажимает кнопку "отмена"	
	1.1 Система возвращает пользователя к просмотру задач
Исключение 2: код не прошел все тесты	

	18.1 Система получает ответ от интерпретатора о том, что код не прошел некоторые из тестов
	18.2 Система уведомляет пользователя о том, что решение неверно
Исключение 3: нет ответа от БД	
	21.1 Система не получает ответа от БД, значит транзакция записи не завершена
	21.2 Система заносит запись в специальную очередь, из которой эта запись позже попадет в БД автоматически
	21.3 Система уведомляет пользователя о том, что при отправке запроса в БД возникла ошибка и ответ пользователя запишется в БД чуть позже

Таблица А.2 — Вариант использования «Решение премиум задачи»

Главный раздел сценария выполнения варианта использования "Изменение данных в ЛК"	
Вариант использования	Изменение данных в ЛК
Актеры	Ученик
Краткое описание	Пользователь переходит в личный профиль и может добавить, удалить или изменить информацию о себе
Цель	Удалить, добавить или изменить какие-либо данные в ЛК
Тип	Зависимый
Ссылки на другие варианты использования	Расширяет: Просмотр ЛК
Типичный ход событий сценария выполнения варианта использования "Изменение данных в ЛК"	
Действия актера	Отклик системы
1. Ученик, находясь в ЛК, кликает на кнопку "Редактировать профиль"	
	2. Система открывает интерфейс изменения профиля, в котором ученик может вносить изменения
3. Ученик меняет нужную ему информацию и нажимает кнопку "Сохранить"	

	исключение 1: Нажата кнопка "Отмена" или пользователь вышел с сайта, не сохранив изменения
	4 Система сохраняет новую информацию, отправляя запрос в БД и внося изменения
	исключение 2: Нет ответа то БД
	5. Система заново запрашивает данные о пользователи, чтобы отобразить обновленные данные
	исключение 2: Нет ответа то БД
	6. Система отображает пользователю его профиль
Исключения сценария выполнения варианта использования "Выбор задания"	
Действия актеров	Отклик системы
Исключение 1: Нажата кнопка "Отмена" или пользователь вышел с сайта, не сохранив изменения	
2.1 Ученик нажал кнопку "Отмена" или вышел из редактора любым другим способом	
	2.2 Система не сохраняет введенные изменения, если таковые были
Исключение 2: Нет ответа то БД	
	2.1 Система не получает ответа от БД
	2.2 Система просит пользователя повторить запрос позже

Таблица А.3 — Вариант использования «Изменение данных в ЛК»

Главный раздел сценария выполнения варианта использования "Подключение подписки"	
Вариант использования	Подключение подписки
Актеры	Ученик, Банк
Краткое описание	Приобрести платную подписку и получить доступ к премиум контенту
Цель	Получить доступ к премиум контенту

Тип	Базовый
Ссылки на другие варианты использования	Расширяет: Решение премиум задачи
Типичный ход событий сценария выполнения варианта использования "Подключение подписки"	
Действия актера	Отклик системы
1. Пользователь переходит на страницу выбора тарифа подписки	
2. Пользователь выбирает тариф и нажимает кнопку "купить"	
	3. Система получает данные о пользователе и его выбор
	4. Система направляет пользователя на страницу оплаты (онлайн касса/банк)
	исключение 1: Невозможно установить соединение с онлайн кассой/банком
	5. Система ожидает ответа
	исключение 2: Транзакция отклонена
	6. Система получает положительный ответ
	7. Система направляет чек на почту пользователя (Опционально, если не предусмотрено онлайн кассой/банком)
	8. Система отправляет sql запрос в бд, для того, чтобы отметить, что пользователь имеет доступ к премиум контенту
	исключение 3: Нет ответа от БД
	9. Система уведомляет пользователя о том, что оплата прошла успешно
Исключения сценария выполнения варианта использования "Подключение подписки"	
Исключение 1: Невозможно установить соединение с онлайн кассой/банком	
Действия актеров	Отклик системы
	4.1 Система не может установить соединение с онлайн кассой/банком

	4.2 Система уведомляет пользователя о том, что сервис оплаты временно не доступен и просит повторить попытку позже, а если ситуация повторится, то обратиться в тех. поддержку
Исключение 2: Транзакция отклонена	
	5.1 Система получает ответ о том, что транзакция отклонена
	5.2 Система уведомляет пользователя о том, что его транзакция отклонена. Если присутствует информация об ошибке, то выводит ее пользователю (Например, недостаточно денежных средств и т.д.)
Исключение 3: Нет ответа от БД	
	8.1 Система не может получить ответ от БД
	8.2 Система уведомляет пользователя о том, что оплата прошла успешно, но подписка активируется в течении некоторого времени. Если подписка не активируется - просьба обратиться в тех. поддержку

