

Инженерная школа информационных технологий и робототехники  
 Направление: «Информационные системы и технологии»  
 Отделение: «Информационные технологии»

### БАКАЛАВРСКАЯ РАБОТА

Тема работы
Проектирование и реализация микросервисной архитектуры веб-платформы для обучения иностранным языкам

УДК 004.451.9:004.75:004.455.1

Студент

Группа	ФИО	Подпись	Дата
8И7А	Бойко Иван Дмитриевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Демин Антон Юрьевич	Кандидат технических наук		

Консультант

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОИТ	Коровкин Виталий Александрович	-		

### КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Маланина Вероника Анатольевна	Кандидат экономических наук		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ООД	Аверкиев Алексей Анатольевич			

### ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко Ирина Валериевна	Кандидат технических наук		

## РЕЗУЛЬТАТЫ ОБУЧЕНИЯ (КОМПЕТЕНЦИИ ВЫПУСКНИКОВ)

по направлению 09.03.02 «Информационные системы и технологии»

Код компетенции	Наименование компетенции
<b>Универсальные компетенции</b>	
УК(У)-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способен осуществлять деловую коммуникацию в устной и письменной форме на государственном и иностранном (-ых) языке
УК(У)-5	Способен воспринимать межкультурное разнообразие общества в социально-историческом, этническом и философском контекстах
УК(У)-6	Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течении сей жизни
УК(У)-7	Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности
УК(У)-8	Способен создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций
<b>Общепрофессиональные компетенции</b>	
ОПК(У)-1	Владеет широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий
ОПК(У)-2	Способен использовать основные законы естественнонаучных дисциплин в профессиональной деятельности, применять методы математического анализа и моделирования, теоретического и экспериментального исследования
ОПК(У)-3	Способен применять основные приемы и законы создания и чтения чертежей и документации по аппаратным и программным компонентам информационных систем
ОПК(У)-4	Понимает сущность и значения информации в развитии современного информационного общества, соблюдает основные требования к информационной безопасности, в том числе защите государственной тайны
ОПК(У)-5	Способен использовать современные компьютерные технологии поиска информации для решения поставленной задачи, критического анализа этой информации и обоснования принятых идей и подходов к решению
ОПК(У)-6	Способен выбирать и оценивать способ реализации информационных систем и устройств (программно-, аппаратно- или программно-аппаратно-) для решения поставленной задачи
<b>Профессиональные компетенции</b>	
ПК(У)-11	Способен к проектированию базовых и прикладных информационных технологий
ПК(У)-12	Способен разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные)
ПК(У)-13	Способен разрабатывать средства автоматизированного проектирования информационных технологий
ПК(У)-14	Способен использовать знание основных закономерностей функционирования биосферы и принципов рационального природопользования для решения задач профессиональной деятельности
ДПК(У)-1	Способен использовать технологии разработки объектов профессиональной деятельности в бизнесе и осуществлять все виды деятельности в условиях экономики информационного общества.

Инженерная школа информационных технологий и робототехники

Направление: «Информационные системы и технологии»

Отделение: «Информационные технологии»

**УТВЕРЖДАЮ:**  
 Руководитель ООП

\_\_\_\_\_  
 (Подпись)    (Дата)    (Ф.И.О.)

### ЗАДАНИЕ

#### на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8И7А	Бойко Ивану Дмитриевичу

Тема работы:

Проектирование и реализация микросервисной архитектуры веб-платформы для обучения иностранным языкам	
Утверждена приказом директора (дата, номер)	05.02.2021 №36-82/с

Срок сдачи студентом выполненной работы:	11.06.2021
--	------------

### ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p><b>Исходные данные к работе</b></p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Объект исследования: информационная система обучения иностранным языкам, основанная на микросервисной архитектуре</p> <p>Предмет исследования: архитектурные подходы разработки веб-приложений, микросервисная архитектура информационной системы обучения иностранным языкам</p> <p>Режим работы: непрерывный</p>
---	---

<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b></p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> <li>1. Аналитический обзор предметной области</li> <li>2. Проектирование платформы</li> <li>3. Разработка платформы</li> <li>4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</li> <li>5. Социальная ответственность</li> </ol>
<p><b>Перечень графического материала</b></p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<p>Презентация формата *.pptx</p>

<p><b>Консультанты по разделам выпускной квалификационной работы</b></p> <p><i>(с указанием разделов)</i></p>	
<p><b>Раздел</b></p>	<p><b>Консультант</b></p>
<p>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p>Маланина Вероника Анатольевна</p>
<p>Социальная ответственность</p>	<p>Аверкиев Алексей Анатольевич</p>
<p><b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b></p>	
<p> </p>	

<p><b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b></p>	<p> </p>
--	----------

**Задание выдал руководитель / консультант (при наличии):**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Демин Антон Юрьевич	Кандидат технических наук		
Ассистент ОИТ	Коровкин Виталий Александрович	-		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8И7А	Бойко Иван Дмитриевич		

Инженерная школа информационных технологий и робототехники  
 Направление: «Информационные системы и технологии»  
 Отделение: «Информационные технологии»  
 Период выполнения: (осенний / весенний семестр 2020 /2021 учебного года)

Форма представления работы:

бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

### КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	11.06.2021
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	Основная часть	75
	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
	Социальная ответственность	10

**СОСТАВИЛ:**

**Руководитель ВКР**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Демин Антон Юрьевич	Кандидат технических наук		

**СОГЛАСОВАНО:**

**Руководитель ООП**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко Ирина Валерьевна	Кандидат технических наук		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8И7А	Бойко Ивану Дмитриевичу

<b>Школа</b>	ИШИТР	<b>Отделение школы (НОЦ)</b>	ОИТ
<b>Уровень образования</b>	Бакалавриат	<b>Направление/специальность</b>	ИСТ

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Оклад инженера – 21760 руб. Оклад руководителя – 33664 руб.
2. Нормы и нормативы расходования ресурсов	Премимальный коэффициент 30%; Коэффициент доплат и надбавок 20%; Районный коэффициент 30%; Коэффициент дополнительной заработной платы 12%; Накладные расходы 16%.
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Коэффициент отчислений на уплату во внебюджетные фонды 30,2%

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Анализ конкурентных технических решений
2. Планирование и формирование бюджета научных исследований	Формирование плана и графика разработки: -определение структуры и трудоемкости работ; -разработка графика Гантта. Формирование бюджета затрат: -затраты на специальное оборудование; -заработная плата (основная и дополнительная); -отчисления на социальные цели; -накладные расходы.
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Определение потенциального эффекта исследования

**Перечень графического материала (с точным указанием обязательных чертежей):**

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. Альтернативы проведения НИ
4. График проведения и бюджет НИ
5. Оценка ресурсной, финансовой и экономической эффективности НИ

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
доцент ОСГН	Маланина Вероника Анатольевна	К.Э.Н.		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8И7А	Бойко Иван Дмитриевич		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8И7А	Бойко Иван Дмитриевич

Школа	Инженерная школа информационных технологий и робототехники	Отделение (НОЦ)	Отделение информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	Информационные системы и технологии

Тема ВКР:

Проектирование и реализация микросервисной архитектуры веб-платформы для обучения иностранным языкам	
<b>Исходные данные к разделу «Социальная ответственность»:</b>	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объект исследования – информационная система обучения иностранным языкам, основанная на микросервисной архитектуре. Рабочая зона – аудитория с естественным и искусственным освещением, оборудованная системой отопления и кондиционирования воздуха. Область применения – любые люди, желающие изучить иностранный язык.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<b>1. Правовые и организационные вопросы обеспечения безопасности:</b> <ul style="list-style-type: none"> <li>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</li> <li>– организационные мероприятия при компоновке рабочей зоны.</li> </ul>	<p>Требования к организации и оборудованию рабочих мест с ПЭВМ согласно требованиям СанПин 2.2.2/2.4.1340-03.</p> <p>Регулирования организации рабочего места при выполнении работы сидя проводятся согласно ГОСТ 12.2.032-78, а также расположение элементов рабочего места согласно ГОСТ 22269–76.</p> <p>Трудовые отношения регулируются согласно ТК РФ ФЗ–197 от 30.12.2001.</p> <p>Опасные и вредные производственные факторы устанавливаются согласно ГОСТ 12.0.003-2015.</p> <p>Нормы естественного, искусственного и совмещенного освещения зданий устанавливаются согласно СП 52.13330.2016 "Естественное и искусственное освещение".</p> <p>Превышение уровня шума рассматривается согласно СанПиН 2.2.2/2.4.1340–03.</p> <p>Опасность поражения электрическим током по ГОСТ 12.1.038–82 и ГОСТ 12.1.019-2017</p> <p>Технический регламент по ПБ и норм пожарной безопасности (НПБ 105-03) и</p>

	удовлетворять требованиям по предотвращению и тушению пожара по ГОСТ 12.1.004-91 и СНиП 21-01-97. Обращение с отходами рассматривается согласно ГОСТ Р 53692–2009.
<b>2. Производственная безопасность:</b> 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	Вредные факторы: 1. отклонение показателей микроклимата (температуры и влажности воздуха) 2. отсутствие или недостаток необходимого естественного и искусственного освещения 3. превышение уровня шума Опасные факторы: 1. опасность поражения электрическим током
<b>3. Экологическая безопасность:</b>	Анализ негативного воздействия на окружающую природную среду: утилизация компьютеров и другой оргтехники Негативное воздействие на гидросферу и атмосферу совершается посредством производства различной оргтехники Воздействие на литосферу в результате образования отходов при поломке оргтехники
<b>4. Безопасность в чрезвычайных ситуациях:</b>	Возможные ЧС: землетрясение, наводнение, пожар Наиболее типичная ЧС: пожар

Дата выдачи задания для раздела по линейному графику

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Аверкиев Алексей Анатольевич	-		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8И7А	Бойко Иван Дмитриевич		



## РЕФЕРАТ

Учебно-исследовательская работа содержит 85 страниц, 36 рисунков, 17 таблиц и 19 источников.

Ключевые слова: микросервисы, проектирование архитектуры, информационная система, веб-платформа, иностранный язык, обучение.

Данная работа является актуальной ввиду роста онлайн-доли рынка образования и увеличения эффективности применения цифровых инструментов для обучения иностранным языкам.

Объектом исследования является информационная система обучения иностранным языкам, основанная на микросервисной архитектуре.

Предмет исследования: архитектурные подходы разработки веб-приложений, микросервисная архитектура информационной системы обучения иностранным языкам

Были спроектированы веб-сервисы проекта согласно выявленным требованиям и используемой микросервисной архитектуре.

Цель работы – исследование и разработка прототипа микросервисной архитектуры веб-платформы обучения иностранным языкам.

Область применения: сфера дистанционного обучения.

Апробация работы: результаты исследований были представлены на следующих конференциях: Международная научно-практическая конференция студентов, аспирантов и молодых ученых «Молодежь и современные информационные технологии» (г. Томск, 2021).

## ОГЛАВЛЕНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ .....	14
ВВЕДЕНИЕ .....	15
1 АНАЛИТИЧЕСКИЙ ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ .....	17
1.1 Актуальность разработки .....	17
1.2 Анализ предметной области .....	18
1.3 Обзор аналогов разрабатываемой платформы .....	19
1.4 Требования для платформы .....	20
1.4.1 Требования к функциональным характеристикам .....	20
1.4.2 Требования к надёжности .....	21
1.4.2 Требования к составу и параметрам технических средств .....	22
2 ПРОЕКТИРОВАНИЕ .....	23
2.1 Архитектурный подход .....	23
2.1.1 Обзор архитектурных подходов .....	23
2.1.1.1 Монолитная архитектура .....	23
2.1.1.2 Микросервисная архитектура .....	25
2.1.2 Выбор архитектурного подхода .....	27
2.2 Сервисы платформы .....	27
2.2.1 Обнаружение сервисов .....	29
2.2.2 Маршрутизация .....	29
2.2.3 Балансировка нагрузки .....	30
2.3 Автоматизация инфраструктуры .....	32
2.3.1 Контейнеризация сервисов .....	32
2.3.1.1 Оркестрация контейнеров .....	32
2.3.2 Непрерывная интеграция и непрерывное развёртывание .....	33

2.4 Система управления базами данных .....	34
2.5 Пользовательский интерфейс .....	36
3 РЕАЛИЗАЦИЯ .....	41
3.1 Используемые технологии .....	41
3.1.1 Веб-инфраструктура Spring MVC .....	41
3.1.2 СУБД .....	42
3.1.3 CI/CD .....	43
3.1.4 Технологии JS .....	43
3.2 Используемые средства разработки .....	43
3.2 Взаимодействие архитектуры .....	44
3.3 Базы данных .....	45
3.4 Контроллеры .....	47
3.5 Объекты передачи данных .....	48
3.6 REST API и запросы .....	50
3.7 Реализация пользовательского интерфейса .....	52
3.8 Реализация авторизации при помощи SSO Keycloak .....	54
4 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ .....	57
4.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения .....	57
4.1.1 Потенциальные потребители результатов исследования .....	57
4.1.2 Анализ конкурентных технических решений .....	57
4.1.3 Технология QuaD .....	58
4.1.4 SWOT-анализ .....	59

4.2 Планирование научно-исследовательских работ.....	61
4.2.1 Структура работ в рамках научного исследования .....	61
4.2.2 Определение трудоемкости выполнения работ .....	61
4.2.3 Разработка графика проведения научного исследования.....	63
4.2.4 Бюджет научно-технического исследования .....	64
4.2.5 Расчет затрат на специальное оборудование для научных (экспериментальных) целей .....	64
4.2.6 Основная заработная плата исполнителей темы.....	65
4.2.7 Дополнительная заработная плата исполнителей темы.....	67
4.2.8 Отчисления во внебюджетные фонды (страховые отчисления) .....	68
4.2.9 Накладные расходы.....	69
4.2.10 Формирование бюджета затрат научно-исследовательского проекта .	69
4.3 Определение потенциального эффекта исследования .....	70
5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ.....	71
5.1 Правовые и организационные вопросы обеспечения безопасности .....	71
5.1.1 Специальные правовые нормы трудового законодательства.....	71
5.1.2 Организационные мероприятия при компоновке рабочей зоны .....	72
5.2 Производственная безопасность.....	74
5.2.1 Вредные производственные факторы .....	75
5.2.1.1 Отклонение показателей микроклимата в помещении .....	75
5.2.1.2 Недостаточная освещенность рабочей зоны .....	76
5.2.1.3 Превышение уровня шума .....	77
5.2.2 Опасные производственные факторы .....	78
5.2.2.1 Опасность поражения электрическим током .....	78
5.3 Экологическая безопасность.....	79

5.4 Безопасность в чрезвычайных ситуациях.....	80
ЗАКЛЮЧЕНИЕ .....	83
СПИСОК ЛИТЕРАТУРЫ.....	84

## **ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

API – Application Programming Interface.

CRUD – Create, Update, Delete (операции создания, обновления и удаления).

MVC – Model View Controller.

MVP – Minimal Value Product.

QuaD – Quality Advisor.

SQL – Structured Query Language.

SWOT – Strengths Weaknesses Opportunities Threats.

БД – база данных.

ГОСТ – государственный стандарт.

ОС – операционная система.

СанПиН – санитарные правила и нормы.

СУБД – система управления базами данных.

ФЗ – федеральный закон.

## ВВЕДЕНИЕ

В современном мире трудно представить жизнь без знания нескольких языков. Кроме тривиальных преимуществ, таких как расширения возможности для путешествий, перспектив карьерного роста и построения новых социальных отношений, знание нескольких языков способствует развитию интеллектуальных способностей, а также раскрытию творческого потенциала и горизонта собственных возможностей.

Данная работа описывает проектирование и реализацию микросервисной архитектуры на примере платформы для обучения иностранным языкам.

Объектом исследования являются архитектурные подходы разработки веб-приложений.

Предметом исследования в данной выпускной квалификационной работе является разработка микросервисной архитектуры веб-платформы изучения иностранным языкам.

Целью данной работы является разработка MVP платформы для изучения иностранного языка, построенной согласно микросервисной архитектуре. Данная платформа обеспечивает возможность обучения иностранным языкам самостоятельно и возможность записи к преподавателю для индивидуальной работы в паре.

Для достижения поставленной цели в квалификационной работе решаются следующие основные задачи:

1. Провести анализ предметной области разработки.
2. Выдвинуть требования к разрабатываемой платформе.
3. Провести проектирование микросервисной архитектуры платформы
4. Разработать базу данных для платформы.
5. Реализовать спроектированные сервисы платформы.
6. Разработать REST API.
7. Реализовать пользовательские интерфейсы.
8. Провести тестирование.

Апробация работы: результаты исследований были представлены на следующих конференциях: Международная научно-практическая конференция студентов, аспирантов и молодых ученых «Молодежь и современные информационные технологии» (г. Томск, 2021).



# 1 АНАЛИТИЧЕСКИЙ ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Актуальность разработки

Спрос на изучение иностранных языков всегда будет высоким. Сегодня знание иностранного языка предоставляет ряд возможностей, которые оказывают существенное влияние на жизнь конкретного человека. Примерами таких возможностей являются эмиграция в другую страну, получение доступа к обширному массиву актуальной информации, международное общение и обмен мнениями. Обучение иностранным языкам как следствие показывает его важность и необходимость для современного человека.

Сегодня в школах и университетах изучаются различные иностранные языки, такие как английский, французский, немецкий, испанский, китайский и многие другие. Несмотря на это, в России английский язык перестал быть обязательным для сдачи на едином государственном экзамене. Отмена экзамена по английскому показывает низкий уровень квалификации учителей в школах и общую неготовность образовательных учреждений качественно обучать учащихся [1]. Данные события указывают на необходимость разработки информационной системы, с помощью которой можно будет самостоятельно изучать иностранные языки.

Существует множество частных языковых школ, которые привлекают к себе клиентов, но в данный момент они не способны полностью удовлетворить спрос. Декабрь 2019 года принёс новые значительные изменения на рынок. Сейчас активно растёт спрос на разные дистанционные услуги, как на самостоятельное обучение, так и на индивидуальные занятия с преподавателем.

Информационные технологии дают возможность полностью переместить школы и курсы обучения английскому языку в онлайн, не теряя при этом эффективности. Онлайн-обучение позволяет сочетать в себе эффективность традиционных школ иностранных языков и удобство использования технологий. Но на данный момент не все онлайн-школы являются такими, и не все удовлетворяют потребности клиентов.

Каждый из двух типов учебной деятельности имеет ряд преимуществ и недостатков, поэтому важно их объединить: самостоятельный и с индивидуально с преподавателем.

Таким образом, задача предоставления обучающимся возможности совмещения на единой веб-платформе двух вышеуказанных типов учебной деятельности, получая все преимущества самостоятельных и индивидуальных занятий является актуальной и высоко востребованной на рынке.

## **1.2 Анализ предметной области**

Для анализа предметной области рассмотрены научные статьи при помощи ресурса Google Scholar. В ходе анализа выделены следующие ключевые слова:

- microservices;
- information platform;
- foreign languages;
- learning and teaching platform;
- e-learning platform;
- system architecture.

Исследования подтверждают, что в России существует устойчивый интерес к онлайн-обучению иностранным языкам. При этом одной из основных проблем является то, что люди могут говорить на своем родном языке, но при этом они не обладают навыками или возможностью обучения других людей [2].

