



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 09.04.01 Информатика и вычислительная техника
ООП/ОПОП Анализ данных в разработке программного обеспечения
Отделение школы (НОЦ) Отделение информационных технологий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРАНТА

Тема работы

Приложение для анализа и экспорта графа сообществ социальных сетей

УДК 004.451:316.472.4

Обучающийся

Группа	ФИО	Подпись	Дата
8ВМ1И	Сунцов Кирилл Сергеевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Ковин Р. В.	к.т.н.		

Language mentor

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Ануфриева Т. Н.			

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Спицына Л. Ю.	к.э.н., доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Антоневич О. А.	к.б.н., доцент		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП, должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Зарубин А. Г.	к.х.н., доцент		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП
по направлению 09.04.01 Информатика и вычислительная техника

Код компетенции	Наименование компетенции
Универсальные компетенции	
УК(У)-1	Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий
УК(У)-2	Способен управлять проектом на всех этапах его жизненного цикла
УК(У)-3	Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели
УК(У)-4	Способен применять современные коммуникативные технологии, в том числе на иностранном (-ых) языке (-ах), для академического и профессионального взаимодействия
УК(У)-5	Способен анализировать и учитывать разнообразие культур в процессе межкультурного взаимодействия
УК(У)-6	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки
Общепрофессиональные компетенции	
ОПК(У)-1	Способен самостоятельно приобретать, развивать и применять математические, естественно-научные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте
ОПК(У)-2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач
ОПК(У)-3	Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями
ОПК(У)-4	Способен применять на практике новые научные принципы и методы исследований
ОПК(У)-5	Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем
ОПК(У)-6	Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования
ОПК(У)-7	Способен адаптировать зарубежные комплексы обработки информации и автоматизированного проектирования к нуждам отечественных предприятий
ОПК(У)-8	Способен осуществлять эффективное управление разработкой программных средств и проектов
Профессиональные компетенции	
ПК(У)-1	Способен разрабатывать и администрировать системы управления базами данных
ПК(У)-2	Способен проектировать сложные пользовательские интерфейсы
ПК(У)-3	Способен управлять процессами и проектами по созданию (модификации) информационных ресурсов
ПК(У)-4	Способен осуществлять руководство разработкой комплексных проектов на всех стадиях и этапах выполнения работ
ПК(У)-5	Способен проектировать и организовывать учебный процесс по образовательным программам с использованием современных образовательных технологий



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 09.04.01 Информатика и вычислительная техника
ООП/ОПОП Анализ данных в разработке программного обеспечения
Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП

А. Г. Зарубин

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

В форме:

магистерской диссертации

Студенту:

Группа	ФИО
8ВМ1И	Сунцову Кириллу Сергеевичу

Тема работы:

«Приложение для анализа и экспорта графа сообществ социальных сетей»

Утверждена приказом директора (дата, номер) от 06.02.2023 г. №34–47/с

Срок сдачи студентом выполненной работы:

01.06.2023 г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе

(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.).

Объектом проектирования и разработки является веб-приложение для анализа и экспорта графа сообществ социальных сетей «ВКонтакте» и «Одноклассники».

Перечень подлежащих исследованию, проектированию и разработке вопросов

(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).

1. Исследование предметной области.
2. Анализ требований к приложению.
3. Проектирование и реализация приложения.
4. Работа над разделом по финансовому менеджменту, ресурсоэффективности и ресурсосбережения.
5. Работа над разделом по социальной ответственности.

Перечень графического материала <i>(с точным указанием обязательных чертежей)</i>	<ol style="list-style-type: none"> 1. Схема архитектуры приложения. 2. Дизайн-макеты интерфейса. 3. Матрица SWOT-анализа. 4. Диаграмма Ганта.
---	---

Консультанты по разделам выпускной квалификационной работы
(с указанием разделов)

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Спицына Л. Ю., доцент
Социальная ответственность	Антоневич О. А., доцент
Английский язык	Ануфриева Т. Н., старший преподаватель

Названия разделов, которые должны быть написаны на русском и иностранном языках:

Software Implementation of the Information System

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	30.01.2023
---	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Ковин Роман Владимирович	к.т.н.		30.01.2023

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ1И	Сунцов Кирилл Сергеевич		30.01.2023



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 09.04.01 Информатика и вычислительная техника
ООП/ОПОП Анализ данных в разработке программного обеспечения
Отделение школы (НОЦ) Отделение информационных технологий
Период выполнения осенний / весенний семестр 2022/2023 учебного года

Форма представления работы:

ВКР магистра

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы**

Обучающегося:

Группа	ФИО
8ВМ1И	Сунцова Кирилла Сергеевича

Тема работы:

Разработка приложения для анализа и экспорта графа сообществ социальных сетей
--

Срок сдачи студентом выполненной работы:	01.06.2023
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
15.02.2023	Исследование предметной области.	20
20.02.2023	Анализ требований к приложению.	20
26.05.2023	Проектирование и реализация приложения.	20
16.05.2023	Работа над разделом по финансовому менеджменту, ресурсоэффективности и ресурсосбережения.	10
22.05.2023	Работа над разделом по социальной ответственности.	10

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Ковин Р. В.	К.Т.Н.		

СОГЛАСОВАНО:

Руководитель ООП

Руководитель ООП, должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Зарубин А. Г.	к.х.н., доцент		

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

Студенту:

Группа		ФИО	
8ВМ1И		Сунцов Кириллу Сергеевичу	
Школа	Инженерная школа информационных технологий и робототехники	Отделение (НОЦ)	Отделение информационных технологий
Уровень образования	магистратура	Направление/специальность	09.04.01 Информатика и вычислительная техника

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

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Бюджет проекта – не более 600000 руб., в т. ч. затраты по оплате труда – не более 500000 руб. Стоимость материальных и интеллектуальных ресурсов определялась согласно прейскурантам компаний Оклад студента-программиста – 45000 р.
2. Нормы и нормативы расходования ресурсов	Накладные расходы 16 %; Районный коэффициент 30%; Норма амортизации ПЭВМ 33,33%; Норма амортизации ПО 20%
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Коэффициент отчислений на выплаты во внебюджетные фонды 30%

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого и инновационного потенциала НТИ	Анализ потенциальных потребителей результатов исследования; Анализ конкурентных технических решений; FAST анализ; SWOT анализ; Оценка готовности проекта к коммерциализации
2. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	Структура работ. План проекта. Бюджет научного исследования. Риски проекта.
3. Определение ресурсной, финансовой, экономической эффективности	Абсолютная эффективность исследования. Сравнительная эффективность исследования.

Перечень графического материала (с точным указанием обязательных чертежей):

1. «Портрет» потребителя результатов НТИ
2. Сегментирование рынка
3. Оценка конкурентоспособности технических решений
4. Матрица SWOT
5. График проведения и бюджет НТИ
6. Оценка ресурсной, финансовой и экономической эффективности НТИ
7. Потенциальные риски

Дата выдачи задания для раздела по линейному графику	30.01.2023
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Спицына Любовь Юрьевна	к.э.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ1И	Сунцов Кирилл Сергеевич		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
8ВМ1И		Сунцову Кириллу Сергеевичу	
Школа	Инженерная школа информационных технологий и робототехники	Отделение (НОЦ)	Отделение информационных технологий
Уровень образования	магистратура	Направление/специальность	09.04.01 Информатика и вычислительная техника

Тема ВКР:

Приложение для анализа и экспорта графа сообществ социальных сетей	
Исходные данные к разделу «Социальная ответственность»:	
<p>Введение Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации</p>	<p><i>Объект исследования:</i> программные средства сбора и анализа данных сообществ социальных сетей <i>Область применения:</i> исследовательская деятельность, маркетинг <i>Рабочая зона:</i> офис <i>Размеры помещения:</i> 3*6 м <i>Количество и наименование оборудования рабочей зоны:</i> ПЭВМ – 1 шт. <i>Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне:</i> написание программного кода</p>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<p>1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения: специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; организационные мероприятия при компоновке рабочей зоны.</p>	<p>1. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 09.03.2021). 2. Федеральный закон от 28 декабря 2013 г. N 426-ФЗ «О специальной оценке условий труда». 3. ГОСТ 12.2.032–78 Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования.</p>
<p>2. Производственная безопасность при разработке: Анализ выявленных вредных и опасных производственных факторов Расчет уровня опасного или вредного производственного фактора</p>	<p>Вредные производственные факторы:</p> <ol style="list-style-type: none"> Недостаточная освещённость рабочей зоны. Отсутствие или недостаток естественного света. Повышенный уровень шума. Повышенный уровень электромагнитных излучений. Повышенная напряжённость электрического поля. Производственные факторы, связанные с аномальными микроклиматическими параметрами воздушной среды на местонахождении работающего. Статические перегрузки. <p>Опасные производственные факторы: производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий.</p> <p>Требуемые средства коллективной и индивидуальной защиты от выявленных факторов: наушники, устройства для вентиляции и очистки воздуха, источники света, устройства улавливания и очистки воздуха и жидкостей; Расчет: повышенный уровень шума.</p>
<p>3. Экологическая безопасность при разработке проектного решения</p>	<p>Воздействие объекта на атмосферу, гидросферу, селитебную зону не происходит.</p>

	В работе проведён анализ воздействия на литосферу (образование отходов при выходе из строя ПК, возникновения отходов при печати и утилизации ламп).
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения:	<p>Возможные ЧС:</p> <ol style="list-style-type: none"> 1. техногенные (взрывы, пожары, обрушение помещений); 2. природные (наводнения, ураганы); 3. социальные (террор). <p>Наиболее типичная ЧС: пожар.</p>

Дата выдачи задания для раздела по линейному графику	01.03.2023
--	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Антоневич Ольга Алексеевна	к.б.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ1И	Сунцов Кирилл Сергеевич		

РЕФЕРАТ

Выпускная квалификационная работа 122 страниц, 6 рисунков, 31 таблица, 52 источника, 10 приложений.

Ключевые слова: веб-приложение, социальные сети, теория графов, библиотека VivaGraph, .NET 6.

Целью работы является создание интерактивной визуализации графа сообществ социальных сетей ВКонтакте и Одноклассники.

Объектом исследования данной работы являются социальные сети и сообщества, которые на них существуют. Предметом исследования является разработка приложения для анализа и экспорта графа сообществ социальных сетей.

В результате исследования было создано веб-приложение, предназначенное для визуализации и анализа взвешенного графа сообществ социальной сети ВКонтакте. Оно позволяет определить количество общих членов между двумя сообществами, что может быть полезно для анализа аудитории сообществ. Если два сообщества с разной тематикой имеют много общих членов, это может указывать на то, что эти тематики привлекают одну и ту же аудиторию.

Практическая значимость работы состоит в том, что созданное приложение может использоваться как основа для коммерческих и некоммерческих маркетинговых проектов, направленных на комплексный анализ интересов аудитории сообществ социальных сетей.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	15
ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ	18
1.1. Существующие средства анализа аудитории.....	19
1.2. Анализ требований.....	20
1.2.1. Функциональные требования	20
1.2.2. Нефункциональные требования	20
1.2.3. Требования к установке и развертыванию:.....	21
ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	22
2.1. Используемые методы анализа	22
2.2. Разработка архитектуры приложения	23
2.3. Существующие виды архитектуры	23
2.4. Описание архитектуры приложения	24
2.5. Выбор используемых технологий	26
2.6. Описание механизма сбора данных	28
2.6.1. Описание работы программного интерфейса сервиса «ВКонтакте»	28
2.6.2. Описание работы программного интерфейса сервиса «Одноклассники»	28
2.7. Обезличивание данных.....	29
ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ	31
3.1. Реализация серверной части	31
3.1.1. Организация взаимодействия с программным интерфейсом «ВКонтакте»	31

3.1.2. Организация взаимодействия с программным интерфейсом «Одноклассники»	33
3.2. Реализация клиентской части	33
3.2.1. Выбор библиотеки визуализации графов.....	33
3.2.2. Налаживание взаимодействия с серверной частью	33
3.2.3. Реализация функции анализа графа.....	35
3.3. Реализация функции экспорта графа	38
3.4. Контейнеризация веб-приложения.....	39
3.5. Хранение исходного кода веб-приложения.....	39
3.6. Сборка веб-приложения	42
3.7. Публикация веб-приложения.....	44
3.8. Выводы по разделу.....	44
ГЛАВА 4. ФИНАНСОВЫЙ	МЕНЕДЖМЕНТ,
РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ	46
Введение.....	46
4.1. Оценка коммерческого и инновационного потенциала НТИ.....	46
4.1.1. Потенциальные потребители результатов исследования	46
4.1.2. Анализ конкурентных технических решений.....	48
4.1.3. SWOT анализ.....	50
4.1.4. Оценка готовности проекта к коммерциализации	53
4.2. Инициация проекта	55
4.2.1. Цели и результат проекта.....	55
4.2.2. Организационная структура проекта.....	57
4.2.3. Ограничения и допущения проекта	58
4.3. Планирование управления НТИ	58

4.3.1. Иерархическая структура работ	59
4.3.2. План проекта	59
4.3.3. Бюджет НТИ.....	63
4.4. Риски проекта	69
4.5. Определение ресурсной, финансовой, экономической эффективности	70
4.6. Выводы по разделу.....	72
ГЛАВА 5. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ	74
Введение.....	74
5.1. Правовые и организационные вопросы обеспечения безопасности. 74	
5.1.1. Особенности законодательного регулирования проектных решений	74
5.1.2. Организационные мероприятия при компоновке рабочей зоны	75
5.2. Производственная безопасность.....	76
5.2.1. Недостаточная освещённость рабочей зоны; отсутствие или недостаток естественного света.....	77
5.2.2. Повышенный уровень шума.....	78
5.2.3. Повышенный уровень электромагнитных излучений	80
5.2.4. Производственные факторы, связанные с аномальными микроклиматическими параметрами воздушной среды на местонахождении работающего	80
5.2.5. Статические перегрузки.....	81
5.2.6. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов	82
5.3. Экологическая безопасность.....	83
5.3.1. Воздействие на литосферу.....	83

5.4. Безопасность в чрезвычайных ситуациях.....	84
5.4.1. Пожарная безопасность.....	84
5.5. Выводы по разделу.....	85
ЗАКЛЮЧЕНИЕ	88
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	89
ПРИЛОЖЕНИЯ.....	94
ПРИЛОЖЕНИЕ А Листинг агрегатора пакетных запросов.....	94
ПРИЛОЖЕНИЕ Б Листинг кэширующего модуля	95
ПРИЛОЖЕНИЕ В Листинг модуля взаимодействия с VK API.....	97
ПРИЛОЖЕНИЕ Г Листинг класса NetworkGraph.....	99
ПРИЛОЖЕНИЕ Д Листинг класса VkNodesLoader	101
ПРИЛОЖЕНИЕ Е Листинг скрипта анализа графа	102
ПРИЛОЖЕНИЕ Ж Листинг страницы анализа примера графа.....	103
ПРИЛОЖЕНИЕ З Листинг файла конфигурации CI/CD.....	106
ПРИЛОЖЕНИЕ И (справочное).....	107
ПРИЛОЖЕНИЕ К Диаграмма активностей	122

АННОТАЦИЯ

В настоящее время социальные сети являются важным маркетинговым каналом. При подборе тематики контента сообщества (группы, канала) может быть важно учитывать не только общие данные об аудитории, такие как возраст и место жительства, но и тематики других сообществ, на которые подписаны пользователи. В рамках работы производится разработка средства сбора, анализа и визуализации данных о других сообществах, на которые подписаны члены целевой аудитории какого-либо сообщества.

Результатом работы является веб-приложение, позволяющее построить граф сообществ социальной сети ВКонтакте, опубликованное в сети Интернет и доступное пользователям.

В пояснительной записке 122 страниц, 6 рисунков, 31 таблица, 52 источника, 10 приложений.

ВВЕДЕНИЕ

Каждый день миллионы пользователей по всему миру проводят в социальных сетях многие часы, общаясь, делаясь информацией и покупая товары. Это делает социальные сети одним из самых мощных маркетинговых инструментов, которые доступны в настоящее время. В связи с этим, важно не только привлекать новых пользователей, но и укреплять связь с уже имеющимися, для чего необходимо понимать их интересы и потребности. Подбор тематики контента является одним из ключевых факторов, который может повлиять на успех маркетинговой кампании в социальных сетях.

Традиционный подход к подбору тематики контента основывается на общих данных об аудитории, таких как возраст, пол и место жительства [1]. Однако, эти данные могут быть недостаточными для определения релевантности контента для конкретного сообщества в социальной сети. Это связано с тем, что пользователи социальных сетей могут подписываться на различные сообщества, которые могут быть не связаны напрямую с их личными данными. В результате, контент, который был бы релевантным для аудитории, основываясь только на общих данных, может не привлекать достаточно внимания и не иметь нужного влияния.

Для решения проблемы определения интересов и потребностей целевой аудитории в социальных сетях, в рамках данной работы производится разработка приложения для анализа и экспорта графа сообществ. Это приложение будет собирать данные о сообществах, на которые подписаны пользователи из целевой аудитории, анализировать их и представлять в удобном виде для использования в маркетинговых кампаниях. Такой подход позволит лучше понимать интересы и потребности целевой аудитории, подбирать более релевантную тематику контента и, в итоге, увеличивать эффективность маркетинговой кампании в социальных сетях.

Для реализации проекта будет использоваться язык программирования Python, JavaScript и C#, а также модули и библиотеки, такие как NetworkX, WebAPI, jQuery. Предполагается, что разработанное приложение будет иметь

простой и понятный интерфейс, который позволит пользователям легко собирать и анализировать данные о сообществах в социальных сетях, используя графы.

Разделом дискретной математики, изучающим графы, является теория графов. В ней граф представляется как множество вершин (узлов), соединенных ребрами или дугами [2]. Теория графов применяется при анализе систем, состоящих из набора вершин и связей между ними. Такими системами являются компьютерные сети, сети банковских операций, биологические сети, социальные сети и многие другие. Имея схожую структуру, они могут быть исследованы с использованием похожих универсальных технологий, одной из которых является теория графов.

Полезным свойством теории графов является наглядность, поскольку граф можно визуализировать. Визуализация помогает получить качественную и количественную оценку сетей.

Целью работы является создание интерактивной визуализации графа сообществ социальных сетей ВКонтакте и Одноклассники.

Задачами работы являются:

1. Исследовать предметную область.
2. Изучить работу программного интерфейса «ВКонтакте» и «Одноклассников».
3. Реализовать серверную часть веб-приложения.
4. Реализовать клиентскую часть веб-приложения.
5. Реализовать функцию анализа графа.
6. Реализовать функцию экспорта графа.
7. Контейнеризировать веб-приложение.
8. Опубликовать веб-приложение.

Объектом исследования данной работы являются социальные сети и сообщества, которые на них существуют. В качестве конкретного примера объектом исследования может выступать популярная социальная сеть ВКонтакте и её сообщества.

Предметом исследования является разработка приложения для анализа и экспорта графа сообществ социальных сетей.

ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Социальные сети стали неотъемлемой частью нашей жизни, их значимость в повседневной коммуникации и обмене информацией высока. В настоящее время более половины населения Земли используют социальные сети, что делает их привлекательным каналом для продвижения продуктов и услуг, а также для анализа поведения потребителей.

Анализ социальных сетей и их пользователей имеет широкий спектр применений, включая маркетинг, социологические исследования, мониторинг общественного мнения, анализ взаимодействия между людьми и т.д. [3] Социальные сети также стали важным источником информации для различных организаций и компаний, которые используют их для сбора и анализа данных, необходимых для принятия решений.

Одним из главных преимуществ социальных сетей является возможность собирать данные о пользователе и анализировать его предпочтения и поведение. Так, например, многие социальные сети предоставляют информацию о пользователе, такую как его возраст, местоположение, интересы, причины посещения сайта, а также о том, какой контент ему нравится.

Однако, для анализа социальных сетей необходимо учитывать такие факторы, как разнообразие социальных платформ, аудитории и алгоритмов. Кроме того, при работе с данными пользователей социальных сетей необходимо соблюдать соответствующие правила и этические нормы [4]. В последнее время общественное мнение стало более требовательным к тем, кто работает с данными пользователей, поэтому необходимо обеспечить соответствующую защиту данных и соблюдать правила использования.

Таким образом, анализ социальных сетей имеет большое значение в настоящее время и может применяться в различных областях. В данной работе будет рассмотрен анализ социальных сетей ВКонтакте и Одноклассники, а также разработка инструмента для сбора и анализа данных, необходимых для создания эффективной стратегии маркетинга.

1.1. Существующие средства анализа аудитории

Для анализа аудитории сообщества в социальной сети ВКонтакте используются различные инструменты. Один из основных инструментов — это статистика сообщества, которая доступна администраторам и модераторам [1]. С помощью этого инструмента можно узнать, сколько участников в сообществе, сколько новых участников за определенный период, какие посты получили большую реакцию аудитории, сколько просмотров у каждого поста и какой процент просмотров пришелся на рекламу.