Таблица А.4 — Вариант использования «Подключение подписки»

Главный раздел сценария выполнения варианта использования "Восстановление пароля"	
Вариант использования	Восстановление пароля
Актеры	Ученик забыл пароль от аккаунта и пытается его восстановить
Краткое описание	Ученик забыл пароль от аккаунта и пытается его восстановить
Цель	Восстановить пароль
Тип	Зависимый
Ссылки на другие варианты использования	Расширяет: Авторизация

Типичный ход событий сценария выполнения варианта использования "Восстановление пароля"	
Действия актера	Отклик системы
1. Пользователь заполняет форму для смены пароля	
	2. Система проверяет корректность заполненных данных
	исключение 1: Данные не корректны
	3. Система отправляет письмо с подтверждением на почту
	исключение 2: невозможно отправить письмо на почту
4. Пользователь переходит в почту и подтверждает смену пароля	
	5. Система отправляет запрос в БД, на смену пароля
	исключение 3: Нет ответа от БД
Исключения сценария выполнения варианта использования "Отправить сообщение техподдержке"	
Действия актеров	Отклик системы
Исключение 1: Данные не корректны	
	2.1 Система уведомляет пользователя о том, что введенные данные некорректны
Исключение 2: невозможно отправить письмо на почту	
	3.1 Система уведомляет пользователя о том, что не удастся отправить письмо на почту
Исключение 3: Нет ответа от БД	
	5.1 Система уведомляет пользователя о том, что

Таблица А.5 — Вариант использования «Восстановление пароля»

