

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
 федеральное государственное автономное образовательное учреждение высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
 ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт электронного обучения _____
 Специальность Прикладная информатика
 Кафедра Оптимизации систем управления

ДИПЛОМНЫЙ ПРОЕКТ/РАБОТА

Тема работы
Реализация системы описания и поиска потребностей и возможностей пользователей, основанная на сетевых БД для применения в приложениях к социальным сетям.

УДК 004.65: 004.738.5

Студент

Группа	ФИО	Подпись	Дата
3-8К11	Магсумзянов Артур Шавкетович		09.06.2016

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Эксперт ЦТТ	Черний А. В.		09.06.2016	

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Николаенко В. С.			21.05.16

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Антоневич О. А.	К.Т.Н.		01.06.16

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Фофанов О.Б.	К.Т.Н		10.06.16

Томск – 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
 федеральное государственное автономное образовательное учреждение высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
 ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Институт кибернетики
 Специальность Прикладная информатика (в экономике)
 Кафедра Оптимизации систем управления

УТВЕРЖДАЮ:
 Зав. Кафедрой ОСУ
 Иванов А. М.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

дипломной работы

Студенту:

Группа	ФИО
3-8К11	Магсумзянову Артуру Шавкетовичу

Тема работы:

Реализация системы описания и поиска потребностей и возможностей пользователей, основанная на сетевых БД для применения в приложениях к социальным сетям.	
Утверждена приказом директора (дата, номер)	10.03.2016 №1760/С

Срок сдачи студентом выполненной работы:	10 июня 2016 г.
--	-----------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	1. Документация по реализуемой системе 2. Техническое задание
Перечень подлежащих исследованию, проектированию и разработке вопросов	1. Обзор проектов социальных сетей и поисковых движков 2. Описание проекта 3. Подбор средств и инструментов для реализации проекта 4. Разработка интерфейса 5. Разработка бизнес-логики 6. Тестирование 7. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение 8. Социальная ответственность Приложение А. An overview project with search in graph database.
Перечень графического материала <i>(с точным указанием обязательных чертежей)</i>	презентация в Microsoft Power Point

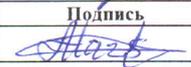
Консультанты по разделам выпускной квалификационной работы <i>(с указанием разделов)</i>	
Раздел	Консультант
Глава 7. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Николаенко В. С.
Глава 8. Социальная ответственность	Антоневич О. А.
Приложение А. An overview project with search in graph database.	
Названия разделов, которые должны быть написаны на русском и иностранном языках:	
Обзор англоязычной литературы по теме ВКР (Приложение А)	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	10 марта 2016 г.
---	------------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Эксперт ЦТТ	Черний А. В.			09.06.2016

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3-8К11	Магсумзянов А.Ш.		09.06.2016