Другим инструментом является аналитика ВКонтакте. Она предоставляет более подробную информацию о целевой аудитории сообщества. С ее помощью можно узнать, сколько женщин и мужчин в аудитории, какой возрастной диапазон покрывает сообщество, где находятся подписчики по географии, какой язык предпочитают и как часто заходят в ВКонтакте.

Социальная сеть «Одноклассники» предлагают свои инструменты для анализа аудитории сообществ. Например, «Статистика страницы» — это панель управления, которая показывает основные метрики сообщества, такие как количества подписчиков, уникальных посетителей, просмотров страницы и др. [5] Также, есть возможность просмотреть динамику подписчиков, просмотров, реакций и комментариев на посты.

Кроме того, Одноклассники предоставляют инструмент «Аудитория сообщества», [6] который помогает анализировать целевую аудиторию сообщества. Этот инструмент отображает демографические данные подписчиков, такие как пол, возраст, регион проживания, а также предоставляет информацию о том, какие тематики интересны подписчикам.

Для анализа контента сообщества можно использовать инструмент «Статистика постов», который показывает количество просмотров, лайков, репостов и комментариев на каждый пост, а также позволяет сравнивать показатели между разными постами [7].

Как можно видеть, встроенные средства анализа аудитории предоставляют довольно широкий спектр информации об аудитории, но не охватывают

интересы аудитории в контексте тематик сообществ, на которые также подписана аудитория.

Кроме встроенных средств социальных сетей, на рынке существуют решения для анализа сообществ в социальных сетях, которые используют данные о подписках пользователей на другие сообщества. Такие сервисы как «POPSTERS», «SOCIALSTATS» и «JAGAJAM» позволяют производить анализ интересов аудитории, используя широкий спектр находящейся в открытом доступе информации [8, 9]. Однако проприетарный характер исходного кода накладывает ограничения на использование и модификацию приложения, в том числе касательно улучшения графического интерфейса пользователя и увеличения функционала.

1.2. Анализ требований

Спецификация требований для рассматриваемого приложения может включать следующие пункты [10].

1.2.1. Функциональные требования

F1. Приложение должно иметь возможность сбора данных из социальных сетей;

F2. Приложение должно иметь возможность анализировать данные о других сообществах, на которые подписаны пользователи;

F3. Приложение должно предоставлять результаты анализа в удобной форме для дальнейшего использования;

F4. Приложение должно иметь возможность экспорта полученных данных в различных форматах.

F5. Приложение не должно помещать чувствительные данные в долговременное хранилище либо выводить их пользователю.

1.2.2. Нефункциональные требования

NF1. Приложение должно иметь высокую скорость обработки и анализа данных (время ответа сервера не более 5 секунд);

NF2. Приложение должно иметь простой и интуитивно понятный пользовательский интерфейс;

NF3. Приложение должно обеспечивать безопасность и конфиденциальность данных пользователей;

NF4. Приложение должно поддерживать масштабируемость для работы с большими объемами данных.

1.2.3. Требования к установке и развертыванию:

I1. Приложение должно корректно работать на последних версиях популярных браузеров, таких как Google Chrome, Mozilla Firefox, Microsoft Edge и Safari.

I2. Адаптивность: приложение должно быть адаптивным и корректно отображаться на устройствах с разными размерами экранов, включая мобильные устройства, планшеты и настольные компьютеры.

I3. Совместимость с различными устройствами ввода: приложение должно быть совместимо с различными устройствами ввода, такими как мыши, клавиатуры, сенсорные экраны.

I4. Приложение должно быть безопасным для использования и обеспечивать защиту пользовательских данных, что может быть достигнуто с помощью использования защищенного соединения, авторизации и аутентификации пользователей и прочих мер безопасности.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1. Используемые методы анализа

Теория графов предоставляет широкий спектр инструментов для анализа различных типов данных, которые могут быть представлены в виде графов.

Центральность (centrality) является мерой важности вершин в графе и может использоваться для анализа социальных сетей, транспортных сетей, биологических сетей, информационных сетей и т. д. [11] Существует несколько видов центральности:

Центральность по степени (degree centrality) — это мера того, сколько связей имеет вершина с другими вершинами в графе. Чем больше связей у вершины, тем более центральной она является.

Близость (closeness centrality) — это мера того, насколько близко каждая вершина к остальным вершинам в графе. Она рассчитывается как обратное значение средней длины кратчайшего пути между вершиной и всеми остальными вершинами в графе. Чем ближе вершина к другим вершинам, тем более центральной она является.

Центральность по посредничеству (betweenness centrality) — это мера того, насколько часто данная вершина является наиболее кратчайшим путем между другими вершинами в графе. Эта мера помогает идентифицировать ключевые вершины, которые играют важную роль в связывании различных частей графа.

Центральность по собственному вектору (eigenvector centrality) — это мера того, какая вершина имеет наибольшее влияние на другие вершины в графе. Она основана на том, что влияние вершины зависит от ее соседей. Если у вершины много влиятельных соседей, то она сама также будет считаться более влиятельной.

Центральность по Кацу (Katz centrality) — это метрика центральности, которая учитывает не только количество непосредственных связей узла, но и важность узлов, с которыми он связан через цепочки более длинных путей. Формально, центральность по Кацу определяется как взвешенная сумма бесконечного ряда матриц смежности, каждая из которых представляет собой

все пути длины от 1 до бесконечности между парами узлов, взвешенных коэффициентами, экспоненциально убывающими в зависимости от длины пути. Коэффициент затухания β определяет вклад дальних связей и может быть подобран экспериментально.

Использование этих мер центральности позволяет выделить ключевые вершины и понять, как они взаимодействуют с другими вершинами в графе. Это может быть полезно для выявления закономерностей в анализе социальных сетей, где центральные актеры могут играть важную роль в распространении информации или формировании мнения.

Кластеризация (clustering) - позволяет определить группы узлов, которые тесно связаны между собой, но имеют меньшие связи с другими группами. Например, можно использовать кластеризацию, чтобы выявить связь между сообществами со схожей тематикой в социальной сети [12].

2.2. Разработка архитектуры приложения

2.3. Существующие виды архитектуры

Существует множество различных видов архитектур приложений, каждая из которых имеет свои преимущества и недостатки. Некоторые из наиболее распространенных видов архитектур включают в себя [13, 14]:

Монолитная архитектура — это традиционный подход к разработке приложений, где все компоненты приложения находятся внутри одного монолитного приложения. Этот подход прост в реализации, но может стать неэффективным в случае, если приложение разрастется до очень большого размера или если нужно масштабировать отдельные компоненты независимо друг от друга.

Архитектура клиент-сервер — это подход, в котором приложение разделено на две части: клиентскую и серверную. Клиентская часть отвечает за представление данных, а серверная часть отвечает за бизнес-логику и работу с базой данных. Этот подход позволяет расширять масштабирование и обеспечивает более высокую безопасность.

Архитектура микросервисов — это подход, в котором приложение разбито на множество мелких сервисов, каждый из которых отвечает за свою узкоспециализированную функциональность. Этот подход позволяет более эффективно масштабировать приложение и упрощает поддержку, но может оказаться сложным в реализации и управлении.

Событийно-ориентированная архитектура — это подход, в котором компоненты приложения взаимодействуют друг с другом через события, которые происходят в системе. Этот подход позволяет разрабатывать более гибкие и отказоустойчивые приложения, но может оказаться сложным в реализации.

Серверная логика в виде функций — это подход, в котором логика приложения разбита на множество мелких функций, которые выполняются на сервере при вызове со стороны клиента. Этот подход позволяет обеспечить более высокую производительность и гибкость, но может быть сложным в реализации и требует более высокого уровня опыта разработчиков.

2.4. Описание архитектуры приложения

В рамках работы используется гибридная архитектура, включающая в себя элементы клиент-серверной и микросервисной архитектуры. Разделение клиента и сервера необходимо ввиду наличия большого пласта программной логики, требующегося на стороне клиента. При этом серверная логика может быть разделена на три микросервиса: сервис сбора и хранения данных, сервис анализа данных и сервис предоставления файлов клиентской части пользователю.

Микросервис сбора и хранения данных отвечает за сбор и хранение данных, которые будут использоваться в приложении.

Основные задачи этого микросервиса включают в себя:

- Сбор данных из разных источников. Например, он может получать данные из различных API, баз данных и т. д.
- Обработка и агрегация данных. Некоторые данные могут требовать обработки, фильтрации или агрегации перед сохранением в базе данных.

- **Хранение данных.** Микросервис должен иметь возможность сохранять данные в базу данных или файловую систему.
- **Управление доступом к данным.** Микросервис должен иметь механизмы для контроля доступа к данным, чтобы обеспечить безопасность и конфиденциальность информации.
- **Масштабирование.** Микросервис должен быть способен масштабироваться, чтобы обеспечить обработку большого объема данных.
- **Мониторинг.** Микросервис должен иметь механизмы для мониторинга своей работы, чтобы оперативно реагировать на возникающие проблемы.

Второй сервис — это сервис анализа данных. Его основной задачей является обработка данных, собранных первым сервисом, и предоставление клиентской части информации о результатах анализа.

Функционал сервиса анализа данных может включать в себя следующие задачи:

- **Агрегация данных:** сбор и обработка данных с различных источников с целью получения полной картины об интересующих параметрах.
- **Обработка и анализ данных:** анализ данных для выявления трендов и зависимостей, обнаружения аномалий и выбросов, а также расчет показателей и метрик.
- **Представление данных:** предоставление полученных результатов в удобном для анализа и понимания виде, например, в виде диаграмм, графиков, таблиц и т. д.
- **Интеграция с другими сервисами:** передача обработанных данных другим сервисам для дальнейшей обработки или использования.

Для обеспечения работы сервиса анализа данных может использоваться различное программное обеспечение, в том числе специализированные библиотеки для обработки данных, базы данных и другие инструменты. Кроме

того, для обеспечения масштабируемости и отказоустойчивости сервиса может использоваться контейнеризация и кластеризация.

Третий микросервис отвечает за предоставление файлов клиентской части пользователю. Он не имеет специфических механизмов обработки запросов от пользователя, и его задачей является предоставление файлов приложения браузеру при открытии сайта.

Упомянутые микросервисы и их взаимодействие описаны в диаграмме активностей (приложение К).

2.5. Выбор используемых технологий

В рамках работы рассматривается создание веб-приложения. Используемые технологии можно разделить на три группы: серверная часть, клиентская часть и доставка.

Для реализации серверной части (backend) используется язык программирования C#, являющийся основным для работы с фреймворком .NET [15]. C# и .NET имеют долгую историю разработки, и являются одними из самых популярных языков и платформ для разработки веб-приложений. Они предоставляют мощные инструменты для создания высокопроизводительных и масштабируемых приложений, включая WebAPI для создания веб-сервисов.

Также для анализа данных используется Python. Он является одним из наиболее популярных языков программирования для научных и исследовательских приложений, в том числе для работы с графами и сетями. NetworkX — это библиотека для работы с графами и сетями на Python, которая предоставляет множество инструментов для анализа графов.

Клиентская часть (frontend) отвечает за взаимодействие пользователя с приложением. Код клиентской части выполняется в веб-браузере на компьютере пользователя. Основными языками в данном случае является классическая комбинация HTML, JavaScript и CSS. Однако для упрощения (и, в конечном итоге, удешевления разработки) существуют инструменты, позволяющие реализовать требуемый функционал без прямой работы с этими языками и

реализующие высокоуровневые абстракции над средствами упомянутых выше языков.

jQuery — это JavaScript-библиотека, которая упрощает работу с HTML-документами, обработку событий и манипулирование DOM-элементами. Она предоставляет мощные инструменты для создания динамических веб-страниц и повышения удобства использования приложений.

LESS — это язык таблиц стилей, который является расширением CSS. LESS обеспечивает множество возможностей для создания более современных и динамичных пользовательских интерфейсов, и повышения производительности веб-приложений.

TypeScript — это язык программирования, который расширяет синтаксис JavaScript и предоставляет возможности для статической типизации. TypeScript облегчает разработку более сложных приложений и повышает их надежность.

За хранение исходного кода и организации совместного использования кода отвечают Системы контроля версий (Version control systems, VCS) [16]. Наиболее универсальной системой контроля версий является Git. В этой системе контроля версий в качестве каталогов для хранения исходных кодов используются Репозитории, которые можно хранить как на собственном компьютере (локально), так и на серверах. Наиболее популярными серверными решениями для хранения Git-репозитория являются GitHub, GitLab и Bitbucket [17]. В данной работе используется Gitlab, поставляемый Томским Политехническим университетом и находящийся на его серверах.

Заключительным этапом в цепочке работы с приложением является его доставка на сервер (хостинг). Данная часть, как и клиентская, имеет современные инструменты. Существует выделенное понятие DevOps, объединяющее механизмы разработки и доставки приложения. Упомянутый ранее GitLab имеет встроенные инструменты для облачной сборки приложений, называемые «Сборочными линиями». Для сборки и запуска приложения при этом используется Docker.

Docker — это платформа для управления контейнеризацией, которая позволяет разработчикам создавать, развертывать и управлять приложениями в контейнерах. Docker обеспечивает универсальность и переносимость приложений между различными окружениями, что позволяет значительно упростить процесс разработки и развертывания. Кроме того, Docker позволяет загружать образы во внешние хранилища, называемые Репозиториями. Наиболее популярным репозиторием Docker-образов является DockerHub [18].

2.6. Описание механизма сбора данных

Сбор данных осуществляется из двух внешних систем: ВКонтакте и Одноклассники.

2.6.1. Описание работы программного интерфейса сервиса «ВКонтакте»

Программным интерфейсом приложения (application programming interface, API) называются набор компонентов программного приложения и их описание, позволяющие программно взаимодействовать с этим приложением [19].

Социальная сеть ВКонтакте предоставляет интерфейс «VK API», который позволяет обращаться к большей части систем социальной сети. Данный интерфейс представляет собой REST API [20], принимающая query-запросы либо POST-запросы с JSON в качестве полезной нагрузки сообщения.

В рамках данной работы используется раздел «Groups» данного API. Точки входа данного раздела позволяют получать список участников сообщества, а также — при наличии прав — добавлять и удалять сообщества в список на персональной странице [21].

2.6.2. Описание работы программного интерфейса сервиса

«Одноклассники»

Для сбора данных об участниках сообщества в Одноклассниках можно использовать ОК API, предоставляемый социальной сетью. С помощью этого API можно получать информацию о профилях пользователей, а также информацию об их друзьях и подписчиках.

Для начала работы с ОК API необходимо получить ключ доступа. Для этого необходимо зарегистрироваться в разработческой консоли Одноклассников и создать свое приложение. Далее, используя полученный ключ доступа, можно отправлять запросы к API и получать необходимую информацию.

2.7. Обезличивание данных

Обезличивание данных позволяет производить анализ набора данных без возможности отследить источники этих данных – страницы пользователей в социальных сетях [22].

Персональные данные в Российской Федерации определяются в Федеральном законе «О персональных данных» № 152-ФЗ как любая информация, относящаяся к определенному или определяемому физическому лицу (субъекту персональных данных).

К персональным данным относятся, например, имя, фамилия, отчество, дата рождения, адрес проживания, контактная информация (телефон, электронная почта), паспортные данные, биометрические данные, информация о социальном и имущественном положении, медицинские данные, данные о работе и образовании и т. д.

Также к персональным данным могут относиться данные, полученные с помощью технических средств, например, IP-адрес компьютера, cookie-файлы, местоположение с помощью GPS, в том числе идентификатор персональной страницы в социальных сетях.

Для выполнения поставленной в работе задачи не требуется хранение непосредственно идентификаторов пользователей, однако важно обеспечить возможность сопоставлять сообщества и их подписчиков. Для выполнения этой задачи используется следующий алгоритм обезличивания персональных данных:

1. Необезличенные данные пользователя (в том числе его имя, идентификатор и список сообществ) выгружаются из социальной сети;
2. Из данных удаляется имя;

3. Идентификатор пользователя хешируется алгоритмом «Стрибог», разработанным Центром защиты информации и специальной связи ФСБ России с участием АО «ИнфоТеКС».

Обезличенные таким образом данные могут быть как сохранены в базу данных, так и предоставлены пользователю приложения.

ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

3.1. Реализация серверной части

3.1.1. Организация взаимодействия с программным интерфейсом «ВКонтакте»

Для работы с VK API необходимо получить токен пользователя с правами «друзей». Для этого было создано приложение «Some app» (app51419292). На основании документации VK API была создана ссылка для получения токена следующего вида:

Листинг 1 – Ссылка для получения токена пользователя

```
https://oauth.vk.com/authorize?client_id=51419292&display=page&redirect_uri=https://oauth.vk.com/blank.html&scope=groups&response_type=token&v=5.131
```

Для работы достаточно короткоживущего токена, поэтому модификатор «offline» указывать не нужно. Поскольку пользовательский токен уже имеет права пользователя, модификатор «users» не требуется. Требуется версия VK API (5.131) и заглушка вместо ссылки переадресации (<https://oauth.vk.com/blank.html>).

После перехода по ссылке и подтверждения был получен адрес сайта с зарегистрированным токеном доступа:

Листинг 2 – Адрес сайта с токеном доступа

```
https://oauth.vk.com/blank.html#access_token=vk1.a. 23PF3XFTUhfOXjUW3-Co4CL07ewjYbdo090LC6Lix42ZF1df8G5bga8HuvJPBgNWJ_f6-bgZMrLUXQUpYt5FQP3QtIvpRZKxR7GToYT8dZKj1T_xoazs_UWYZnIewEY7JMVGeRZj6epZZTaEpioEvBG-kn4rzsCJLY5RkBDNoDh4G2Gpb5p33dJ-SMNfXLoF&expires_in=86400&user_id=768684892
```

Реализация бизнес-логики

Работа серверной части не зависит от выбранной библиотеки визуализации графиков и будет заключаться в приеме запросов от браузера, совершении необходимых запросов к API ВКонтакте и кешировании результатов этих запросов.

Так как часть запросов к API требует разбиения результата на пакеты, вынесем модуль для работы с такими запросами, который бы инкапсулировал логику отправки множественных запросов. Он представлен в Приложении А.

Можно видеть, что запрос списка друзей написан для получения данных фрагментами по 1000 записей. Это связано с ограничением API ВКонтакте на количество данных, получаемых за один раз.

VK API имеет ограничение на количество запросов в секунду. Кроме того, поскольку это внешний сервис, выполнение этих запросов займет довольно много времени. Чтобы исключить излишние внешние запросы, сделаем кеширующую трассировку их результатов. Ожидаемая нагрузка на приложение носит пиковый характер, поэтому можно обойтись кратковременным кэшем в памяти. Он представлен в Приложении Б.

Для взаимодействия модуля VK API с модулем кэширования в основной код приложения были реализованы связывающие их функции, представленные в Приложении В.

Для кэширования результатов используется реализация кэша на основе интерфейса `IDistributedCache`, предоставляемого `.NET`. Разработанная реализация удобна тем, что позволяет работать с кэшем внутри единого контейнера, без установки сторонних кеширующих сервисов.

Между тем, в случае горизонтального масштабирования сервиса может возникнуть потребность в использовании единого кэширующего сервиса для всех экземпляров веб-приложения. Благодаря использованию интерфейса `IDistributedCache` разработчик может подменить кеширующий сервис сторонним, например, `Redis`. Такая подмена позволит использовать единый кэш для всех экземпляров приложения, что в свою очередь не позволит сократить запросы к внешним сервисам.

Для использования `Redis` необходимо настроить сервис и добавить конфигурацию («`AddStackExchangeRedisCache`») в файл `Program.cs`.

3.1.2. Организация взаимодействия с программным интерфейсом

«Одноклассники»

Сбор данных из социальной сети «Одноклассники» в рамках работы почти идентичен таковому с социальной сетью «ВКонтакте», поэтому принципы работы используются те же. Основное отличие заключается в способе получения токена доступа, адресах требуемых методов и формате ответа сервера.

3.2. Реализация клиентской части

3.2.1. Выбор библиотеки визуализации графов

При выборе библиотеки визуализации графов учтем следующие факторы:

- в приложении использован JavaScript без каких-либо фреймворков;
- граф должен быть интерактивным;
- граф должен «красиво» размещать свои элементы, в идеале – поддерживать симуляцию упругости связей с физической точки зрения.

При анализе существующих решений была выделена библиотека Diagraphs, которая соответствует описанным требованиям. Остальные, например, Vis.JS, требовали наличия сторонних фреймворков (React, Angular) [23] либо не предоставляли возможность обработки кликов по вершине графа [24].

Работа с библиотекой была инкапсулирована в отдельный класс, представленный в Приложении Г.

3.2.2. Налаживание взаимодействия с серверной частью

Поскольку взаимодействие с серверной частью производится через REST JSON API посредством HTTP-запросов, оно плохо с архитектурной точки зрения встраивается в JavaScript код. Для решения этой проблемы методы работы с серверной частью были вынесены в отдельный класс. Он представлен в Приложении Д.