Знание иностранного языка стало решающим фактором для получения информации. Изучение лексики и пополнение словарного запаса в целом является основополагающей частью для изучения иностранного языка. Каждый день изобретаются новые устройства для удовлетворения потребностей пользователей. Активный рост использование мобильных телефонов сделало их популярными не только для общения и развлечений, но и для обучения. Мобильные телефоны предоставляют особое преимущество в процессе обучения: они дают возможность изучения за пределами дома в

любое время и в любом месте. Результаты показали, что использование мобильных телефонов в качестве инструмента изучения лексики более эффективно, чем один из традиционных инструментов изучения лексики (ведение бумажных словарей с переводами). В качестве инструмента для изучения новых слов и закрепления уже изученных применялись «карточки», что доказало их эффективность по сравнению с традиционным изучением слов на бумаге [3].

### **1.3 Обзор аналогов разрабатываемой платформы**

Сегодня существует множество аналогов разрабатываемой платформы. Среди наиболее популярных конкурентов можно выделить следующие платформы:

- Инглекс.
- Skyeng.
- Duolingo.
- PuzzleEnglish.
- Lingualeo.

Для проведения сравнительного анализа платформ были выделены следующие критерии:

- Индивидуальные занятия с преподавателем – возможность индивидуального подхода в обучении с применением технологий удалённого общения (Skype, Zoom, собственные технологии) один на один с преподавателем.
- Онлайн курсы – как правило, это курсы, которые содержат в себе материалы на отдельные темы в виде ряда видеоматериалов и вспомогательных к ним текстовых объяснений с заданиями в виде тестов.
- База уроков-упражнений для самостоятельного изучения. Под данной базой понимается применение различной функциональности, направленной на полное самостоятельное обучение. К таким могут относиться теоретические материалы, изучение выбранных слов, прослушивание аудио и последующее прохождение заранее подготовленных тестов и т. д.

– Индивидуальный учебный план – это персональная программа обучения, которая составляется в соответствии с уровнем пользователя, целями и желаемым временем.

На основании приведенных выше критериев, была сформирована таблица после сравнительного анализа аналогов разрабатываемой платформы, которая представлена в таблице 1.

Таблица 1 – Сравнительный анализ функциональных возможностей

	Инглекс	Skyeng	Duolingo	PuzzleEnglish	Lingualeo
Индивидуальные занятия с преподавателем	+	+	–	–	–
Онлайн курсы	–	–	+	+	+
База уроков-упражнений для самостоятельного изучения	±	±	+	+	+
Индивидуальный учебный план	±	±	–	+	+

Каждая из сравниваемых платформ имеет свои достоинства и недостатки, но ни одна платформа не предоставляет все функциональные возможности, описанные выше.

Главная цель разрабатываемой платформы предоставить пользователю каждый из указанных критериев.

## **1.4 Требования для платформы**

### **1.4.1 Требования к функциональным характеристикам**

Разрабатываемая платформа должна предоставлять возможность регистрации двух видов пользователей: преподавателя и ученика.

Возможности роли пользователей «преподаватель»:

- предоставлять возможность создания урока и/или курса, а также инструменты для их изменения и удаления;
- предоставлять возможность создания своего расписания и инструментов для управления им;
- предоставлять возможность просмотра данных о своих учениках.

Возможности роли пользователей «ученик»:

- просмотр списка преподавателей;
- просмотр подробной информации о преподавателях;
- просмотр списка уроков и курсов преподавателей по отдельности;
- запись на урок, курс;
- просмотр списка уроков и курсов текущего преподавателя;
- изучение выбранных слов в различных форматах (карточки, повторение и т.д.);
- просмотр видео на различных языках;
- прослушивание аудио для повышения уровня аудирования;
- просмотр правил грамматики;
- выполнение тестов для проверки своих знаний.

#### **1.4.2 Требования к надёжности**

Система должна сохранять работоспособность и обеспечивать восстановление своих функций при возникновении следующих аварийных ситуаций:

- При сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы должно происходить после перезапуска ОС и запуска необходимых для работы системных служб/сервисов.
- При ошибках, связанных с работой удалённых машин, для поддержания работы конкретного сервиса, что устраняется посредством перераспределения нагрузки на рабочие узлы системы.
- Выполнение резервного копирования данных средствами СУБД ежедневно для предотвращения потери данных при отказе системы в целом или отдельных компонентов системы.

## **1.4.2 Требования к составу и параметрам технических средств**

Требования к техническому обеспечению серверной части во многом зависят от масштаба пользовательского обхвата системы. Минимальные аппаратные требования к серверам, следующие:

- Процессор – x86-совместимый 2 ГГц процессор.
- Объём оперативной памяти – 4 Гб.
- Дисковое пространство – 50 Гб.
- Сетевой адаптер с пропускной возможностью в 100 Мбит/с.

Минимальные требования к техническому и программному обеспечению клиентской части:

- Процессор – x86-совместимый 2 ГГц процессор.
- Объём оперативной памяти – 2 Гб.
- Сетевой адаптер с пропускной возможностью в 100 Мбит/с.
- Браузер на базе Chromium 65.0 и выше.

## 2 ПРОЕКТИРОВАНИЕ

### 2.1 Архитектурный подход

#### 2.1.1 Обзор архитектурных подходов

Сегодня для разработки интернет-решений активно применяются две модели архитектур: монолит и микросервисная

##### 2.1.1.1 Монолитная архитектура

Монолит означает, что система имеет однородное строение. Монолитное приложение описывает одноуровневую систему, в которой различные компоненты объединены в одну программу на единой платформе. Обычно монолитное приложение состоит из базы данных, клиентского пользовательского интерфейса и серверного приложения. Все части программного обеспечения унифицированы, и все его функции управляются в одном месте. Схема монолитной архитектуры представлена на рисунке 1.

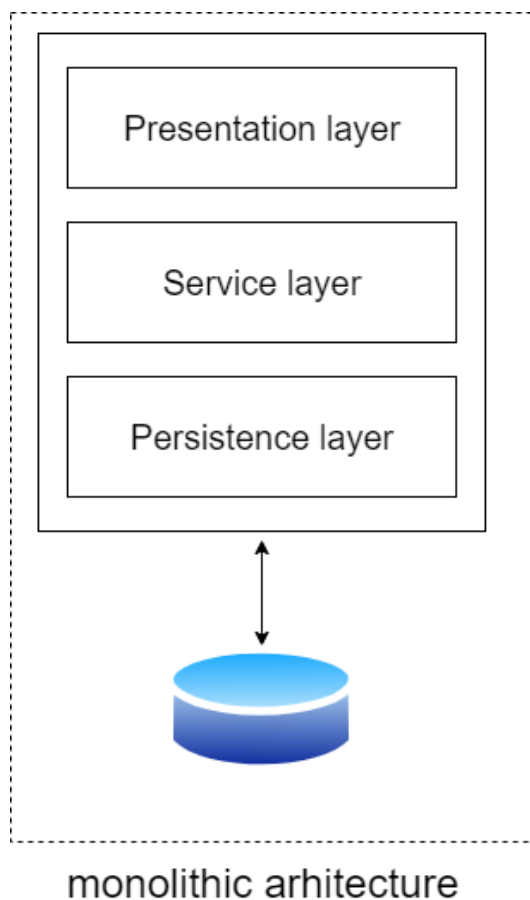


Рисунок 1 – Архитектура монолитной системы

Можно выделить следующие достоинства монолитной архитектуры:

**Упрощённая разработка и развёртывание.** Разработка монолитного приложения относительно проста. Все файлы находятся в одном каталоге, что позволяет быстро находить необходимый компонент и в случае ошибки его исправить. В связи с тем, что при монолитной разработке введётся работа в основном на одном инструменте и разрабатывается на одном языке программирования, не нужно развёртывать изменения или обновления по отдельности.

**Меньше коммуникационных проблем и упрощённое логирование.** Большинство приложений зависят от взаимодействия между компонентами, что усложняет ведение контрольных журналов, логов, ограничивает скорость работы и т. д. Монолитные приложения легче выполняют такие задачи благодаря единой кодовой структуре. К таким приложениям легче добавлять новые компоненты, так как все работает единообразно.

**Лучшая производительность.** Монолитные приложения, предоставляют достаточно быструю связь между компонентами благодаря единой кодовой структуре. Как правило, один запрос со стороны клиента действительно приведёт к единому атомарному действию. Таким образом не будет накладных расходов на взаимодействие с другими частями системы.

Минусами данной архитектуры являются следующие:

**Кодовая база со временем становится громоздкой.** Сложнее внести изменения в такое большое и сложное приложение с очень тесной связью его компонентов. Любое изменение в коде влияет на всю систему, поэтому его необходимо тщательно координировать. Это значительно увеличивает длительность процесса разработки.

**Ограниченная гибкость.** В монолитных приложениях каждое небольшое обновление требует полного повторного развертывания. Таким образом, все разработчики обязаны ждать, пока это не будет сделано. Когда несколько команд работают над одним проектом, гибкость может быть значительно снижена.



Монолитная модель активно используется и не вызывает дополнительных трудностей. Особенно полезной может оказаться такая архитектура для небольших проектов или проектов на начальной стадии разработки, что достаточно важно для проверки новых гипотез в программных продуктах.

### 2.1.1.2 Микросервисная архитектура

Микросервис – это тип сервисно-ориентированной программной архитектуры (SOA), которая ориентирована на создание серии автономных компонентов, составляющих приложение. В отличие от монолитных приложений, построенных как единое неделимое целое, микросервисные приложения состоят из нескольких независимых компонентов, склеенных вместе с помощью API. Сравнение двух архитектур представлено на рисунке 2.

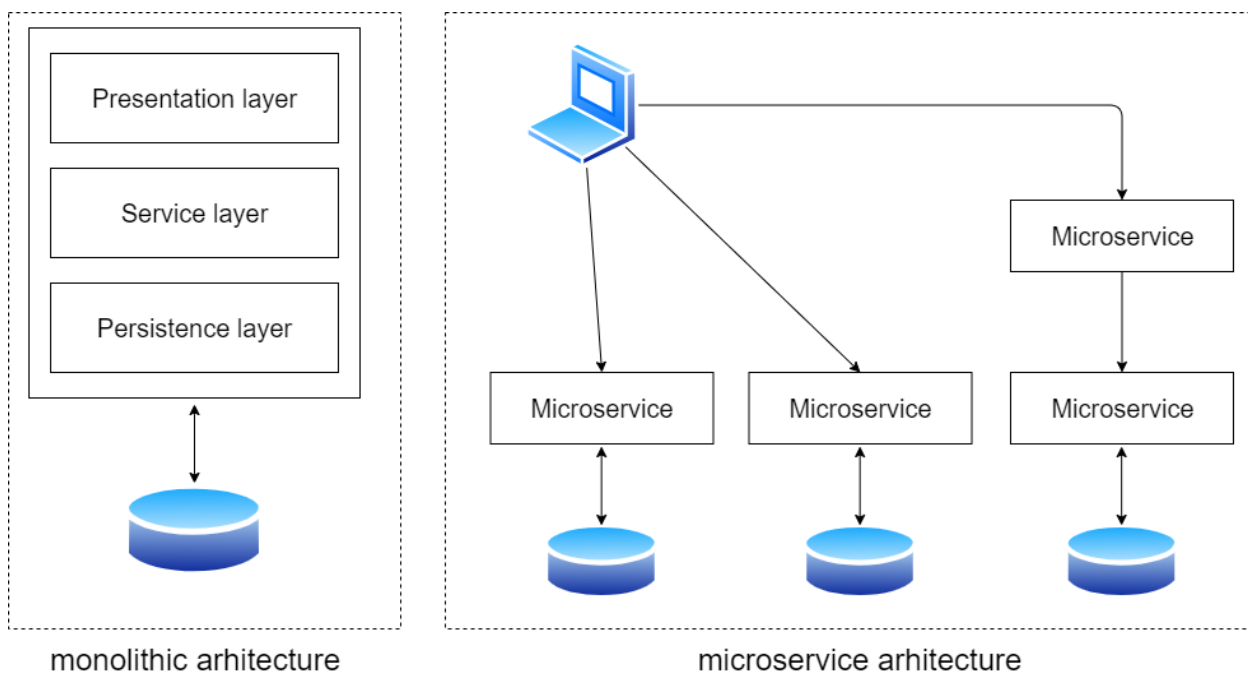


Рисунок 2 – Сравнение архитектурных подходов

Подход микросервисов фокусируется в основном на бизнес-приоритетах и возможностях, тогда как монолитный подход организован вокруг технологических уровней, пользовательских интерфейсов и баз данных. Подход с использованием микросервисов стал популярным,

поскольку всё больше и больше предприятий стараются быстрее выводить свои новые решения на рынок и оптимизировать ресурсы разработки.

К плюсам микросервисной архитектуры можно отнести следующее:

**Легко разрабатывать, тестировать и развертывать.** Самым большим преимуществом микросервисов перед другими типами архитектур является то, что небольшие отдельные сервисы можно создавать, тестировать и развертывать независимо. Поскольку модуль развертывания невелик, он упрощает и ускоряет разработку и выпуск в промышленную среду.

**Гибкость.** Все сервисы можно развёртывать и обновлять независимо, что предоставляет разработчикам большую гибкость. Ошибка в одном микросервисе влияет только на конкретный сервис и не влияет на всё приложение. Кроме того, гораздо легче добавлять новые функции в микросервисное приложение, чем в монолитное.

**Возможность горизонтального масштабирования.** Каждый элемент можно масштабировать независимо. Таким образом, весь процесс более рентабелен и эффективен по времени, чем при использовании монолитов, когда всё приложение необходимо масштабировать, даже если в этом нет необходимости. Кроме того, у каждого монолита есть ограничения с точки зрения масштабируемости, существует ряд ограничений по количеству пользователей.

В минусы данного подхода следует отнести следующие пункты:

**Сложность.** Поскольку архитектура микросервисов является распределенной системой, необходимо выбирать и настраивать соединения между модулями и базами данных. Кроме того, поскольку такое приложение включает наборы отдельных сервисов, все они должны развёртываться независимо друг от друга.

Таким образом, микросервисная архитектура более выгодна для сложных и динамично меняющихся приложений. Она предлагает эффективные решения для управления сложной системой различных функций

и услуг в одном приложении. Микросервисы идеальны, когда речь идет о платформах, охватывающих сложные и связанные бизнес-процессы.

### **2.1.2 Выбор архитектурного подхода**

Платформа для изучения иностранных языков является достаточно сложным приложением, которое требует реализации различных несвязанных между собой функций и услуг. Микросервисы добавляют уникальную ценность за счет упрощения сложности систем. Разбивая приложение на множество более мелких частей, уменьшая дублирование и связи между частями, можно получить более понятное, масштабируемое и изменяемое веб-приложение в целом.

Сегодня организации, стремящиеся к максимальной производительности, гибкости и улучшению качества обслуживания клиентов, выходят за рамки монолитных приложений и используют микросервисы, чьи слабосвязанные архитектуры ускоряют разработку, тестирование и внедрение, тем самым уменьшая операционные расходы на развитие и поддержку веб-системы предприятий [6].

### **2.2 Сервисы платформы**

Сервис должен быть достаточно маленьким, чтобы его могла разработать небольшая команда, а также быстро и легко протестировать. Полезным руководящим принципом объектно-ориентированного проектирования (OOD) является принцип единой ответственности (SRP). SRP определяет ответственность класса как причину изменения и заявляет, что у класса должна быть только одна причина для изменения. Имеет смысл применять SRP также к дизайну сервисов и проектировать их согласно ему.

Приложение также должно быть декомпозировано таким образом, чтобы большинство новых и измененных требований затрагивали только один сервис. Это связано с тем, что изменения, затрагивающие несколько сервисов, требуют координации работы между несколькими командами, что замедляет разработку. Еще один полезный принцип OOD – это общий принцип закрытия (CCP), который гласит, что классы, которые изменяются по одной и той же

причине, должны находиться в одном пакете. Например, два класса реализуют разные аспекты одного и того же бизнес-правила. Цель состоит в том, чтобы разработчикам было необходимо лишь изменить код только в небольшом количестве компонентов. Такой подход имеет смысл при разработке сервисов, поскольку он поможет гарантировать, что каждое изменение повлияет только на один сервис.

Декомпозиция приложения производилась посредством паттерна разделения по бизнес-возможностям. Набор бизнес-возможностей приложения декомпозирован на отдельные сервисы:

- Сервис работы с данными аккаунтов пользователей: хранит в себе данные пользователей, их пароли и тип авторизации.

- Сервис для изучения слов: взаимодействует со словарями пользователей, в которые были добавлены слова для дальнейшего изучения и повторения слов.

- Сервис для занятий учеников с преподавателями: предоставляет возможность записи и e-mail уведомления для каждого участника.

- Сервис, который отвечает за тестирование знаний учеников. Тестирование представляет собой последовательные наборы заданий, которые позволяют проверить себя обучающему и закрепить изученный материал.

Помимо данных сервисов обязательными для функционирования системы являются:

- Сервис авторизации: представляет собой SSO для единой авторизации во всех существующих сервисах, находящийся вместе с сервисом аккаунтов.

- Сервис маршрутизации запросов и балансировки нагрузки между сервисами, а также сервис в себе содержит само веб-приложение.

- Сервис для обнаружения сервисов: обнаруживает рабочие сервисы для сбора информации о них и предоставления информации для Ribbon о работоспособности каждого экземпляра сервисов.

Для организации взаимодействия сервисов между собой за основу был взят стек технологий, представленный компанией Netflix, а именно Spring Cloud Netflix.

Данный проект обеспечивает интеграцию Netflix OSS для приложений Spring Boot через автоконфигурацию и привязку к среде Spring и другим идиомам модели программирования Spring. С его помощью возможно построить большие распределенные системы, а также его использует непосредственно сама компания Netflix. Предоставляемые шаблоны включают в себя обнаружение сервисов (Eureka), прерыватель (Hystrix), интеллектуальную маршрутизацию (Zuul) и балансировку нагрузки на стороне клиента (Ribbon) [5].

### **2.2.1 Обнаружение сервисов**

Eureka Server – это приложение, которое содержит информацию обо всех клиентских сервисных приложениях. Другими словами Eureka Server – это сервер имён или реестр сервисов, который предоставляет имена каждому микросервису.

Каждый микросервис регистрируется на сервере Eureka, и Eureka знает все клиентские приложения, работающие на каждом порту и IP-адресе. Он регистрирует микросервисы и отдает их ip другим микросервисам. Таким образом, каждый сервис, регистрируясь в Eureka, отправляет эхо-запрос серверу, чтобы сообщить, что он активен.

Обнаружение сервисов – один из ключевых принципов архитектуры на основе микросервисов. Попытка вручную настроить каждого клиента или какую-либо форму соглашения может быть трудной и нестабильной. Сервер можно настроить и развернуть для обеспечения высокой доступности, при этом каждый сервер реплицирует состояние зарегистрированных служб на другие.

### **2.2.2 Маршрутизация**

В качестве API Gateway используется Zuul, ответственный за маршрутизацию HTTP-запросов. В основе Zuul лежит серия фильтров,

которые способны выполнять ряд действий во время маршрутизации HTTP-запросов и ответов.

Существует несколько стандартных типов фильтров, соответствующих типичному жизненному циклу запроса. Жизненный цикл запроса представлен на рисунке 3.