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ по ООП

Код результата	Результат обучения (выпускник должен быть готов)
<i>ПРОФЕССИОНАЛЬНЫЕ компетенции</i>	
ПК-1	Способен использовать нормативные правовые документы в профессиональной деятельности.
ПК-2	Способен при решении профессиональных задач анализировать социально-экономические проблемы и процессы с применением методов системного анализа и математического моделирования.
ПК-3	Способен использовать основные законы естественнонаучных дисциплин в профессиональной деятельности и эксплуатировать современное электронное оборудование и информационно-коммуникационные технологии в соответствии с целями образовательной программы бакалавра.
ПК-4	Способен ставить и решать прикладные задачи с использованием современных информационно-коммуникационных технологий.
ПК-5	Способен осуществлять и обосновывать выбор проектных решений по видам обеспечения информационных систем.
ПК-6	Способен документировать процессы создания информационных систем на всех стадиях жизненного цикла.
ПК-7	Способен использовать технологические и функциональные стандарты, современные модели и методы оценки качества и надежности при проектировании, конструировании и отладке программных средств.
ПК-8	Способен проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе, участвовать в реинжиниринге прикладных и информационных процессов.
ПК-9	Способен моделировать и проектировать структуры данных и знаний, прикладные и информационные процессы.
ПК-10	Способен применять к решению прикладных задач базовые алгоритмы обработки информации, выполнять оценку сложности алгоритмов, программировать и тестировать программы.
ПК-11	Способен принимать участие в создании и управлении ИС на всех этапах жизненного цикла.
ПК-12	Способен эксплуатировать и сопровождать информационные системы и сервисы.
ПК-13	Способен принимать участие во внедрении, адаптации и настройке прикладных ИС.
ПК-14	Способен принимать участие в реализации профессиональных коммуникаций в рамках проектных групп, презентовать результаты проектов и обучать пользователей ИС.
ПК-15	Способен проводить оценку экономических затрат на проекты по информатизации и автоматизации решения прикладных задач.
ПК-16	Способен оценивать и выбирать современные операционные среды и информационно-коммуникационные технологии для информатизации и автоматизации решения прикладных задач и создания ИС.
ПК-17	Способен применять методы анализа прикладной области на концептуальном, логическом, математическом и алгоритмическом уровнях.
ПК-18	Способен анализировать и выбирать методы и средства обеспечения информационной безопасности.
ПК-19	Способен анализировать рынок программно-технических средств, информационных продуктов и услуг для решения прикладных задач и создания информационных систем.
ПК-20	Способен выбирать необходимые для организации информационные ресурсы и источники знаний в электронной среде.
ПК-21	Способен применять системный подход и математические методы в формализации решения прикладных задач.
ПК-22	Способен готовить обзоры научной литературы и электронных информационно-образовательных ресурсов для профессиональной деятельности.
<i>Универсальные компетенции</i>	
ОК-1	Способен использовать, обобщать и анализировать информацию, ставить цели и находить пути их достижения в условиях формирования и развития информационного общества.
ОК-2	Способен логически верно, аргументировано и ясно строить устную и письменную речь, владеть навыками ведения дискуссии и полемики.
ОК-3	Способен работать в коллективе, нести ответственность за поддержание партнерских, доверительных отношений.
ОК-4	Способен находить организационно-управленческие решения и готов нести за них ответственность.
ОК-5	Способен самостоятельно приобретать и использовать в практической деятельности новые знания и умения, стремиться к саморазвитию.

ОК-6	Способен осознавать социальную значимость своей будущей профессии, обладать высокой мотивацией к выполнению профессиональной деятельности.
ОК-7	Способен понимать сущность и проблемы развития современного информационного общества.
ОК-8	Способен работать с информацией в глобальных компьютерных сетях.
ОК-9	Способен свободно пользоваться русским языком и одним из иностранных языков на уровне, необходимом для выполнения профессиональных задач.
ОК-10	Способен использовать методы и средства для укрепления здоровья и обеспечения полноценной социальной и профессиональной деятельности.
ОК-11	Способен уважительно и бережно относиться к историческому наследию и культурным традициям, толерантно воспринимать социальные и культурные различия.
ОК-12	Способен использовать Гражданский кодекс Российской Федерации, правовые и моральные нормы в социальном взаимодействии и реализации гражданской ответственности.
ОК-13	Способен понимать сущность и значение информации в развитии современного информационного общества, сознавать опасности и угрозы, возникающие в этом процессе, соблюдать основные требования информационной безопасности, в том числе защиты государственной тайны.
ОК-14	Способен применять основные методы защиты производственного персонала и населения от возможных последствий аварий, катастроф, стихийных бедствий, технику безопасности на производстве.

Реферат

Выпускная квалификационная работа состоит из 71 страниц, 13 рисунков, 19 таблиц, 6 источников, 1 приложения.

Ключевые слова: Графовая СУБД, потребности, возможности, поиск связей, взаимодействие, .Net C#, ASP MVC

Объектом исследования является потребности и возможности пользователей, графовая СУБД для применения в приложениях к социальным сетям.

Цель работы – реализация системы описания и поиска потребностей и возможностей пользователей, основанная на сетевых БД для применения в приложениях к социальным сетям.

В процессе исследования проводилось изучение возможностей графовой СУБД Neo4J, создавалась система для описания и поиска взаимосвязей, информирования пользователей о новых потребностях других пользователей системы.

В результате исследования была создана ИС с помощью, которой пользователи могут описать потребности и возможности, найти возможных исполнителей или помочь в исполнении другому пользователю.