ПРИЛОЖЕНИЕ Б ДИАГРАММЫ ПОСЛЕДОВАТЕЛЬНОСТИ

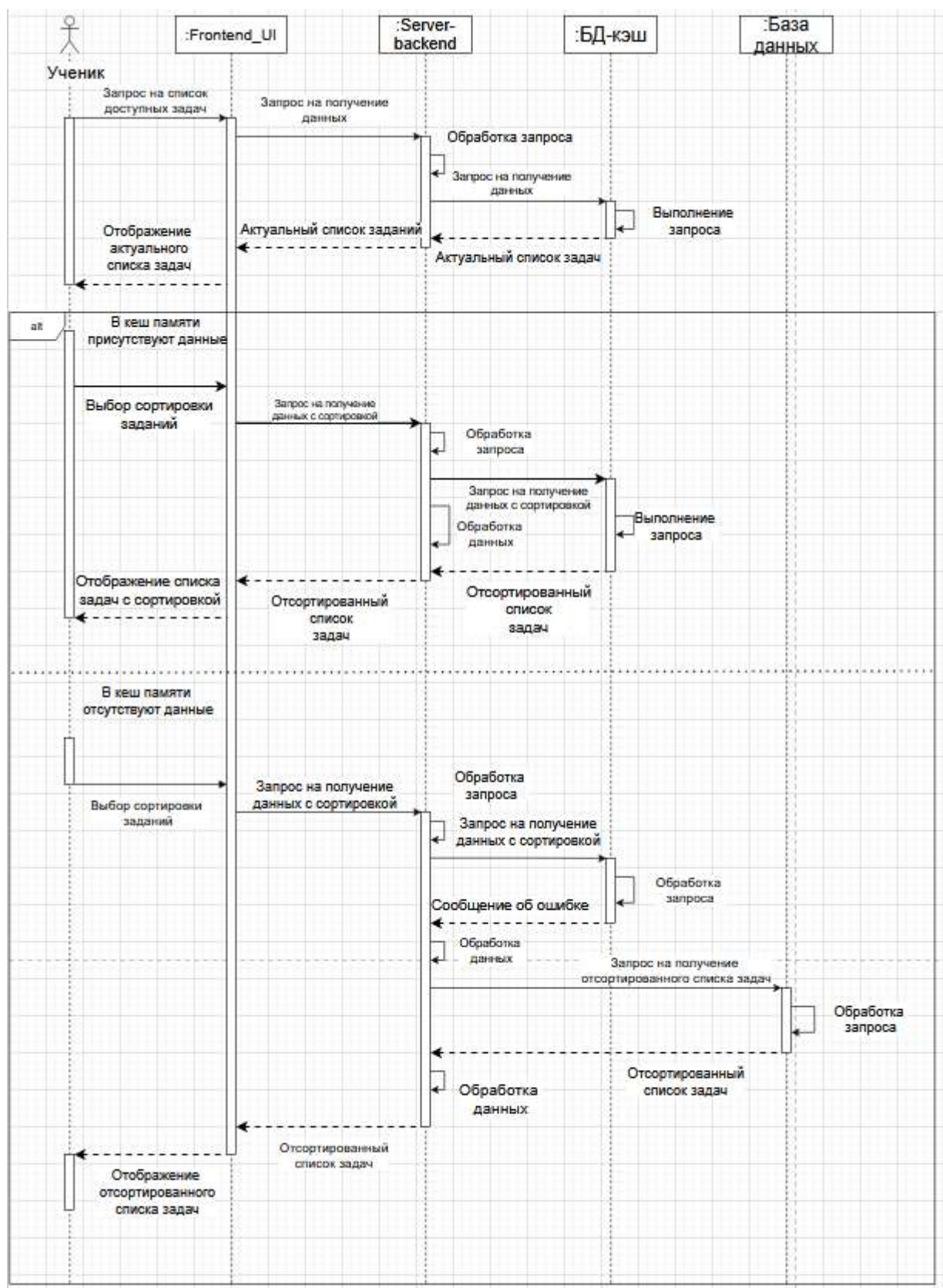


Рисунок Б.1 - Диаграмма последовательности «Просмотр задач»

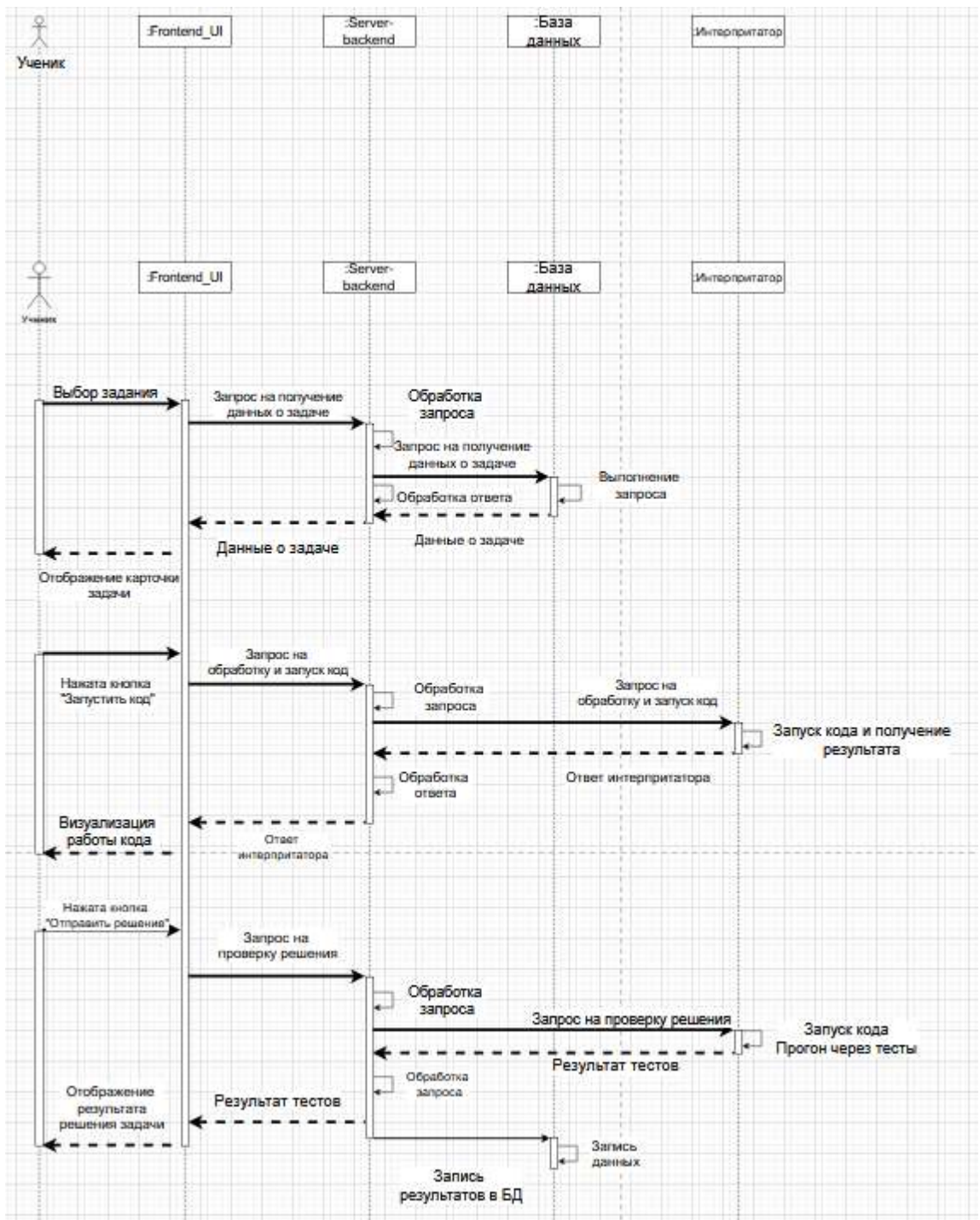


Рисунок Б.2 - Диаграмма последовательности «Решение задач»