Используемые в приложении точки входа могут быть определены следующим образом.

Листинг 3 – Описание точке входа

```
/Vape/GetFriends/{id}:
  get:
    summary: "* VkApi/GetFriends/{id}"
    parameters:
      - name: "id"
        in: "path"

/VkApi/GetGroupInfo/{id}:
  get:
    summary: "* VkApi/GetGroupInfo/{id}"
    parameters:
      - name: "id"
        in: "path"

/VkApi/GetUserInfo/{id}:
  get:
    summary: "* VkApi/GetUserInfo/{id}"
    parameters:
      - name: "id"
        in: "path"
```

Результаты являются вершинами и связями графа. Их модели определяются следующим образом.

Листинг 4 – Модели вершин и узлов

```
using System;
using VkNet.Model;

namespace VkGraphBuilder.BusinessLogic
{
    [Serializable]
    public class NodeModel
    {
        public long Id { get; set; }
        public string Name { get; set; }
        public string ImageSrc { get; set; }
        public string Extras { get; set; }
        public string Link { get; set; }
    }

    [Serializable]
    public class EdgeModel
    {
        public long FromId { get; set; }
        public long ToId { get; set; }
        public uint Weight { get; set; }
    }
}
```

Пример графа, строящегося таким образом, показан на рис. 1.

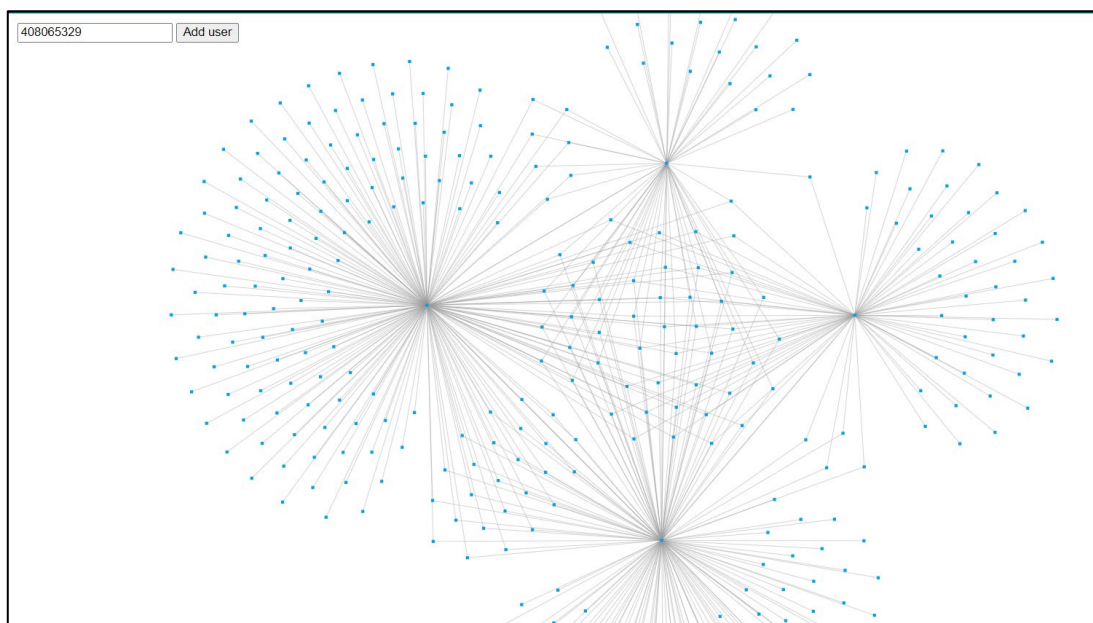


Рисунок 1 – Граф с четырьмя полностью «загруженными» вершинами

На представленном графе можно видеть наличие кластеров общих друзей между двумя, тремя и четырьмя загруженными профилями.

3.2.3. Реализация функции анализа графа

Теория графов предоставляет множество инструментов для анализа графов [25]. Например, алгоритм выявления изолированных вершин может быть использован для определения сообществ, у которых нет общих участников с другими представленными на графе сообществами. Напротив, алгоритм определения степеней вершин может быть использован для определения сообществ, имеющих множество общих участников с другими сообществами. Алгоритмы определения центральности как по отдельности, так и вместе позволяют сделать определенные выводы о весе данного сообщества в графе сообществ [24, 25].

Анализ упомянутых выше показателей может быть полезен как для отдельных вершин, так и для всех вершин графа одновременно. В данной работе реализованы обе этих возможности: значения частных показателей выводятся в панели информации о вершине графа, а значения общих показателей графа – на отдельной странице, открывающейся по кнопке «Open statistics page».



Рисунок 2 – Кнопки управления графом

Для вычисления статистических показателей используется библиотека NetworkX языка Python. Для этого создана отдельная точка входа, которая отвечает за анализ графа с использованием данной библиотеки.

Листинг 5 – Код точки входа анализатора графа

```
public class AnalyzerController : Controller
{
    private readonly ILogger<AnalyzerController> logger;

    public AnalyzerController(
        ILogger<AnalyzerController> logger,
        NetworkXWrapper.NetworkXWrapper networkXWrapper)
    {
        NetworkXWrapper = networkXWrapper;
        this.logger = logger;
    }

    private NetworkXWrapper.NetworkXWrapper NetworkXWrapper { get; }

    [HttpPut]
    public async Task<AnalyzingResult> Analyze([FromBody] List<EdgeModel> edges)
    {
        var calcStatisticsAsync = await
NetworkXWrapper.CalcStatisticsAsync(edges);
        return calcStatisticsAsync;
    }
}
```

Приведенный выше код передает полученный граф в обработку скрипту на языке Python (приведен в приложении E) и возвращает результаты анализа.

В клиентской части веб-приложения степень вершины отображается на карточке информации о вершине.



Рисунок 3 – Данные о вершине графа и результатами анализа

Статистика по степеням вершин графа выводится на отдельной странице при нажатии на кнопку «Open statistics page». Пример статистики по графу в целом приведен в листинге Приложении Ж.

Центральностью вершин называется метрика, отражающая «важность» вершин графа и их близость к центру этого графа. Существует несколько формул для вычисления центральности, такие как degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, Katz centrality, harmonic centrality.

Вычислим эти метрики с помощью скрипта из Приложения Е. Результатом вычисления каждой метрики является словарь «Идентификатор вершины – Значение».

Пример визуализации этих метрик для единичной вершины приведен на рисунке 3.

В Приложении Ж приведен пример вывода идентификаторов вершин, имеющих наибольшие значения по каждой из метрик для графа, состоящего из 900 вершин. Из данных можно сделать вывод, что ни одна из вершин не изолирована, при этом существуют 4 вершины, имеющих наибольшее количество связей. Вершина с самым высоким значением степени вершины имеет связи с более чем половиной вершин графа. При этом существует по

крайней мере 16 вершин, имеющих связи со всеми четырьмя рассматриваемыми вершинами.

Упомянутые выше 4 вершины имеют наибольшие значения центральности в графе. Вершина с наибольшим degree centrality имеет наибольшее количество связей, в то время как вершина с наибольшим Katz centrality имеет наибольшее количество не только прямых, но и второстепенных связей с другими вершинами. В рамках предметной области это говорит о наличии наибольшего количества сообществ со смежными тематиками у данного сообщества по сравнению с другими [26].

3.3. Реализация функции экспорта графа

Существует множество инструментов анализа графов. Для возможности открытия сформированного веб-приложением графа этими инструментами необходимо предусмотреть возможность экспорта графа в файл.

Для экспорта был выбран формат файла TGF, используемый приложением «yEd Graph editor». Данный формат представляет собой 3D-SSV [27] (Space-Separated Values), состоящий из двух секций – описания связей и описания вершин. Секции разделены символом решетки («#»). Секция описания связей представляет собой пары чисел – идентификаторов вершин, секция описания вершин – пары «идентификатор вершины – название». Пример файла приведен ниже.

Листинг 6 – Пример экспортируемого файла

```
045ef7d3-0000-0000-0000-000000000000 qq
045ef7d2-0000-0000-0000-000000000000 eee
045ef7d5-0000-0000-0000-000000000000 DELETED
045ef7d6-0000-0000-0000-000000000000 www
045ef7d7-0000-0000-0000-000000000000 eeee
045ef7d8-0000-0000-0000-000000000000 rrrr
045ef7d9-0000-0000-0000-000000000000 tttt
045ef7da-0000-0000-0000-000000000000 yyy
045ef7db-0000-0000-0000-000000000000 hhhh
#
045ef7d3-0000-0000-0000-000000000000 045ef7d3-0000-0000-0000-000000000000
045ef7d8-0000-0000-0000-000000000000 045ef7d2-0000-0000-0000-000000000000
045ef7db-0000-0000-0000-000000000000 045ef7d2-0000-0000-0000-000000000000
045ef7db-0000-0000-0000-000000000000 045ef7d8-0000-0000-0000-000000000000
```

3.4. Контейнеризация веб-приложения

Контейнеризация веб-приложения является общепринятой практикой для развертывания веб-приложений на серверах сторонних поставщиков услуг размещения приложений. Контейнеризация позволяет, с одной стороны, изолировать веб-приложение от окружения поставщика услуг, а с другой стороны – не расходовать ресурсы на полную виртуализацию веб-приложения и среды исполнения (как в случае с использованием виртуальных машин) [31].

В работе для контейнеризации веб-приложения используется docker. Для настройки контейнеризации создадим Dockerfile следующего вида.

Листинг 7 – Файл конфигурации Docker

```
FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build

WORKDIR /src
COPY . .

WORKDIR "/src/WebUI"
RUN dotnet restore "WebUI.csproj"
RUN dotnet build "WebUI.csproj" -c Release -o /app/build
RUN dotnet publish "WebUI.csproj" -c Release -o /app/publish

WORKDIR /app/publish
RUN apt update -y && apt install python3 python3-venv -y \
    python3 -m venv venv \
    /venv/bin/python3 -m pip install -f NetworkXWrapper/requirements.txt

FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
EXPOSE 80
EXPOSE 443

WORKDIR /app
COPY --from=build /app/publish .
ENTRYPOINT ["dotnet", "VkGraphBuilder.WebUI.dll"]
```

В приведенном файле выделены две основные секции – конфигурация образа сборки и конфигурация образа среды исполнения. Такое разделение необходимо потому, что сборка подразумевает использование большего количества ресурсов, чем исполнение собранного кода, и разделение помогает избежать наличия в результирующем контейнере «лишних» файлов.

3.5. Хранение исходного кода веб-приложения

Использование Git-репозитория для хранения исходного кода приложений является распространенной и рекомендуемой практикой в современной

разработке программного обеспечения. В данном разделе будет представлено обоснование и описание преимуществ использования Git-репозитория для хранения и управления исходным кодом вашего веб-приложения.

Git – это распределенная система контроля версий (Version Control System, VCS), которая предоставляет разработчикам возможность эффективно управлять историей изменений в исходном коде проекта. Основные преимущества использования Git включают:

1. История изменений и восстановление: Git сохраняет полную историю изменений в вашем проекте, позволяя вам просматривать, анализировать и восстанавливать предыдущие версии кода. Это дает вам возможность отслеживать, какие изменения были внесены, кто и когда их внес, и легко вернуться к предыдущим состояниям кода.
2. Ветвление и слияние: Git предоставляет мощные функции ветвления и слияния, что позволяет разработчикам создавать независимые ветки для работы над новыми функциями или исправлением ошибок. Ветвление позволяет вам разрабатывать и тестировать новые функции, не затрагивая основную ветку разработки, а слияние позволяет объединить изменения из разных веток в одну, обеспечивая целостность кодовой базы.
3. Совместная работа: Git облегчает совместную работу нескольких разработчиков над проектом. Каждый разработчик может создавать свои собственные ветки, вносить изменения и делиться своим кодом с другими участниками команды. Git предоставляет инструменты для управления конфликтами при слиянии изменений разных разработчиков.
4. Удобство и гибкость: Git имеет простой и интуитивно понятный интерфейс командной строки, а также множество графических пользовательских интерфейсов (GUI). Это делает Git доступным для разработчиков с различным уровнем опыта. Кроме того, Git-репозитории могут быть легко интегрированы с различными инструментами разработки, такими как интегрированные среды разработки (IDE) или

системы непрерывной интеграции и развертывания (CI/CD), что обеспечивает удобство и гибкость в работе с кодом.

Для работы с кодом был создан репозиторий GraphBuilder на сервере GitLab Томского Политехнического университета. В качестве основной ветки разработки используется ветка «main». В процессе работы создавались также ветки для демонстративных целей, содержащие так называемый «минимально жизнеспособный продукт», (minimum viable product, MVP). В последствии находящийся в этих ветках исходный код был объединен с кодом ветки «main».

Объединение веток можно производить несколькими способами: с использованием командной строки и утилиты «git» (см. листинг 8) и с помощью веб-интерфейса GitLab.

Листинг 8 – Команды для объединения веток

```
git switch main # выбираем целевую ветку в качестве текущей
git merge mvp_v2 # сливаем изменения из ветки mvp_v2 в ветку main
git push origin main # загружаем актуальное состояние ветки на сервер GitLab
```

Для слияния с помощью веб-интерфейса нужно создать так называемый «запрос на слияние». Для этого нужно:

1. Открыть список веток в данном репозитории.
2. Нажать кнопку «Запрос на слияние» справа от целевой ветки.
3. Ввести название запроса на слияние, описание и заполнить другие поля при необходимости.
4. Нажать кнопку «Создать запрос на слияние».
5. В созданном запросе нажать «Слить».

Следует отметить, что данная последовательность действий является упрощенной и служит только для слияния собственных изменений между собственными ветками. При работе в команде следует также получить одобрение данного запроса на слияние от других участников команды.

3.6. Сборка веб-приложения

С целью автоматизации этапов сборки и проверки веб-приложений существуют «сборочные линии». Они предоставляют средства для автоматизации различных этапов, начиная от сборки и компиляции кода до развертывания приложения на сервере. Главная цель использования сборочных линий заключается в ускорении процесса разработки, обеспечении надежности и повышении качества финального продукта.

Сборочные линии GitLab обеспечивают выполнение трех основных этапов:

1. Сборка приложения.
2. Тестирование приложения.
3. Публикация приложения.

Статус того или иного этапа отражается соответствующим индикатором на панели сборочной линии (см. рис. 4).



Рисунок 4 – Панель сборочной линии

Сборочную линию можно настроить как через веб-интерфейс GitLab, так и с помощью специально оформленного файла в репозитории. В данной работе использовалась именно настройка с помощью файла, так как позволяет автоматизировать перенастройку компонентов и ее возврат в предыдущее состояние в случае сбоя.

Для настройки сборочной линии GitLab использует файл «.gitlab-ci.yml». Данный файл определяет конфигурацию процесса непрерывной интеграции и развертывания (CI/CD) и содержит набор инструкций для автоматического выполнения различных этапов при сборке и тестировании приложения.

В файле «.gitlab-ci.yml» можно настроить следующие аспекты:

1. Определение этапов сборки. Файл «.gitlab-ci.yml» позволяет определить последовательность этапов сборки и их зависимости. Например, можно определить этапы компиляции кода, запуска автоматических тестов, сборки контейнера и развертывания на сервере.
2. Настройка окружений и переменных. В файле «.gitlab-ci.yml» можно задать переменные окружения, которые будут использоваться в процессе сборки и развертывания. Это может включать информацию о базе данных, ключах API или других конфиденциальных данных, а также параметры, необходимые для конкретного окружения разработки.
3. Управление запуском и параллельностью. Файл «.gitlab-ci.yml» позволяет настроить условия запуска задач, такие как срабатывание по определенным событиям (например, загрузка в репозиторий) или запуск по расписанию. Кроме того, можно определить параллельное выполнение задач для ускорения процесса сборки и тестирования.
4. Интеграция с внешними сервисами. В файле «.gitlab-ci.yml» можно указать настройки интеграции с внешними сервисами, такими как системы управления базами данных, системы отслеживания ошибок или системы уведомлений. Это позволяет автоматизировать взаимодействие с различными инструментами, используемыми в процессе разработки.
5. Управление результатами и уведомлениями. Файл «.gitlab-ci.yml» позволяет определить, как работать с результатами сборки и тестирования. Это может включать определение условий успешного или неуспешного прохождения каждого этапа, сохранение артефактов сборки (например, бинарных файлов или отчетов о тестировании) и настройку уведомлений о результатах сборки, например, отправка уведомлений на электронную почту или интеграцию с системами уведомлений.

Настройка с помощью файла «.gitlab-ci.yml» обладает большой гибкостью и мощными возможностями настройки процесса CI/CD. Она позволяет разработчикам определить и управлять шагами сборки и развертывания веб-

приложения в соответствии с требованиями и особенностями проекта. Это значительно упрощает процесс разработки, ускоряет выкладку новых изменений и обеспечивает стабильность и надежность приложения.

Для настройки CI/CD в разработанном приложении использовался файл, содержащий следующие этапы:

1. Авторизация в DockerHub.
2. Формирование docker-тега. В качестве тега используется название текущей ветки либо, если текущая ветка «main», то используется ключевое слово «latest».
3. Сборка docker-образа с указанием тега.
4. Загрузка собранного docker-образа DockerHub.

Полный листинг файла приведен в Приложении 3. Можно видеть, что данный файл использует не конкретные значения, а переменные. Эти переменные позволяют использовать один и тот же файл для сборки в нескольких средах.

3.7. Публикация веб-приложения

В качестве поставщика услуг размещения веб-приложений был выбран сервис, отслеживающий обновления Docker-образа в заданном репозитории. При обновлении образа данный сервис обновляет приложение. Стоит отметить, что данный функционал поддерживается многими хостингами веб-приложений [7], что позволяет при необходимости оперативно опубликовать приложение в другом хостинге.

3.8. Выводы по разделу

Было разработано и реализовано веб-приложение GraphBuilder, которое позволяет анализировать сообщества в социальной сети ВКонтакте. Приложение собирает данные о сообществе, его участниках и связанных сообществах, и строит взвешенный граф, который может быть использован для анализа аудитории сообществ.

Программа GraphBuilder была оптимизирована, и время загрузки приложения составляет всего 16 мс. Анализ графа занимает 0,2 мс на каждую

вершину, что в упрощенном виде означает, что граф с 1000 вершинами может быть проанализирован за 200 мс. В приложении также реализованы функции подсчета степеней вершин и их центральности. Пользователь может просмотреть аналитические и статистические данные о степенях вершин и центральности графа.

Анализ примера графа показал, что существуют 4 вершины с наибольшим количеством связей. Вершина с самой высокой степенью имеет связи с более чем половиной вершин графа, и по крайней мере 16 вершин связаны с этими 4 вершинами. Эти 4 вершины также имеют наибольшую центральность в графе. Вершина с наибольшей степенью имеет максимальное количество связей, а вершина с наибольшей центральностью имеет не только прямые, но и второстепенные связи с другими вершинами. Эти результаты указывают на наличие большего количества сообществ со схожей тематикой, связанных с данным сообществом по сравнению с другими.

ГЛАВА 4. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

Введение

В данном разделе работы проведен анализ конкурентоспособности системы и процесса проектирования. Были оценены альтернативные варианты процесса исследования и выделены возможности проекта для решения поставленных задач и проблем. Также был проведен расчет эффективности и рисков в финансовой и социальной сферах.

Анализ конкурентоспособности позволил определить преимущества и недостатки разрабатываемой системы и процесса проектирования по сравнению с аналогичными решениями на рынке. Это позволяет получить обоснование и осознание преимуществ и конкурентных позиций проекта.

Оценка альтернативных вариантов процесса исследования позволяет выбрать оптимальный подход для достижения поставленных целей. Выделение возможностей проекта для решения задач и проблем способствует определению потенциальных выгод и преимуществ, которые может принести проект.

Расчет эффективности и рисков в финансовой и социальной сферах помогает оценить финансовую устойчивость проекта, его потенциальную доходность, а также социальное воздействие и возможные риски.

4.1. Оценка коммерческого и инновационного потенциала НТИ

Перед началом планирования и определения ресурсного и экономического потенциала разработки интерактивной визуализации графа сообществ в социальных сетях, необходимо уделить особое внимание оценке коммерческого потенциала и перспективности нового продукта в целом. Это включает анализ и определение целевого сегмента рынка, на котором компания будет ориентироваться при продаже данного продукта.

4.1.1. Потенциальные потребители результатов исследования

Целевая аудитория результатов разработки веб-приложения включает маркетинговые команды, которые заинтересованы в создании собственных

инструментов анализа аудитории на основе проектов с открытым исходным кодом. Веб-приложение предлагает комплексный подход, включающий три ключевых этапа анализа данных.

Первый этап заключается в сборе и обезличивании данных о смежных сообществах. Веб-приложение осуществляет сбор информации о сообществах, на которые подписаны пользователи из целевой аудитории, и обрабатывает их таким образом, чтобы сохранить конфиденциальность личных данных. Этот этап позволяет собрать достоверные и актуальные данные о целевой аудитории для последующего анализа.