Основные конструктивные, технологические и технико-эксплуатационные характеристики: используется сервер под управлением Windows, веб приложение можно запустить с помощью Internet Information Services, используется платформа .Net, язык программирования C#.

Степень внедрения: находится в стадии тестирования бета версии сайта

Область применения: социальные сети и поисковые системы.

Экономическая эффективность/значимость работы есть большой интерес к проекту, пользователям нравится идея и сайт, экономическая эффективность высокая.

В будущем планируется развивать данный сервис для выполнения основных задач ИС.

Перечень условных обозначений, единиц и терминов

GET-запрос – тип HTTP-запроса, состоящий только из HTTP-заголовков, тело у него отсутствует.

HTTP – HyperText Transfer Protocol (протокол передачи гипертекста). Протокол прикладного уровня передачи данных в первую очередь в виде текстовых сообщений.

POST-запрос – тип HTTP-запроса, в котором пара значение/переменная передается в теле заголовка.

SQL (структурированный язык запросов) – стандартный не процедурный язык, который ориентирован на обработку данных.

XML – eXtensible Markup Language. Текстовый формат, который предназначен для хранения структурированных данных, для обмена информацией между программами.

Авторизация – процесс входа в систему.

База данных (БД) – программное обеспечение для хранения, обработки и получения информации. Информация, сохраненная в БД, хранится там до тех пор, пока она не будет удалена, независимо от выключения аппаратного обеспечения или остановки программных задач.

ИС – информационная система

ПК – персональный компьютер

ПО – программное обеспечение

СУБД – система управления базами данных

Оглавление

Реферат	6
Перечень условных обозначений, единиц и терминов	7
Введение	10
1. Обзор проектов социальных сетей и поисковых сервисов	11
2. Описание проекта.....	16
3. Подбор средств и инструментов для реализации проекта	18
3.1. Рассмотрены классические СУБД.....	18
3.2. Обоснование нерациональности использования классических СУБД	19
3.3. Обоснование выбора графовой БД.....	20
3.4. Обоснование выбора платформы и инструмента разработки.....	22
3.4.1 Среда .NET Framework	22
3.4.2 Особенности CLR	24
3.4.3 .NET Framework – Библиотека классов	27
3.4.4 Инструмент Visual Studio	28
3.4.5 Язык программирования C# – отличия и преимущества	29
3.4.6 Технология ASP .NET MVC 4.0.....	30
4. Разработка интерфейса	34
4.1. Инструменты, используемые для улучшения интерфейса	34
4.2. Проработка механизмов взаимодействия пользователя с системой	35
4.2.1 Информация о пользователе	35
4.2.2 Список контактов первого круга	36
4.2.3 Список желаний, страницы для редактирования.....	37
4.2.4 Список возможностей.	37
4.2.5 Пользовательский чат.....	38
4.2.6 Раздел новостей о потребностях друзей.....	38
4.2.7 Раздел переданных просьб	39
5. Разработка бизнес-логики.....	40
5.1. Модели.....	40
5.2. Алгоритмы поиска.....	41
6. Тестирование	44

7. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	45
7.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.	45
7.1.1 Потенциальные потребители результатов исследования	45
7.1.2. Анализ конкурентных технических решений	46
7.1.3. Технология QuaD.....	46
7.1.4. SWOT – анализ	47
7.2. Определение возможных альтернатив проведения научных исследований.	49
7.3. Планирование научно-исследовательских работ.	49
7.3.1. Структура работ в рамках научного исследования.	49
7.3.2. Определение трудоемкости выполнения работ	51
7.3.3 Разработка графика проведения проектирования.....	51
7.4 Бюджет научно-технического исследования (НТИ).	57
7.4.1. Материальные затраты.	57
7.4.2. Расчет затрат на оборудование для научно-экспериментальных работ.	58
7.4.3. Основная заработная плата исполнителей темы.....	59
7.4.4 Дополнительная заработная плата исполнителей темы	63
7.4.5. Отчисления во внебюджетные фонды (страховые отчисления)	63
7.4.6. Накладные расходы	64
7.4.7. Формирование бюджета затрат научно-исследовательского проекта	64
7.5. Определение ресурсной, финансовой, бюджетной, социальной и экономической эффективности исследования	65
Заключение	68
Список использованных источников	69
Приложение А	70

Введение

Информационный мир всё больше внедряется в нашу жизнь, предлагая нам более комфортные условия использования реальных услуг. Вместе с тем у людей появляются более новые желания и необходимость в хороших сервисах, для решения различных задач. Примером может быть сервисы по аренде ближайшей такси-машины, поисковые сервисы для поиска различных типов данных, информационно развлекательные ресурсы в виде онлайн газет.