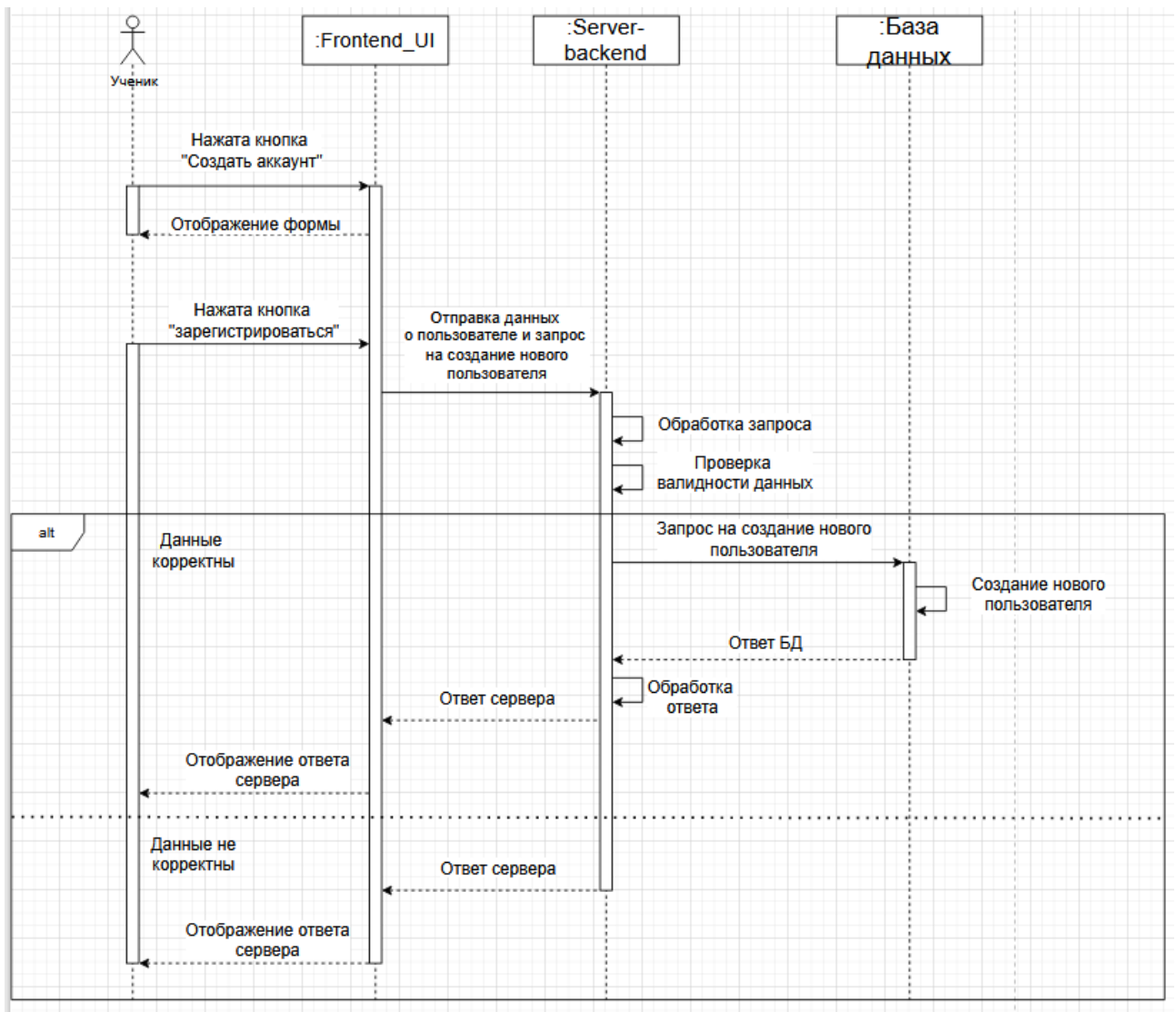


Рисунок Б.3 - Диаграмма последовательности «Регистрация»