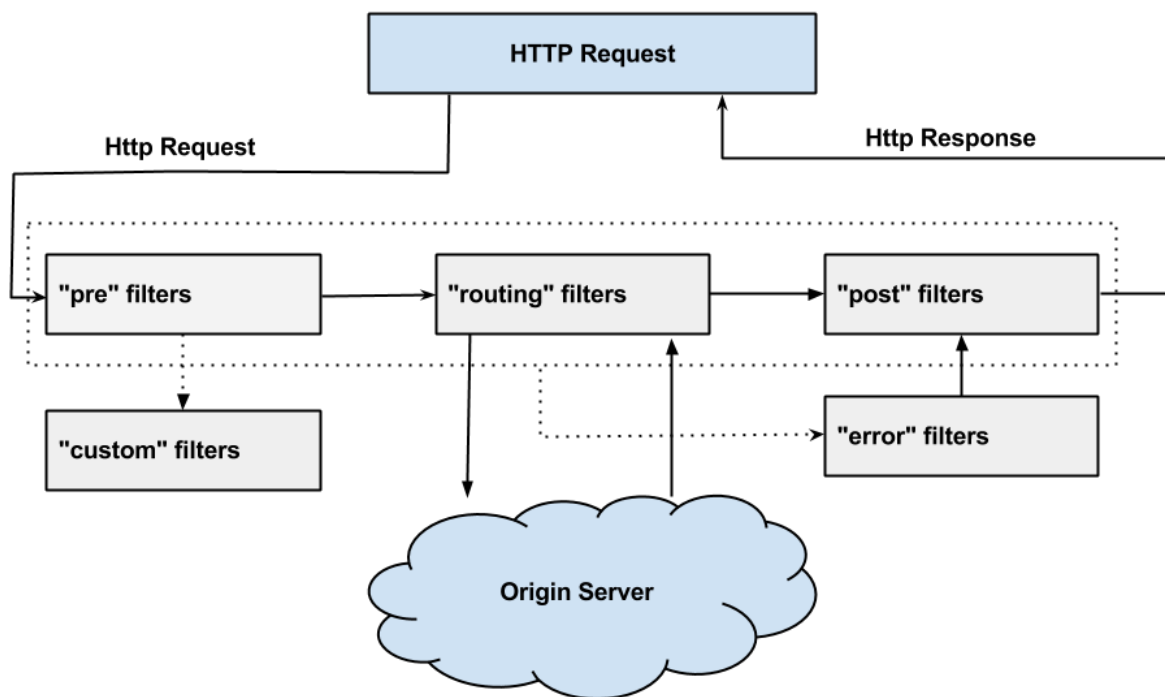


Рисунок 3 – Жизненный цикл запроса

К одному из важных фильтров системы можно отнести аутентификацию пользователей. При помощи фильтра перед получением доступа к защищённому ресурсу Zuul перенаправит запрос на сервер аутентификации для выдачи или проверки JWT-токена.

### 2.2.3 Балансировка нагрузки

Для балансировки нагрузки используется Ribbon. Данный инструмент является компонентом, расположенным на стороне клиента (client-side). Он решает какой сервер будет вызван. Схема работы балансировщика нагрузки на стороне клиента представлена на рисунке 4.

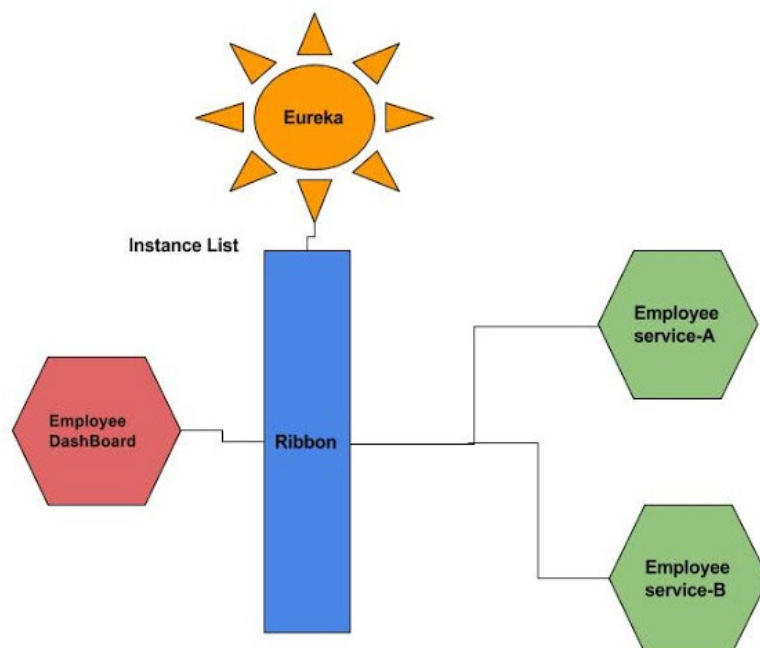


Рисунок 4 – Схема работы балансировщика нагрузки

Ribbon, как и другие балансировщики нагрузки, для отправки запроса получает на вход список серверов, которые могут предоставить необходимую информацию. Сервера на входе фильтруются по доступности и другим критериям перед выбором.

Ключевым критерием, используемым для измерения состояния зоны (по географическому расположению), является среднее количество активных запросов. Это общее количество невыполненных запросов в зоне, разделенное на количество доступных целевых экземпляров. Эта метрика очень эффективна, когда тайм-аут происходит медленно в плохой зоне.

Ribbon рассчитывает и использует статистику всех доступных зон. Если среднее количество активных запросов для любой зоны достигло настроенного порога, эта зона будет удалена из списка активных серверов. Если более чем одна зона достигла порогового значения, то зона с наиболее активными запросами на сервер будет удалена. После того, как худшая зона отброшена, другая зона будет выбрана среди остальных с вероятностью, пропорциональной ее количеству экземпляров. Иными словами, решения о балансировке нагрузки, связанные с каждой зоной, принимаются в реальном времени, а актуальная статистика помогает сделать оптимальный выбор.

## **2.3 Автоматизация инфраструктуры**

### **2.3.1 Контейнеризация сервисов**

Контейнеризация — это подход к разработке программного обеспечения, при котором приложение или служба, их зависимости и конфигурация (абстрактные файлы манифеста развертывания) упаковываются вместе в образ контейнера. Контейнерное приложение может тестироваться как единое целое и развертываться как экземпляр образа контейнера в операционной системе узла.

Контейнеры предоставляют ряд таких преимуществ, как изоляция, переносимость, гибкость, масштабируемость и контроль, на протяжении всего жизненного цикла приложения. Самым важным преимуществом из них является изоляция среды разработки от рабочей среды.

Платформа контейнеризации Docker стала именем нарицательным после её перевода в открытый доступ компанией Google. Docker – это специализированная система, которая позволяет пользователям легко упаковывать, распространять и управлять приложениями в контейнерах. Другими словами, это проект с открытым исходным кодом, который автоматизирует развертывание приложений внутри программных контейнеров.

Docker действительно облегчает создание, развертывание и запуск приложений с использованием контейнеров, а контейнеры позволяют разработчику упаковывать приложение со всеми необходимыми ему частями, такими как библиотеки и другие зависимости, и отправлять все это как один пакет. Таким образом, разработчик может быть уверен, что приложение будет работать на любом другом компьютере под управлением операционной системой GNU/Linux независимо от каких-либо пользовательских настроек, которые может иметь машина.

#### **2.3.1.1 Оркестрация контейнеров**

При реализации микросервисной архитектуры важным элементом является быстрое горизонтальное масштабирование компонентов различных



используемых сервисов. Для развёртывания контейнеров между множеством физических или виртуальных машин используется система оркестрации контейнеров Kubernetes.

### 2.3.2 Непрерывная интеграция и непрерывное развёртывание

Непрерывная интеграция (Continuous Integration, CI) и непрерывная поставка (Continuous Delivery, CD) представляют собой набор принципов и практик, которые позволяют чаще и надежнее развёртывать изменения программного обеспечения.

Процесс CI – это процесс, в котором программное обеспечение разрабатывается из нескольких источников и автоматически интегрируется с помощью установленной процедуры.

Процесс CD – это процесс, который обычно следует за процессом интеграции. Он принимает только что объединенную версию и обычно выполняет правила для развёртывания в производственную среду.

Концепция CI/CD позволяет выпускать новые версии продуктов, как только они будут разработаны для выпуска.

Для применения CI/CD за основу взят инструмент Jenkins с открытым исходным кодом.

Рассмотрим схему применения CI/CD. Схема применения CI/CD представлена на рисунке 5.

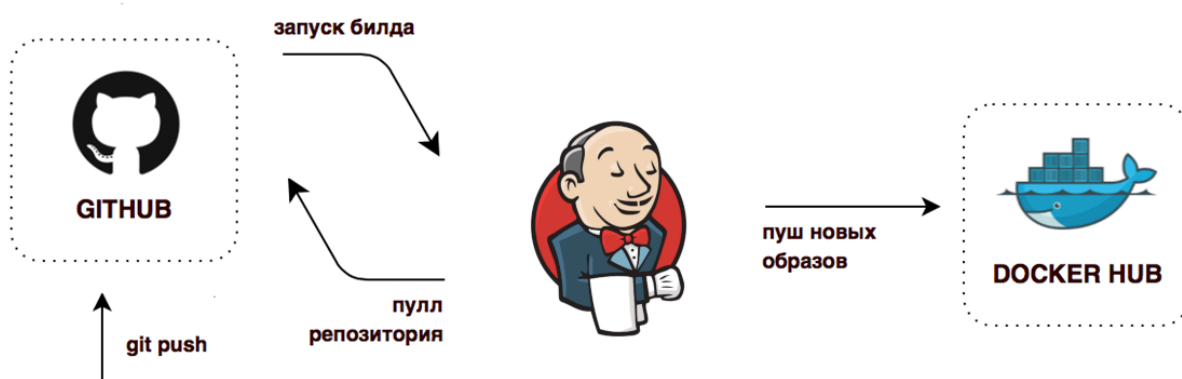


Рисунок 5 – Схема применения CI/CD

При выпуске новой версии продукта, исходный код выкладывается на удалённый репозиторий. Jenkins имеет интегрированный механизм для отслеживания изменений в репозитории (в качестве примера использовался Github). После изменений в репозитории Jenkins забирает новую версию и производит ряд указанных действий для сборки приложения.

Платформа Docker имеет собственную систему для хранения образов – Docker Hub. Он позволяет хранить только что собранные версии приложений. Версионность может быть необходима для применения образов в среде разработки и производственной среде.

Jenkins также поддерживает автоматическое развёртывание при применении инструмента оркестрации. Для проверки работоспособности и загруженности системы удобнее проводить развёртывание приложения вручную в тестовых средах.

## **2.4 Система управления базами данных**

Для хранения данных была выбрана СУБД PostgreSQL. Для сервисов словаря и тестирования знаний были выделены ниже представленные сущности, а также при помощи программного продукта DbSchema, предназначенного для визуального проектирования баз данных, по результатам анализа предметной области созданы логические модели базы данных.

Для сервиса словаря выделены следующие сущности:

- Word – сущность представляет собой слова, которые пользователь добавляет к себе в словарь для дальнейшего изучения.
- WordExample – сущность представляет пример использования слова в контексте с другими словами, например, в разговорной речи или литературе.
- Collection – сущность коллекции имеет возможность добавления слов для упорядоченного хранения их пользователем.

- User – сущность пользователей, которая необходима для привязки сущностей к конкретному пользователю и сохранения возможности независимой работы сервиса в распределённой системе.

- Authority – сущность является ролью пользователя в системе.

Логическая модель БД сервиса словаря представлена на рисунке 6.

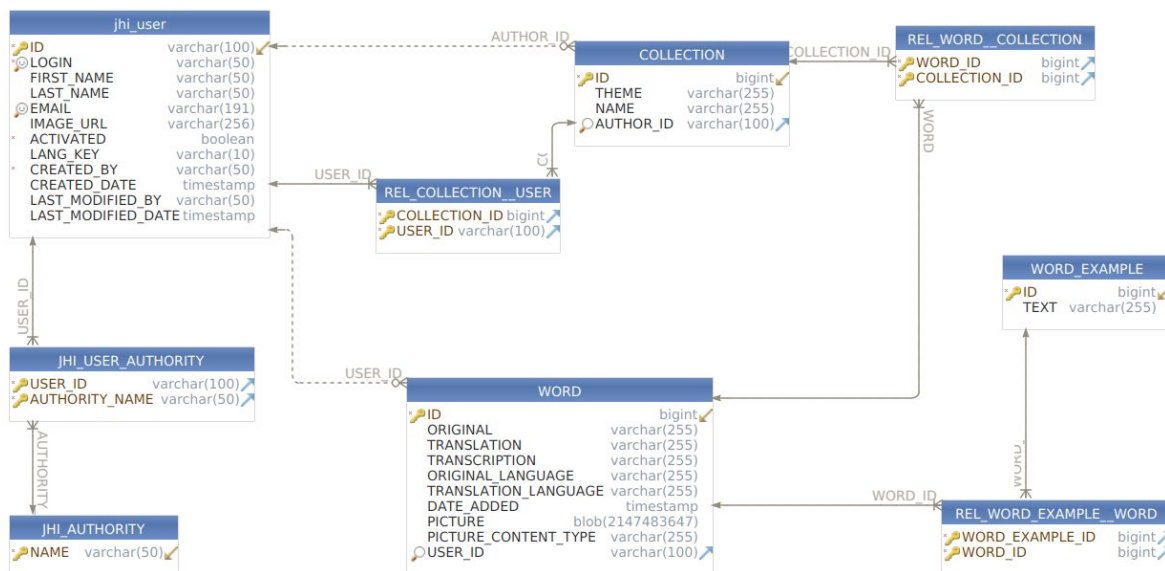


Рисунок 6 – Логическая модель БД сервиса словаря

Для сервиса тестирования знаний выделены следующие сущности:

- KnowledgeTest – сущность представляет собой тест для оценки текущих/полученных знаний пользователем. За этой сущностью закрепляются конкретные задания.

- Task – сущность заданий. Задания содержат в себе формулировку вопроса и за конкретным заданием прикрепляются возможные ответы.

- TaskType – сущность, в которой определяется тип задания.

- Answer – сущность, представляющая собой вариант ответа на задание (может быть правильным или ошибочным).

- User – аналогичная сущность пользователя модели БД словаря.

- Authority – аналогичная сущность роли модели БД словаря.

Логическая модель БД сервиса тестирования знаний представлена на рисунке 7.

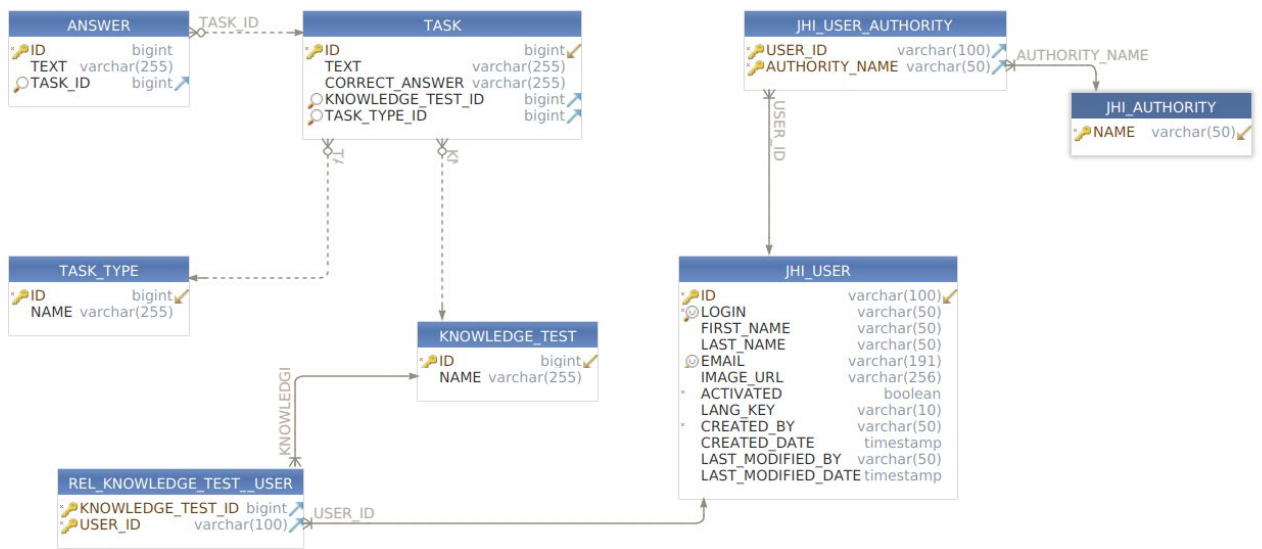


Рисунок 7 – Логическая модель БД сервиса тестирования знаний

## 2.5 Пользовательский интерфейс

Пользовательский интерфейс должен предоставлять возможность работы со словарём, с его коллекциями, тестированием знаний и возможность записи на индивидуальное занятие с преподавателем. На рисунке 8 представлен макет авторизация пользователя на платформе.

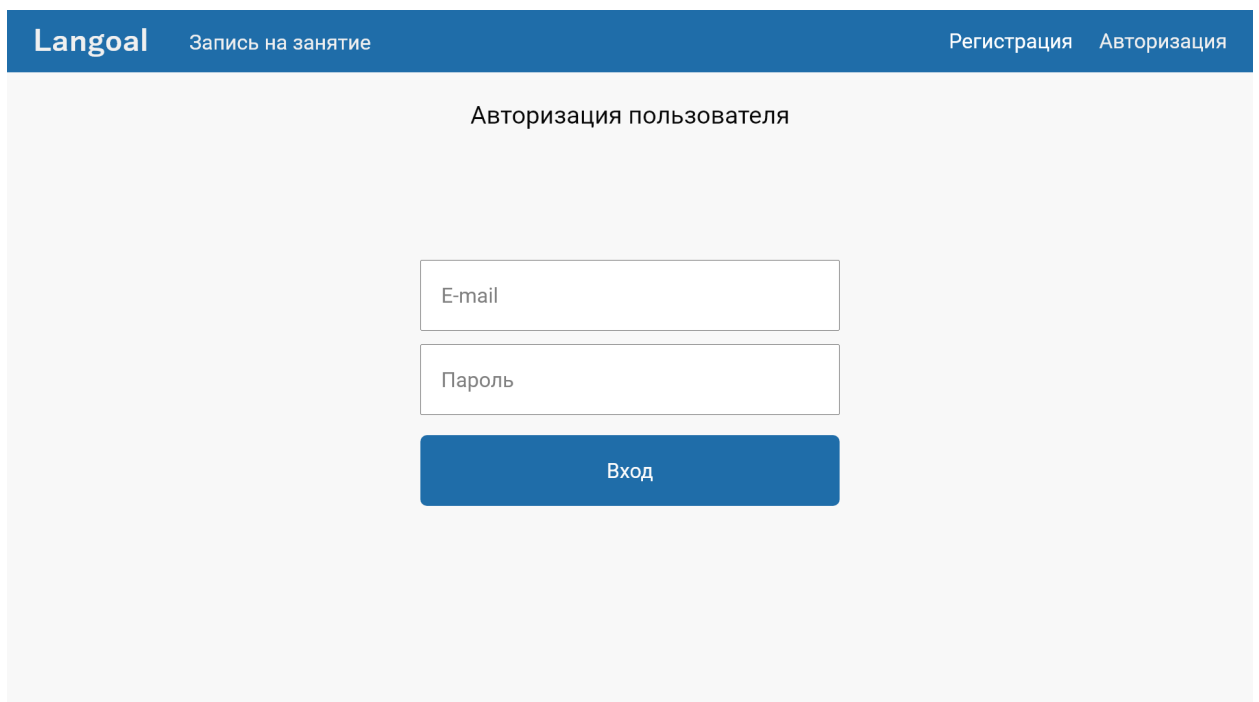


Рисунок 8 – Макет страницы авторизации пользователя

Для записи на индивидуальное занятие с преподавателем выбран формат ближайших двух дней и интервального выбора времени каждые полчаса. Макет представлен на рисунках 9–10.