Предметной областью данной работы является ИС, для взаимодействия людей и реализации желаний пользователей данной системы с возможностью поиска связей через графы.

Проблемой является то, что ИС использует большие объёмы данных введенные пользователями и должна научиться их анализировать, самообучаться и находить связи. Пользователям будет доступна функция передачи другим пользователям вне найденных связей, тем самым обучая систему.

Целью работы является создание ИС, которая будет построена на графах и графовой базе данных, использовать современные средства и языки разработки, создать приятную для использования систему и интерфейс, где основной целью ставится создание платформы для потребностей одних пользователей и возможностей других, с целью их соединения и взаимодействия, для решения общих для них задач.

С практической точки зрения проект может использоваться как социальная сеть или использовать другие социальные сети и поисковые системы для своей работы.

1. Обзор проектов социальных сетей и поисковых сервисов

На данный момент практически невозможно встретить людей, которые бы не пользовались сервисами поисковых систем и социальными сетями.

Поисковые сервисы это ИС, предназначенная для поиска информации. Одно из наиболее известных применений поисковых систем, веб-сервисы для поиска текстовой или графической информации в Интернете. [1]

Для поиска информации с помощью поисковой системы пользователь формулирует поисковый запрос. Работа поисковой системы заключается в том, чтобы по запросу пользователя найти документы, содержащие либо указанные ключевые слова, либо слова, как-либо связанные с ключевыми словами. При этом поисковая система генерирует страницу результатов поиска. Такая поисковая выдача может содержать различные типы результатов, например: веб-страницы, изображения, аудиофайлы.

"Google" это самая популярная поисковая система в мире, её название давно уже стало нарицательным. Доля рынка, которую занимает "Google" среди остальных поисковиков, превышает 60 процентов. Данным сервисом индексируется около восьми миллиардов страниц, при этом запросы могут приниматься примерно на 190 мировых языков. По многим параметрам "Google" не имеет равных в своем сегменте.

"Yahoo" на следующем по популярности месте расположился поисковик "Yahoo!", чья доля на рынке составляет около 13 процентов. Как утверждают ряд достаточно серьезных источников, данная поисковая система является самым посещаемым ресурсом глобальной сети (более трех миллиардов запросов за сутки).

"Bing" ресурс, который является собственностью "Microsoft", входит в тройку самых используемых в мире, а так же встроен в стандартные браузеры операционной системы "Windows10". В число характерных черт данного поисковика входит оформление заглавной странички, которая меняется каждый день, удобный поиск видеозаписей, демонстрация дополнительной информации

по запросам. Специалисты отмечают, что данный ресурс обладает большим потенциалом.

"Яндекс" это популярнейший русскоязычный поисковик. Кроме того, он является наиболее крупным неанглоязычным ресурсом глобальной сети. Еще одна цифра – он расположился на седьмом месте среди всех поисковых машин. "Яндекс" позволяет настраивать поиск с максимальной точностью, ограничивать поисковый охват по территориальному принципу или по дате, настраивать число предлагаемых ссылок, которые выходят на странице. Безусловно, это лучшая русскоязычная поисковая система.

"Rambler" этот поисковик появился на свет еще в далеком 1996 году. Он позволяет производить поиск по морфологическому принципу. Выдача ссылок происходит исходя из популярности сайтов.

"Nigma" одной из характерных черт этого поисковика является то, что страницы могут находиться как по поисковому индексу, так и по индексу других поисковых систем. Помимо этого, данная поисковая система позволяет решать различные математические задачи, осуществлять построение графиков функций и так далее. По этой причине этот ресурс имеет достаточную популярность среди студентов и старшеклассников.