Второй этап включает агрегацию и анализ собранных данных. Веб-приложение проводит обработку и структуризацию данных, чтобы выявить основные тренды, характеристики и взаимосвязи между сообществами. Аналитические инструменты позволяют выявить интересы и предпочтения целевой аудитории, а также определить наиболее релевантную тематику контента для маркетинговых кампаний.

Третий этап состоит в визуализации данных графа смежных сообществ. Веб-приложение предоставляет удобный и интуитивно понятный интерфейс, позволяющий визуализировать связи между сообществами в виде графа. Это помогает наглядно представить структуру и взаимосвязи между сообществами, что облегчает понимание интересов и потребностей целевой аудитории.

Такой комплексный подход к анализу данных позволяет маркетинговым командам эффективно исследовать аудиторию и принимать обоснованные решения в процессе разработки маркетинговых стратегий. Веб-приложение предоставляет ценные инсайты и инструменты для выбора релевантной тематики контента и повышения эффективности маркетинговых кампаний в социальных сетях.

Целевой рынок можно разделить на три сегмента, по характеру применения разработки:

- Научные и образовательные учреждения – коммерческие и некоммерческие научнопрактические разработки;

- Маркетинговые компании – коммерческие научнопрактические разработки;
- Администраторы сообществ социальных сетей практическое использование.

Карта сегментирования рынка выглядит следующим образом (таблица 1).

Таблица 1 – Сегментированная карта рынка

Использование продукции Группа потребителей	Некоммерческая разработка	Коммерческая разработка	Практическое использование
ВУЗы и НИИ			
Маркетинговые компании			
Администраторы сообществ социальных сетей			

ТПУ		POPSTERS, агентство комплексного интернет-маркетинга «Crystal»		Администратор сообщества «Подслушано ТПУ»	
-----	--	---	--	---	--

Согласно карте сегментирования рынка, можно сделать вывод об относительной свободе сегмента разработки программных средств анализа аудитории на основании тематик смежных сообществ.

4.1.2. Анализ конкурентных технических решений

Исследование существующих решений в области анализа и визуализации данных в социальных сетях позволяет выявить конкурентоспособность разрабатываемого веб-приложения путем выявления слабых и сильных сторон этих решений. Анализ слабых сторон помогает определить потенциальные проблемы и улучшения, в то время как выделение сильных сторон позволяет

использовать успешные функциональности в разработке продукта. Это обеспечивает создание конкурентоспособного веб-приложения с улучшенными возможностями анализа и визуализации данных для маркетинговых команд.

Для оценки конкурентоспособности проектируемого решения была составлена оценочная карта (таблица 2), в которой приведены результаты сравнительного анализа решения с конкурентными решениями, где индексы «ф» соответствует собственной разработке, «к1» и «к2» - POPSTERS и SOCIALSTATS соответственно.

Таблица 2 – Оценочная карта критериев эффективности конкурентных технических решений

Показатели конкурентоспособности	Вес критерия	Балльная оценка			Конкурентоспособность		
		Бф	Бк1	Бк2	Кф	Кк1	Кк2
Технические критерии оценки ресурсоэффективности							
Надёжность алгоритмов	0,1	3	5	5	0,3	0,5	0,5
Потребность в вычислительных мощностях	0,05	5	5	5	0,25	0,25	0,25
Простота эксплуатации	0,1	3	4	4	0,3	0,4	0,4
Масштабируемость	0,3	5	2	2	1,5	0,6	0,6
Многопрофильность	0,1	4	5	4	0,4	0,5	0,4
Экономические критерии оценки эффективности							
Конкурентоспособность продукта	0,1	3	5	5	0,3	0,5	0,5
Уровень проникновения на рынок	0,05	0	4	5	0	0,2	0,25
Цена	0,1	5	3	4	0,5	0,3	0,4

Показатели конкурентоспособности	Вес критерия	Балльная оценка			Конкурентоспособность		
		Бф	Бк1	Бк2	Кф	Кк1	Кк2
Послепродажное обслуживание	0,1	3	5	5	0,3	0,5	0,5
Итого	1	31	38	39	3,85	3,75	3,8

В целом разработка уступает индустриальным конкурентам, но разрыв позволяет предположить о возможности продукта конкурировать с ними. Кроме того, по отдельным пунктам: масштабируемость и цена, – разработка опережает конкурентов.

4.1.3. SWOT анализ

SWOT-анализ — это инструмент для оценки проекта, который выявляет его сильные и слабые стороны внутри и возможности и угрозы снаружи. Это позволяет разработать стратегии для использования преимуществ, преодоления ограничений и максимизации возможностей, минимизации угроз.

В процессе проведения SWOT-анализа была построена матрица, каждый квадрант которой соответствует сильным сторонам, слабым сторонам, возможностям и угрозам (таблица 3).

Таблица 3 – SWOT-матрица

	<p>Сильные стороны:</p> <p>С1. Визуализация многомерных данных</p> <p>С2. Повышение вовлеченности и интереса пользователей</p> <p>С3. Демонстрация влияния и распространение идей в обществе</p>	<p>Слабые стороны:</p> <p>Сл.1 Ограничения в масштабировании</p> <p>Сл.2 Неполная информация о связях и взаимодействиях</p>
<p>Возможности:</p> <p>В1. Возможность определения трендов и паттернов в сообществах</p>		

<p>В2. Интеграция с другими инструментами и аналитикой</p> <p>В3. Персонализация и адаптация</p>		
<p>Угрозы:</p> <p>У1. Неполная информация о связях и взаимодействиях</p> <p>У2. Ограничения в доступе к данным или изменения в API</p>		

В рамках второго этапа проведём анализ соответствия сильных и слабых сторон проекта внешним условиям окружающей среды. Анализ представлен ниже в виде интерактивных матриц, сильное соответствие отмечено знаком «+», слабое знаком «-» (таблицы 4–7).

Таблица 4 – Матрица сильных сторон и возможностей проекта

Сильные стороны				
Возможности		C1	C2	C3
	B1	+	-	+
	B2	+	+	+
	B3	+	+	+

Направления реализации сильных сторон и возможностей: B1C1, B1C3, B2B1, B2C2, B2C3, B3C1, B3C2, B3C3.

Таблица 5 – Матрица слабых сторон и возможностей проекта

Слабые стороны			
Возможности		Сл1	Сл2
	B1	+	-
	B2	-	+
	B3	-	-

Направления реализации слабых сторон и возможностей: B1Сл1, B2Сл2.

Таблица 6 – Матрица сильных сторон и угроз проекта

Сильные стороны	
-----------------	--

Угрозы		C1	C2	C3
	У1	+	–	–
	У2	–	+	+

Направления реализации слабых сторон и возможностей: У1С1, У2С1, У2С3.

Таблица 7 – Матрица слабых сторон и угроз проекта

Слабые стороны			
Угрозы		Сл1	Сл2
	У1	+	+
	У2	–	–

Направления реализации слабых сторон и возможностей: У1Сл1, У2Сл2.

В рамках третьего этапа построим итоговую матрицу SWOT анализа, результат представлен в таблице 8.

Таблица 8 – Итоговая матрица SWOT

	<p>Сильные стороны: С1. Визуализация многомерных данных С2. Повышение вовлеченности и интереса пользователей С3. Демонстрация влияния и распространение идей в обществе</p>	<p>Слабые стороны: Сл.1 Ограничения в масштабировании Сл.2 Неполная информация о связях и взаимодействиях</p>
<p>Возможности: В1. Возможность определения трендов и паттернов в сообществах В2. Интеграция с другими инструментами и аналитикой В3. Персонализация и адаптация</p>	<p>В1С1, В1С3, В2В1, В2С2, В2С3, В3С1, В3С2, В3С3 – Интегрировать визуализацию графа сообществ с другими инструментами и аналитикой, такими как алгоритмы машинного обучения или анализ социальных сетей, для получения более глубоких и полезных инсайтов.</p>	<p>В1Сл1 – Развивать алгоритмы и методы для определения трендов и паттернов в сообществах, чтобы помочь пользователям лучше понимать их эволюцию и динамику. В2Сл2 – Предоставить возможность пользователям настраивать и адаптировать</p>

		визуализацию графа в соответствии с их потребностями, предоставляя гибкость и персонализацию.
<p>Угрозы:</p> <p>У1. Неполная информация о связях и взаимодействиях</p> <p>У2. Ограничения в доступе к данным или изменения в API</p>	<p>У1С1 – Работать над улучшением сбора данных о связях и взаимодействиях в сообществах, искать дополнительные источники информации для обеспечения более полного представления графа сообществ.</p> <p>У2С1, У2С3 – Следить за изменениями в API социальных сетей и поддерживать обновления, чтобы гарантировать доступность и актуальность данных для визуализации.</p>	<p>У1Сл1, У2Сл2 – Инвестировать в создание интерактивных функций и возможностей, которые повышают вовлеченность и интерес пользователей, обеспечивая более привлекательный и удобный пользовательский опыт.</p>

Исходя из проведенного SWOT-анализа, можно сделать вывод о том, что проблемы, которые могут возникнуть в ходе данного исследовательского проекта, могут быть преодолены с использованием его сильных сторон. Важно обратить внимание на улучшение технологий, основываясь на данных, полученных в результате наблюдений за конкурентами. Этот подход позволит оптимизировать проект и обеспечить его успешную реализацию.

4.1.4. Оценка готовности проекта к коммерциализации

Для оценки готовности проекта определим показатели по вопросам в таблице 10. Оценка проводится по пятибалльной шкале. При оценке научного проекта: 1 балл – не проработано, 2 балла – проработка слабая, 3 балла – выполнено, качество посредственное, 4 балла – удовлетворительное качество, 5 баллов –

качество подтверждено сторонним специалистом. При оценке знаний разработчика: 1 балл – не знаю, 2 балла – только теоретические знания, 3 балла – теоретические знания с практическими примерами, 4 балла – умею, практикую, 5 баллов – могу консультировать по вопросу. (таблица 9)

Таблица 9 – Таблица оценки готовности научного проекта к коммерциализации

№ п/п	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
1.	Определен имеющийся научно-технический задел	4	5
2.	Определены перспективные направления коммерциализации научно-технического задела	3	3
3.	Определены отрасли и технологии (товары, услуги) для предложения на рынке	4	4
4.	Определена товарная форма научно-технического задела для представления на рынок	2	3
5.	Определены авторы и осуществлена охрана их прав	4	4
6.	Проведена оценка стоимости интеллектуальной собственности	3	3
7.	Проведены маркетинговые исследования рынков сбыта	3	3
8.	Разработан бизнес-план коммерциализации научной разработки	3	3
9.	Определены пути продвижения научной разработки на рынок	3	3

№ п/п	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
10	Разработана стратегия (форма) реализации научной разработки	3	3
11	Проработаны вопросы международного сотрудничества и выхода на зарубежный рынок	2	2
12	Проработаны вопросы использования услуг инфраструктуры поддержки, получения льгот	3	3
13	Проработаны вопросы финансирования коммерциализации научной разработки	4	4
14	Имеется команда для коммерциализации научной разработки	3	3
15	Проработан механизм реализации научного проекта	4	5
	ИТОГО БАЛЛОВ	48	51

В соответствии с полученными результатами можно сделать вывод, что перспективность данного проекта выше среднего.

4.2. Инициация проекта

В данном этапе фиксируются начальные цели, содержание и финансовые ресурсы. Определяются заинтересованы стороны, которые могут повлиять на конечный результат проекта. Эта информация закрепляется в уставе проекта.

4.2.1. Цели и результат проекта

Сначала определим заинтересованные стороны (таблица 10). Заинтересованные стороны – это лица или организации, которые активно

заинтересованы и/или могут быть как положительно, так и отрицательно затронуты в результате проекта.

Таблица 10 – Заинтересованные стороны проекта

Заинтересованные стороны проекта	Ожидания заинтересованных сторон
Томский политехнический университет	Изучение распределения и динамики сообществ, выявления важных узлов и подграфов, а также анализа влияния сообществ на поведение пользователей.
Агентство комплексного интернет-маркетинга «Crystal»	Определение целевых аудиторий, исследование взаимосвязей и влияния сообществ на принятие решений о рекламных стратегиях и партнерствах.

В таблице 11 представлена информация о иерархии целей проекта и критерия достижения целей.

Таблица 11 – Цели и результаты проекта

Цели проекта	<p>Понимание и анализ сообществ: позволяет исследователям, аналитикам и социологам лучше понять структуру сообществ, взаимосвязи между его участниками.</p> <p>Улучшение пользовательского опыта: возможность исследовать данные и взаимодействовать с графом сообществ.</p> <p>Визуализация социального влияния: наглядное представление социального влияния и распространения идей в сообществе.</p>
Ожидаемые результаты проекта	<p>Разработка и предоставление инструмента визуализации</p> <p>Сдача выпускной квалификационной работы</p>

Критерии приемки результата проекта	Разработана интерактивная визуализация графа сообществ социальных сетей
Требования к результату проекта	Выполнены все поставленные задачи Разработанное приложение соответствует заявленным характеристикам

4.2.2. Организационная структура проекта

В таблице 12 представлены участники команды научного проекта, а также их роли, функции и трудозатраты.

Таблица 12 – Рабочая группа

№ п/п	ФИО, основное место работы, должность	Роль в проекте	Функция	Трудозатраты, час.
1	Ковин Роман Владимирович, доцент отделения информационных технологий	Научный руководитель	Постановка целей и задач Разработка календарного плана работы Оценка полученных результатов	50
2	Сунцов Кирилл Сергеевич, магистрант	Разработчик	Изучение и анализ предметной области Разработка приложения Оформление пояснительной записки	800
3	Спицына Любовь Юрьевна, к.э.н., Доцент Отделения социально-гуманитарных наук, Школа базовой инженерной подготовки	Эксперт проекта	Консультирование по разделу выпускной квалификационной работы «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение».	48
4	Антоневич Ольга Алексеевна, к.б.н., Доцент Отделения общетехнических дисциплин, Школа базовой инженерной подготовки	Эксперт проекта	Консультирование по разделу выпускной квалификационной работы «Социальная ответственность».	40

№ п/п	ФИО, основное место работы, должность	Роль в проекте	Функция	Трудозатраты, час.
5	Ануфриева Татьяна Николаевна, Старший преподаватель Отделения иностранных языков, Школа базовой инженерной подготовки	Эксперт проекта	Консультации по разделу на английском языке выпускной квалификационной работы.	16
ИТОГО				954

4.2.3. Ограничения и допущения проекта

Ограничения проекта — это все факторы, которые могут ограничивать степень свободы членов команды проекта. Эти факторы перечислены в таблице 13.

Таблица 13 – Ограничения проекта

Фактор	Ограничения / Допущения
Источник финансирования	Средства ТПУ
Сроки реализации проекта	30.01.2023 — 27.06.2023
Дата утверждения плана проекта	03.03.2023
Дата завершения	27.06.2023
Продолжительность работы над проектом в день	8 часов
Максимальный бюджет проекта	600000 рублей

4.3. Планирование управления НТИ

Для планирования хода работ НТИ определим структуру работ, распланируем контрольные события и определим календарный план проекта, который выразим в диаграмме Ганта.

4.3.1. Иерархическая структура работ

При организации проекта был определен перечень необходимых работ в виде иерархической структуры работ проекта.



Рисунок 5 – Иерархическая структура работ проекта

4.3.2. План проекта

Для обоснованного планирования разделения труда и рабочего времени каждого участника процесса внедрения проекта был составлен полный перечень работ, представленный в таблице 14. Рабочие дни были рассчитаны с учетом шестидневной рабочей недели и праздничных дней Российской Федерации.

Ниже приведен список участников проекта:

- Научный руководитель (Руководитель проекта) – Ковин Роман Владимирович (НР).
- Эксперт по разделу Финансового Менеджмента – Спицына Любовь Юрьевна (ФМ).
- Эксперт по разделу Социальной Ответственности – Антонец Ольга Алексеевна (СО).
- Эксперт по разделу на английском языке – Ануфриева Татьяна Николаевна (АЧ).
- Исполнитель по проекту (Магистрант) – Сунцов Кирилл Сергеевич (С).

Используя данную информацию, можно распределить работы и определить рабочие времена для каждого участника, с учетом их компетенций и ролей в проекте. Это обеспечит эффективную координацию и выполнение задач в рамках проекта.

Определим ключевые события проекта, их даты и результаты в таблице 14.

Таблица 14 – Календарный план проекта

№	Название	Длительность, раб. дни	Дата начала работ	Дата окончания работ	Состав участников
1	Инициация ВКР				
1.1	Выбор темы исследования	5	30.01.2023	03.02.2023	НР, С
1.2	Постановка целей и задач	3	04.02.2023	07.02.2023	НР, С
1.3	Разработка спецификации ВКР	6	08.02.2023	14.02.2023	С
1.4	Подготовка технической спецификации	3	15.02.2023	17.02.2023	С
1.5	Анализ актуальности	7	18.02.2023	28.02.2023	С
1.6	Разработка календарного плана	3	01.03.2023	03.03.2023	НР, С
2	Исследование предметной области и технологий				
2.1	Анализ предметной области	9	04.03.2023	15.03.2023	С
2.2	Изучение литературы	9	16.03.2023	24.03.2023	С
2.3	Исследования существующих алгоритмов анализа графов сообществ	6	25.03.2023	31.03.2023	С
2.4	Анализ требований к исходным данным	4	01.04.2023	05.04.2023	С
3	Разработка и исследование технической части				
3.1	Реализация клиентской части	15	06.04.2023	22.04.2023	С
3.2	Реализация серверной части	5	23.04.2023	28.04.2023	С
3.3	Реализация развертывания	10	29.04.2023	13.05.2023	С
4	Подведение итогов работы				
4.1	Оценка и сравнение результатов анализа графов сообществ	3	14.05.2023	17.05.2023	НР, С

№	Название	Длительность, раб. дни	Дата начала работ	Дата окончания работ	Состав участников
4.2	Составление пояснительной записки	16	18.05.2023	05.06.2023	С
4.3	Составление раздела "Финансовый менеджмент"	6	06.06.2023	13.06.2023	ФМ, С
4.4	Составление раздела "Социальная ответственность"	5	14.06.2023	19.06.2023	СО, С
4.5	Составление раздела на английском языке	2	20.06.2023	21.06.2023	АЧ, С
4.6	Подготовка к защите ВКР	5	22.06.2023	27.06.2023	НР, С

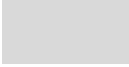




В рамках научного проекта был разработан и составлен календарный график, представленный в виде диаграммы Ганта.

Диаграмма Ганта – это графическое представление плана проекта, которое отражает хронологическую последовательность задач, событий и этапов. Диаграмма Ганта представляет собой столбчатую диаграмму, где каждая полоса представляет собой задачу или активность, а горизонтальные полосы представляют промежутки времени. Каждая полоса имеет длину, которая отражает продолжительность задачи или активности (таблица 15).

Таблица 15 – Диаграмма Ганта по проекту

№	Исполнители	Тк, раб. дни	Продолжительность выполнения работ																	
			Январь			Февраль			Март			Апрель			Май			Июнь		
			1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1.1	НР	2																		
	С	3																		
1.2	НР	3																		
	С																			
1.3	С	6																		
1.4	С	3																		
1.5	С	7																		

№	Исполнители	Тк, раб. дни	Продолжительность выполнения работ																	
			Январь			Февраль			Март			Апрель			Май			Июнь		
			1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1.6	НР	3																		
	С																			
2.1	С	9																		
2.2	С	9																		
2.3	С	6																		
2.4	С	4																		
3.1	С	15																		
3.2	С	5																		
3.3	С	10																		
4.1	НР	3																		
	С																			
4.2	С	16																		
4.3	С	6																		
	ФМ																			
4.4	С	5																		
	СО																			
4.5	С	2																		
	АЧ																			
4.6	НР	5																		
	С																			

-  НР, Научный руководитель
-  ФМ, Эксперт по разделу Финансового Менеджмента
-  СО, Эксперт по разделу Социальной Ответственности
-  АЧ, Эксперт по разделу на английском языке
-  С, Исполнитель по проекту (Магистрант)

4.3.3. Бюджет НТИ

Бюджет научного исследования должен в полной мере отражать все планируемые расходы на его выполнение. Бюджет формируется по следующим статьям: сырье и материалы, специальное оборудование, основная заработная плата, дополнительная заработная плата, отчисления в социальные фонды, командировки, оплата работ сторонних организаций.

4.3.3.1. Сырьё и материалы

Разработка программного обеспечения проводится на персональном компьютере без использования материального сырья. Расчет данной статьи расходов не требуется.

4.3.3.2. Специальное оборудование

В качестве специального оборудования подразумеваются затраты на приобретение ПК для разработки (таблица 16).