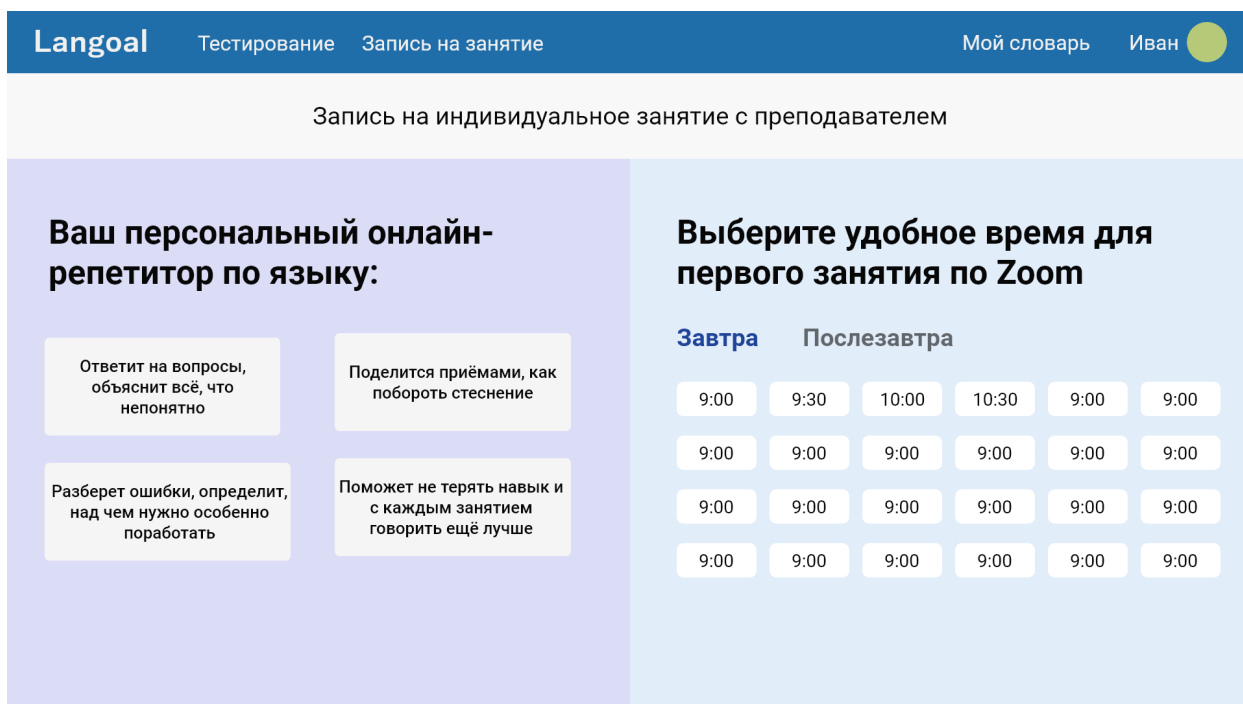


Рисунок 9 – Макет страницы для записи на индивидуальное занятие

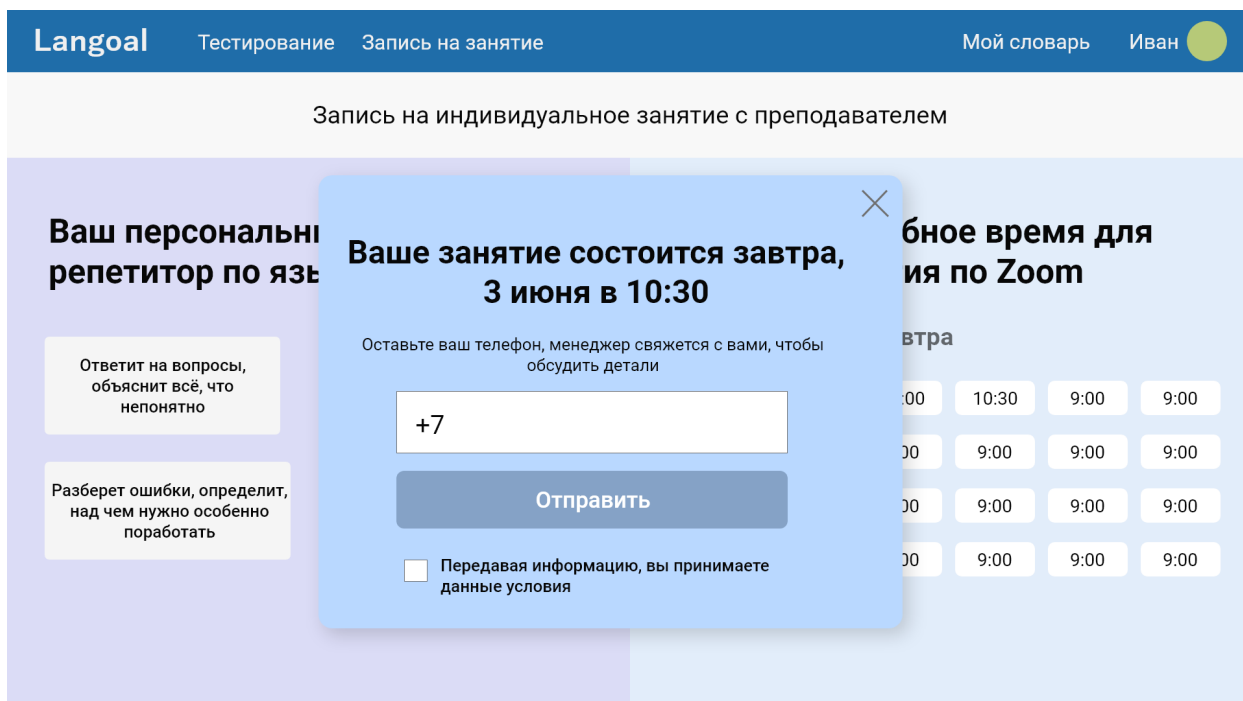


Рисунок 10 – Макет ввода номера для подтверждения индивидуального занятия

Далее на рисунках 11–12 представлены макеты для словаря, коллекций и слов соответственно.

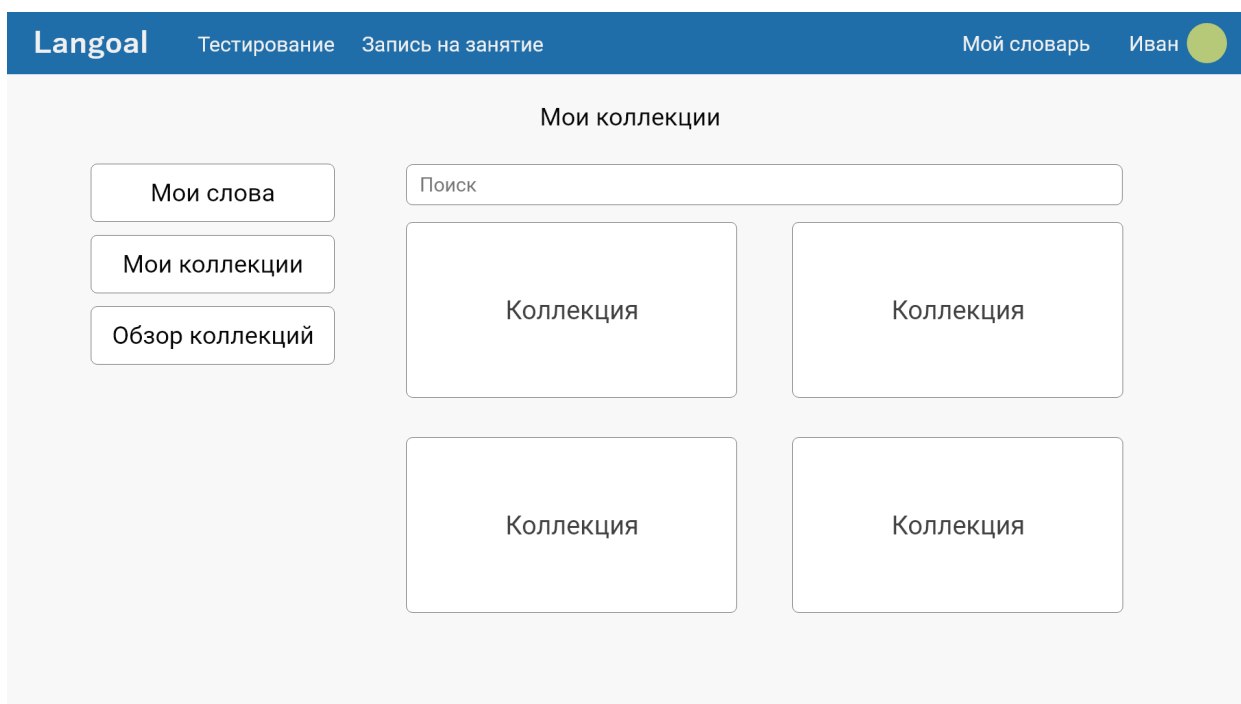


Рисунок 11 – Макет списка коллекций

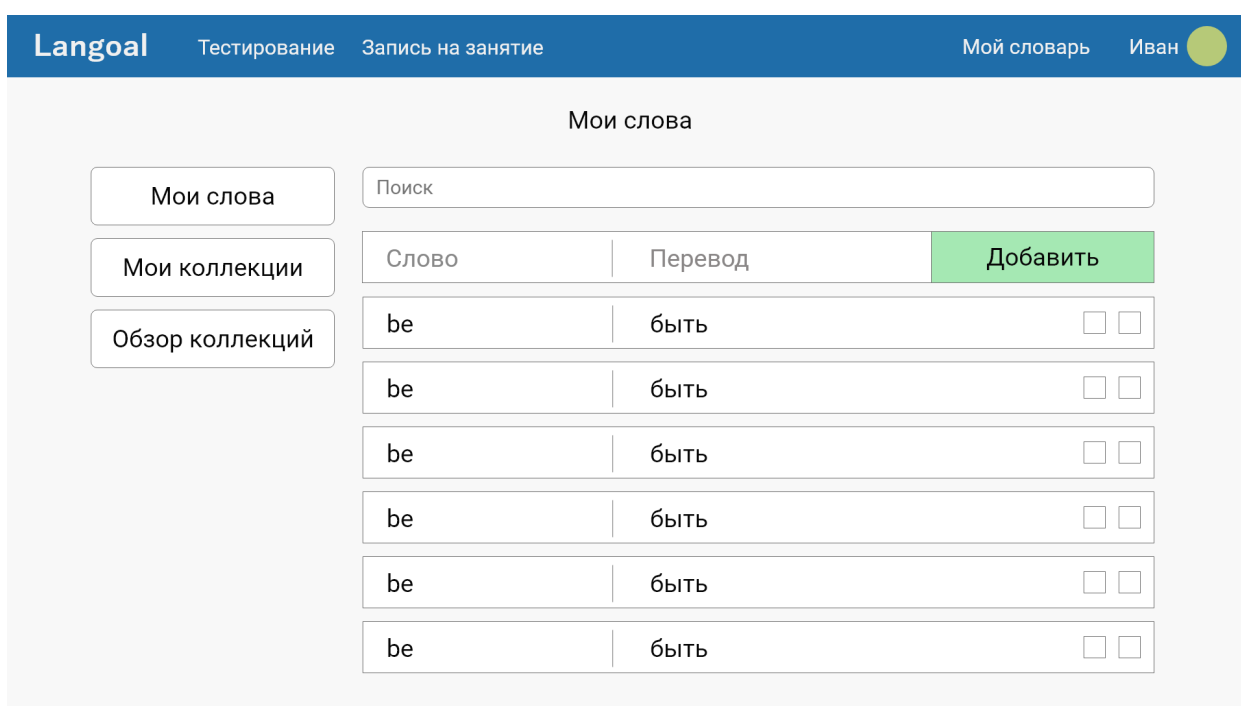


Рисунок 12 – Макет списка слов

Главная страница платформы представляет собой набор информации о ресурсе. Макет главной страницы сайта представлен на рисунке 13.



Рисунок 13 – Макет главной страницы сайта

На личной странице пользователя представлен инструментарий для редактирования профиля. Макет личного кабинета пользователя представлен на рисунке 14.

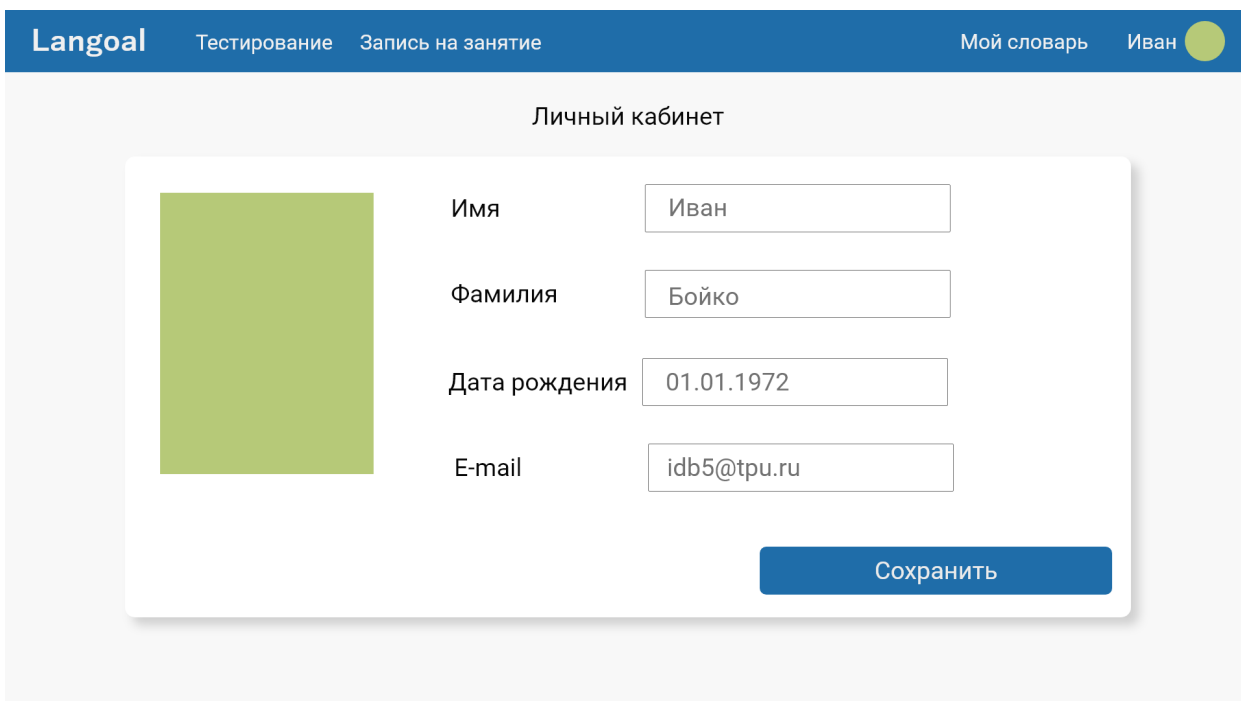


Рисунок 14 – Макет личного кабинета

Важной составляющей проекта являются тесты для проверки знаний пользователей ресурса. На рисунке 15 представлена сетка тестов, которая может быть отсортирована по тегам.

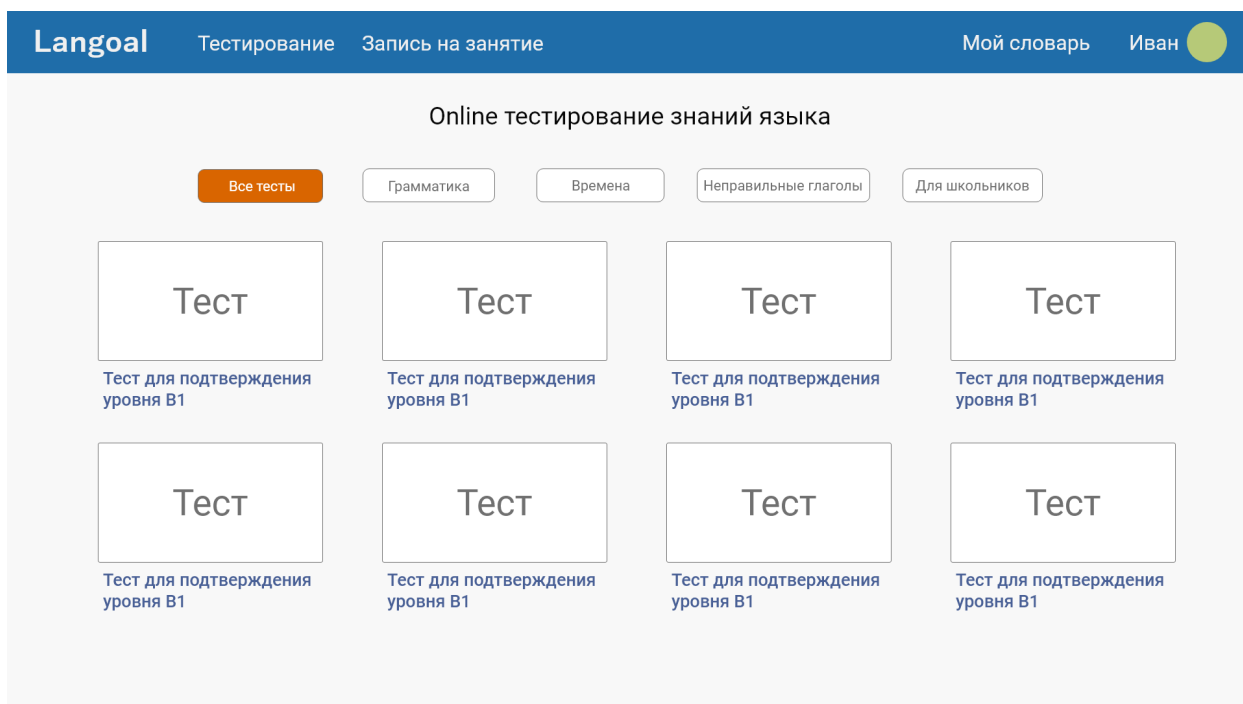


Рисунок 15 – Макет страницы с тестами

Тесты должны иметь линейность прохождения и варианты ответов. Макет прохождения теста представлен на рисунке 16.