Социальная сеть - это платформа, онлайн-сервис и веб-сайт, предназначенные для построения, отражения и организации социальных взаимоотношений в Интернете.

Социальные сети являются мощным инструментом маркетинговых исследований, поскольку пользователи добровольно публикуют информацию о себе, своих взглядах, интересах, предпочтениях и так далее. Ввиду этого рекламодатели могут весьма четко определять, каких именно пользователей заинтересует их объявление, и направить свои рекламные объявления конкретным пользователям, в зависимости от информации в их профилях (возраст, пол, место жительства и прочее). [2]

"Вконтакте" самая популярная русскоязычная социальная сеть. Величина ежедневной аудитории этого сервиса превышает 60 млн, что примерно

соответствует уровню самого популярного российского поисковика "Яндекс". Основными пользователями "ВКонтакте" являются молодые люди в возрасте от 17 до 35 лет, которые заходят сюда не только пообщаться, но и послушать музыку, посмотреть фильмы и клипы, посмотреть интересные картинки и фотографии. Помимо этого, "ВКонтакте" получила популярность благодаря огромному числу различных «пабликов» и страниц знаменитых людей и музыкальных групп, на которых публикуются все самые горячие новости. Эти «паблики» позволяют пользователям создавать группы по интересам, узнавать свежую информацию, делиться фотографиями и делать приглашения друг другу на концерты и прочие события.

"Одноклассники" на данный момент этот сервис занимает второе место среди социальных сетей в России, без больших шансов подняться вверх и без особого риска спуститься вниз. "Одноклассники" были одной из первых социальных сетей в России и до сих пор остаются любимыми среди определенной части населения. В основном это касается представителей среднего и зрелого возраста. Главной «фишкой» "Одноклассников" считается возможность пользования специальной валютой, которая позволяет приобретать подарки, производить смену элементов интерфейса, а также управлять учетной записью.

"Facebook" эта американская социальная сеть на данный момент в мире считается номер один. Но при этом в нашей стране она обладает достаточно ограниченной популярностью из-за того, что не позволяет в полной мере осуществлять просмотр и прослушивание фильмов, видеоклипов и аудиозаписей. Кроме того, люди, которые регулярно пользуются "Facebook", считают, что этот сервис обладает перегруженным интерфейсом, что осложняет привыкание и применение возможностей социальной сети.

"Twitter" аудитория этого сервиса в России сравнительно малочисленна и составляет всего два-три миллиона в сутки. Конечно, если проводить сравнение с другими ресурсами, суточный трафик "Твиттера" будет просто огромным. Но вот на фоне остальных социальных сетей эти несколько миллионов кажутся

жалкими копейками. Действительно, небольшие блоги, который является сутью "Твиттера", нравится далеко не всем.

"Мой мир" это российская социальная сеть выполнена в самом традиционном ключе, являясь местом для поиска новых друзей, чтобы затем находится в курсе событий из их жизни, просматривать их свежие фотографии и видеоролики. Данный сервис позволяет вести блоги, общаться посредством комментариев, писать в гостевую книгу и так далее. Год назад "Мой мир" несколько изменил свой формат, начав позиционировать себя как социальная сеть для общения с пользователями с похожими взглядами и вкусами.

Все вышеперечисленные сервисы не могут выполнить поставленной задачи, по поиску возможных исполнителей потребностей пользователей. Используя поисковые сервисы пользователь можно самостоятельно проводить поиск по каждой из потребностей, и основываясь на результатах поисковых систем, выбирать место для реализации. Однако, выбранное место не имеет возможности выполнить потребность бесплатно или за другую услугу взамен, и может не совсем точно найти подходящую тему, если потребность не является материальной. В том числе и социальные сети, где на основной ленте новостей можно добавить свои потребности, и возможно кто то из друзей увидит и может помочь сам или посоветовать кого то другого. Процесс описания, поиска и исполнения потребностей в социальных сетях не удобен, не безопасен и не точен.