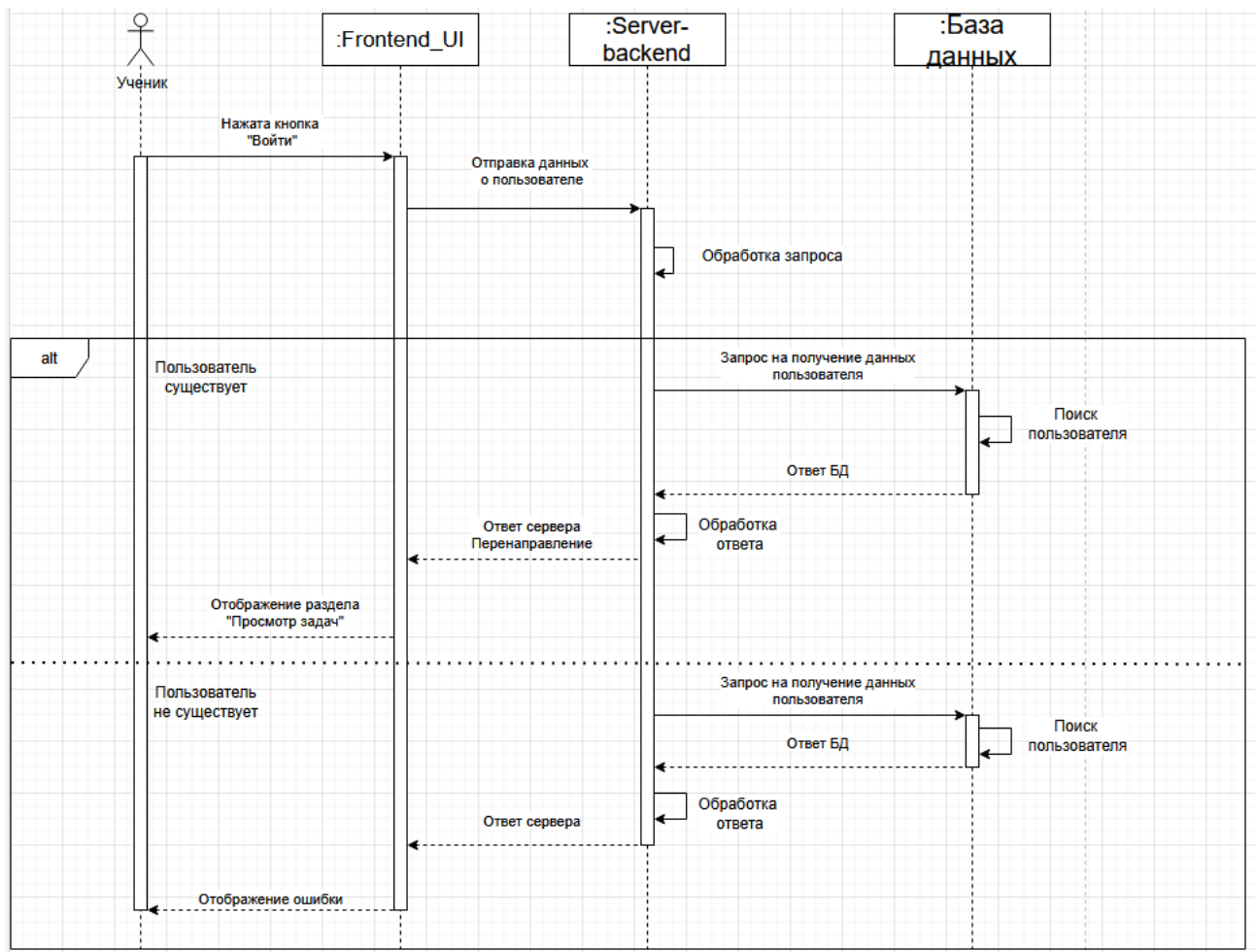


Рисунок Б.4 - Диаграмма последовательности «Авторизация»