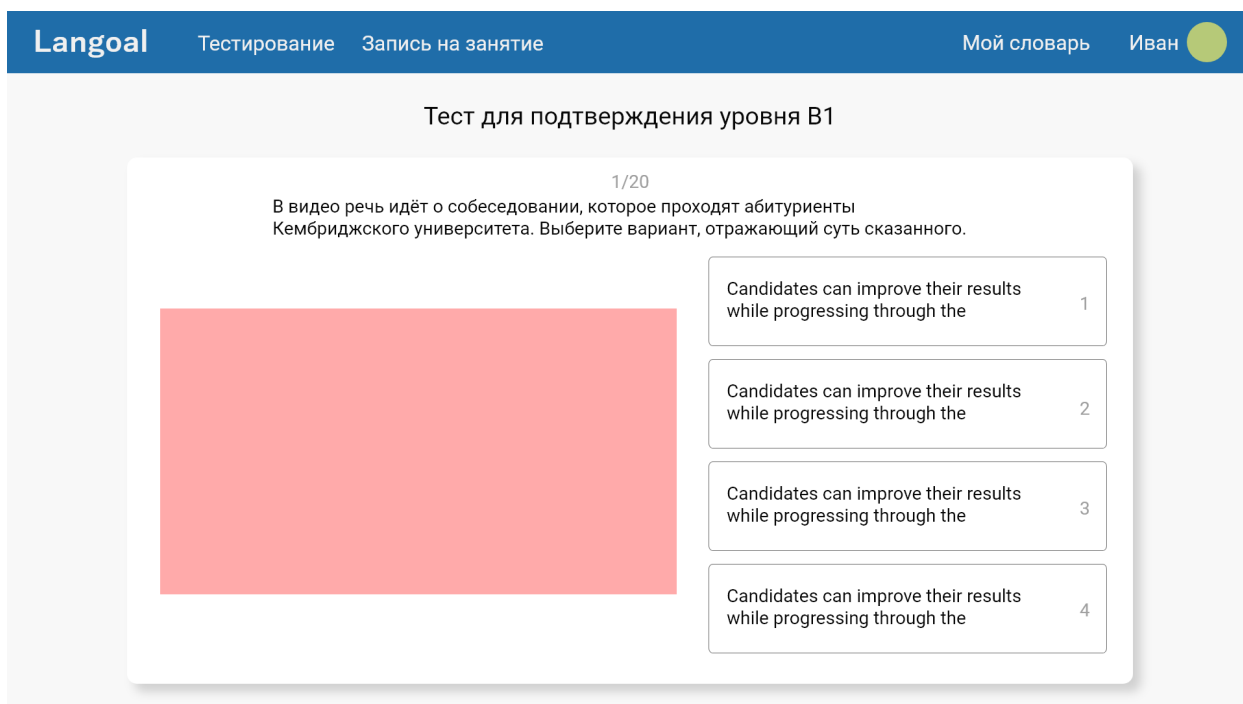


Рисунок 16 – Макет прохождения теста



## 3 РЕАЛИЗАЦИЯ

### 3.1 Используемые технологии

#### 3.1.1 Веб-инфраструктура Spring MVC

В качестве основного фреймворка разработки используется Spring, а именно стек OSS Netflix Spring. Веб-инфраструктура Spring MVC, как и многие другие веб-инфраструктуры MVC, основана на запросах и разработана на основе центрального сервлета, который отправляет запросы контроллерам и предлагает другие функциональные возможности, облегчающие разработку веб-приложений. Одной из главных составляющих Spring является DispatcherServlet, который отвечает за правильную координацию запросов по обработчикам. Он полностью интегрирован с контейнером Spring IoC и, таким образом, позволяет использовать все остальные функции Spring.

Процесс обработки Spring MVC от получения запроса до возвращения ответа показан на следующей диаграмме (Рис. 17).

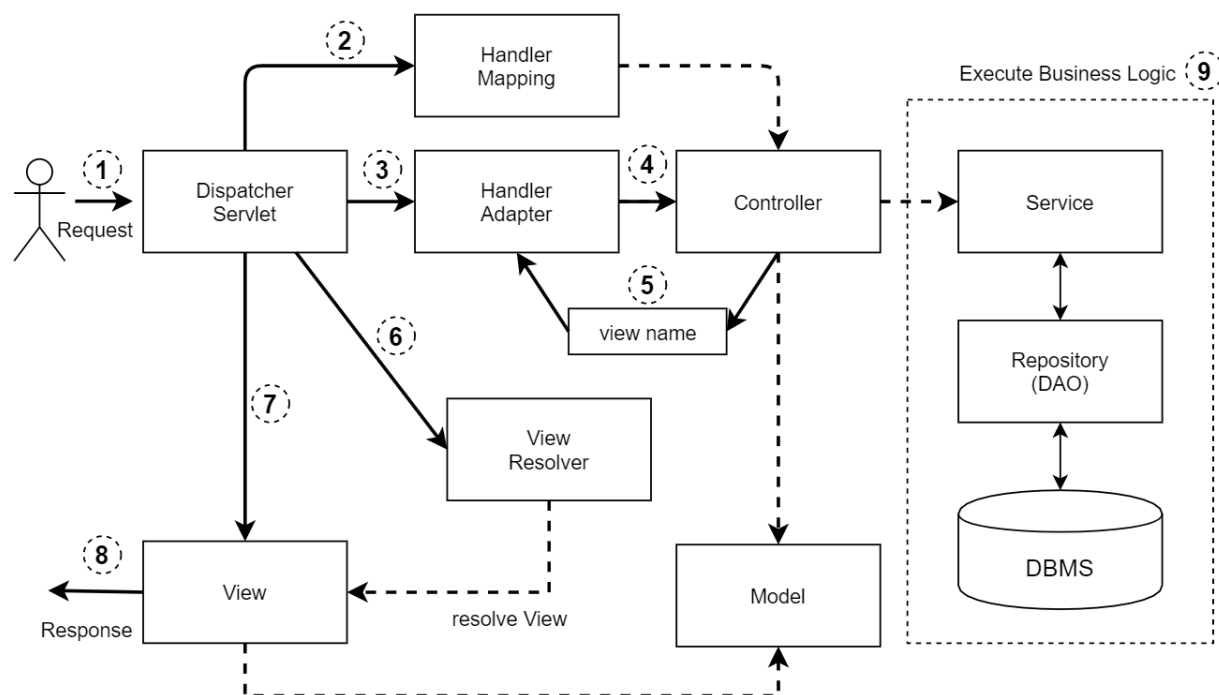


Рисунок 17 – Процесс обработки Spring MVC

1. DispatcherServlet получает запрос;

2. DispatcherServlet выполняет задачу выбора подходящего контроллера для HandlerMapping. HandlerMapping выбирает контроллер, который отвечает на входящий запрос URL и возвращает выбранный Handler и Controller на DispatcherServlet;
3. DispatcherServlet рассылает задачу выполнения бизнес-логики Controller к HandlerAdapter;
4. HandlerAdapter инициализирует работу Controller.
5. Контроллер выполняет бизнес-логику, устанавливает результат обработки Model и возвращает логическое имя представления на HandlerAdapter.
6. DispatcherServlet определяет какое представление необходимо использовать на основании полученного имени в ViewResolver.
7. DispatcherServlet направляет процесс рендеринга возвращенному View.
8. View показывает данные из Model и возвращает ответ.
9. При необходимости работы с базой данных Controller вызывает Service, который посредством Data Access Object взаимодействует с базой данных.

Таким способом происходит работа Spring MVC, но так как за возвращение конечному пользователю View отвечает Vue.js, то необходимости в использовании View Resolver нет.

В качестве инструмента сборки и для её оптимизации используется Gradle. Особенностью Gradle является тесная интеграция с инструментами разработки и сервером непрерывной интеграции Jenkins.

### **3.1.2 СУБД**

В качестве СУБД для разработки платформы выбрана PostgreSQL. PostgreSQL является объектно-реляционной системой управления базами данных. СУБД обладает высокой надёжностью и производительностью. PostgreSQL поддерживает транзакции (ACID) и имеет встроенный механизм

репликации БД, а также данная СУБД позволяет создавать собственные типы данных и индексов.

### **3.1.3 CI/CD**

Для непрерывной интеграции используется система Jenkins. Jenkins – это сервер автоматизации с открытым исходным кодом. Jenkins позволяет автоматизировать процесс разработки программного обеспечения. Он также управляет процессами доставки программного обеспечения на протяжении всего жизненного цикла, включая сборку, тестирование и развёртывание.

### **3.1.4 Технологии JS**

Для разработки пользовательского интерфейса используется JavaScript-фреймворк с открытым исходным кодом Vue.js (v2). Данный фреймворк позволяет строить представления на основе технологии SPA (single page application), что даёт возможность перехода между страницами без их полной перезагрузки.

## **3.2 Используемые средства разработки**

Во время разработки проекта были использованы следующие средства и инструменты:

- Adobe XD – использовался для проектирования пользовательского интерфейса для различных платформ, основан на векторной графике.
- DrawIO – использовался для построения графических схем в процессе разработки и общего представления работы алгоритмов.
- JetBrains IntelliJ IDEA – интегрированная среда разработки для языка программирования Java. Является основным инструментом для разработки данного проекта.
- VSCode – текстовый редактор для исходного кода, который поддерживает множество плагинов и расширений для оптимизации разработки, а также поддерживает возможности отладки исходного кода.
- Toad Data Modeler и DbSchema использовались для проектирования и наглядного представления сущностей баз данных в концептуальном, логическом и физическом видах.

– Postman – позволяет создавать коллекции запросов и тестировать их поочерёдно, что упрощает проверку работоспособности каждого доступного запроса в готовом API.

Разработка платформы производилась под управлением операционной системы GNU/Linux Fedora 34.

### 3.2 Взаимодействие архитектуры

На основе информации, полученной в процессе проектирования, была построена микросервисная архитектура. Она представляет собой:

- Registry.
- SSO.
- Gateway.
- WordService.
- KnowledgeTestService.

Полученная архитектура представлена на рисунке 18.

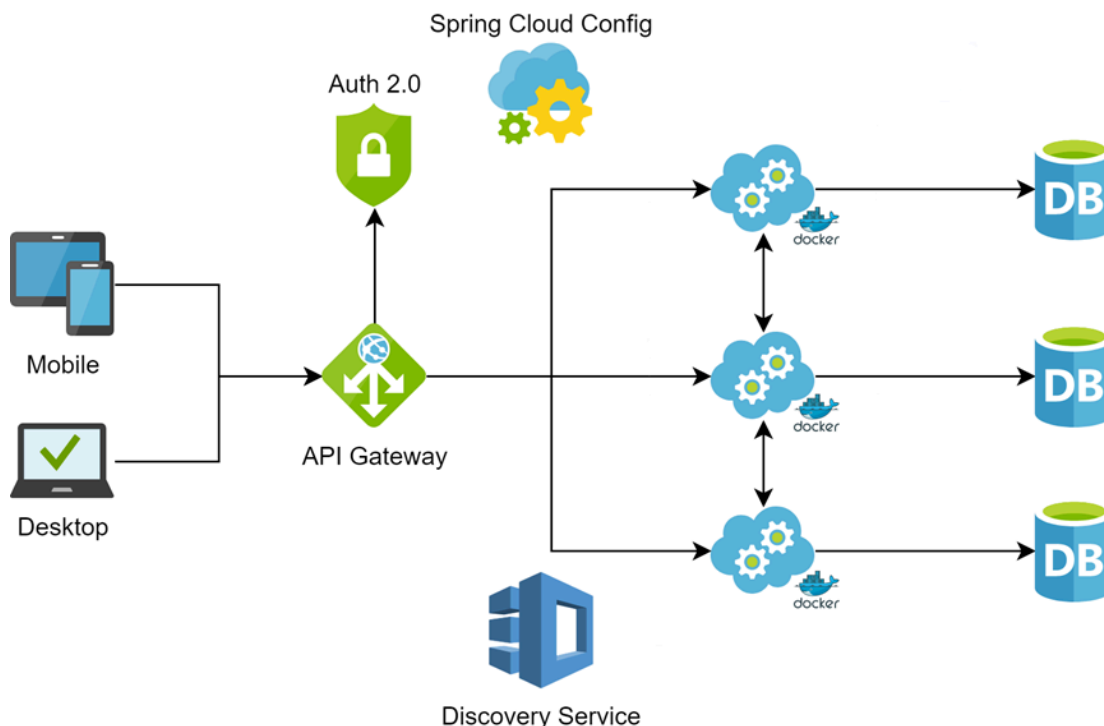


Рисунок 18 – Архитектура платформы

Сервисы успешно запускаются на локальной машине и взаимодействуют между собой. Запущенные сервисы представлены на рисунке 19.

```
registry
registry x keycloak x gateway x word x test x
e : Instance 'testservice:48a25c46aef4d43c8c71a8c4d182f4df' already registered
2021-04-30 00:52:14.303 DEBUG 17350 --- [ry-scheduling-1] i.g.j.r.service.ZuulUpdaterService
e : Exit: updateZuulRoutes() with result = null
2021-04-30 00:52:19.303 DEBUG 17350 --- [ry-scheduling-1] i.g.j.r.service.ZuulUpdaterService
e : Enter: updateZuulRoutes() with argument[s] = []
2021-04-30 00:52:19.304 DEBUG 17350 --- [ry-scheduling-1] i.g.j.r.service.ZuulUpdaterService
e : Checking instance gateway:22989585fb3bfc9179c1a89b452fe6f6 - http://85.143.78.49:8080/
2021-04-30 00:52:19.304 DEBUG 17350 --- [ry-scheduling-1] i.g.j.r.service.ZuulUpdaterService
e : Instance 'gateway:22989585fb3bfc9179c1a89b452fe6f6' already registered
2021-04-30 00:52:19.304 DEBUG 17350 --- [ry-scheduling-1] i.g.j.r.service.ZuulUpdaterService
e : Checking instance service1:8e39255e559be5aa37ae13a11d28b129 - http://85.143.78.49:8081/
2021-04-30 00:52:19.304 DEBUG 17350 --- [ry-scheduling-1] i.g.j.r.service.ZuulUpdaterService
e : Instance 'service1:8e39255e559be5aa37ae13a11d28b129' already registered
2021-04-30 00:52:19.304 DEBUG 17350 --- [ry-scheduling-1] i.g.j.r.service.ZuulUpdaterService
e : Checking instance testservice:48a25c46aef4d43c8c71a8c4d182f4df - http://85.143.78.49:8082/
2021-04-30 00:52:19.304 DEBUG 17350 --- [ry-scheduling-1] i.g.j.r.service.ZuulUpdaterService
e : Instance 'testservice:48a25c46aef4d43c8c71a8c4d182f4df' already registered
2021-04-30 00:52:19.304 DEBUG 17350 --- [ry-scheduling-1] i.g.j.r.service.ZuulUpdaterService
e : Exit: updateZuulRoutes() with result = null
```

Рисунок 19 – Запущенные сервисы

### 3.3 Базы данных

На основе логических моделей базы данных были описаны сущности с помощью инструмента Hibernate ORM. Описание сущностей позволяет выполнять миграцию в базу данных, поэтому нет необходимости в описании отдельных файлов для интеграции моделей в существующую БД (Рис. 20).

```

@Entity
@Table(name = "collection")
public class Collection implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "sequenceGenerator")
    @SequenceGenerator(name = "sequenceGenerator")
    private Long id;

    @Column(name = "theme")
    private String theme;

    @Column(name = "name")
    private String name;

    @ManyToOne
    private User author;

    @ManyToMany
    @JoinTable(
        name = "rel_collection_user",
        joinColumns = @JoinColumn(name = "collection_id"),
        inverseJoinColumns = @JoinColumn(name = "user_id")
    )
    private Set<User> users = new HashSet<>();

    @ManyToMany(mappedBy = "collections")
    @JsonIgnoreProperties(value = { "user", "collections", "wordExamples" }, allowSetters = true)
    private Set<Word> words = new HashSet<>();
}

```

Рисунок 20 – Сущность коллекции слов

Для описания моделей используются аннотации. Аннотации начинаются с символа `@`. Рассмотрим более подробно работу и смысловую нагрузку используемых аннотаций.

`@Entity` – указывает для Hibernate, что база данных должна иметь таблицу с тем же именем класса. Также у каждой сущности есть аннотация `@Id`, которая отмечает идентификатор объекта.

`@Table` – позволяет указать детали таблицы, которая будет использоваться для сохранения сущности в базе данных. Аннотация предоставляет четыре атрибута, переопределяющих имя таблицы, её каталог и схему, а также позволяет применять уникальные ограничения для столбцов в таблице.

`@Id` – первичный ключ. Первичный ключ может быть отдельным полем или комбинацией нескольких полей в зависимости от структуры

таблицы. По умолчанию аннотация `@Id` автоматически определяет наиболее подходящую стратегию генерации первичного ключа, которую можно использовать.

`@GeneratedValue` – использует стратегию генерации ключей по умолчанию.

`@Column` – используется для указания сведений о столбце, в который будет отображаться поле.

`@NotNull` – аннотированный элемент не может иметь значение `null`.

`@Min` – указывает минимальное ограничение на числовой аннотированный элемент (или строковое представление числового значения).

`@NotEmpty` – аннотированный элемент показывает, что строка не может быть пустой. Как правило используется совместно с аннотацией `@NotNull`.

### **3.4 Контроллеры**

Контроллеры выполняют важную роль маппинга, то есть распределения запросов по определённым сервисам, которые в свою очередь взаимодействуют с моделями и получают необходимые данные для выполнения своей задачи.

Каждый сервис содержит необходимую реализацию для выполнения поставленной задачи, указанной в документации к API. В первую очередь сервис каждой сущности обязан выполнять CRUD-операции для возможности взаимодействия со структурой БД, остальные методы для сервисов выполняются для каждой отдельной задачи.

На рисунке 21 представлен один из входов в контроллере, который выполняет задачи изменения сущности слова по его уникальному идентификатору.

```

@PutMapping("/{id}")
public ResponseEntity<WordDTO> updateWord(
    @PathVariable(value = "id", required = false) final Long id, @RequestBody WordDTO wordDTO)
    throws URISyntaxException {
    log.debug("REST request to update Word : {}, {}", id, wordDTO);
    if (wordDTO.getId() == null) {
        throw new BadRequestAlertException("Invalid id", ENTITY_NAME, "idnull");
    }
    if (!Objects.equals(id, wordDTO.getId())) {
        throw new BadRequestAlertException("Invalid ID", ENTITY_NAME, "idinvalid");
    }

    if (!wordRepository.existsById(id)) {
        throw new BadRequestAlertException("Entity not found", ENTITY_NAME, "idnotfound");
    }

    WordDTO result = wordService.save(wordDTO);
    return ResponseEntity
        .ok()
        .headers(HeaderUtil.createEntityUpdateAlert(
            applicationName,
            enableTranslation: true,
            ENTITY_NAME, wordDTO.getId().toString()))
        .body(result);
}

```

Рисунок 21 – Пример пути контроллера для изменения слова

### 3.5 Объекты передачи данных

DTO (Data Transfer Object) представляет собой класс для представления сущностей в данные, которые передаются между сервисами для их обработки.