В интернете существуют аналоги разрабатываемой ИС, примером может быть сайт "Drugdrugu.ru" это место, где можно рассказать о своем желании или наоборот, предложить сделать что-то для других, что позволяет назвать "Drugdrugu.ru " сайтом исполнения желаний (потребностей). Данный сайт устроен таким образом, что в интерфейсе на главной странице есть два столбца, слева для списка всех желаний, справа для предложений что-то сделать. При старте подобных сервисов можно ожидать, что многие напишут простые и понятные желания, например "хочу машину, хочу большой дом", а среди предложений можно было бы увидеть множество платных услуг. Но как

оказалось сервис был востребован, и пользователи писали много оригинальных желаний, среди которых были и простые, например "Хочу научиться играть на скрипке, Хочу найти людей из снов". Среди предложений что-то сделать тоже было много интересных и так как никто не устанавливал условия о бесплатности, то часть предложений были коммерческие, а часть были готовы сделать что-то хорошее бесплатно, например "Придумаю как поздравить друга, Могу выслушать если есть проблемы". Пользователи положительно отзывались о сервисе, но со временем количество коммерческих предложений стало слишком большим и найти среди всех их нужное предложение было очень сложно, в связи с чем данный сервис закрылся в середине 2015 года.

Многие пользователи похожих ИС положительно отозвались об идеи, и рады поучаствовать, в тестировании и развитии подобных сервисов.

2. Описание проекта

Общая идея работы ИС является получение, обработка, хранение, выдача информации пользователям, которые получают данные, принимают решения, вносят новую информацию.

Люди имеют множество желаний и ищут возможность их реализовать, в короткие сроки и с минимальным затратам сил. Часто оказывается, что у кого-то из контактов или второго круга контактов данного человека, есть возможность для реализации определённых желаний. Но человек может не знать, что один из контактов может помочь. Многим известна поговорка "человек знаком со всем миром через несколько рукопожатий", что означает пользователи ИС могут быть взаимосвязаны и можно создать ИС, которая будет объединять пользователей для решения общих целей, для пары или группы пользователей.

Задачей разрабатываемой ИС является описание и сбор потребностей и возможностей пользователей, с разными уровнями доступа контактов, поиск среди первого круга или следующих уровней кругов контактов, тех кто является возможным исполнителем или помощником в выполнении конкретного определённых потребностей пользователей, информировать контакты пользователей о новых потребностях, с целью вовлечь пользователей в процесс улучшения поиска и дать возможность в ручном режиме указывать пути до исполнителя.

Система построена на исследованной и выявленной психологами модели, которая нацелена на поддержание комфортных, экологических взаимоотношений между пользователями по модели ЛПСВ, т.е для реализующего потребность человека должно удовлетворяться три критерия:

субъективно выгодно - действие должно быть выгодно человеку;

просто - элементарно в исполнении;

легко - не должно вызвать у человека психологической сложности;

Система подразумевает учёт баланса добрых дел, так как одному участнику процесса взаимоотношений не уютно регулярно безответно

выполнять добрые дела, соответственно система ведёт учёт и предоставляет пользователям информацию о том где, кому и сколько добрых дел было сделано, что обеспечивает комфорт взаимоотношений. Это реализуется с помощью контроля, что на 3-4 добрых дела с одной стороны, другая сторона должна сделать как минимум 1 доброе дело, исполнив потребность участника.

Система обязует пользователей отчитываться о результатах исполнения потребностей, таким образом система ведёт рейтинг пользователя, что бы выявить насколько хорошо было выполнена потребность пользователя, и для предоставления возможности обратной связи, тем людям, которые выполняют потребности пользователя. Другие пользователи увидев рейтинг могут судить о компетентности оппонента, и принимать решение передавать или возлагать выполнение потребностей на данного пользователя.

3. Подбор средств и инструментов для реализации проекта

3.1. Рассмотрены классические СУБД

Классическим СУБД можно отнести реляционные БД. В реляционных базах данных данные собраны в таблицы, которые в свою очередь состоят из столбцов и строк, на пересечении которых расположены ячейки. Запросы к таким базам данных возвращает данные в виде таблицы. Данные в одних таблицах, как правило, связаны с данными других таблиц, откуда и произошло название "реляционные".

С использованием реляционных СУБД можно реализовать разрабатываемый проект, и использовать таблицы для хранения классов в виде записей в таблицах, а связи между пользователями сделать как класс и все их записать в одну таблицу.