Таблица 16 – Список специального оборудования

№ п/п	Наименование	Кол-во, шт.	Цена, единицы, руб.	Общая стоимость, руб.
1	ПК	1	80 000,00	80 000,00
Итого:				80 000,00

4.3.3.3. Основная заработная плата

Заработная плата работника включает основную заработную плату, и дополнительную заработную плату:

$$Z_{зп} = Z_{осн} + Z_{доп},$$

где $Z_{осн}$ - основная заработная плата;

$Z_{доп}$ - дополнительная заработная плата (12–20% от $Z_{осн}$).

Основная заработная плата руководителя (лаборанта, инженера) от предприятия (при наличии руководителя от предприятия) рассчитывается по следующей формуле:

$$Z_{осн} = Z_{дн} \cdot T_p$$

где $Z_{\text{осн}}$ - основная заработная плата одного работника, руб.;

T_p - продолжительность работ, выполняемых научно-техническим работником, раб. дн. (таблица);

$Z_{\text{дн}}$ - среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_{\text{м}} \cdot M}{F_{\text{д}}}$$

где $Z_{\text{м}}$ - месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года: при отпуске в 48 раб. дня $M = 10,4$ месяца, 6-дневная неделя;

$F_{\text{д}}$ - действительный годовой фонд рабочего времени научно-технического персонала, рабочие дни (таблица 17).

Таблица 17 – Баланс рабочего времени

Показатели рабочего времени	Работник
Календарное число дней	365
Количество нерабочих дней	
• выходные дни	54
• праздничные дни	11
Потери рабочего времени	
• отпуск	48
• невыходы по болезни	0
Действительный годовой фонд рабочего времени	252

Месячный должностной оклад работника:

$$Z_{\text{м}} = Z_{\text{тс}} \cdot (1 + k_{\text{пр}} + k_{\text{д}}) \cdot k_{\text{р}}$$

где $Z_{\text{тс}}$ - заработная плата по тарифной ставке, руб.;

$k_{\text{пр}}$ - премиальный коэффициент, равный 0,3;

$k_{\text{д}}$ - коэффициент доплат и надбавок составляет примерно 0,2–0,5 (в НИИ и на промышленных предприятиях – за расширение сфер обслуживания, за профессиональное мастерство, за вредные условия: 15–20 % от $Z_{\text{тс}}$);

k_p - районный коэффициент, равный 1,3 (для Томска).

Для предприятий, не относящихся к бюджетной сфере, тарифная заработная плата (оклад) рассчитывается по тарифной сетке, принятой на данном предприятии. Расчёт основной заработной платы приведен в таблице 18.

Таблица 18 – Расчёт основной заработной платы

№	Наименование этапа	Исполнители по категориям	Трудоёмкость., чел.-дн.	Заработная плата, приходящаяся на один чел.- дн., руб	Всего заработная плата по тарифу, руб.
1	Инициация ВКР	НР	12	6469,08	77628,96
2	Инициация ВКР	С	18	2995,17	53913,06
3	Исследование предметной области и технологий	С	28	2995,17	83864,76
4	Разработка и исследование технической части	С	30	2995,17	89855,10
5	Подведение итогов работы	НР	8	6469,08	51752,64
6	Подведение итогов работы	ФМ	6	4627,79	27766,74
7	Подведение итогов работы	СО	5	4627,79	23138,95
8	Подведение итогов работы	АЧ	2	4627,79	9255,58
9	Подведение итогов работы	С	37	2995,17	110821,29
Итого:					527997,08

Далее произведем расчет месячной заработной платы (таблица 19).

Таблица 19 – Расчет месячной заработной платы

Исполнитель	З _б , руб.	k _{пр}	k _д	k _р	З _м , руб.	З _{дн} , руб.	T _{раб} , дн.	З _{осн} , руб.
Научный руководитель	52 700	1,1	1,118	1,3	156 750,88	6 469,08	20	129 381,60
Эксперт по разделу Финансового Менеджмента	37 700				112 134,88	4 627,79	6	27 766,74
Эксперт по разделу Социальной Ответственности	37 700				112 134,88	4 627,79	5	23 138,95
Эксперт по английской части	37 700				112 134,88	4 627,79	2	9 255,58
Исполнитель по проекту (Магистрант)	24 400				72 575,36	2 995,17	113	338 454,21

4.3.3.4. Дополнительная заработная плата

Расчет дополнительной заработной платы ведется по следующей формуле:

$$З_{\text{доп}} = k_{\text{доп}} \cdot З_{\text{осн}},$$

где $k_{\text{доп}}$ - коэффициент дополнительной заработной платы.

Примем коэффициент равный 0,10 и произведем расчет (таблица 20).

Таблица 20 – Расчет заработной дополнительной платы

Исполнитель	Основная заработная плата, руб.	Дополнительная заработная плата, руб.	Зарплата исполнителя, руб.
Научный руководитель	129 381,60	12 938,16	142 319,76
Эксперт по разделу Финансового Менеджмента	27 766,74	2 776,67	30 543,41
Эксперт по разделу Социальной Ответственности	23 138,95	2 313,90	25 452,85

Эксперт по английской части	9 255,58	925,56	10 181,14
Исполнитель по проекту (Магистрант)	338 454,21	33 845,42	372 299,63
Итого:		52 799,71	580 796,79

Таким образом, общая сумма дополнительной заработной платы равна 52 799,71 руб.

4.3.3.5. Отчисления во внебюджетные фонды

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$З_{внеб} = k_{внеб} \cdot (З_{осн} + З_{доп})$$

где $k_{внеб}$ - коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Результаты расчета приведены отчислений и общий итог по фонду заработной платы приведены в таблице 21.

Таблица 21 – Общий итог по зарплатному фонду

Исполнитель	Коэффициент отчислений во внебюджетные фонды	Основная заработная плата, руб.	Дополнительная заработная плата, руб.	Отчисления во внебюджетные фонды
Научный руководитель	0,3	129 381,60	12 938,16	42 695,93
Эксперт по разделу Финансового Менеджмента		27 766,74	2 776,67	9 163,02
Эксперт по разделу Социальной Ответственности		23 138,95	2 313,90	7 635,85
Эксперт по английской		9 255,58	925,56	3 054,34

части				
Исполнитель по проекту (Магистрант)		338 454,21	33 845,42	111 689,89

Таким образом, общая сумма отчислений на социальные нужды равна 174 239,04 руб.

4.3.3.6. Научные и производственные командировки

Данный вид работ не запланирован, расчёт статьи расходов не требуется.

4.3.3.7. Контрагентные расходы

Привлечение сторонних организаций не запланировано, расчёт статьи расходов не требуется.

4.3.3.8. Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т. д. Их величина определяется по следующей формуле:

$$Z_{\text{накл}} = k_{\text{накл}} \cdot (Z_{\text{осн}} + Z_{\text{доп}})$$

где $k_{\text{накл}}$ - коэффициент, учитывающий накладные расходы.

Примем коэффициент равным 0,16. Сумма накладных расходов представлена в таблице 22.

Таблица 22 – Расчет накладных расходов

Исполнитель	Основная заработная плата, руб.	Дополнительная заработная плата, руб.	Накладные расходы
Научный руководитель	129 381,60	12 938,16	22 771,16
Эксперт по разделу Финансового Менеджмента	27 766,74	2 776,67	4 886,95

Исполнитель	Основная заработная плата, руб.	Дополнительная заработная плата, руб.	Накладные расходы
Эксперт по разделу Социальной Ответственности	23 138,95	2 313,90	4 072,46
Эксперт по английской части	9 255,58	925,56	1 628,98
Исполнитель по проекту (Магистрант)	338 454,21	33 845,42	59 567,94

Таким образом, общая сумма накладных расходов равна 92 927,49 руб.

4.3.3.9. Бюджет НТИ

Сведём статьи, рассчитанные в предыдущих пунктах, в таблицу 23.

Таблица 23 – Итоговый бюджет проекта

Статьи							
Сырье и материалы, руб.	Специальное оборудование, руб.	Основная заработная плата, руб.	Дополнительная заработная плата, руб.	Отчисления на социальные нужды, руб.	Научные и производственные, руб. командировки, руб	Контрагентные расходы, руб.	Накладные расходы, руб.
0,00	80 000,00	527 997,08	52 799,71	174 239,04	0,00	0,00	92 927,49
Итог:							509007,89

4.4. Риски проекта

Составим реестр возможных рисков, сведём его в таблицу 24.

Таблица 24 – Реестр рисков проекта

№	Риск	Потенциальное воздействие	Вероятность наступления (1–5)	Влияние риска (1–5)	Уровень риска	Способы смягчения риска	Условия наступления
1	Несоответствие используемых библиотек заявленным характеристикам	Увеличение сроков разработки	2	5	Средний	Привлечение средств помощи в программировании для изучения альтернативных библиотек	Ошибки в документации
2	Критические ошибки приложения	Отказ приложения	1	5	Средний	Использование средств повышения надежности и автоматического тестирования	Ошибки в разработке приложения
3	Срыв сроков разработки	Увеличение сроков разработки	2	2	Средний	Привлечение дополнительных специалистов	Нарушение сроков разработки

4.5. Определение ресурсной, финансовой, экономической эффективности

Для определения эффективности НТИ необходимо рассчитать интегральный показатель финансовой эффективности и интегральный показатель эффективности.

Интегральный финансовый показатель определяются по следующей формуле:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}},$$

где $I_{\text{финр}}^{\text{исп.}i}$ - интегральный финансовый показатель разработки;

Φ_{pi} - стоимость i -го варианта исполнения;

Φ_{max} - максимальная стоимость исполнения научно-исследовательского проекта (в т. ч. аналоги).

Определение интегральных показателей эффективности проведём в сравнении со следующими аналогами:

1. POPSTERS.

2. SOCIALSTATS.

Результаты вычислений интегрального показателя представлены в таблице 25.

Таблица 25 – Расчёт интегрального финансового показателя

Продукт	Φ_{pi}	Φ_{max}	$I_{фин}^{исп}$
Разрабатываемое приложение	509007,89	2000000,00	0,25
POPSTERS	1200000,00		0,60
SOCIALSTATS	2000000,00		1,00

Интегральный показатель ресурсоэффективности вариантов исполнения объекта определяется по формуле:

$$I_{pi} = \sum a_i \cdot b_i,$$

где I_{pi} - интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки;

a_i - весовой коэффициент i -го варианта исполнения разработки;

b_i - балльная оценка i -го варианта исполнения разработки, которая устанавливается экспертным путем по выбранной шкале оценивания.

Результаты вычислений представлены в таблице 26.

Таблица 26 – Сравнительная оценка характеристик продуктов

Критерии	Весовой коэффициент параметра	НТИ	POPSTERS	SOCIALSTATS
Надежность алгоритмов	0,1	3	4	4
Быстродействие	0,1	2	4	5
Удобство	0,1	3	4	4
Масштабируемость	0,4	5	3	2
Многопрофильность	0,3	4	4	3
ИТОГО	1	4	3.6	3

Сравнительная эффективность проекта определяется по формуле:

$$\mathcal{E}_{cp} = \frac{I_{\text{фин}}^{\text{исп}}}{I_{pi}}$$

где \mathcal{E}_{cp} - сравнительная эффективность проекта;

$I_{\text{фин}}^{\text{исп}}$ - интегральный показатель разработки;

I_{pi} - интегральный технико-экономический показатель аналога.

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта (таблица 27) и выбрать наиболее целесообразный вариант из предложенных.

Таблица 27 – Интегральные показатели

	$I_{\text{фин}}^{\text{исп}}$	I_{pi}	$I_{\text{исп}}$
Разрабатываемое приложение	0,25	4	16
POPSTERS	0,60	3,6	6
SOCIALSTATS	1,00	3	3

Таким образом сравнительная эффективность проекта с аналогами:

1. В сравнении с POPSTERS $\mathcal{E}_{cp} = \frac{16}{6} = 2.67$.
2. В сравнении с SOCIALSTATS $\mathcal{E}_{cp} = \frac{16}{3} = 5.33$.

4.6. Выводы по разделу

В ходе работы были выявлены направления, способствующие возможному развитию проекта с целью повышения его конкурентоспособности. Анализ позволил выявить потенциальные улучшения, которые могут принести значительные преимущества в экономическом плане, повысить эффективность проекта и снизить связанные с ним риски.

Предлагаемые направления развития проекта основываются на результатах анализа рынка, конкурентов и потребностей целевой аудитории. Они включают в себя возможности улучшения функциональности и производительности приложения, расширение его возможностей анализа данных, усовершенствование пользовательского интерфейса и оптимизацию процессов взаимодействия с сообществами в социальных сетях.

Разработка проекта является целесообразной, поскольку предполагаемые улучшения и развитие позволят повысить его конкурентоспособность на рынке и привлечь больше пользователей и клиентов. Экономическая успешность проекта оценивается на основе потенциального роста доходов, снижения затрат и достижения финансовой устойчивости.

Также проанализированы риски, связанные с разработкой проекта, и разработаны стратегии и меры для их снижения и управления. Это поможет обеспечить успешную реализацию проекта, достижение его целей и максимизацию результативности в долгосрочной перспективе.

ГЛАВА 5. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Введение

В данном разделе анализируются опасные факторы труда и определяются меры организационного, технического и правового характера, необходимые для предотвращения негативных последствий в процессе работы разработчика программного обеспечения.

Выпускная квалификационная работа по разработке приложения для анализа и визуализации графа сообществ социальных сетей осуществляется в офисном помещении размером 4 м в ширину, 5 м в длину и 3 м в высоту. Площадь помещения составляет 20 м².

Работа выполняется на компьютере в помещении, где присутствует искусственное и естественное освещение. В среднем одновременно в помещении работает от 1 до 3 человек.

Цель работы состоит в обеспечении безопасности и комфорта работы разработчика программного обеспечения в офисных условиях. В ходе исследования будет проанализировано влияние эргономических аспектов, освещения, вентиляции, санитарной гигиены, шума, электромагнитных полей и других факторов на здоровье и благополучие работников.

Результаты исследования позволят разработать рекомендации по оптимизации рабочего места и внедрению мер профилактики с целью создания безопасных и комфортных условий работы для разработчика программного обеспечения.

5.1. Правовые и организационные вопросы обеспечения безопасности

5.1.1. Особенности законодательного регулирования проектных решений

Режим труда и отдыха для работы с персональным компьютером регламентируется соответствующими нормативными актами, такими как Трудовой кодекс Российской Федерации [33] и «ГОСТ 21889–76 Система «Человек-машина». Кресло человека-оператора» [35].

Работа в офисе, включающая работу с персональным компьютером, относится ко второй категории тяжести труда. При этом условия внешней производственной среды и физическая, умственная и нервно-эмоциональная нагрузка должны быть оптимальными. Согласно указанным нормативным актам, продолжительность рабочего дня работников не должна превышать 40 часов в неделю.

Важно соблюдать определенные рекомендации и перерывы при работе за компьютером. Продолжительность непрерывной работы без регламентированного перерыва не должна превышать 1 час. Рекомендуется делать перерывы продолжительностью 10–15 минут каждые 45–60 минут работы. Во время таких перерывов полезно выполнить комплекс упражнений и проветрить помещение.

Также эффективными являются нерегламентированные перерывы, известные как микропаузы, длительностью 1–3 минуты. Регламентированные перерывы и микропаузы могут использоваться для выполнения упражнений и гимнастики для глаз и пальцев рук, а также для проведения массажа. Рекомендуется менять комплексы упражнений каждые 2–3 недели.

Не рекомендуется превышать 6 часов работы за компьютером за одну смену. Для обеспечения соответствия компьютерного оборудования нормам безопасности, проводится производственный контроль и надзор со стороны производителя. Эксплуатирующие предприятия также следят за характеристиками используемой техники.

5.1.2. Организационные мероприятия при компоновке рабочей зоны

Согласно ГОСТ 12.2.032–78 «Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования» [36], рабочий стол для работы с персональным компьютером может иметь любую конструкцию, соответствующую современным эргономическим требованиям. Он должен обеспечивать комфортное размещение необходимого оборудования с учетом его количества, размеров и характера выполняемой работы.

Ниже приведена таблица 28, которая отражает выполнение требований для данного рабочего места согласно ГОСТ 21889–76 [35] и ГОСТ 12.2.032–78 [36].

Таблица 28 – Требования к организации рабочего места при работе с ПЭВМ

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

Можно видеть, что рабочее место соответствует перечисленным требованиям.

5.2. Производственная безопасность

Ниже приведен перечень опасных и вредных факторов, характерных для производственной среды в виде таблицы (таблица 29).

Таблица 29 – Опасные и вредные факторы при выполнении работ по разработке программы

Факторы (ГОСТ 12.0.003–2015) [46]	Нормативные документы
Недостаточная освещённость рабочей зоны; отсутствие или недостаток естественного света	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23–05–95 [37]

Повышенный уровень шума	ГОСТ 12.1.003–83 Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности [38]
Повышенный уровень электромагнитных излучений	ГОСТ 12.1.006–84 «Система стандартов безопасности труда (ССБТ). Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля» [39] Повышенная или пониженная влажность воздуха
Производственные факторы, связанные с аномальными микроклиматическими параметрами воздушной среды на местонахождении работающего	СанПиН 1.2.3685–21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания [34]
Статические перегрузки, умственные перегрузки, перегрузки анализаторов	МР 2.2.9.2311 – 07 «Профилактика стрессового состояния работников при различных видах профессиональной деятельности» [40]
Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий	ГОСТ 12.1.030–81 Система стандартов безопасности труда (ССБТ). Электробезопасность. Защитное заземление. Зануление [41] ГОСТ 12.1.038–82 Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов [42].

5.2.1. Недостаточная освещённость рабочей зоны; отсутствие или недостаток естественного света

Рабочее место разработчика находится на 5-м этаже здания и имеет естественное освещение в кабинете. Однако основное освещение в аудитории обеспечивается с использованием общего искусственного освещения. В процессе разработки программного обеспечения разработчик постоянно работает за персональным компьютером (ПК).

Согласно СП 52.13330.2016 «Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95», при работе с ПК необходимо обеспечить оптимальные условия освещения, чтобы избежать возникновения бликов на экране ПК. Это достигается путем правильной организации

искусственного освещения в помещении, при котором освещенность на экране ПК не превышает 300 люксов. Такие рекомендации связаны с обеспечением комфортного восприятия информации и предотвращением негативного влияния яркого освещения на зрительную функцию разработчика.

Работа с ПК относится к категории зрительных работ Б, что означает, что основная нагрузка связана с восприятием и обработкой информации на экране. Это подразумевает, что для поддержания зрительного комфорта и предотвращения напряжения зрительного аппарата, включая утомление глаз, важно обеспечить соответствующие условия освещения на рабочем месте разработчика.

5.2.2. Повышенный уровень шума

Звуковые колебания, производимые движущимися частями механизмов и приборов, могут оказывать воздействие на здоровье человека. Громкие звуки могут привести к проблемам со слухом, а длительное воздействие шума со значением выше 80 децибел (дБ) может вызвать потерю слуха или его ухудшение. Чувствительность к монотонным звукам является индивидуальным показателем. Однако, постоянно повторяющиеся шумы на рабочем месте могут вызывать проблемы, связанные с нервной системой и органами слуха.

В данном исследовании основным источником шума на рабочем месте является системный блок персонального компьютера (ПК). Внутри системного блока находится система охлаждения, которая состоит из вентиляторов. В процессе работы эти вентиляторы генерируют непрерывный гул (гудение), что создает шумовую обстановку на рабочем месте.

Постоянный уровень шума оказывает влияние на работоспособность и концентрацию человека. Однако, рабочее место, представленное в данной работе, соответствует нормам ГОСТ 12.1.003–83 «Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности» [38]. Это помещение обладает минимальным уровнем шума при программировании и разработке программного обеспечения для планирования производственных процессов.

Эквивалентный уровень звука для эффективной длительности рабочего дня вычисляется по формуле 1 [38].

$$L_p = 10 \lg \left(\frac{1}{N} \sum_{n=1}^N 10^{0.1 \times L_{p,m}} \right), \quad (1)$$

где $L_{p,m}$ эквивалентный уровень звука n-го выборочного измерения;

n номер выборочного измерения;

N число выборочных измерений.

Эквивалентный уровень звука за 8-часовой рабочий день для работников группы равного шумового воздействия рассчитывают по формуле 2.

$$L_{EX} = L_p + 10 \lg \left(\frac{T_e}{T_0} \right), \quad (2)$$

где T_e эффективная длительность рабочего дня;

T_0 базовая длительность рабочего дня.

Замеры уровней звука показали следующие результаты (таблица 30).