Для каждой сущности был написан данный класс, что является обязательным при работе с сущностями. На рисунках 22-23 представлен пример DTO для сущности слова.



```

public class WordDTO implements Serializable {

    private Long id;

    private String original;

    private String translation;

    private String transcription;

    private String originalLanguage;

    private String translationLanguage;

    private ZonedDateTime dateAdded;

    @Lob
    private byte[] picture;

    private String pictureContentType;
    private UserDTO user;

    private Set<CollectionDTO> collections = new HashSet<>();
}

```

Рисунок 22 – DTO сущности Word

```

@Override
public String toString() {
    return "WordDTO{" +
        "id=" + getId() +
        ", original='" + getOriginal() + "'" +
        ", translation='" + getTranslation() + "'" +
        ", transcription='" + getTranscription() + "'" +
        ", originalLanguage='" + getOriginalLanguage() + "'" +
        ", translationLanguage='" + getTranslationLanguage() + "'" +
        ", dateAdded='" + getDateAdded() + "'" +
        ", picture='" + Arrays.toString(getPicture()) + "'" +
        ", user='" + getUser() + "'" +
        ", collections=" + getCollections() +
        "}";
}

```

Рисунок 23 – Метод toString DTO Word для представления данных JSON-формате

Хорошей практикой является использование паттерна маппер для изменения представления данных «на лету». Маппер позволяет переводить данные из типа DTO в реляционную структуру БД. Данный способ изолирует реализацию от разработчика для предотвращения проблем и ошибок во время разработки. Пример маппера представлен на рисунке 24.

```

@Mapper(componentModel = "spring", uses = { UserMapper.class, CollectionMapper.class })
public interface WordMapper extends EntityMapper<WordDTO, Word> {
    @Mapping(target = "user", source = "user", qualifiedByName = "id")
    @Mapping(target = "collections", source = "collections", qualifiedByName = "idSet")
    WordDTO toDto(Word s);

    @Named("idSet")
    @BeanMapping(ignoreByDefault = true)
    @Mapping(target = "id", source = "id")
    Set<WordDTO> toDtoIdSet(Set<Word> word);

    @Mapping(target = "removeCollection", ignore = true)
    Word toEntity(WordDTO wordDTO);
}

```

Рисунок 24 – Пример маппера

Сервис сущности Word имеет метод «save», который использует в себе маппер для преобразования DTO в реляционное представление и обратно (Рис. 25).

```

public WordDTO save(WordDTO wordDTO) {
    log.debug("Request to save Word : {}", wordDTO);
    Word word = wordMapper.toEntity(wordDTO);
    word = wordRepository.save(word);
    return wordMapper.toDto(word);
}

```

Рисунок 25 – Применение DTO и маппера

### 3.6 REST API и запросы

Для работы с данными необходимыми для функционирования системы реализован REST API сервисов, с помощью которого можно выполнять все CRUD-операции. Далее рассмотрим выполненные запросы сервиса словаря:

GET запросы:

- Получение коллекции/коллекций слов.
- Получение примера/примеров для слова.
- Получение слов/слова.

POST запросы:

- Добавление коллекции слов.
- Добавление примера слова.

- Добавление слова.

PUT/PATCH запросы:

- Изменение коллекции слова.
- Изменение примера слова.
- Изменение слова.

DELETE запросы:

- Удаление коллекции слов.
- Удаление примера слова.
- Удаление слова.

Рассмотрим выполненные запросы сервиса тестов:

GET запросы:

- Получение ответа/ответов.
- Получение теста/тестов.
- Получение задания/заданий.

POST запросы:

- Добавление ответа.
- Добавление теста.
- Добавление задания.

PUT/PATCH запросы:

- Изменение ответа.
- Изменение теста.
- Изменение задания.

DELETE запросы:

- Удаление ответа.
- Удаление теста.
- Удаление задания.

Работоспособность запросов проверялась с помощью Curl средствами веб-интерфейса Swagger. На рисунке 26 представлен пример поступающего запроса на сервер и ответа на него.

```
Curl
curl -X 'GET' \
'http://localhost:8080/services/service1/api/collections' \
-H 'accept: */*' \
-H 'X-XSRF-TOKEN: a60d3a31-1553-4810-88c0-66cdf222b31'

Request URL
http://localhost:8080/services/service1/api/collections

Server response
Code    Details
200     Response body
[
  {
    "id": 1,
    "theme": "index Wooden",
    "name": "South",
    "author": null,
    "users": []
  },
  {
    "id": 2,
    "theme": "feed turn-key",
    "name": "Strategist Maldives bandwidth-monitored",
    "author": null,
    "users": []
  },
]
```

Рисунок 26 – GET-запрос получения всех коллекций слов

### 3.7 Реализация пользовательского интерфейса

Для реализации пользовательского интерфейса на основе SPA (single page application) используется JavaScript-фреймворк с открытым исходным кодом Vue.js.

Для предоставления возможности пользователю переходить по страницам сайта были прописаны маршруты. Разделения доступа происходит на уровне микросервисов, к которым имеется доступ только после авторизации. Маршруты представлены на рисунке 27.

```

Vue.use(Router);

// prettier-ignore
export default new Router({
  mode: 'history',
  routes: [
    {
      path: '/',
      name: 'Home',
      component: Home
    },
    {
      path: '/forbidden',
      name: 'Forbidden',
      component: Error,
      meta: { error403: true }
    },
    {
      path: '/not-found',
      name: 'NotFound',
      component: Error,
      meta: { error404: true }
    },
  ],
});

```

Рисунок 27 – Маршруты перехода по страницам

Пример реализации «шапки», которая располагается на каждой странице вверху пользовательской страницы, представлен на рисунке 28.

```

<template>
  <b-navbar data-cy="navbar" toggleable="md" type="dark" class="jh-navbar">
    <b-navbar-brand class="logo" b-link to="/">
      <span class="logo-img"></span>-->
      <span class="navbar-title">Langoal</span>
    </b-navbar-brand>
    <b-navbar-toggle
      right
      class="jh-navbar-toggler d-lg-none"
      href="javascript:void(0);"
      data-toggle="collapse"
      target="header-tabs"
      aria-expanded="false"
      aria-label="Toggle navigation"
    >
      <font-awesome-icon icon="bars" />
    </b-navbar-toggle>

    <b-collapse is-nav id="header-tabs">
      <b-navbar-nav class="ml-auto">
        <b-nav-item to="/" exact...>
          <b-nav-item-dropdown right id="entity-menu" v-if="authenticated" active-class="active" class="pointer" data-cy="entity">
            <span slot="button-content" class="navbar-dropdown-menu">
              <font-awesome-icon icon="th-list" />
              <span class="no-bold" v-text="$t('global.menu.entities.main')">Entities</span>
            </span>
          </b-nav-item-dropdown>
          <b-nav-item-dropdown...>
          <b-nav-item-dropdown id="languagesNavBarDropdown" right v-if="languages && Object.keys(languages).length > 1"...>
          <b-nav-item-dropdown...>
        </b-navbar-nav>
      </b-collapse>
    </b-navbar>
  </template>

<script lang="ts" src="./jhi-navbar.component.ts"></script>

```

Рисунок 28 – Пример реализации «шапки»

### 3.8 Реализация авторизации при помощи SSO Keycloak

Для авторизации используется SSO Keycloak. Данный инструмент предоставляет возможность взаимодействия со шлюзом посредством API и имеет достаточно гибкую настройку под различные нужды разработчика. На рисунке 29 представлена консоль администрирования SSO Keycloak.

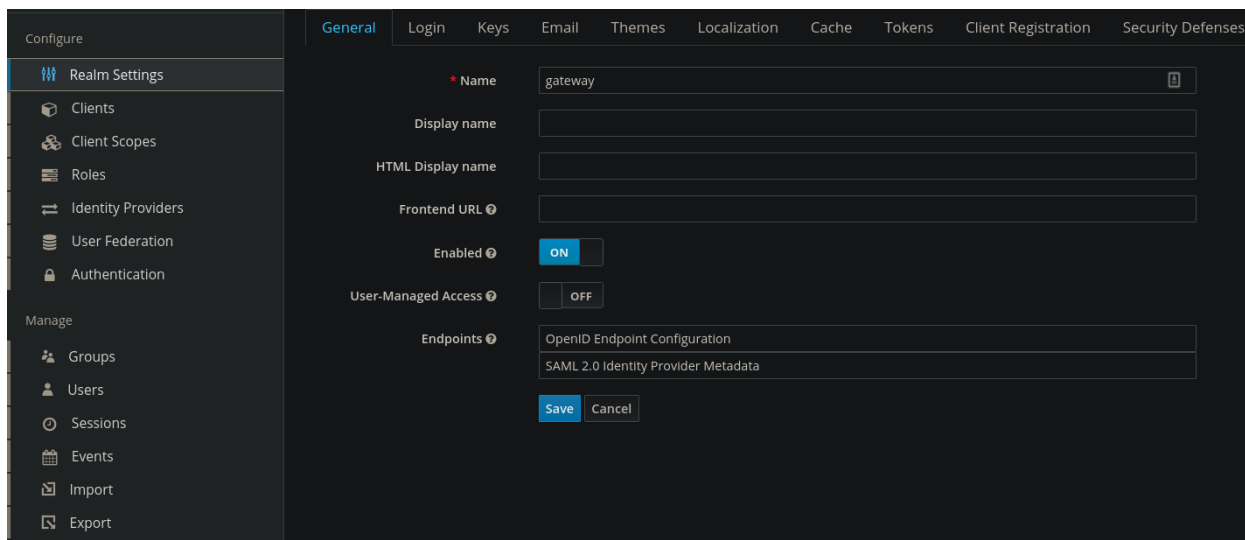


Рисунок 29 – Консоль администрирования SSO Keycloak

При нажатии кнопки авторизации пользователя переадресует на страницу авторизации Keycloak. При вводе верных данных пользователь будет обратно направлен на страницу, с которой совершал авторизацию. Код переадресации представлен на рисунке 30. Страница авторизации представлена на рисунке 31.

```
public login(loc : Location = window.location) {
  const port = loc.port ? ':' + loc.port : '';
  let contextPath = location.pathname;
  if (contextPath.endsWith('accessdenied')) {
    contextPath = contextPath.substring(0, contextPath.indexOf('accessdenied'));
  }
  if (contextPath.endsWith('forbidden')) {
    contextPath = contextPath.substring(0, contextPath.indexOf('forbidden'));
  }
  if (!contextPath.endsWith('/')) {
    contextPath = contextPath + '/';
  }

  loc.href = `://${loc.hostname}${port}${contextPath}oauth2/authorization/oidc`;
}
```

Рисунок 30 – Переадресация на страницу авторизации

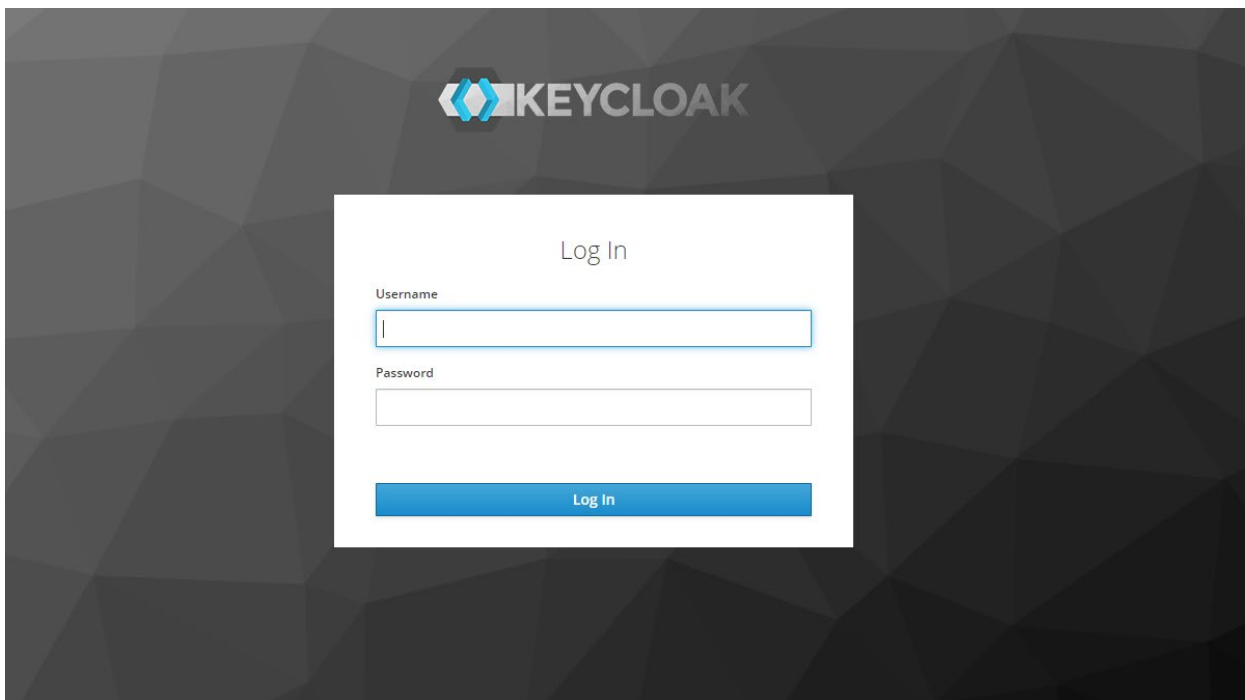


Рисунок 31 – Страница авторизации Keycloak

Для работы с токеном описана его сущность. Ее поля представлены на рисунке 32.

```
public class OAuthIdpTokenResponseDTO implements Serializable {  
  
    @JsonProperty("token_type")  
    private String tokenType;  
  
    private String scope;  
  
    @JsonProperty("expires_in")  
    private Long expiresIn;  
  
    @JsonProperty("ext_expires_in")  
    private Long extExpiresIn;  
  
    @JsonProperty("expires_on")  
    private Long expiresOn;  
  
    @JsonProperty("not-before-policy")  
    private Long notBefore;  
  
    private UUID resource;  
  
    @JsonProperty("access_token")  
    private String accessToken;  
  
    @JsonProperty("refresh_token")  
    private String refreshToken;  
  
    @JsonProperty("id_token")  
    private String idToken;  
  
    @JsonProperty("session_state")  
    private String sessionState;  
  
    @JsonProperty("refresh_expires_in")  
    private String refreshExpiresIn;  
}
```

Рисунок 32 – Сущность токена авторизации

Для работы с пользователем и отправки запросов созданы необходимые функции для получения текущего авторизованного пользователя и проверки действительности аутентификации пользователя на каждой странице платформы, что соответственно представлены на рисунках 33 и 34.

```
/**
 * Get the login of the current user.
 *
 * @return the login of the current user.
 */
public static Mono<String> getCurrentUserLogin() {
    return ReactiveSecurityContextHolder
        .getContext()
        .map(SecurityContext::getAuthentication)
        .flatMap(authentication -> Mono.justOrEmpty(extractPrincipal(authentication)));
}
```

Рисунок 33 – Получение текущего авторизованного пользователя

```
/**
 * Check if a user is authenticated.
 *
 * @return true if the user is authenticated, false otherwise.
 */
public static Mono<Boolean> isAuthenticated() {
    return ReactiveSecurityContextHolder
        .getContext() Mono<SecurityContext>
        .map(SecurityContext::getAuthentication) Mono<Authentication>
        .map(Authentication::getAuthorities) Mono<Collection<capture of ? extends GrantedAuthority>>
        .map(authorities -> authorities.stream().map(GrantedAuthority::getAuthority).noneMatch(AuthoritiesConstants.ANONYMOUS::equals));
}
```

Рисунок 34 – Проверка аутентификации пользователя



## **4 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ**

### **4.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения**

#### **4.1.1 Потенциальные потребители результатов исследования**

В рамках данной работы выполняется исследование, проектирование и разработка микросервисной архитектуры платформы для обучения иностранным языкам. Данная платформа обеспечивает возможность обучения иностранным языкам самостоятельно и возможность записи к преподавателю для индивидуальной работы в паре. Разбивая приложение на множество более мелких частей, уменьшая дублирование и связи между частями, можно получить более понятное, масштабируемое и изменяемое веб-приложение в целом. Данная платформа обеспечивает возможность обучения иностранным языкам самостоятельно и возможность записи к преподавателю для индивидуальной работы в паре. Особый интерес представляется для обычных пользователей, которые получают быстродействующее и безотказное приложение.

#### **4.1.2 Анализ конкурентных технических решений**

При рассмотрении различных архитектур проектирования систем, в качестве конкурентных компаний можно выделить:

- компании, системы которых построены на основе монолитной архитектуры (K1)
- компании, предоставляющие решения с использованием микросервисной архитектуры на базе облачных технологий (K2)

Анализ конкурентных технических решений был проведен с помощью оценочной карты, представленной в таблице 2.

Таблица 2. Оценочная карта критериев эффективности конкурентных технических решений (разработок)

<b>Критерии оценки</b>		<b>Баллы</b>	<b>Конкурентоспособность</b>
------------------------	--	--------------	------------------------------

	<b>Вес критерия</b>	Бф	Бк1	Бк2	Кф	Кк1	Кк2
<b>Технические критерии оценки ресурсоэффективности</b>							
Простота в эксплуатации	0,1	4	5	3	0,4	0,5	0,3
Инновационная привлекательность	0,15	4	2	5	0,6	0,3	0,75
Надёжность	0,1	5	4	5	0,5	0,4	0,5
Удобство в эксплуатации	0,05	4	5	3	0,2	0,25	0,15
Функциональная мощность	0,15	4	3	5	0,6	0,45	0,75
<b>Экономические критерии оценки эффективности</b>							
Послепродажное обслуживание	0,15	4	5	3	0,6	0,75	0,45
Предполагаемый срок эксплуатации	0,1	4	3	4	0,4	0,3	0,4
Цена	0,2	4	5	2	0,8	1	0,4
<b>Итого</b>	<b>1</b>	<b>33</b>	<b>35</b>	<b>30</b>	<b>4,1</b>	<b>3,95</b>	<b>3,7</b>

Анализ конкурентных решений определяется по формуле 1:

$$K = \sum V_i \times B_i, \quad (1)$$

где К – конкурентоспособность научной разработки или конкурента;

$V_i$  – вес показателя (в долях единицы);

$B_i$  – балл  $i$ -го показателя.

Основываясь на рассчитанных значениях, можно предположить, что инновационность разработки и функциональная мощность продукта в сочетании с выгодной ценой формирует конкурентное преимущество. Существующим альтернативам не хватает более низкой конечной стоимости в одном случае (микросервисная архитектура на базе облачных технологий) и функциональной мощности в другом случае (монолитная архитектура).

#### 4.1.3 Технология QuaD

Технология QuaD (QUality ADvisor) представляет собой гибкий инструмент измерения характеристик, описывающих качество новой разработки и ее перспективность на рынке и позволяющие принимать решение целесообразности вложения денежных средств в научно-исследовательский проект.

Анализ имеющихся конкурентных продуктов необходимо проводить с достаточной регулярностью, так как рынок ИТ находится в постоянном и активном движении на сегодняшний день. Данный анализ позволяет производить изменения текущего внедрения, чтобы сделать его наиболее перспективным и конкурентоспособным в дальнейшем. Результат QuaD-анализа представлен в таблице 3.

Таблица 3. Оценка критериев в соответствии с технологией QuaD

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение	Средневзвешенное значение
Простота в эксплуатации	0,1	80	100	0,8	8
Инновационная привлекательность	0,15	90	100	0,9	13,5
Надёжность	0,1	90	100	0,9	9
Удобство в эксплуатации	0,05	70	100	0,7	3,5
Функциональная мощность	0,15	90	100	0,9	13,5
Послепродажное обслуживание	0,15	60	100	0,6	9
Предполагаемый срок эксплуатации	0,1	90	100	0,9	9
Цена	0,2	90	100	0,9	18
<b>Итого:</b>					<b>83,5</b>

Средневзвешенное значение показателя качества и перспективности научной разработки равен 83,5 (попадает в промежуток 80–100), то есть такая разработка считается перспективной.

#### 4.1.4 SWOT-анализ

SWOT-анализ – метод стратегического планирования, заключающийся в выявлении факторов внутренней и внешней среды организации и разделении их на четыре категории:

- Strengths (сильные стороны);
- Weaknesses (слабые стороны);
- Opportunities (возможности);
- Threats (угрозы).