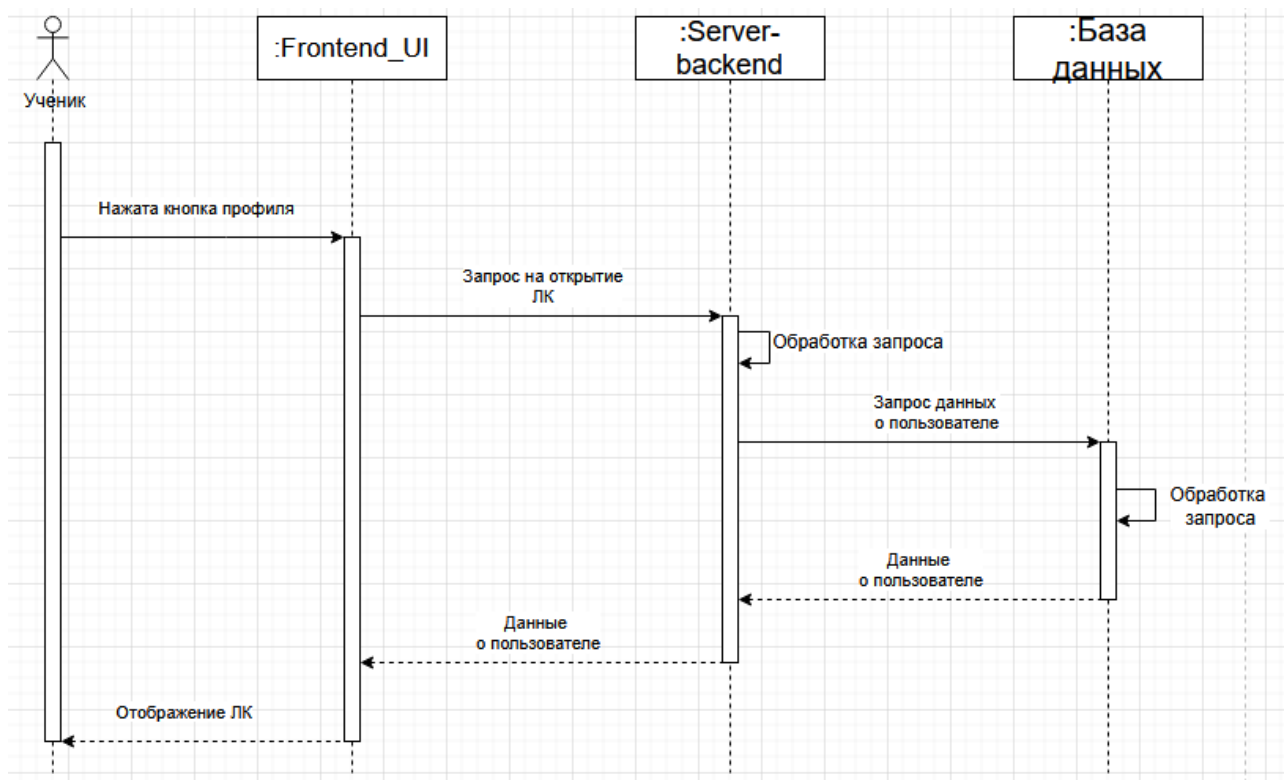


Рисунок Б.5 - Диаграмма последовательности «Просмотр ЛК»