Таблица 30 – Данные измерений, дБ

Номер замера	Значение, дБ
1	54,2
2	55,1
3	50,0
4	62,0
5	51,2
6	51,3

Согласно формулам 1, 2,

$$L_p = 10 \lg \left(\frac{1}{6} 10^{0.1 \times 54.2} + \frac{1}{6} 10^{0.1 \times 55.1} + \frac{1}{6} 10^{0.1 \times 50} + \frac{1}{6} 10^{0.1 \times 62} + \frac{1}{6} 10^{0.1 \times 51.2} + \frac{1}{6} 10^{0.1 \times 51.3} \right) = 56.26,$$

$$L_{EX} = 56.26 + 10 \lg \frac{8}{8} = 56.26$$

Уровень звука на рабочем месте, который не превышает 60 децибел (дБ), соответствует допустимому уровню шума [38] для легкой физической нагрузки

и напряженности трудового процесса легкой и средней степеней. Это означает, что данный уровень шума находится в пределах безопасных значений для работников, выполняющих как легкую физическую нагрузку, так и работу средней интенсивности.

5.2.3. Повышенный уровень электромагнитных излучений

Уровень электромагнитных излучений регулируется ГОСТ 12.1.006–84 «Система стандартов безопасности труда (ССБТ). Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля» [39]. В повседневной жизни большинство людей не ощущает влияния электромагнитных излучений. Чтобы эти излучения оказывали влияние на здоровье и самочувствие организма, мощность источника должна быть значительно выше обычных рабочих уровней.

Основными источниками электромагнитного поля в рабочей среде ПЭВМ являются системный блок, в котором располагается процессор, и экран монитора. На рабочем месте установлены ПЭВМ с жидкокристаллическими мониторами, которые излучают электромагнитные волны. Однако эти излучения не представляют угрозы для здоровья человека, даже при длительной работе с ними.

5.2.4. Производственные факторы, связанные с аномальными микроклиматическими параметрами воздушной среды на местонахождении работающего

В соответствии с требованиями СанПиН 1.2.3685–21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» [34], давление, температура и влажность воздуха оказывают влияние на здоровье работников, а также на общее самочувствие, работоспособность и выполнение поставленных задач.

Требования к параметрам микроклимата определены СанПиН 1.2.3685–21 и приведены в таблице 31.

Таблица 31 – Требования к параметрам микроклимата для работ категории

Ia

Период года	Категория работ	Температура воздуха, °С		Относительная влажность воздуха, %
		Ниже оптимальных величин	Выше оптимальных величин	
Теплый	Ia	21,0 – 22,9	25,1 – 28,0	15 – 75
Холодный	Ia	20,0 – 21,9	24,1 – 25,0	15 – 75

Большая часть работ выполнялась в холодный период года при средней температуре 23оС и относительной влажности 70%.

В данном случае рабочее место может иметь повышенную влажность из-за своего расположения. Для регулирования влажности в помещении, в стенах аудитории установлена механическая вентиляция, оснащенная вентилятором, которая обеспечивает отвод воздуха из помещения. Это позволяет снизить уровень влажности и создать более комфортные условия для работников.

Кроме того, аудитория подключена к системе центрального городского отопления, что позволяет регулировать температуру в помещении в зимние месяцы. Это обеспечивает поддержание оптимального уровня тепла и комфортных условий для работников, способствуя их работоспособности и эффективности труда.

5.2.5. Статические перегрузки

Работа за компьютером в современном мире стала неотъемлемой частью многих профессий. Разработчики программ, инженеры и другие специалисты проводят множество часов за персональными компьютерами (ПК). Однако такая работа сопровождается некоторыми проблемами, которые могут влиять на здоровье и работоспособность.

Одной из проблем является статическая нагрузка на мышцы, вызванная необходимостью принимать одну и ту же позу в течение продолжительного времени. Неправильная поза, постоянный наклон вперед и другие факторы могут вызывать боли в шее и спине.

Работа с большим объемом информации, анализ данных, инженерные исследования и разработка программных продуктов требуют высокой концентрации внимания. Это может негативно сказываться на зрении, так как глаза являются основными зрительными анализаторами человека. Постоянное напряжение глазных мышц и монотонность работы с визуальной информацией могут вызывать усталость и снижение остроты зрения. Длительное время, проведенное за дисплеем, может также вызывать стресс.

5.2.6. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов

Для обеспечения электробезопасности на рабочем месте применяются организационные и технические мероприятия. Согласно ГОСТ 12.1.030–81 «Система стандартов безопасности труда (ССБТ). Электробезопасность. Защитное заземление. Зануление» [41], помещение, в котором расположено рабочее место, относится к категории помещений без повышенной опасности. Оно характеризуется сухостью, отсутствием пыли, наличием непроводящих полов и нормальной температурой воздуха. Температурный режим, влажность воздуха и химическая среда не способствуют разрушению изоляции электрооборудования.

Защита от электрического тока на рабочем месте осуществляется с помощью изоляции токопроводящих частей, так что все провода находятся в изолированном состоянии. Электрические устройства, включая процессор ПЭВМ, расположены в защитных коробках.

Короткое замыкание представляет собой соединение двух точек с разными потенциалами, что приводит к увеличению тока и выделению большого количества тепла. В результате короткого замыкания может возникнуть пожар, а люди, находящиеся рядом с источником короткого замыкания, могут понести травмы.

Короткое замыкание на рабочем месте может быть вызвано неисправностью проводки или при работе с компьютером, когда внутри корпуса создается разность фаз, и ток может повредить электросеть. Для защиты электрической

сети от короткого замыкания установлены устройства защитного отключения (УЗО), такие как автоматические выключатели и предохранители. Кроме того, в помещении установлены датчики дыма, которые при возникновении пожара, вызванного коротким замыканием, оповещают всё здание о начавшемся пожаре. Таким образом, рабочее место полностью защищено от возможного короткого замыкания.

В рабочем пространстве присутствует множество устройств, работающих от электрического тока и изготовленных из материалов, которые могут накапливать статический заряд. Это может привести к возникновению разности потенциалов от статического электричества, и прикосновение человека к заряженным объектам может вызвать травмы, ожоги или пожар.

Для обеспечения защиты оборудования и персонала применяются меры общего заземления электропроводки. Общее заземление предусматривает соединение всех электроустановок и электрооборудования с заземляющим проводником, который в свою очередь связан с землей [41, 49]. Это позволяет отводить статический заряд и предотвращать его скопление на оборудовании и других поверхностях.

Также для уменьшения накопления статического заряда и снижения риска возгорания применяются увлажняющие устройства. Увлажнение воздуха способствует уменьшению электростатического заряда, так как влажный воздух хорошо проводит электричество. Это помогает снизить вероятность возникновения разности потенциалов и связанных с этим опасностей [48].

5.3. Экологическая безопасность

5.3.1. Воздействие на литосферу

Когда ПЭВМ и сопутствующая оргтехника выходят из строя или устаревают, они относятся к IV классу опасности и требуют специальной утилизации. Это связано с тем, что данные устройства содержат опасные вещества и могут нанести вред окружающей среде, если их неправильно утилизировать. Для минимизации влияния на окружающую среду необходимо проводить специальные процедуры утилизации, которые соответствуют требованиям ГОСТ

Р 53692–2009 «Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов» [43].

Согласно ГОСТ 17.4.3.04-85 «Охрана природы (ССОП). Почвы. Общие требования к контролю и охране от загрязнения», юридические лица имеют право утилизировать оргтехнику только после прохождения процедуры полного списания, подтвержденного актом.

Томский политехнический университет, как юридическое лицо, также заботится о безопасной утилизации перегоревших люминесцентных ламп. Технический персонал собирает эти лампы, которые затем передаются в центр по переработке, обладающему лицензией на сбор и переработку данного вида отходов. Такая практика позволяет снизить негативное влияние на окружающую среду и обеспечить безопасность персонала.

Для утилизации макулатуры в Томском политехническом университете установлены специальные контейнеры. В эти контейнеры помещаются отработавшие печатные издания, офисная бумага и другие изделия из переработанной целлюлозы. После этого они транспортируются в пункты сбора макулатуры, где происходит их утилизация. Такая практика способствует уменьшению объема отходов и устойчивому использованию ресурсов.

5.4. Безопасность в чрезвычайных ситуациях

5.4.1. Пожарная безопасность

Согласно ГОСТ 12.1.004–91 «Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Общие требования» [44], рабочее помещение, включая аудиторию, относится к категории В, что предполагает наличие производств, связанных с обработкой или применением твердых горючих веществ и материалов.

Для обеспечения пожарной безопасности в помещении аудитории необходимо проводить меры пожарной профилактики и активной пожарной защиты. Пожарная профилактика включает комплекс мероприятий, направленных на предотвращение возникновения пожара или уменьшение его последствий.

В случае возникновения пожара в аудитории необходимо немедленно принять следующие меры: отключить электропитание, вызвать по телефону пожарную команду, осуществить эвакуацию всех присутствующих и приступить к ликвидации пожара с помощью огнетушителей. При небольшом очаге пламени можно также использовать подручные средства для прекращения доступа воздуха к источнику возгорания. Эвакуация должна осуществляться в соответствии с планом эвакуации.

Пожар является чрезвычайной ситуацией, которая представляет опасность для людей, находящихся в помещении. Для обнаружения пожара и своевременного оповещения всех присутствующих, в аудитории должны быть установлены датчики дыма, которые при обнаружении задымления активируют систему общего оповещения во всем здании.

В случае возникновения пожара необходимо:

- незамедлительно сообщить об это в пожарную охрану;
- принять меры по эвакуации людей;
- отключить электроэнергию, приступить к тушению пожара первичными средствами пожаротушения.

Для тушения пожаров в офисных помещениях следует использовать ручные углекислотные огнетушители (ОУ-5) и внутренний пожарный кран противопожарного водопровода. Они предназначены для тушения начальных возгораний различных веществ и материалов, за исключением веществ, которые горят без доступа воздуха. Огнетушители должны всегда быть в исправном состоянии и готовы к использованию. Кроме того, нужно соблюдать запрет на использование химических пенных огнетушителей для тушения пожаров в офисных помещениях [44].

5.5. Выводы по разделу

В ходе исследования по социальной ответственности были рассмотрены правовые и организационные аспекты обеспечения безопасности на рабочем месте, а также проведены анализ вредных и опасных факторов, которые оказывают влияние на здоровье человека, находящегося в офисе.

Основываясь на результате анализа, были выделены и определены средства защиты, которые необходимы для обеспечения безопасности работников и снижения воздействия вредных факторов. Эти средства включают в себя меры по улучшению эргономики рабочего места, использование персональных средств защиты, таких как защитные очки или наушники, а также проведение обучения и информирования сотрудников о правилах безопасности и применении средств защиты.

Согласно ПУЭ [50], помещение, в отношении электробезопасности, относится к первому классу - "помещения без повышенной опасности".

В соответствии с правилами охраны труда для эксплуатации электроустановок, персонал должен иметь I группу по электробезопасности. I группа присваивается работником из числа электротехнического персонала, имеющего группу III по электробезопасности, или специалистом по охране труда, имеющим группу IV по электробезопасности или выше, назначенным распоряжением руководителя организации [51].

Лаборатория, в которой проводилось исследование, относится к категории Ia по СанПиН 1.2.3685–21 (работы, производимые сидя и сопровождающиеся незначительным физическим напряжением) [34].

Помещение, в котором проводилось исследование, относится к группе В1-В4. В нем находятся горючие и трудногорючие жидкости, твёрдые горючие и трудногорючие вещества и материалы (включая пыль и волокна), а также вещества и материалы, способные гореть только при взаимодействии с водой, кислородом воздуха или друг с другом, при условии, что помещения, в которых они находятся или обрабатываются, не относятся к категории А или Б [47].

Одним из важных аспектов исследования было выявление бытовых отходов, которые негативно влияют на окружающую среду. Для обеспечения экологической безопасности и снижения негативного влияния на окружающую среду, эти отходы утилизируются в соответствии с требованиями ГОСТ. Согласно постановлению "Об утверждении критериев отнесения объектов, оказывающих негативное воздействие на окружающую среду, к объектам I, II,

III и IV категорий", исследуемый объект оказывает незначительное негативное воздействие на окружающую среду и относится к объектам III категории [52].

Расчет уровня звука на рабочем месте показал, что уровень звука не превышает 40 децибел (дБ), что соответствует допустимому уровню шума для легкой физической нагрузки и труда легкой либо средней интенсивности.

Итак, исследование по социальной ответственности привело к выявлению вредных и опасных факторов на рабочем месте, определению средств защиты, а также проведению утилизации бытовых отходов согласно требованиям ГОСТ, что способствует обеспечению безопасности работников и снижению негативного влияния на окружающую среду.

ЗАКЛЮЧЕНИЕ

В работе рассмотрена проблема анализа интересов целевой аудитории на основании участия членов анализируемого сообщества в других сообществах. Было спроектировано и реализовано веб-приложение GraphBuilder с использованием языков программирования C#, Python и JavaScript. Разработанное веб-приложение принимает на вход анализируемое сообщество, собирает данные о нем, его участниках и других сообществах, в которые входят участники. Затем оно обезличивает собранные данные, визуализирует и анализировать взвешенный граф сообществ социальной сети ВКонтакте.

Строящийся взвешенный граф сообществ позволяет определить количество общих членов двух сообществ и может быть использован, например, для анализа аудитории сообществ. При наличии большого количества общих членов у двух сообществ, имеющих разные тематики, можно сделать вывод, что эти тематики имеют одинаковую аудиторию.

Время загрузки приложения благодаря оптимизации составляет 16 мс, время анализа графа имеет рост 0,2 мс/вершину, то есть граф с 1000 вершин анализируется 200 мс.

Были реализованы функции подсчета степеней вершин графа и их центральности, реализована возможность открыть страницу с аналитическими и статистическими данными по степеням вершин графа и центральности вершин графа.

Дальнейшее развитие приложения может включать расширение списка поддерживаемых социальных сетей, улучшение пользовательского интерфейса и автоматический анализ графа на предмет тематик смежных сообществ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ВКонтакте. Статистика сообществ: анализ аудитории: Управление Страницей бизнеса [Электронный ресурс] – Режим доступа: <https://vk.com/biz/article/statistika>
2. Уилсон Р. Введение в теорию графов / Пер. с англ. Слинкина А. А. 1977 —
—
М., 1977. — С. 16. — 480 с.
3. Golbeck J. Analyzing the Social Web. — Newnes, 2013. — 290 с. — ISBN 978-0-12-405531-5.
4. Щеглова И. А. Этические и правовые аспекты использования данных из социальных медиа / И. А. Щеглова // Вестник Томского государственного университета. 2018. № 431. С. 81-87. [Электронный источник] – Режим доступа:
<http://vital.lib.tsu.ru/vital/access/manager/Repository/vtls:000632044>
5. Одноклассники. Статистика сообществ [Электронный ресурс] – Режим доступа: <https://help.ok.ru/statistics-page>
6. Одноклассники. Бизнес-аккаунт [Электронный ресурс] – Режим доступа: <https://ok.ru/businessmanager>
7. Одноклассники. Помощь: как посмотреть статистику показов поста [Электронный ресурс] – Режим доступа: <https://ok.ru/help/kak-posmotret-statistiku-pokazov-posta>
8. Под прицелом: Popsters - сервис для аналитики сообществ и страниц социальных сетей // СРА журнал “Где Трафик?”. - 2015. [Электронный источник] — — Режим доступа: https://gdetraffic.com/Obzory/Pod_pritselom_Popsters
9. Писаревский В.Г. Православные сообщества в пространстве интернет-ресурсов и их влияние на социальные сети [Электронный ресурс] – Режим доступа: <https://www.dissercat.com/content/pravoslavnye-soobshchestva-v-prostranstve-internet-resursov-i-ikh-vliyanie-na-sotsialnye-set> (дата обращения: 05.03.2023).

10. ГОСТ Р ИСО/МЭК 15288-2005 Информационная технология. Системная инженерия. Процессы жизненного цикла систем.
11. Ньюман М.Дж. Networks: An Introduction. — Oxford, UK: Oxford University Press, 2010.
12. Хайдуков Д. С. Применение кластерного анализа в государственном управлении // Философия математики: актуальные проблемы. — М.: МАКС Пресс, 2009. — 287 с.
13. Вивин А.А. Кратко о типах архитектур программного обеспечения, и какую из них выбрали мы для IaaS-провайдера, 2018 [Электронный источник] — Режим доступа: <https://habr.com/ru/company/1cloud/blog/424911/>
14. Proglib. Монолитная vs Микросервисная архитектура, 2019 [Электронный источник] — Режим доступа: <https://proglib.io/p/monolitnaya-vs-mikroservisnaya-arhitektura-2019-09-16>
15. Нейглер Д. C# 2.0 for C++ and Java programmer: conference workshop // Journal of Computing Sciences in Colleges – 2007. – №22/5 [Электронный ресурс] – Режим доступа: <https://dl.acm.org/doi/10.5555/1229688.1229689>
16. Кустов, А. А. Особенности использования систем контроля версий и управления совместной разработкой в отделе разработки программного обеспечения / А. А. Кустов // Научный электронный журнал "Меридиан". – 2019. – № 11(29). – С. 192-194.
17. Тэглайн. (2018). Рейтинг сервисов-репозиторий для хранения кода 2018. <https://tagline.ru/source-code-repository-rating/> (дата обращения: 15.04.2023)
18. Fairbanks, J., Tharigonda, A., & Eisty, N. U. (2023). Analyzing the Effects of CI/CD on Open Source Repositories in GitHub and GitLab. Доклады на 20-й Международной конференции IEEE/ACIS по программному обеспечению, инженерии, управлению и приложениям (SERA 2022). arXiv:2303.16393 [Электронный ресурс] — Режим доступа: <https://doi.org/10.48550/arXiv.2303.16393>

19. О компании // ВКонтакте [Электронный ресурс] URL: <https://vk.com/about>. (Дата обращения: 01.12.2022).
20. М. Меджуи, М. Амундсен, Р. Митра. Непрерывное развитие API. Правильные решения в изменчивом технологическом ландшафте. — СПб.: «Питер», 2021. — 272 с. — ISBN 978-5-4461-1232-6.
21. М. Masse. REST API Design Rulebook. — O'Reilly Media, 2011. — 272 с. — ISBN 9781449310509.
22. Learning API // ВКонтакте [Электронный ресурс] URL: https://vk.com/dev/first_guide. (Дата обращения: 01.12.2022).
23. ГОСТР ИСО/МЭК 27038. Информационные технологии. Методы обеспечения безопасности. Требования и методы электронного цензурирования от 2016 г. N 3403: дата введения 2016-05-05. — URL: <https://meganorm.ru/Data/622/62203.pdf> (дата обращения: 05.12.2022). — Текст: электронный.
24. Grammar and style // Docker reference [Электронный ресурс] URL: <https://docs.docker.com/contribute/style/grammar/>. (Дата обращения: 02.12.2022).
25. A. Thomas. Data Visualization with JavaScript 1st Edition, Kindle Edition. — No Starch Press, 2015. — 381 с. — ASIN: B00V20IFDG.
26. E. Ma, M. Seth. Network Analysis Made Simple — 2021. — 381 с.[Электронный ресурс] URL: <https://ericmjl.github.io/Network-Analysis-Made-Simple/>.Дата обращения: 05.11.2022.
27. D. Krackhardt. Assessing the Political Landscape: Structure, Cognition, and Power in Organizations // Administrative Science Quarterly. — 1990. — Июнь (т. 35, вып. 2). — doi:10.2307/2393394. — JSTOR 2393394.
28. A. Dekker. Conceptual Distance in Social Network Analysis // Journal of Social Structure. — 2005. — Т. 6, вып. 3.
29. L. Katz A New Status Index Derived from Sociometric Index // Psychometrika. — 1953.

30. M. Roughan; J. Tuke The Hitchhikers Guide to Sharing Graph Data, 2015 3rd International Conference on Future Internet of Things and Cloud, IEEE, doi:10.1109/ficloud.2015.76
31. J. Turnbull. The Docker Book: Containerization is the new virtualization // James Turnbull; 2nd edition. — 2014. — ASIN: B00LRROT14.
32. N. Middleton. Heroku: Up and Running: Effortless Application Deployment and Scaling // O'Reilly Media; 1st edition. — 2013. — ISBN: 978-1449341398.
33. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022).
34. СанПиН 1.2.3685–21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания.
35. ГОСТ 21889–76 Система «Человек-машина». Кресло человека-оператора.
36. ГОСТ 12.2.032–78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования.
37. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95.
38. ГОСТ 12.1.003–83 «Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности».
39. ГОСТ 12.1.006–84 «Система стандартов безопасности труда (ССБТ). Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля».
40. МР 2.2.9.2311–07 «Профилактика стрессового состояния работников при различных видах профессиональной деятельности».
41. ГОСТ 12.1.030–81 «Система стандартов безопасности труда (ССБТ). Электробезопасность. Защитное заземление. Зануление».
42. ГОСТ 12.1.038–82 Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов.