Среди самых популярных клиент-серверных СУБД есть представители: Microsoft SQL Server от компании Microsoft, Oracle RDBMS от одноимённой компании Oracle, MySQL разработанная в MySQL AB, PostgreSQL и многие другие. Многие клиент-серверные СУБД разрабатывались с целью использоваться преимущества централизованного управления, поддержки высокой надёжности, доступности и безопасности. Большинство из них используются сейчас в крупных коммерческих проектах, в социальных сетях, в некоторых поисковых системах для табличного хранения данных с быстрой возможностью изъятия по ключевым словам.

Рассматривая достоинство и недостатки СУБД, можно увидеть, что большинство из них, сами уже являются довольно крупными проектами, имеющие хорошие показатели по производительности и безопасности, возможности резервного копирования и восстановления после сбоя, позволяющие сохранять системные данные для анализа производительности и поиска узких мест и ошибок. В таблице 3.1 представлены наиболее популярные СУБД и их достоинства и недостатки.

Таблица 3.1 - Достоинства и недостатки классических СУБД

Сервер	Достоинства	Недостатки
Microsoft SQL Server	Развитый продукт, производительный, недорогой. Есть возможность делать резервные копии, логирование, кластеризацию. Простота в изучении и большое количество пользователей умеющих работать	Только для платформы компании Microsoft. Сложность в масштабировании.
Oracle Database	Большое множество расширяемых дополнений. Имеет широкие возможности по кластеризации. Механизм пакетной обработки запросов с подтверждением. Кроссплатформенность.	Сложность в поддержке, Отсутствие специалистов, Стоимость
MySQL	Простой, используется в небольших проектах. Бесплатный.	Мало дополнительных возможностей.

3.2. Обоснование нерациональности использования классических СУБД

В разрабатываемом проекте действующими лицами являются пользователи, которые будут создают связи между собой. Количество связей может быть довольно большим, а так как в проекте используется поиск между желаними и возможностями пользователей, то необходимо обойти все контакты рассматриваемого пользователя, контакты тех кто у данного пользователя в контактах, и далее обойти контакты всех кто вероятно попадёт в цепочку. Получается, что количество запросов к связям будет огромным, что приведёт к падению скорости. Ещё надо учитывать, что данные связей могут меняться и одновременный доступ к записи и считыванию может привести к блокированию потока на более длительное время.

Если взять реляционную СУБД Microsoft SQL Server и провести запрос одной записи из таблицы с более 100,000 записей, ответ будет выдан в среднем за 25 миллисекунд. Для реализации задач в разрабатываемом проекте нужно пройти по всем связям пользователей и чем больше у пользователей связей, тем медленнее будет производиться поиск. При среднем количестве связей более 100 контактов у одного пользователя и более 100 желаний и возможностей у каждого, поиск с использованием реляционных СУБД будет очень затруднительным, по скорости каждый поиск может занять более 1сек для поиска цепочек каждого желания.

3.3. Обоснование выбора графовой БД

Графовая база данных это разновидность баз данных с реализацией сетевой модели в виде графа и его связей. В разрабатываемом проекте используется СУБД Neo4J это графовая база данных, разработана в Neo Technologies в 2003 году, имеет открытый исходный код, реализованный на языке Java. Neo4j имеет свой собственный формат хранилища узлов (node) и связей (relationship). Оригинальное представление графа, в отличие от моделирования хранилища на реляционной базе данных, позволяет применять дополнительную оптимизацию в случае данных со сложной структурой. [4]

Преимущество использования графовых СУБД:

1. Возможность быстрого подключения к новому проекту
2. Простота добавления осмысленных связей
3. Высокая скорость
4. Низкое потребление ресурсов ПК

База данных NEO4J была выбрана, потому что:

1. является наиболее используемой из графовых СУБД
2. имеет достаточно хорошую скорость
3. может содержать неограниченное количество объектов и связей
4. лёгкая в соединении и использовании
5. простой язык запросов
6. имеется визуальный просмотр созданных объектов и связей.