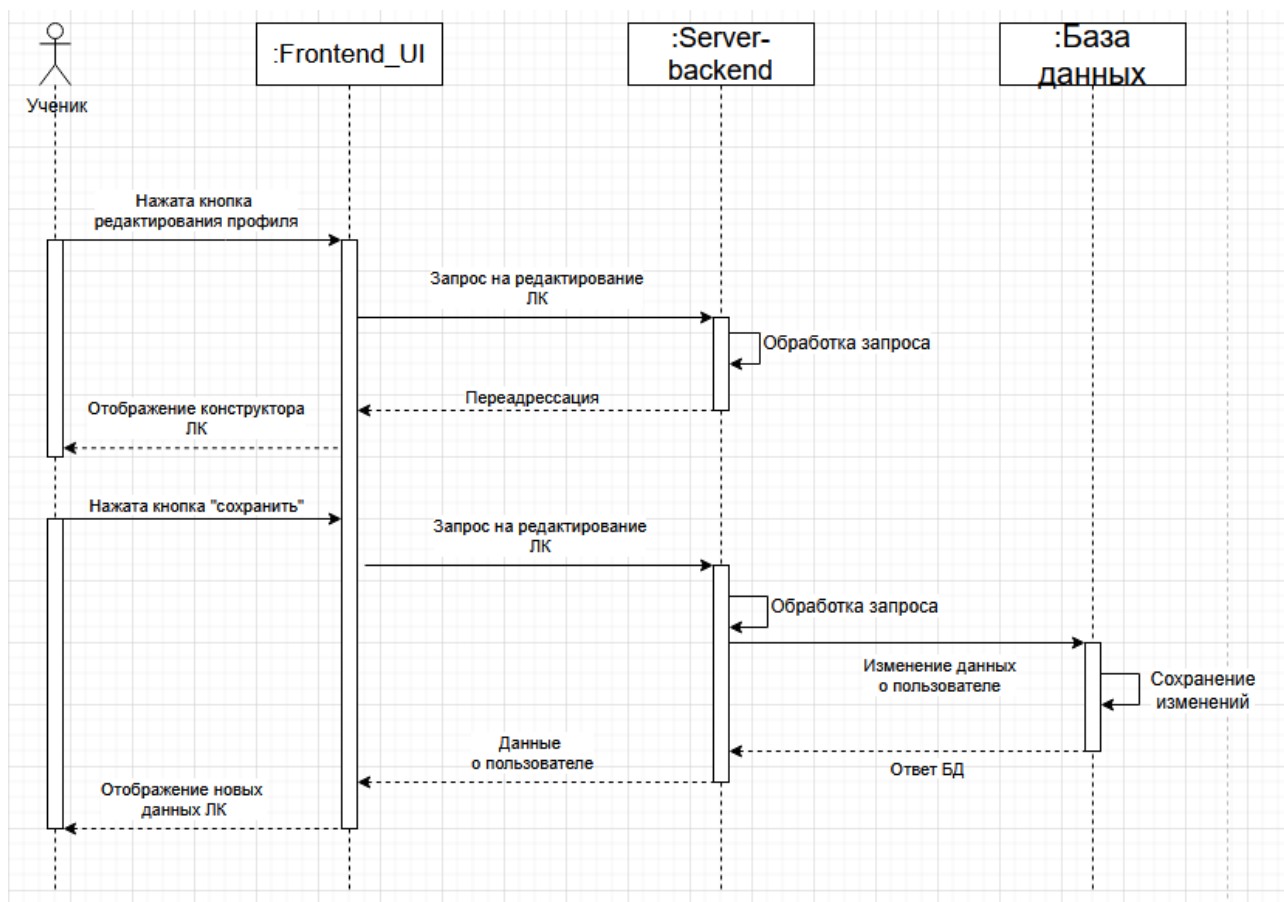


Рисунок Б.6 - Диаграмма последовательности «Изменение ЛК»

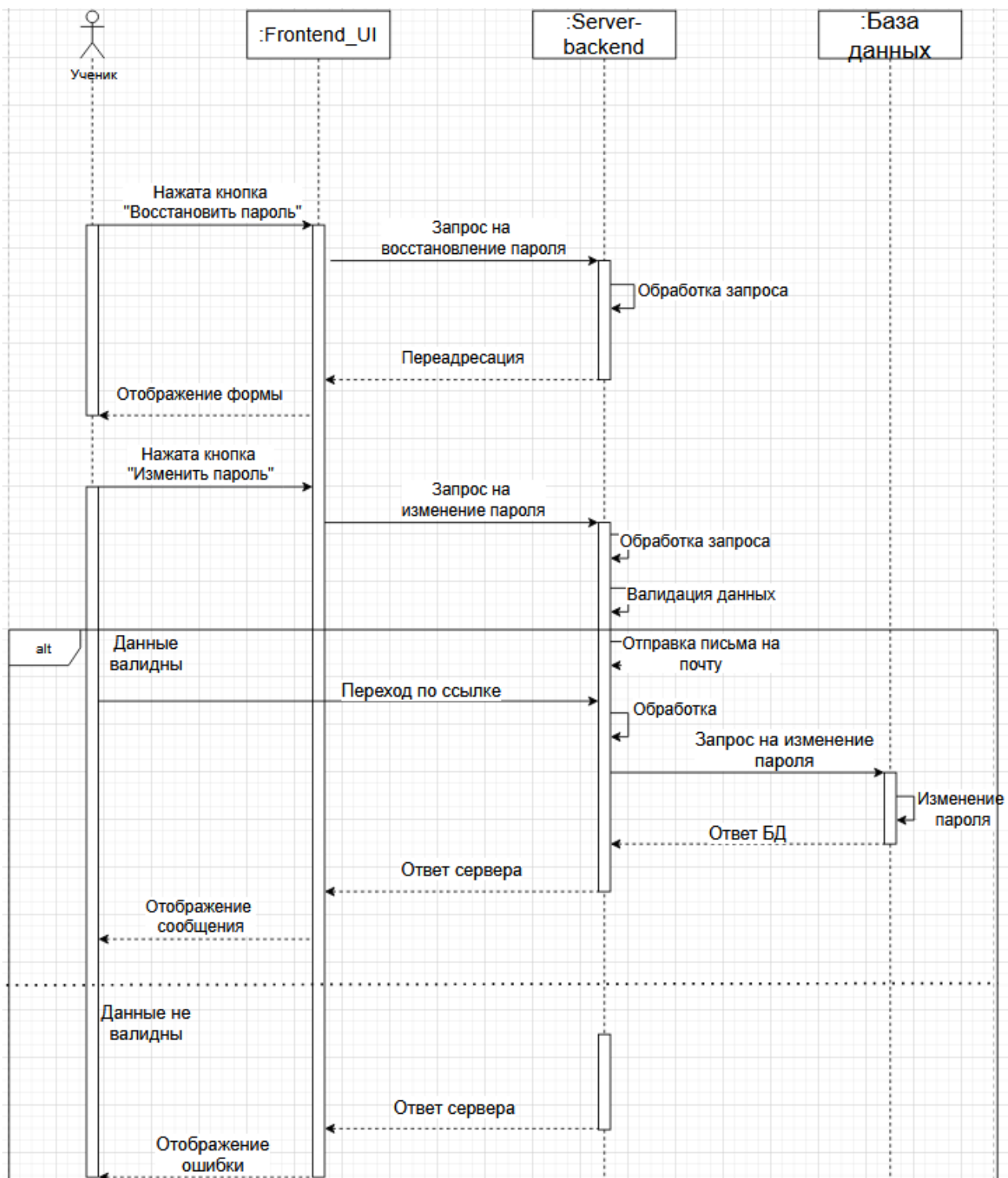


Рисунок Б.7 - Диаграмма последовательности «Восстановление пароля»