43. ГОСТ Р 53692–2009 Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов.
44. ГОСТ 12.1.004–91 «Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Общие требования».
45. ГОСТ ISO 9612–2016 Акустика. Измерения шума для оценки его воздействия на человека. Метод измерений на рабочих местах (с Поправкой).
46. ГОСТ 12.0.003–2015 Опасные и вредные производственные факторы. Классификация. Перечень опасных и вредных факторов.
47. СП 12.13130.2009 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности.
48. ГОСТ Р 53734.1–2014 Электростатика. Часть 1. Электростатические явления. Физические основы, прикладные задачи и методы измерения.
49. ГОСТ Р 58882–2020 Заземляющие устройства. Системы уравнивания потенциалов. Заземлители. Заземляющие проводники.
50. Правила устройства электроустановок (ПУЭ). Седьмое издание.
51. Правила по охране труда при эксплуатации электроустановок от 15.12.2020 N 903н (с изм. от 29.04.2022).
52. Приказ Об утверждении критериев отнесения объектов, оказывающих негативное воздействие на окружающую среду, к объектам I, II, III и IV категорий от 31.12.2020 года N 2398 (с изм. от 07.10.2021).

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Листинг агрегатора пакетных запросов

(справочное)

```
using System.Collections.Generic;
using System.Threading.Tasks;
using VkNet.Abstractions;
using VkNet.Model;
using VkNet.Model.RequestParams;
using VkNet.Utils;

namespace VkGraphBuilder.BusinessLogic
{
    public static class VkApiExtensions
    {
        public static async Task<VkCollection<User>> GetAllMembersAsync(
            this IGroupsCategory groupsCategory,
            string groupId)
        {
            const ushort batchSize = 1000;
            var groupsGetMembersParams = new GroupsGetMembersParams
            {
                GroupId = groupId,
                Offset = 0,
                Count = batchSize,
            };

            List<User> members = new();
            while (true)
            {
                groupsGetMembersParams.Offset = members.Count;
                var batch = await
groupsCategory.GetMembersAsync(groupsGetMembersParams);
                if (batch.Count == 0)
                {
                    break;
                }

                members.AddRange(batch);
            }

            return new VkCollection<User>((ulong)members.Count, members);
        }
    }
}
```

ПРИЛОЖЕНИЕ Б

Листинг кэширующего модуля

(справочное)

```
using System;
using System.Threading.Tasks;
using Microsoft.Extensions.Caching.Distributed;
using Newtonsoft.Json;

namespace VkGraphBuilder.BusinessLogic
{
    public static class DistributedCacheExtensions
    {
        public static async Task<T[]> GetOrAddArrayAsync<T>(
            this IDistributedCache distributedCache,
            string key,
            Func<Task<T[]>> factory)
            where T : class
        {
            var value = await distributedCache.GetObjectAsync<T[]>(key);
            if (value is not null)
            {
                return value;
            }

            try
            {
                value = await factory();
            }
            catch
            {
                // pass
            }

            value ??= Array.Empty<T>();

            await distributedCache.SetObjectAsync(key, value);

            return value;
        }

        private static async Task<T> GetObjectAsync<T>(this IDistributedCache
distributedCache, string key)
            where T : class
        {
            string serializedValue = await distributedCache.GetStringAsync(key);
            if (serializedValue is null)
            {
                return null;
            }

            var value = JsonConvert.DeserializeObject<T>(serializedValue);

            return value;
        }

        private static async Task SetObjectAsync<T>(this IDistributedCache
distributedCache, string key, T value)
            where T : class
```

```
{
    var serializedValue = JsonConvert.SerializeObject(value);
    await distributedCache.SetStringAsync(key, serializedValue);
}
}
```


ПРИЛОЖЕНИЕ В

Листинг модуля взаимодействия с VK API

(справочное)

```
using System;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.Extensions.Caching.Distributed;
using VkNet.Enums.Filters;
using VkNet.Model;
using VkNet.Model.RequestParams;

namespace VkGraphBuilder.BusinessLogic
{
    public class CachedVkApiClient
    {
        public CachedVkApiClient(VkApiClientFactory vkApiClientFactory,
            IDistributedCache distributedCache)
        {
            VkApiClientFactory = vkApiClientFactory;
            DistributedCache = distributedCache;
        }

        private VkApiClientFactory VkApiClientFactory { get; }

        private IDistributedCache DistributedCache { get; }

        public async Task<User> Users_GetAsync(long userId)
        {
            var fetchingResult = await DistributedCache.GetOrAddArrayAsync(
                $"{nameof(Users_GetAsync)}_{userId}",
                async () =>
                {
                    using var vkApi = await VkApiClientFactory.GetInstanceAsync();

                    var users = await vkApi.Users.GetAsync(
                        new[] { userId },
                        ProfileFields.City | ProfileFields.BirthDate |
ProfileFields.Sex | ProfileFields.Photo200);

                    return users.ToArray();
                });

            return fetchingResult.SingleOrDefault();
        }

        public Task<User[]> Friends_GetAsync(long userId)
        {
            return DistributedCache.GetOrAddArrayAsync(
                $"{nameof(Friends_GetAsync)}_{userId}",
                async () =>
                {
                    using var vkApi = await VkApiClientFactory.GetInstanceAsync();

                    var users = await vkApi.Friends.GetAsync(new FriendsGetParams
                    {
                        UserId = userId,
                        Fields = ProfileFields.City
                            | ProfileFields.BirthDate
```


ПРИЛОЖЕНИЕ Г

Листинг класса NetworkGraph

(справочное)

```
class NetworkGraph {
  constructor(containerId) {
    const container = document.getElementById(containerId);

    this.graph = Viva.Graph.graph();
    const graphics = Viva.Graph.View.svgGraphics();

    graphics.node((node) => {
      const ui = Viva.Graph.svg("rect")
        .attr("width", 10)
        .attr("height", 10)
        .attr("fill", "#00a2e8");

      ui.addEventListener('click', () => this.onSelect(node));

      return ui;
    });

    graphics.link(function (link) {
      return Viva.Graph.svg('line')
        .attr('stroke', '#999')
        .attr('data-weight', link.data?.weight || '')
        .attr('id', link.id);
    }).placeLink(function (linkUI, fromPos, toPos) {
      linkUI.attr('x1', fromPos.x)
        .attr('y1', fromPos.y)
        .attr('x2', toPos.x)
        .attr('y2', toPos.y);

      if (!linkUI.attr('data-weight')) {
        return;
      }

      const textId = "text_for_" + linkUI.id;
      const prevTextElement = document.getElementById(textId);
      prevTextElement?.parentElement.removeChild(prevTextElement);
      const el = Viva.Graph.svg('text')
        .attr('id', textId)
        .attr('x', ((fromPos.x + toPos.x) / 2).toString())
        .attr('y', ((fromPos.y + toPos.y) / 2).toString());
      el.textContent = linkUI.attr('data-weight');
      linkUI.parentElement.appendChild(el);
    });

    const layout = Viva.Graph.Layout.forceDirected(
      this.graph,
      {
        springLength: 80,
        springCoeff: 1e-4,
        dragCoeff: .05,
        gravity: -60,
        theta: .5,
      });
  }
}
```

```

        const renderer = Viva.Graph.View.rendererer(this.graph, {container, graphics,
layout});
        renderer.run();
    }

    graph = null;
    onSelect = null;

    setEventHandler(onSelect) {
        this.onSelect = onSelect;
    }

    getNodeById(nodeId) {
        return this.graph.getNode(nodeId);
    }

    addNodes(nodes) {
        this.graph.beginUpdate();
        for (const node of nodes) {
            this.graph.addNode(node.id, node);
        }
        this.graph.endUpdate();
    }

    addEdges(edges) {
        this.graph.beginUpdate();
        for (const edge of edges) {
            this.graph.addLink(edge.fromId, edge.toId, edge);
        }
        this.graph.endUpdate();
    }

    getAllNodeIds() {
        const nodes = [];
        this.graph.forEachNode(node => {
            nodes.push(node.id)
        });

        return nodes;
    }
}

```

ПРИЛОЖЕНИЕ Д

Листинг класса VkNodesLoader

(справочное)

```
class VkNodesLoader {
  async loadUser(userId) {
    const fetchResult = await fetch(`/VkApi/GetUserInfo?userId=${userId}`);
    return fetchResult.json();
  }

  async loadFriends(userId) {
    const fetchResult = await fetch(`/VkApi/GetFriends?userId=${userId}`);
    return fetchResult.json();
  }

  async loadGroup(groupId, neighbourGroupIds) {
    let ids = "";
    for (const id of neighbourGroupIds) {
      ids += `&neighbourGroupIds=${id}`
    }

    const fetchResult = await
fetch(`/VkApi/GetGroupInfo?groupId=${groupId}&neighbourGroupIds=${ids}`);
    return fetchResult.json();
  }
}
```

ПРИЛОЖЕНИЕ Е

Листинг скрипта анализа графа

(справочное)

```
import networkx
import json

from networkx import Graph

class EdgeModel:
    fromId: str
    toId: str
    weight: str

    @staticmethod
    def from_json(json_dct):
        obj = EdgeModel()
        obj.fromId = json_dct["FromId"]
        obj.toId = json_dct["ToId"]
        obj.weight = json_dct["Weight"] if "Weight" in json_dct else 0
        return obj

def main() -> None:
    import os
    json_edges = os.environ['VKGRAPH_INPUT']

    if len(json_edges) < 10:
        raise Exception('Unexpected length of args.')

    edges: list[EdgeModel] = json.loads(json_edges, object_hook=EdgeModel.from_json)

    graph: Graph = networkx.Graph()
    for edge in edges:
        graph.add_edge(edge.fromId, edge.toId, weight=edge.weight)

    result: dict[str, dict[str, float]] = {
        "degree": {node: val for (node, val) in networkx.degree(graph)},
        "degree_centrality": networkx.degree_centrality(graph),
        "betweenness_centrality": networkx.betweenness_centrality(graph),
        "closeness_centrality": networkx.closeness_centrality(graph),
        "eigenvector_centrality": networkx.eigenvector_centrality(graph),
        "katz_centrality": networkx.katz_centrality_numpy(graph),
        "harmonic_centrality": networkx.harmonic_centrality(graph),
    }

    print(json.dumps(result))
    return

main()
```

ПРИЛОЖЕНИЕ Ж

Листинг страницы анализа примера графа

(справочное)

```
Graph statistics
```

```
Isolated nodes:
```

```
Degree statistics:
```

```
Min degree: 1
```

```
Max degree: 505
```

```
Average degree: 2.5436046511627906
```

```
Median degree: 1
```

```
List of top-20 nodes ordered by 'degree':
```

```
0c947048-0000-0000-0000-000000000000 (value: 505)
18529531-0000-0000-0000-000000000000 (value: 150)
0f9d0074-0000-0000-0000-000000000000 (value: 136)
0a58cadc-0000-0000-0000-000000000000 (value: 89)
087f4e24-0000-0000-0000-000000000000 (value: 4)
08885cc9-0000-0000-0000-000000000000 (value: 4)
0a6f56a0-0000-0000-0000-000000000000 (value: 4)
0addf423-0000-0000-0000-000000000000 (value: 4)
0b5b4a96-0000-0000-0000-000000000000 (value: 4)
0bfd02d2-0000-0000-0000-000000000000 (value: 4)
0dc8fcde-0000-0000-0000-000000000000 (value: 4)
0f9f5776-0000-0000-0000-000000000000 (value: 4)
10afbda3-0000-0000-0000-000000000000 (value: 4)
11bda403-0000-0000-0000-000000000000 (value: 4)
11f484da-0000-0000-0000-000000000000 (value: 4)
12e7d16f-0000-0000-0000-000000000000 (value: 4)
13b1a556-0000-0000-0000-000000000000 (value: 4)
1733209c-0000-0000-0000-000000000000 (value: 4)
19cd902e-0000-0000-0000-000000000000 (value: 4)
```

```
List of top-20 nodes ordered by 'degreeCentrality':
```

```
0c947048-0000-0000-0000-000000000000 (value: 0.73508006)
18529531-0000-0000-0000-000000000000 (value: 0.2183406)
0f9d0074-0000-0000-0000-000000000000 (value: 0.19796215)
0a58cadc-0000-0000-0000-000000000000 (value: 0.12954876)
087f4e24-0000-0000-0000-000000000000 (value: 0.0058224164)
08885cc9-0000-0000-0000-000000000000 (value: 0.0058224164)
0a6f56a0-0000-0000-0000-000000000000 (value: 0.0058224164)
0addf423-0000-0000-0000-000000000000 (value: 0.0058224164)
0b5b4a96-0000-0000-0000-000000000000 (value: 0.0058224164)
0bfd02d2-0000-0000-0000-000000000000 (value: 0.0058224164)
0dc8fcde-0000-0000-0000-000000000000 (value: 0.0058224164)
0f9f5776-0000-0000-0000-000000000000 (value: 0.0058224164)
10afbda3-0000-0000-0000-000000000000 (value: 0.0058224164)
11bda403-0000-0000-0000-000000000000 (value: 0.0058224164)
11f484da-0000-0000-0000-000000000000 (value: 0.0058224164)
12e7d16f-0000-0000-0000-000000000000 (value: 0.0058224164)
13b1a556-0000-0000-0000-000000000000 (value: 0.0058224164)
1733209c-0000-0000-0000-000000000000 (value: 0.0058224164)
19cd902e-0000-0000-0000-000000000000 (value: 0.0058224164)
```

List of top-20 nodes ordered by 'betweennessCentrality':

```
0c947048-0000-0000-0000-000000000000 (value: 0.84723264)
0f9d0074-0000-0000-0000-000000000000 (value: 0.22755034)
18529531-0000-0000-0000-000000000000 (value: 0.19451986)
0a58cadc-0000-0000-0000-000000000000 (value: 0.11436783)
087f4e24-0000-0000-0000-000000000000 (value: 0.001497077)
08885cc9-0000-0000-0000-000000000000 (value: 0.001497077)
0a3deb02-0000-0000-0000-000000000000 (value: 0.001497077)
0a414f94-0000-0000-0000-000000000000 (value: 0.001497077)
0a6f56a0-0000-0000-0000-000000000000 (value: 0.001497077)
0a953100-0000-0000-0000-000000000000 (value: 0.001497077)
0acf6fed-0000-0000-0000-000000000000 (value: 0.001497077)
0addf423-0000-0000-0000-000000000000 (value: 0.001497077)
0b3091af-0000-0000-0000-000000000000 (value: 0.001497077)
0b5b4a96-0000-0000-0000-000000000000 (value: 0.001497077)
0bfd02d2-0000-0000-0000-000000000000 (value: 0.001497077)
0cfee462-0000-0000-0000-000000000000 (value: 0.001497077)
0d324983-0000-0000-0000-000000000000 (value: 0.001497077)
0d8ea7b8-0000-0000-0000-000000000000 (value: 0.001497077)
0dc8fcde-0000-0000-0000-000000000000 (value: 0.001497077)
```

List of top-20 nodes ordered by 'closenessCentrality':

```
0c947048-0000-0000-0000-000000000000 (value: 0.7574421)
18529531-0000-0000-0000-000000000000 (value: 0.5612745)
0f9d0074-0000-0000-0000-000000000000 (value: 0.5549273)
087f4e24-0000-0000-0000-000000000000 (value: 0.50145984)
08885cc9-0000-0000-0000-000000000000 (value: 0.50145984)
0a6f56a0-0000-0000-0000-000000000000 (value: 0.50145984)
0addf423-0000-0000-0000-000000000000 (value: 0.50145984)
0b5b4a96-0000-0000-0000-000000000000 (value: 0.50145984)
0bfd02d2-0000-0000-0000-000000000000 (value: 0.50145984)
0dc8fcde-0000-0000-0000-000000000000 (value: 0.50145984)
0f9f5776-0000-0000-0000-000000000000 (value: 0.50145984)
10afbda3-0000-0000-0000-000000000000 (value: 0.50145984)
11bda403-0000-0000-0000-000000000000 (value: 0.50145984)
11f484da-0000-0000-0000-000000000000 (value: 0.50145984)
12e7d16f-0000-0000-0000-000000000000 (value: 0.50145984)
13b1a556-0000-0000-0000-000000000000 (value: 0.50145984)
1733209c-0000-0000-0000-000000000000 (value: 0.50145984)
19cd902e-0000-0000-0000-000000000000 (value: 0.50145984)
1a1c9a62-0000-0000-0000-000000000000 (value: 0.50145984)
```

List of top-20 nodes ordered by 'eigenvectorCentrality':

```
0c947048-0000-0000-0000-000000000000 (value: 0.6603747)
18529531-0000-0000-0000-000000000000 (value: 0.2061074)
0f9d0074-0000-0000-0000-000000000000 (value: 0.15503572)
0a58cadc-0000-0000-0000-000000000000 (value: 0.10195482)
087f4e24-0000-0000-0000-000000000000 (value: 0.04748763)
08885cc9-0000-0000-0000-000000000000 (value: 0.04748763)
0a6f56a0-0000-0000-0000-000000000000 (value: 0.04748763)
0addf423-0000-0000-0000-000000000000 (value: 0.04748763)
0b5b4a96-0000-0000-0000-000000000000 (value: 0.04748763)
0bfd02d2-0000-0000-0000-000000000000 (value: 0.04748763)
0dc8fcde-0000-0000-0000-000000000000 (value: 0.04748763)
0f9f5776-0000-0000-0000-000000000000 (value: 0.04748763)
10afbda3-0000-0000-0000-000000000000 (value: 0.04748763)
11bda403-0000-0000-0000-000000000000 (value: 0.04748763)
```



```
11f484da-0000-0000-0000-000000000000 (value: 0.04748763)
12e7d16f-0000-0000-0000-000000000000 (value: 0.04748763)
13b1a556-0000-0000-0000-000000000000 (value: 0.04748763)
1733209c-0000-0000-0000-000000000000 (value: 0.04748763)
19cd902e-0000-0000-0000-000000000000 (value: 0.04748763)
```

List of top-20 nodes ordered by 'katzCentrality':

```
0f9d0074-0000-0000-0000-000000000000 (value: 0.44790465)
04d09532-0000-0000-0000-000000000000 (value: 0.05858799)
05b16bf0-0000-0000-0000-000000000000 (value: 0.05858799)
0995b12a-0000-0000-0000-000000000000 (value: 0.05858799)
09ebcd41-0000-0000-0000-000000000000 (value: 0.05858799)
0a1fd19f-0000-0000-0000-000000000000 (value: 0.05858799)
0a3b7502-0000-0000-0000-000000000000 (value: 0.05858799)
0a762c48-0000-0000-0000-000000000000 (value: 0.05858799)
0acf4186-0000-0000-0000-000000000000 (value: 0.05858799)
0b2a4042-0000-0000-0000-000000000000 (value: 0.05858799)
0ba22689-0000-0000-0000-000000000000 (value: 0.05858799)
0c467fc4-0000-0000-0000-000000000000 (value: 0.05858799)
0c564f92-0000-0000-0000-000000000000 (value: 0.05858799)
0c56e195-0000-0000-0000-000000000000 (value: 0.05858799)
0c5ca880-0000-0000-0000-000000000000 (value: 0.05858799)
0c7f12f0-0000-0000-0000-000000000000 (value: 0.05858799)
0cbd8ae8-0000-0000-0000-000000000000 (value: 0.05858799)
0cdb5864-0000-0000-0000-000000000000 (value: 0.05858799)
0ce04f6a-0000-0000-0000-000000000000 (value: 0.05858799)
```

List of top-20 nodes ordered by 'harmonicCentrality':

```
0c947048-0000-0000-0000-000000000000 (value: 589.6667)
18529531-0000-0000-0000-000000000000 (value: 418.5)
0f9d0074-0000-0000-0000-000000000000 (value: 411.5)
1b298455-0000-0000-0000-000000000000 (value: 345.5)
0a6f56a0-0000-0000-0000-000000000000 (value: 345.5)
1d322b49-0000-0000-0000-000000000000 (value: 345.5)
087f4e24-0000-0000-0000-000000000000 (value: 345.5)
12e7d16f-0000-0000-0000-000000000000 (value: 345.5)
0dc8fcde-0000-0000-0000-000000000000 (value: 345.5)
08885cc9-0000-0000-0000-000000000000 (value: 345.5)
11bda403-0000-0000-0000-000000000000 (value: 345.5)
19cd902e-0000-0000-0000-000000000000 (value: 345.5)
1733209c-0000-0000-0000-000000000000 (value: 345.5)
1a1c9a62-0000-0000-0000-000000000000 (value: 345.5)
10afbda3-0000-0000-0000-000000000000 (value: 345.5)
0addf423-0000-0000-0000-000000000000 (value: 345.5)
13b1a556-0000-0000-0000-000000000000 (value: 345.5)
0bfd02d2-0000-0000-0000-000000000000 (value: 345.5)
11f484da-0000-0000-0000-000000000000 (value: 345.5)
```

ПРИЛОЖЕНИЕ 3

Листинг файла конфигурации CI/CD

(справочное)

```
docker-build:
  image: docker:latest
  stage: build
  services:
    - docker:dind
  before_script:
    - docker login -u "$REGISTRY_USER" -p "$REGISTRY_PASSWORD" $REGISTRY
  script:
    - |
      if [[ "$COMMIT_BRANCH" == "main" ]]; then
        tag=""
        echo "Running on default branch '$DEFAULT_BRANCH': tag = 'latest'"
      else
        tag=":$COMMIT_REF_SLUG"
        echo "Running on branch '$COMMIT_BRANCH': tag = $tag"
      fi
    - docker build --pull -t "$REGISTRY_IMAGE${tag}" .
    - docker push "$REGISTRY_IMAGE${tag}"
  rules:
    - if: $COMMIT_BRANCH
      exists:
        - Dockerfile
```

ПРИЛОЖЕНИЕ И
(справочное)

Software Implementation of the Information System

Обучающийся

Группа	ФИО	Подпись	Дата
8ВМ1И	Сунцов Кирилл Сергеевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Ковин Р. В.	к.т.н., доцент		

Консультант-лингвист отделения иностранных языков ШБИП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Ануфриева Т. Н.			