Сильные (S) и слабые (W) стороны являются факторами внутренней среды объекта анализа, (то есть тем, на что сам объект способен повлиять);

возможности (О) и угрозы (Т) являются факторами внешней среды (то есть тем, что может повлиять на объект извне и при этом не контролируется объектом).

В рамках данного анализа выявлены сильные и слабые стороны проекта, а также его возможности и угрозы. Все параметры SWOT анализа данного проекта представлены в таблице 4.

Таблица 4. SWOT-анализ проекта

		Внутренние факторы	
		Сильные стороны проекта: 1. Устойчивость системы 2. Быстродействие системы 3. Гибкость конфигурации отдельных сервисов 4. Многофункциональность предлагаемых сервисов	Слабые стороны проекта: 1. Сложность поддержки системы 2. Необходимость длительной поддержки 3. Повышенная стоимость конечного продукта 4. Недостаточная проработка дизайна приложения системы
Внешние факторы	<b>Возможности:</b> 1. Продажа готового продукта для других организаций под заказ 2. Выведение продукта на мировой рынок 3. Рост спроса на рынке онлайн-приложений для обучения 4. Приоритетная политика правительства РФ по поддержке малого и среднего бизнеса	1. Востребованность как среди конечных потребителей, так и среди крупных организаций 2. Отсутствие рекламы 3. Повышенная безопасность данных пользователей	1. Завышенные цены конкурентов помогут повысить количество лояльных клиентов и репутацию организации 2. Повышенная стоимость нивелируется высоким спросом и качеством предоставляемого продукта
	<b>Угрозы:</b> 1. Неокупаемость проекта в связи с высокими затратами на поддержку системы 2. Несоответствие ожиданиям клиента 3. Снижение спроса на обучение иностранным языкам 4. Появление конкурентов со стороны крупных компаний	1. Интуитивно понятный дизайн, а также возможность его изменить может повысить посещаемость ресурса 2. Рост клиентов ресурса привлекает новых клиентов, что позволит привлечь новых пользователей из других систем	1. Малые вычислительные мощности серверов могут уменьшить посещаемость сайта 2. Неблагоприятные изменения в гос. законах могут повлечь убытки (федеральный закон о персональных данных) 3. Признание организации иностранным агентом в связи с оттоком населения из страны

SWOT-анализ показал, что риски, связанные с устранением несоответствий на всех этапах эксплуатации продукта, должны занимать ключевую роль в расчете выгоды предлагаемой разработки. Общая

картина говорит о том, что, несмотря на угрозы и слабые стороны проекта, разработка обладает конкурентным преимуществом и является перспективной.

## 4.2 Планирование научно-исследовательских работ

### 4.2.1 Структура работ в рамках научного исследования

Важным этапом проведения научно-исследовательских работ является необходимость планирования работ, которое включает в себя определение полного перечня работ, а также их распределение между всеми исполнителями проекта. Исполнителями проекта являются студент и научный руководитель. Научный руководитель определяет цели и задачи для студента, направляет и контролирует работу его работу, оценивает результаты проделанной работы и дает рекомендации студенту. Студент полностью отвечает за выполняемую работу. В таблице 5 представлен перечень работ, а также распределение исполнителей по ним.

Таблица 5. Перечень работ и распределение исполнителей

№ работы	Наименование работы	Исполнители работы
1	Выбор научного руководителя бакалаврской работы	Бойко И.Д. Коровкин В.А.
2	Составление и утверждение темы бакалаврской работы	Бойко И.Д. Коровкин В.А.
3	Составление календарного плана-графика выполнения бакалаврской работы	Бойко И.Д.
4	Подбор и изучение литературы по теме бакалаврской работы	Бойко И.Д. Коровкин В.А.
5	Анализ предметной области	Бойко И.Д.
6	Проектирование информационной системы	Бойко И.Д.
7	Разработка клиентского приложения системы	Бойко И.Д.
8	Разработка сервера системы	Бойко И.Д.
9	Согласование выполненной работы с научным руководителем	Бойко И.Д. Коровкин В.А.
10	Выполнение других частей работы (финансовый менеджмент, социальная ответственность)	Бойко И.Д.
11	Подведение итогов, оформление работы	Бойко И.Д.

### 4.2.2 Определение трудоемкости выполнения работ

Как правило, трудовые затраты образуют основную часть стоимости исследования, поэтому важным этапом планирования научно-исследовательской деятельности является определение трудоемкости работ.

Определение трудоёмкости выполнения работ осуществляется на основе экспертной оценки ожидаемой трудоёмкости выполнения каждой работы путём расчёта длительности работ в рабочих и календарных днях каждого этапа работ.

Трудоёмкость оценивается по следующей формуле 2:

$$t_{ож\ i} = \frac{3t_{mini} + 2t_{max\ i}}{5}, \quad (2)$$

где  $t_{ож\ i}$  – это ожидаемая трудоёмкость  $i$ -ой работы (чел.-дни),

$t_{mini}$  – это минимально возможная трудоёмкость выполнения заданной  $i$ -ой работы (оптимистическая оценка) (чел.-дни),

$t_{max\ i}$  – это максимально возможная трудоёмкость выполнения заданной  $i$ -ой работы (пессимистическая оценка) (чел.-дни).

После оценки ожидаемой трудоёмкости работ, производится определение продолжительности каждой работы в рабочих днях по формуле 3:

$$T_{pi} = \frac{t_{ож\ i}}{ч_i}, \quad (3)$$

где  $T_{pi}$  – это продолжительность одной работы (раб. дни),

$t_{ож\ i}$  – это ожидаемая трудоёмкость выполнения одной работы (чел.-дни),

$ч_i$  – это численность исполнителей, выполняющих одновременно одну и ту же работу на этом этапе (чел.).

Для того чтобы в дальнейшем построить график работ с помощью диаграммы Ганта, необходимо также произвести перевод длительности работ из рабочих дней в календарные по формуле 4:

$$T_{ki} = T_{pi} \times k_{кал}, \quad (4)$$

где  $T_{ki}$  – это продолжительность выполнения  $i$ -й работы в календарных днях;

$T_{pi}$  – это продолжительность выполнения  $i$ -й работы в рабочих днях;

$k_{кал}$  – это коэффициент календарности, равный 1,48.

Коэффициент календарности  $k_{кал}$  рассчитывается по формуле 5:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}}, \quad (5)$$

где  $k_{\text{кал}}$  – коэффициент календарности;

$T_{\text{кал}}$  – количество календарных дней в году;

$T_{\text{кал}}$  – количество выходных дней в году;

$T_{\text{пр}}$  – количество праздничных дней в году.

С учётом того, что календарных дней в 2021 году 365, а сумма выходных и праздничных дней составляет 118 дней, коэффициент календарности равен  $k_{\text{кал}} = 1,48$ .

#### 4.2.3 Разработка графика проведения научного исследования

В таблице 6 приведены временные показатели научного исследования.

Таблица 6. Временные показатели проведения научного исследования

Наименование работы	Исполнители работы	Трудоёмкость работ, чел-дни		Длительность работ, дни		
		tmin	tmax	тож	Тр	Тк
Выбор научного руководителя бакалаврской работы	Бойко И.Д. Коровкин В.А.	1	2	1,4	1	1
Составление и утверждение темы бакалаврской работы	Бойко И.Д. Коровкин В.А.	1	3	1,8	1	1
Составление календарного плана-графика выполнения бакалаврской работы	Бойко И.Д.	1	2	1,4	1	1
Подбор и изучение литературы по теме бакалаврской работы	Бойко И.Д. Коровкин В.А.	5	10	7	4	6
Анализ предметной области	Бойко И.Д.	5	10	7	7	10
Проектирование информационной системы	Бойко И.Д.	8	14	10,4	10	15
Разработка клиентского приложения системы	Бойко И.Д.	5	12	7,8	8	12
Разработка сервера системы	Бойко И.Д.	6	12	8,4	8	12
Согласование выполненной работы с научным руководителем	Бойко И.Д. Коровкин В.А.	2	2	2	1	1
Выполнение других частей работы (финансовый менеджмент, социальная ответственность)	Бойко И.Д.	5	12	7,8	8	12
Подведение итогов, оформление работы	Бойко И.Д.	8	14	10,4	10	15
Итого	Бойко И.Д.	47	93	65,4	59	86

	Коровкин В.А.	9	17	12,2	7	9
--	---------------	---	----	------	---	---

На основе данных таблицы 6 был построен календарный план-график, представленный на рисунке 35.

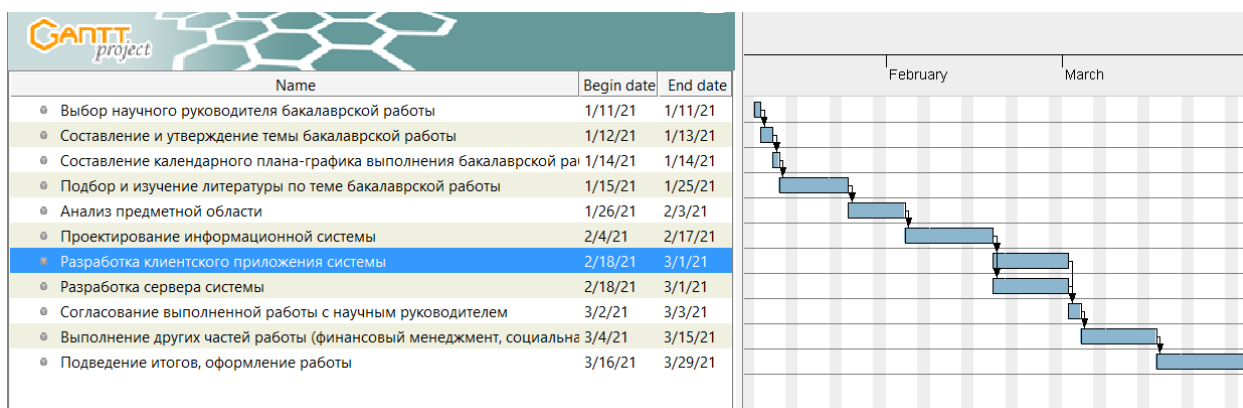


Рисунок 35 – Календарный план-график проведения работ

#### 4.2.4 Бюджет научно-технического исследования

В состав бюджета входит стоимость всех расходов, необходимых для выполнения работ по проекту. При формировании бюджета используется группировка затрат по следующим статьям:

- затраты на специальное оборудование;
- основная заработная плата исполнителей;
- дополнительная заработная плата исполнителей;
- отчисления во внебюджетные фонды;
- накладные расходы.

#### 4.2.5 Расчет затрат на специальное оборудование для научных (экспериментальных) целей

Данная статья затрат включает в себя затраты на приобретение специального оборудования. Также в эту статью включаются затраты по доставке и монтажу оборудования, равные 15% от его стоимости.

В ходе работы над проектом использовалось оборудование, имеющееся у исполнителей, соответственно необходим расчет его амортизации.



При создании информационной системы были использованы два персональных компьютера, стоимость 40000 и 60000 рублей, суммарная стоимость 100000 рублей.

Расчет амортизации ПК: первоначальная стоимость ПК 100000 рублей; срок полезного использования для машин офисных код 330.28.23.23 составляет 36 месяцев. Планируемое время использования ПК для написания ВКР - 6 месяцев.

Норма амортизация основных средств линейным способом рассчитывается по формуле 6:

$$A_n = \frac{1}{n} \times 100\%, \quad (6)$$

$n$  – установленный срок в месяцах;

$A_m$  – норма амортизации.

Тогда расчет амортизации ПК:

Норма амортизации:

$$A_n = \frac{1}{n} \times 100\% = \frac{1}{36} \times 100\% = 2,78\%$$

Ежемесячные амортизационные отчисления:

$$A_m = 100000 \times 0,0278 = 2780 \text{ рублей}$$

Итоговая сумма амортизации основных средств:

$$A = 2780 \times 6 = 16680 \text{ рублей}$$

Таким образом, сумма затрат на специальное оборудование составляет 16680 рублей, в виде амортизационных отчислений.

#### **4.2.6 Основная заработная плата исполнителей темы**

Данная статья затрат включает основную заработную плату, премии и доплаты всех исполнителей проекта. В качестве исполнителей проекта выступают студенты и научный руководитель.

Зарботная плата рассчитывается по формуле 7:

$$Z_{зп} = Z_{осн} + Z_{доп}, \quad (7)$$

где  $Z_{зп}$  – заработная плата исполнителя;

$Z_{осн}$  – основная заработная плата исполнителя;

Здоп – дополнительная заработная плата исполнителя (12%-15% от размера основной заработной платы).

Основную заработную плату можно получить по формуле 8:

$$Z_{\text{осн}} = Z_{\text{дн}} \times T_p \times (1 + K_{\text{пр}} + K_{\text{д}}) \times K_r \quad (8)$$

где  $Z_{\text{дн}}$  – среднедневная заработная плата, руб.

$K_{\text{пр}}$  – премиальный коэффициент (0,3);

$K_{\text{д}}$  – коэффициент доплат и надбавок (0,2-0,5);

$K_r$  – районный коэффициент (для Томска 1,3);

$T_p$  – продолжительность работ, выполняемых работником, раб. дни

где  $Z_{\text{дн}}$  – среднедневная заработная плата исполнителя;

$T_p$  – продолжительность работ, выполняемых исполнителем.

Среднедневную заработную плату можно получить по формуле 9:

$$Z_{\text{дн}} = \frac{Z_m \times M}{F_d}, \quad (9)$$

где  $Z_m$  – месячный должностной оклад исполнителя, рубли;

$M$  – количество месяцев работы равно:

При отпуске в 24 рабочих дня  $M = 11,2$  месяца, 5 – дневная неделя;

При отпуске в 48 рабочих дней  $M = 10,4$  месяца, 6 дневная неделя;

$F_d$  – действительный годовой фонд рабочего времени персонала по разработке.

Должностные оклады исполнителей проекта согласно приказу ТПУ представлены в таблице 7.

Таблица 7. Месячные должностные оклады исполнителей

Исполнитель	Районный коэффициент (для Томска)	Размер месячного должностного оклада без учета коэффициента, рубли
Научный руководитель (должность – доцент, степень – кандидат технических наук)	1,3	33664
Студент (ассистент, без степени)	1,3	21760

Баланс рабочего времени для 6-дневной рабочей недели представлен в таблице 8.

Таблица 8. Баланс рабочего времени (для 6-дневной недели)

Показатели рабочего времени	Дни
Календарные дни	365
Нерабочие дни (праздники/выходные)	118
Действительный годовой фонд рабочего времени	247

На основе формулы 8 и таблиц 7–8 была рассчитана среднедневная заработная плата:

$$Z_{\text{дн}}(\text{студент}) = \frac{21760 \times 10,4}{247} = 927,48 \text{ рубля}$$

$$Z_{\text{дн}}(\text{научный руководитель}) = \frac{33664 \times 10,4}{247} = 1434,86$$

Расчет затрат на основную заработную плату приведен в таблице 9.

Таблица 9. Затраты на основную заработную плату

Исполнители	Здн, руб.	Кпр	Кд	Кр	Тр	Зосн, руб.
Студент	927,48	0,3	0,2	1,3	59	106706,57
Научный руководитель	1434,86	0,3	0,2	1,3	7	19585,84
<b>Итого:</b>						126292,41

Итоговая сумма затрат на основную заработную плату составила 126292,41 руб.

#### 4.2.7 Дополнительная заработная плата исполнителей темы

Данная статья расходов учитывает величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда и выплат, связанных с обеспечением гарантий и компенсаций.

Расчёт дополнительной заработной платы осуществляется по формуле 10:

$$Z_{\text{доп}} = k_{\text{доп}} \times Z_{\text{осн}}, \quad (10)$$

где  $Z_{\text{доп}}$  – дополнительная заработная плата, рубли;

$k_{\text{доп}}$  – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12 – 0,15);

$Z_{\text{осн}}$  – основная заработная плата, рубли.

Расчет затрат на дополнительную заработную плату приведен в таблице 10.

Таблица 10. Затраты на дополнительную заработную плату

Исполнители	Зосн, руб.	Кдоп	Здоп, руб.
Студент	106706,57	0,12	12804,79
Научный руководитель	19585,84	0,12	2350,30
<b>Итого:</b>			15155,09

Итоговая сумма затрат на дополнительную заработную плату составила 15155,09 руб.

#### 4.2.8 Отчисления во внебюджетные фонды (страховые отчисления)

К отчислениям во внебюджетные фонды относятся отчисления:

- отчисления органам государственного социального страхования (ФСС);
- отчисления в пенсионный фонд (ПФ);
- отчисления медицинского страхования (ФФОМС).

Сумма отчислений во внебюджетные фонды рассчитывается на основе затрат на оплату труда исполнителей и может быть вычислена по формуле 11.

$$Z_{\text{внеб}} = k_{\text{внеб}} \times (Z_{\text{осн}} + Z_{\text{доп}}), \quad (11)$$

где  $k_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и др.);

Размер коэффициента определяется законодательно и в настоящее время согласно Федеральному закону от 24.07.2009 №212-ФЗ установлен в размере 30,2%.

Расчет затрат на отчисления во внебюджетные фонды приведен в таблице 11.

Таблица 11. Отчисления во внебюджетные фонды

Исполнители	Зосн, руб.	Здоп, руб.	Квнеб	Звнеб, руб.
Студент	106706,57	12804,79	0,302	36092,43
Научный руководитель	19585,84	2350,30	0,302	6624,71
<b>Итого:</b>				42717,14

Итоговая сумма отчислений во внебюджетные фонды составила 42434,25 руб.

#### 4.2.9 Накладные расходы

Накладные расходы – расходы на организацию, управление и обслуживание процесса производства товара, оказания услуги; носят комплексный характер. Накладные расходы вычисляются по формуле 12:

$$Z_{\text{нкл}} = \text{сумма статей}(1 - 5) \times k_{\text{нр}}, \quad (12)$$

где  $k_{\text{нр}}$  – коэффициент накладных расходов (16% от суммы затрат, подсчитанных выше).

Расчет накладных расходов приведен в таблице 12.

Таблица 12. Расчет накладных расходов

Статьи затрат	Сумма, руб.
Затраты на специальное оборудование	16680
Затраты на основную заработную плату	126292,41
Затраты на дополнительную заработную плату исполнителям проекта	15155,09
Затраты на отчисления во внебюджетные фонды	42717,14
Коэффициент накладных расходов	0,16
<b>Накладные расходы</b>	<b>32372,77</b>

Итоговая сумма накладных расходов составила 32372,77 руб.

#### 4.2.10 Формирование бюджета затрат научно-исследовательского проекта

После того, как была подсчитана каждая из статей расходов, можно приступить к формированию общего бюджета затрат проекта. Итоговый бюджет затрат представлен в таблице 13.

Таблица 13. Расчет бюджета затрат НТИ

Наименование статьи	Сумма, руб.	Удельный вес, %
Затраты на специальное оборудование	16680	7,2
Затраты на основную заработную плату	126292,41	54,3
Затраты на дополнительную заработную плату	15155,09	6,5
Страховые взносы	42717,14	18,2
Накладные расходы	32372,77	13,8
<b>Общий бюджет</b>	<b>233217,41</b>	<b>100</b>

Таким образом, общий бюджет НТИ составляет 233217,41 рубля.