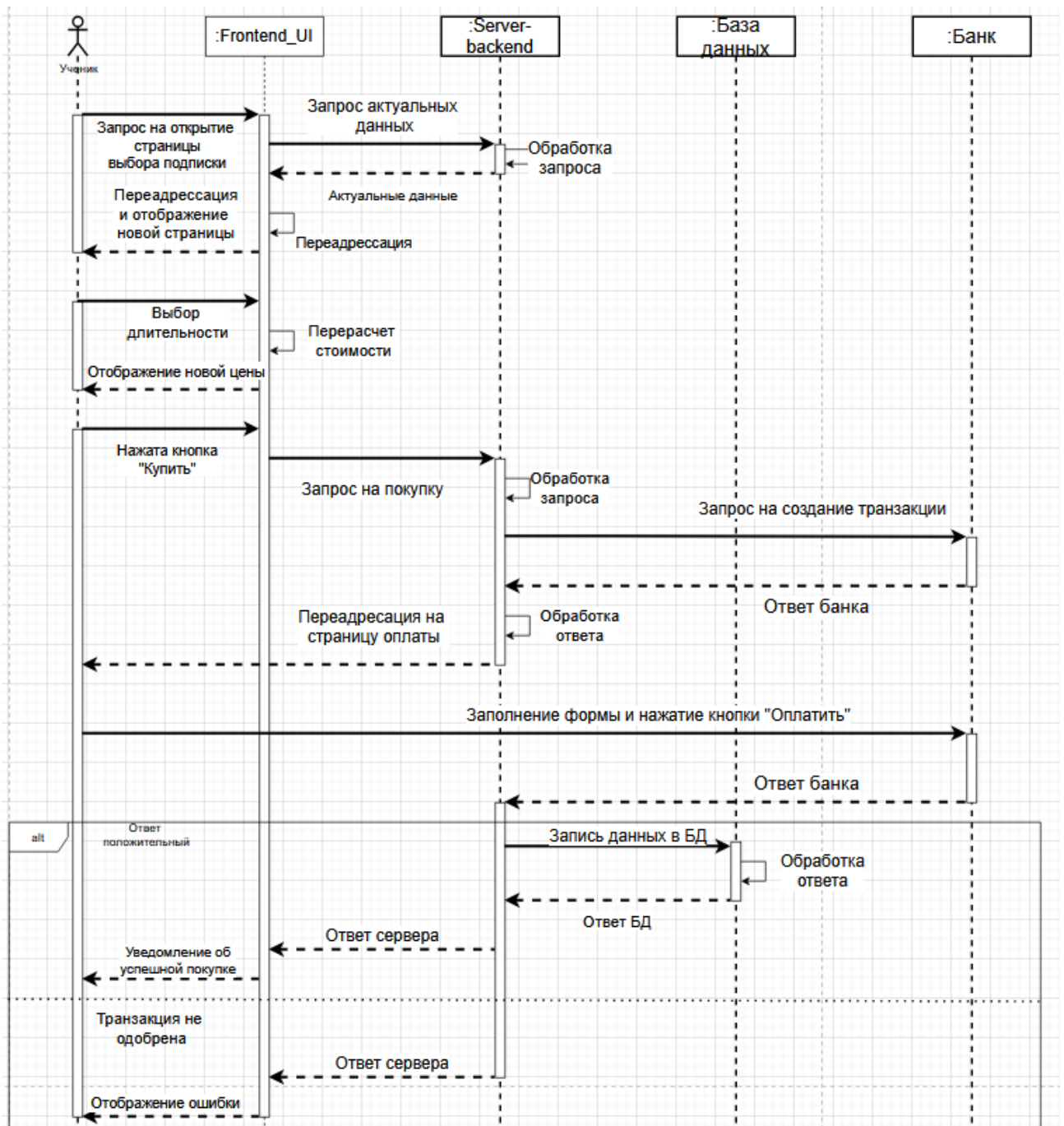


Рисунок Б.8 - Диаграмма последовательности «Подключение подписки»