1. Development of the Server-Side Component

1.1. Organization of Interaction with the "VKontakte" API

To obtain a user token with the "friends" permission for working with the VK API, you need to follow these steps:

Create an application on the VKontakte Developer portal. In this case, the application name is "Some app" (app51419292).

Refer to the VK API documentation to construct a URL for obtaining the token.

Listing 9 – Link for request an access token

```
https://oauth.vk.com/authorize?client_id=51419292&display=page&redirect_uri=https://o  
auth.vk.com/blank.html&scope=groups&response_type=token&v=5.131
```

There is no need to specify the 'offline' modifier as a short-lived token is sufficient for the application to work properly. Since the user token already has the user permissions, the 'users' modifier is not required. The required VK API version is 5.131. We also must use the placeholder (<https://oauth.vk.com/blank.html>) instead of the redirect link.

After following the link and confirming, a website address was obtained with the registered access token:

Listing 10 – Link and an access token

```
https://oauth.vk.com/blank.html#access_token=vk1.a. 23PF3XFTUhfXjUW3-  
Co4CL07ewjYbdo090LC6Lix42ZF1df8G5bga8HuvJPBGNWJ_f6-  
bgZMrLUXQUpYt5FQP3QtIvpRZKxR7GTToYT8dZKj1T_xoazs_UWYZnIewEY7JMVGeRZj6epZZTaEpioEvBG-  
kn4rzsCJLY5RkBDNoDh4G2Gpb5p33dJ-SMnfXLoF&expires_in=86400&user_id=768684892
```

Implementation of Business Logic

The server-side implementation is independent of the chosen graph visualization library and involves handling requests from the browser, making necessary requests to the VKontakte API, and caching the results of these requests.

Since some API requests require result pagination, a module was developed to handle such requests. This module was encapsulate the logic for sending multiple requests. Please refer to Appendix A for the detailed presentation of this module.

It is observed that the request for the list of friends is designed to retrieve data in chunks of 1000 records. This is due to the limitation imposed by the VKontakte API on the amount of data that can be obtained in a single request.

The VK API also has restrictions on the number of requests per second. Additionally, since it is an external service, executing these requests can take a considerable amount of time. To avoid unnecessary external requests, a caching mechanism was implemented to trace and store the results. As that the expected application load is peak-based, a short-term in-memory cache can be utilized. The detailed presentation of this module is presented in Appendix A.

To facilitate the interaction between the VK API module and the caching module, linking functions have been implemented in the main application code. The implementation of this caching mechanism is presented in Appendix B.

1.2. Organization of Interaction with the "Odnoklassniki" API

The process of data retrieval from the "Odnoklassniki" social network is almost identical to that of the VKontakte social network. Therefore, the same principles of operation are applied. The main difference lies in the method of obtaining an access token, the addresses of the required methods, and the response format of the server.

When working with the "Odnoklassniki" API, the following steps need to be followed:

1. Create an application on the "Odnoklassniki" Developer portal to obtain the necessary credentials.
2. Authenticate the user and obtain an access token using the appropriate authentication flow specified by the "Odnoklassniki" API.
3. Use the obtained access token to make API requests to retrieve data such as user profiles, friends, and posts.
4. Parse and process the API responses, adhering to the response format specific to the "Odnoklassniki" API.

It is important to note that while the principles of interaction with the "Odnoklassniki" API are similar to the VKontakte API, the specific details such as

endpoint addresses, parameter formats, and response structures may differ. Referring to the "Odnoklassniki" API documentation provided the necessary information for proper implementation.

By applying similar principles and adapting the implementation to the "Odnoklassniki" API, data can be collected from the "Odnoklassniki" social network as part of the overall system functionality.

2. Development of the Client-Side Component

2.1. Selection of a Graph Visualization Library

When choosing a graph visualization library, the following factors have been taken into account:

- The application utilizes JavaScript without any frameworks.
- The graph should be interactive.
- The graph should have aesthetically pleasing layouts and ideally support simulations of the elasticity of connections from a physical perspective.

After analyzing existing solutions, the Diagraphs library has been identified as meeting the specified requirements. Other libraries, such as Vis.JS, required the presence of third-party frameworks (React, Angular) [23] or did not provide the ability to handle vertex click events [24].

The functionality of working with the selected library has been encapsulated in a separate class, which is presented in Appendix G.

6.2. Establishing Interaction with the Server-Side Component

Since the interaction with the server-side component is performed through a REST JSON API using HTTP requests, it is not architecturally well-suited for embedding within JavaScript code. To address this issue, the server-side interaction methods have been encapsulated in a separate class, presented in Appendix D.

The entry points used in the application can be defined as follows:

Listing 11 – Description of entry points

<pre>/Vape/GetFriends/{id}: get:</pre>
--

```

summary: "* VkApi/GetFriends/{id}"
parameters:
- name: "id"
  in: "path"

/VkApi/GetGroupInfo/{id}:
get:
summary: "* VkApi/GetGroupInfo/{id}"
parameters:
- name: "id"
  in: "path"

/VkApi/GetUserInfo/{id}:
get:
summary: "* VkApi/GetUserInfo/{id}"
parameters:
- name: "id"
  in: "path"

```

The results, which represent the nodes and edges of the graph, are defined as follows:

Listing 12 – Models of nodes and edges

```

using System;
using VkNet.Model;

namespace VkGraphBuilder.BusinessLogic
{
    [Serializable]
    public class NodeModel
    {
        public long Id { get; set; }
        public string Name { get; set; }
        public string ImageSrc { get; set; }
        public string Extras { get; set; }
        public string Link { get; set; }
    }

    [Serializable]
    public class EdgeModel
    {
        public long FromId { get; set; }
        public long ToId { get; set; }
        public uint Weight { get; set; }
    }
}

```

An illustrative example of a graph constructed using this approach is depicted in Figure 6.

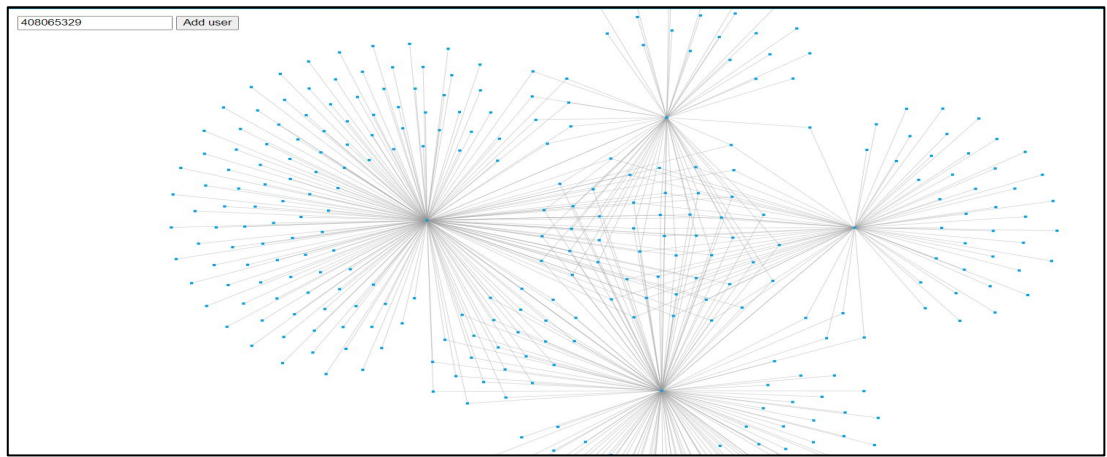


Fig 6 – A graph with four fully 'loaded' nodes

Clusters of mutual friends can be observed among two, three, and four fully loaded profiles (Figure 6).

2.2. Implementation of Graph Analysis Function

Graph theory provides a wide range of tools for graph analysis [25]. For instance, an algorithm for identifying isolated nodes can be utilized to determine communities that have no common members with other communities represented in the graph. Conversely, the algorithm for determining node degrees can be employed to identify communities that share many members with other communities. Algorithms for measuring centrality, both individually and collectively, allow for meaningful insights into the significance of a community within the graph of communities [24, 25].

Analyzing the mentioned indicators can be valuable for individual nodes as well as for all nodes in the graph simultaneously. This work incorporates both possibilities: the values of individual indicators are displayed in the node information panel, while the overall graph indicators are presented on a separate page accessed by clicking the "Open statistics page" button.

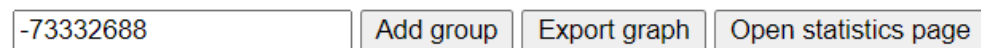


Figure 7 – Graph control buttons

To compute statistical indicators, the Python library NetworkX is utilized. For this purpose, a separate entry point, responsible for graph analysis using this library, is created.

Listing 13 – Source code of entry point of the graph analysis tool

```
public class AnalyzerController : Controller
{
    private readonly ILogger<AnalyzerController> logger;

    public AnalyzerController(
        ILogger<AnalyzerController> logger,
        NetworkXWrapper.NetworkXWrapper networkXWrapper)
    {
        NetworkXWrapper = networkXWrapper;
        this.logger = logger;
    }

    private NetworkXWrapper.NetworkXWrapper NetworkXWrapper { get; }

    [HttpPut]
    public async Task<AnalyzingResult> Analyze([FromBody] List<EdgeModel> edges)
    {
        var calcStatisticsAsync = await
NetworkXWrapper.CalcStatisticsAsync(edges);
        return calcStatisticsAsync;
    }
}
```

The above code passes the obtained graph for processing to a Python script (provided in Appendix E) and returns the analysis results. In the client-side of the web application, the node degree is displayed on the node information card.



Figure 8 – The information about the graph nodes and the analysis results

The statistics for node degrees in the graph are displayed on a separate page when clicking the "Open statistics page" button. An example of overall graph statistics is provided in Appendix H.

Node centrality refers to a metric that reflects the "importance" of nodes in a graph and their proximity to the center of the graph. There are several formulas for computing centrality, such as degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, Katz centrality, and harmonic centrality.

We can calculate these centrality metrics using the script provided in Appendix E. The result of computing each centrality metric is a dictionary with "Node ID - Value" pairs.

An example visualization of these metrics for a single node is shown in Figure 3.

Appendix H presents an example output of the node IDs with the highest values for each centrality metric in a graph consisting of 900 nodes. Based on this, we can conclude that none of the nodes are isolated, while there are 4 nodes with the highest number of connections. The node with the highest degree centrality has connections with over half of the nodes in the graph. Additionally, there are at least 16 nodes that have connections with all four of the analyzed nodes.

The mentioned 4 nodes have the highest centrality values in the graph. The node with the highest degree centrality has the most connections, while the node with the highest Katz centrality has the most direct and indirect connections with other nodes. Within the context of the domain, this indicates that this community has the highest number of related thematic communities compared to others [26].

3. Implementation of Graph Export Function

There are numerous graph analysis tools available. To allow the graph generated by the web application to be opened by these tools, it is necessary to provide the ability to export the graph to a file.

For exporting, the TGF (Trivial Graph Format) file format used by the "yEd Graph editor" has been selected. This format is a 3D-SSV (Space-Separated Values) format [27], consisting of two sections: edge descriptions and node descriptions. The sections are separated by the hash symbol ('#'). The edge description section consists of pairs of numbers representing the identifiers of the nodes, while the node description section consists of pairs of "node identifier - node name". An example file is provided below.

Listing 14 – Example of an exported file

```
045ef7d3-0000-0000-0000-000000000000 qqq
045ef7d2-0000-0000-0000-000000000000 eee
045ef7d5-0000-0000-0000-000000000000 DELETED
045ef7d6-0000-0000-0000-000000000000 www
045ef7d7-0000-0000-0000-000000000000 eeee
045ef7d8-0000-0000-0000-000000000000 rrrr
045ef7d9-0000-0000-0000-000000000000 tttt
045ef7da-0000-0000-0000-000000000000 yyy
045ef7db-0000-0000-0000-000000000000 hhhh
#
045ef7d3-0000-0000-0000-000000000000 045ef7d3-0000-0000-0000-000000000000
045ef7d8-0000-0000-0000-000000000000 045ef7d2-0000-0000-0000-000000000000
045ef7db-0000-0000-0000-000000000000 045ef7d2-0000-0000-0000-000000000000
045ef7db-0000-0000-0000-000000000000 045ef7d8-0000-0000-0000-000000000000
```

By exporting the graph to the TGF file format, users can utilize various analysis tools to examine and manipulate the graph data.

4. Containerization of the Web Application

Containerization of web applications is a widely adopted practice for deploying web applications on third-party hosting service providers. Containerization allows for isolating the web application from the hosting environment while avoiding the overhead of fully virtualizing the web application and its execution environment (as in the case of using virtual machines) [31].

In this work, Docker is used for containerizing the web application. To configure the containerization, a Dockerfile of the following structure was created.

Listing 15 – Docker configuration file

```
FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build

WORKDIR /src
COPY . .

WORKDIR "/src/WebUI"
RUN dotnet restore "WebUI.csproj"
RUN dotnet build "WebUI.csproj" -c Release -o /app/build
RUN dotnet publish "WebUI.csproj" -c Release -o /app/publish

WORKDIR /app/publish
RUN apt update -y && apt install python3 python3-venv -y \
    python3 -m venv venv \
    /venv/bin/python3 -m pip install -f NetworkXWrapper/requirements.txt

FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
EXPOSE 80
EXPOSE 443

WORKDIR /app
```

```
COPY --from=build /app/publish .  
ENTRYPOINT ["dotnet", "VkGraphBuilder.WebUI.dll"]
```

The provided file consists of two main sections: the build image configuration and the runtime image configuration. This separation is necessary because the build process typically requires more resources compared to the execution of the built code. The separation helps to avoid including unnecessary files in the resulting container.

In the build image configuration section, the necessary steps are taken to build the application, such as installing dependencies and copying the application files. This section focuses on creating an environment for compiling and preparing the application.

In the runtime image configuration section, the focus is on creating a lightweight environment to execute the built application. It typically includes copying the compiled application files from the build image and configuring the necessary runtime settings.

By separating the build and runtime configurations, unnecessary files and dependencies from the build environment are excluded from the final runtime image. This helps to optimize the size and performance of the container, as well as ensure that only the essential files and resources are included.

5. Storing the source code of a web application

Using Git repositories to store and manage the source code of applications is a common and recommended practice in modern software development. In this section, we provided an explanation and description of the benefits of using Git repositories for storing and managing the source code of the web application.

Git is a distributed version control system (VCS) that enables developers to effectively manage the history of changes in a project's source code. The main advantages of using Git include:

1. **Change History and Recovery:** Git preserves the complete history of changes in the project, allowing programmers to view, analyze, and restore previous versions of the code. This enables programmers to track what changes have been made, who made them, and easily revert to previous code states.

2. **Branching and Merging:** Git provides powerful branching and merging features, allowing developers to create independent branches for working on new features or bug fixes. Branching allows programmers to develop and test new features without affecting the main development branch, while merging enables programmers to combine changes from different branches into one, ensuring code integrity.
3. **Collaboration:** Git facilitates collaboration among multiple developers working on a project. Each developer can create their own branches, make changes, and share their code with other team members. Git provides tools for managing conflicts when merging changes from different developers.
4. **Convenience and Flexibility:** Git has a simple and intuitive command-line interface and a variety of graphical user interfaces (GUIs). This makes Git accessible to developers with different levels of experience. Additionally, Git repositories can be easily integrated with various development tools, such as integrated development environments (IDEs) or continuous integration and deployment (CI/CD) systems, providing convenience and flexibility in working with code.

For working with the code, a repository named "GraphBuilder" was created on the GitLab server of Tomsk Polytechnic University. The "main" branch is used as the main development branch. During the development process, branches were also created for demonstration purposes, containing the so-called "minimum viable product" (MVP). Later, the code in these branches was merged with the code in the "main" branch.

Merging branches can be performed using various methods, including command-line operations using the "git" utility (see Listing 16) or through the GitLab web interface.

Listing 16 – Branches merging commands

```
git switch main
git merge mvp_v2
git push origin main
```

To perform a merge using the web interface, a developer needs to create a "merge request." To do this, follow these steps:

1. Open the branch list in the repository.
2. Click the "Merge request" button next to the target branch.
3. Enter the merge request title, description, and fill in other fields if necessary.
4. Click the "Create merge request" button.
5. In the created merge request, click "Merge."

It is important to note that this sequence of actions is simplified and intended for merging changes between branches. When working in a team, it is necessary to obtain approval for the merge request from other team members.

6. Building the Web Application

To automate the various stages of building and testing web applications, there are "build pipelines." They provide tools for automating different stages, starting from code compilation and building to deploying the application on a server. The main goal of using build pipelines is to speed up the development process, ensure reliability, and improve the quality of the final product.

GitLab build pipelines facilitate three main stages:

1. **Application build:** This stage involves compiling and building the application code. It ensures that the application is correctly packaged and ready for further testing and deployment.
2. **Application testing:** In this stage, various tests are executed to verify the functionality and quality of the application. This may include unit tests, integration tests, and performance tests, among others. The test results help identify and address any issues or bugs in the application.
3. **Application deployment:** The final stage involves deploying the application to a server or hosting environment, making it accessible for end-users. This stage ensures that the application is properly deployed and ready for use.

The status of each stage is reflected by the corresponding indicator on the build pipeline dashboard (see Figure 8). This allows developers to track the progress and

status of the build process, ensuring transparency and visibility into the different stages of application development.



Figure 9 – Build pipeline dashboard

The CI/CD pipeline can be configured both through the GitLab web interface and by using a dedicated configuration file in the repository. In this project, configuration using a file was chosen because it allows for the automation of component reconfiguration and easy rollback to a previous state in case of failure.

To configure the CI/CD pipeline in GitLab, the file ".gitlab-ci.yml" is used. This file defines the configuration for the Continuous Integration and Deployment (CI/CD) process and contains a set of instructions for automatically executing various stages during the build and testing of the application.

The ".gitlab-ci.yml" file can be used to configure the following aspects:

1. Defining build stages: The ".gitlab-ci.yml" file allows programmers to define the sequence of build stages and their dependencies. For example, a programmer can define stages for code compilation, running automated tests, building a container, and deploying to a server.
2. Setting up environments and variables: The file enables programmers to define environment variables that can be used during the build and deployment process. This can include information about databases, API keys, or other sensitive data, as well as parameters specific to the development environment.
3. Managing job triggering and parallelism: The ".gitlab-ci.yml" file allows programmers to configure conditions for job triggering, such as triggering on specific events (e.g., repository push) or scheduled runs. Additionally, programmers can define parallel execution of jobs to accelerate the build and testing process.

4. Integrating with external services: The file enables programmers to specify integration settings with external services, such as database management systems, error tracking systems, or notification systems. This automates the interaction with different tools used in the development process.
5. Managing results and notifications: The ".gitlab-ci.yml" file allows programmers to define how to handle build and test results. This can include defining conditions for successful or failed stages, storing build artifacts (e.g., binary files or test reports), and configuring notifications about build results, such as sending email notifications or integrating with notification systems.

Configuring the CI/CD pipeline using the ".gitlab-ci.yml" file provides great flexibility and powerful customization options for the CI/CD process. It allows developers to define and manage the build and deployment steps according to their project's requirements and specific needs. This simplifies the development process, accelerates the release of new changes, and ensures the stability and reliability of the application.

In the developed application, the CI/CD was configured using a file that contained the following stages:

1. Authentication to DockerHub.
2. Forming the Docker tag. The tag is derived from the current branch name, or if the current branch is "main," the keyword "latest" is used.
3. Building the Docker image with the specified tag.
4. Pushing the built Docker image to DockerHub.

The complete listing of the file can be found in Appendix 3. It can be observed that the file uses variables instead of specific values. These variables allow for the reuse of the same file for building in multiple environments.

6. Web-application deployment

A service that monitors updates to a Docker image in a specified repository was chosen as the web application hosting provider. When an image update occurs, this service automatically updates the application. It is worth noting that this functionality

is supported by many web application hosting providers [7], enabling prompt publication of the application on a different hosting platform if necessary.

ПРИЛОЖЕНИЕ К

Диаграмма активностей

(справочное)