### **4.3 Определение потенциального эффекта исследования**

В рамках данного раздела был проведен комплексный SWOT-анализ проекта, который позволил выявить его сильные и слабые стороны, а также определить соответствие его характеристик внешним факторам. Более того, были выявлены возможные пути дальнейшего развития и совершенствования проекта для повышения его конкурентоспособности. Также была произведена оценка качества и перспективности данного проекта.

Ещё одним ключевым моментом, рассмотренным в данном разделе, является планирование работ, выполняемых в рамках проекта. Составленный список необходимых работ с распределением исполнителей, а также вычисленная трудоемкость, позволили построить наглядный план-график работ в виде диаграммы Ганта. Это позволило рационально распорядиться временными ресурсами, отведенными на выполнение проекта.

Кроме того, был определен бюджет проекта. Определены затраты на заработные платы исполнителям с учетом страховых отчислений, рассчитаны накладные расходы, а также амортизация оборудования, задействованного в процессе выполнения проекта.

## **5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ**

### **Введение**

Объектом исследования являются архитектурные подходы разработки веб-приложений и разработка веб-платформы изучения иностранным языкам.

Выпускная квалификационная работа представляет собой реализацию микросервисной архитектуры на примере платформы для обучения иностранным языкам.

В данном разделе выпускной квалификационной работы рассмотрены опасные и вредные факторы, оказывающие влияние на производственную деятельность программиста. Также рассмотрены вопросы техники пожарной безопасности, правила поведения во время чрезвычайных ситуаций и способы предотвращения их возникновения.

### **5.1 Правовые и организационные вопросы обеспечения безопасности**

#### **5.1.1 Специальные правовые нормы трудового законодательства**

В трудовом кодексе РФ содержатся основные положения отношений между организацией и сотрудниками, включая оплату и нормирование труда, выходных, отпуска и так далее. Работа в офисе относится ко второй категории тяжести труда – работы выполняются при оптимальных условиях внешней производственной среды и при оптимальной величине физической, умственной и нервно-эмоциональной нагрузки.

Нормальная продолжительность рабочего времени не может превышать 40 часов в неделю. При работе в ночное время продолжительность рабочей смены сокращается на один час [7].

Сокращенная продолжительность рабочего времени устанавливается:

- для работников в возрасте до шестнадцати лет -не более 24 часов в неделю;
- для работников в возрасте от шестнадцати до восемнадцати лет - не более 35 часов в неделю;

- для работников, являющихся инвалидами I или II группы не более 35 часов в неделю;
- для работников, условия труда на рабочих местах которых по результатам специальной оценки условий труда отнесены к вредным условиям труда 3 или 4 степени или опасным условиям труда, -не более 36 часов в неделю.

Согласного трудовому кодексу РФ продолжительность рабочего времени конкретного работника устанавливается трудовым договором на основании отраслевого (межотраслевого) соглашения и коллективного договора с учетом результатов специальной оценки условий труда.

Работа в ночное время считается с 22 часов до 6 часов. Продолжительность работы (смены) в ночное время сокращается на один час без последующей отработки. Не сокращается продолжительность работы (смены) в ночное время для работников, которым установлена сокращенная продолжительность рабочего времени, а также для работников, принятых специально для работы в ночное время, если иное не предусмотрено коллективным договором [7].

### **5.1.2 Организационные мероприятия при компоновке рабочей зоны**

Выполнение работ производится в основном в сидячем положении, поэтому следует выполнять все рекомендуемые требования ГОСТ 12.2.032–78 о рабочем месте при выполнении работ сидя. Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и т. д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы [8].

Конструкцией производственного оборудования и рабочего места должно быть обеспечено оптимальное положение работающего, которое достигается регулированием:

1. высоты рабочей поверхности, сиденья и пространства для ног;



2. высоты сиденья и подставки для ног (при нерегулируемой высоте рабочей поверхности).

В соответствии с ГОСТ 12.2.032-78. «Система стандартов безопасности труда». Рабочее место при выполнении работ сидя» рабочий стол может быть любой конструкции, отвечающей современным требованиям эргономики и позволяющей удобно разместить на рабочей поверхности оборудование с учетом его количества, размеров и характер выполняемой работы [8].

Выполнение требований на данном рабочем месте отражено ниже в таблице 14, согласно СанПиН 2.2.2/2.4.1340-03 и ГОСТ 12.2.032-78.

Таблица 14 – Требования к организации рабочего места при работе с ПЭВМ

<b>Требование</b>	<b>Требуемое значение</b>	<b>Значение параметров в помещении</b>
Высота рабочей поверхности стола	Регулируемая высота (680-800мм) Нерегулируемая высота (725мм)	Нерегулируемая высота (700мм)
Рабочий стул	Подъемно-поворотный, регулируемый по высоте и углу наклона спинки	Соответствует
Расположение монитора от глаз пользователя	600-700мм	Соответствует

Средства отображения информации необходимо группировать и располагать группы относительно друг друга в соответствии с последовательностью их использования. При этом средства отображения информации необходимо размещать в пределах групп так, чтобы последовательность их использования осуществлялась слева направо или сверху вниз.

Лицевые поверхности мониторов должны располагаться перпендикулярно взору работника. Допускаемое отклонение от этой плоскости – не более 45°; допускаемый угол отклонения линии взора от

нормальной - не более 25° для стрелочных индикаторов и 30° для индикаторов с плоским изображением [9]

Клавиатуру следует располагать на поверхности стола на расстоянии 100—300 мм от края, обращенного к пользователю, или на специальной регулируемой по высоте рабочей поверхности, отделенной от основной столешницы [10].

## **5.2 Производственная безопасность**

Первопричиной всех травм и заболеваний, связанных с процессом труда, является неблагоприятное воздействие на организм занятого трудом человека тех или иных факторов производственной среды и трудового процесса. Это воздействие, приводящее в различных обстоятельствах к различным результирующим последствиям, зависит от наличия в условиях труда того или иного фактора, его потенциально неблагоприятных для организма человека свойств, возможности его прямого или опосредованного действия на организм, характера реагирования организма в зависимости от интенсивности и длительности воздействия (экспозиции) данного фактора. Производственные факторы являются частным случаем факторов окружающей человека среды обитания и человеческой деятельности, связанных и (или) порождаемых производственной и трудовой деятельностью.

Неблагоприятные производственные факторы по результирующему воздействию на организм работающего человека подразделяют:

- на вредные производственные факторы, то есть факторы, приводящие к заболеванию, в том числе усугубляющие уже имеющиеся заболевания;
- опасные производственные факторы, то есть факторы, приводящие к травме, в том числе смертельной [11].

В таблице 15 представлены возможные вредные и опасные факторы, возникающие при работе за ПК.

Таблица 15 – Вредные и опасные факторы по ГОСТ 12.0.003-2015, возникающие при работе за персональным компьютером

Наименование видов работ	Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
<b>Вредные факторы</b>		
Работа за персональным компьютером	Отклонение показателей микроклимата (температуры и влажности воздуха)	СанПиН 2.2.2/2.4.1340-03 СанПиН 2.2.4.548-96
	Отсутствие или недостаток необходимого естественного и искусственного освещения	СП 52.13330.2016
	Превышение уровня шума	СанПиН 2.2.2/2.4.1340-03
<b>Опасные факторы</b>		
Работа за персональным компьютером	Опасность поражения электрическим током	ГОСТ 12.1.038-82 ГОСТ 12.1.019-2017 ГОСТ Р 12.1.019-2009

### 5.2.1 Вредные производственные факторы

#### 5.2.1.1 Отклонение показателей микроклимата в помещении

Показателями, характеризующими микроклимат, являются: температура воздуха, температура поверхностей (ограждающих конструкций, устройств, технологического оборудования), влажность воздуха, скорость движения воздуха, тепловое облучение (при наличии источников лучистого тепла).

Оптимальные микроклиматические условия установлены по критериям оптимального теплового и функционального состояния человека. Они обеспечивают общее и локальное ощущение теплового комфорта в течение 8-часовой рабочей смены при минимальном напряжении механизмов терморегуляции, не вызывают отклонений в состоянии здоровья, создают предпосылки для высокого уровня работоспособности и являются предпочтительными на рабочих местах.

Работа программиста относится к категории Ia, которые производятся сидя и сопровождаются незначительным физическим напряжением. Оптимальные параметры микроклимата на рабочих местах должны

соответствовать величинам, приведённым в таблице 16, применительно к выполнению работ в холодный и тёплый периоды года [12].

Таблица 16 – Оптимальные параметры микроклимата на рабочих местах

Период года	Категория работ по уровню энергозатрат, Вт	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia (до 139)	22-24	21-25	60-40	0,1
Тёплый	Ia (до 139)	23-25	22-26	60-40	0,1

В кабинете поддерживается температура равная 22-24 С°, при относительной влажности в 50-60%, что является нормой по требованиям СанПиН 2.2.2/2.4.1340-03. Для поддержки такого состояния необходимо проводить в помещении ежедневную влажную уборку и систематическое проветривание [10].

#### **5.2.1.2 Недостаточная освещенность рабочей зоны**

Рабочие столы следует размещать таким образом, чтобы мониторы были расположены боковой стороной к источникам естественного света. Следует применять системы комбинированного освещения.

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300–500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк.

Следует ограничивать прямую блескость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м<sup>2</sup>. Светильники местного освещения должны иметь непросвечивающий отражатель с защитным углом не менее 40 градусов.

В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ) [13].

### 5.2.1.3 Превышение уровня шума

Звуковое давление – переменная составляющая давления воздуха или газа, возникающая в результате звуковых колебаний, Па.

Допустимый уровень шума – это уровень, который не вызывает у человека значительного беспокойства и существенных изменений показателей функционального состояния систем и анализаторов, чувствительных к шуму [14].

Характеристикой постоянного шума на рабочих местах являются уровни звукового давления в дБ в октавных полосах со среднегеометрическими частотами 31,5; 63; 125; 250; 500; 1000; 2000; 4000; 8000 Гц, определяемые по формуле:

$$L = 20 \lg \frac{P}{P_0} \quad (1)$$

где  $P$  – среднеквадратичная величина звукового давления, Па;

$P_0$  – исходное значение звукового давления в воздухе, равное  $2 \times 10^{-5}$  Па.

Предельно допустимые уровни звукового давления в октавных полосах частот, уровни звука и эквивалентные уровни звука для трудовой деятельности программиста, разработанные с учетом категории тяжести и напряженности труда, представлены в таблице 17.

Таблица 17 – Предельно допустимые уровни звукового давления для программиста

Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука и эквивалентные уровни звука (дБА)
31,5	63	125	250	500	1000	2000	4000	8000	
86	71	61	54	49	45	42	40	38	<b>50</b>

Для исследуемого объекта (персональный компьютер) основными источниками шумов являются составляющие компьютера:

- вентилятор блока питания;
- вентилятор центрального процессора;
- вентилятор на высокопроизводительной видеокарте.

Для снижения уровня шума в помещении обеспечена вентиляция системного блока. Со стороны вентиляционных отверстий системного блока обеспечено не менее 20-30 см свободного пространства.

## **5.2.2 Опасные производственные факторы**

### **5.2.2.1 Опасность поражения электрическим током**

При работе за компьютером необходимо учитывать требования электробезопасности, так как это может привести к негативным последствиям, таким как, поражение электрическим током и возникновение пожара.

Электробезопасность должна обеспечиваться:

- конструкцией электроустановок и архитектурно-планировочными решениями;
- организацией технологических процессов;
- техническими способами и средствами защиты;
- организационными и техническими мероприятиями при производстве работ;
- электрозащитными средствами, средствами защиты от электрических и магнитных полей и другими средствами индивидуальной защиты, применяемыми при эксплуатации электроустановок;
- организацией технического обслуживания электроустановок.

Электрический ток оказывает на человека термическое, электролитическое, биологическое и механическое воздействие. Действие электрического тока на человека приводит к травмам или гибели людей. Для переменного тока частотой 50 Гц допустимое значение напряжения прикосновения составляет 2 В, а силы тока –0,3 мА, для тока частотой 400 Гц, соответственно –2 В и 0,4 мА, для постоянного тока –8 В и 1 мА [15].

В помещении используются для питания приборов напряжение 220 В переменного тока с частотой 50 Гц. Обязательны следующие предосторожности:

- перед началом работы убедиться, что выключатели, розетки закреплены и не имеют оголенных токоведущих частей;
- не включать в сеть компьютеры и другую оргтехнику со снятыми крышками;
- при обнаружении неисправности компьютера необходимо выключить его и отключить от сети;
- запрещается загромождать рабочее место лишними предметами;
- при несчастном случае необходимо немедленно отключить питание электроустановки, вызвать скорую помощь и оказать пострадавшему первую помощь до прибытия врача, согласно правилам;
- дальнейшее продолжение работы возможно только после устранения причины поражения электрическим током.

### **5.3 Экологическая безопасность**

Большое количество процессов, операций и материалов, используемых при изготовлении электронных средств, являются источниками огромного количества веществ, имеющих неблагоприятное воздействие на человека и биосферу. При изготовлении элементной базы, электронных изделий, при обработке, выращивании полупроводниковых кристаллов, при изготовлении интегральных схем, в процессе гальванического производства утилизация исходных материалов часто происходит с низким коэффициентом использования, огромное количество их идет в отходы, попадая в атмосферу, гидросферу, загрязняя почву. Таким образом, наряду с истощением природных запасов дефицитных материалов происходит загрязнение окружающей среды, что ведет к губительным последствиям для отдельных экосистем и биосферы в целом.

На данном рабочем месте выявлен предполагаемый источник загрязнения окружающей среды, а именно воздействие на литосферу в результате образования отходов при поломке предметов вычислительной техники и оргтехники.

Согласно ГОСТ Р 53692-2009, вышедшее из строя ПЭВМ и сопутствующая оргтехника относится к IV классу опасности и подлежит специальной утилизации.

Первым этапом является утилизация обезвреженных (инертных) отходов. Во время утилизации может быть произведена переработка бракованных или вышедших из употребления видов продукции, изделий, их составных частей и отходов от них путем разборки (разукрупнения), переплавки, использования других технологий с обеспечением рециркуляции (восстановления) органической и неорганической составляющих [16].

Вторым этапом является безопасное размещение отходов I-IV классов опасности на соответствующих полигонах или уничтожение.

#### **5.4 Безопасность в чрезвычайных ситуациях**

Возможными чрезвычайными ситуациями на рабочем месте могут быть землетрясение, наводнение, пожар и другие. Наиболее типичной чрезвычайной ситуацией является пожар, так как специфика работы заключается во взаимодействии с вычислительной техникой, что может являться причиной возгорания и угрожать жизни людей.

Причинами возгорания при работе с вычислительной техникой могут быть:

- токи короткого замыкания;
- перегрев и дальнейшее воспламенение;
- неисправность работы техники;
- повреждённая электросеть.

Причинами возникновения КЗ могут быть ошибки при проектировании, старение изоляции, увлажнение изоляции и механические перегрузки. Пожарная опасность при перегрузках – чрезмерное нагревание отдельных элементов, которое может происходить при ошибках проектирования в случае длительного прохождения тока, превышающего номинальное значение.



Рабочее место программиста должно соответствовать требованиям ФЗ Технический регламент по ПБ и норм пожарной безопасности (НПБ 105-03) [17] и удовлетворять требованиям по предотвращению и тушению пожара по ГОСТ 12.1.004-91 [18] и СНиП 21-01-97 [19].

В случае возникновения пожара в здании автоматически срабатывают датчики дыма, и звуковая система оповещает всех сотрудников о немедленной эвакуации из здания, и далее сотрудники направляются на выход в соответствии с планом эвакуации при пожарах и других ЧС. На этаже находится два огнетушителя и пожарных крана, а также два эвакуационных выхода. План эвакуации при пожаре и других ЧС представлен на рисунке 36.



Рисунок 36 – План эвакуации при пожаре и других ЧС

## Вывод

В данном разделе были рассмотрены правовые нормы трудового законодательства, организационные мероприятия при компоновке рабочей зоны, различные вредные и опасные факторы в производственной среде, а также рассмотрена типичная чрезвычайная ситуация, учитывая специфику

выпускной квалификационной работы при работе с вычислительной техникой. Каждый подраздел составлен с учётом государственных стандартов, сводов правил, трудового кодекса РФ и санитарных норм.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работой были проанализированы существующие платформы и их возможности для конечного клиента, описаны функциональные и технические требования к платформе, показаны преимущества выбранной архитектуры для разработки разрабатываемой платформы, описаны сервисы платформы и спроектирована микросервисная архитектура платформы для обучения иностранным языкам на основе стека технологий Spring Cloud Netflix.

Стек, находящийся в основе проекта, позволит разработать архитектуру, которая сможет горизонтально и вертикально масштабироваться и распределять нагрузку между действующими сервисами.

Разработана структура баз данных для сервисов словаря и тестирования знаний пользователя. Для каждой сущности выполнены CRUD-операции, что позволяет управлять базой данных с помощью внешних HTTP-запросов.

Сконфигурировано взаимодействие между сервисами для получения метрик и передачи данных от пользователя до конечного сервиса и обратно, а также настроен механизм защиты API от неавторизованных пользователей и запросов со сторонних клиентов. На данный момент пользователь не сможет выполнить запрос с клиента, который не внесён реестр разрешённых клиентов.

## СПИСОК ЛИТЕРАТУРЫ

1. Проблемы преподавания английского языка: глобальные перспективы и местные реалии <https://www.elibrary.ru/item.asp?id=41441018> (дата обращения 31.05.2021)
2. Participant behavior and content of the online foreign languages learning and teaching platform <https://www.sciencedirect.com/science/article/pii/S0747563215003155> (дата обращения 31.05.2021)
3. A Comparison of Undergraduate Students' English Vocabulary Learning: Using Mobile Phones and Flash Cards <https://eric.ed.gov/?id=EJ898010> (дата обращения 31.05.2021)
4. Pattern: Decompose by business capability <https://microservices.io/patterns/decomposition/decompose-by-business-capability.html> (дата обращения 31.05.2021)
5. Netflix OSS, Spring Cloud, or Kubernetes? How About All of Them! <https://blog.christianposta.com/microservices/netflix-oss-or-kubernetes-how-about-both/> (дата обращения 31.05.2021)
6. Rubygarage URL: <https://rubygarage.org/blog/monolith-soa-microservices-serverless> (дата обращения 25.12.2020);
7. «Трудовой кодекс Российской Федерации» от 30.12.2001 N 197-ФЗ (ред. от 01.04.2019)
8. ГОСТ 12.2.032-78 «Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования»
9. ГОСТ 22269-76 «Рабочее место оператора. Взаимное расположение элементов рабочего места»
10. СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» (с изменениями на 21 июня 2016 года).

11. ГОСТ 12.0.003-2015 «Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация»
12. СанПиН 2.2.4.548-96. «Гигиенические требования к микроклимату производственных помещений».
13. СП 52.13330.2016 «Естественное и искусственное освещение»
14. СН 2.2.4/2.1.8.562-96 «Физические факторы производственной среды. Физические факторы окружающей природной среды»
15. ГОСТ Р 12.1.019-2009. «Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты».
16. ГОСТ Р 53692-2009 «Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов»
17. НПБ 105-03 «Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности».
18. ГОСТ 12.1.004-91 «Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Общие требования (с Изменением N 1)».
19. СНиП 21-01-97\* «Пожарная безопасность зданий и сооружений (с Изменениями N 1, 2)».