Визуальный просмотр объектов, связей и их атрибутов можно осуществлять с помощью веб интерфейса, который имеет доступ к локальному хосту с портом 7474 и передаёт данные с помощью API. В последствии вызов API инициализирует метод для выполнения команды пользователя и может выводить результаты запросов в браузер. Пример показан на рисунке 1.

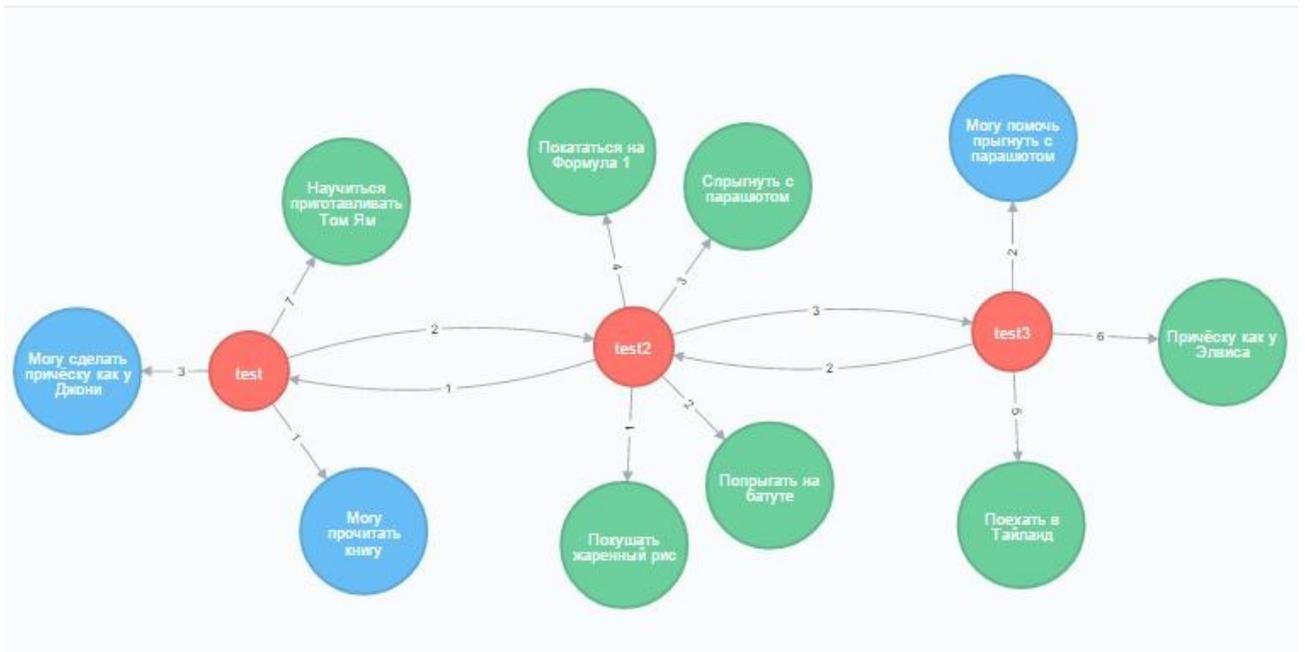


Рисунок 1 - Пример отображения данных из Neo4J API в окне браузера.

Примеры запросов:

1) `match (n:User {ID:2}) return n.name` // возвращает имя пользователя с ID = 2.

2) `match (n:User)-[r:`Know`]-() where n.name = 'Misha' return count(r)`

// возвращает количество цепочек с именем начального пользователя Misha

Графовая СУБД, позволяет иметь динамические классы с разными атрибутами, но имеющими часть общих атрибутов. Если данных этого атрибута нету в записи в БД, то устанавливается значение по умолчанию, для строковых данных это NULL, для числовых ноль.

Поиск с использованием связей проходит быстрее, при сотне пользователей с 10 контактами, 10 желаниями и 10 возможностями поиск всех возможных цепочек занимает менее 20 секунд, и последующая обработка данных на языке C# может занять ещё небольшое время, в зависимости от количества цепочек и данных в них.

Например для запроса с поиском совпадающего текста желания и возможности, где участвует пользователь с ID=1, скорость поиска составила 175мс. У каждого из пользователей в цепи, есть 10 связей с другими пользователями.

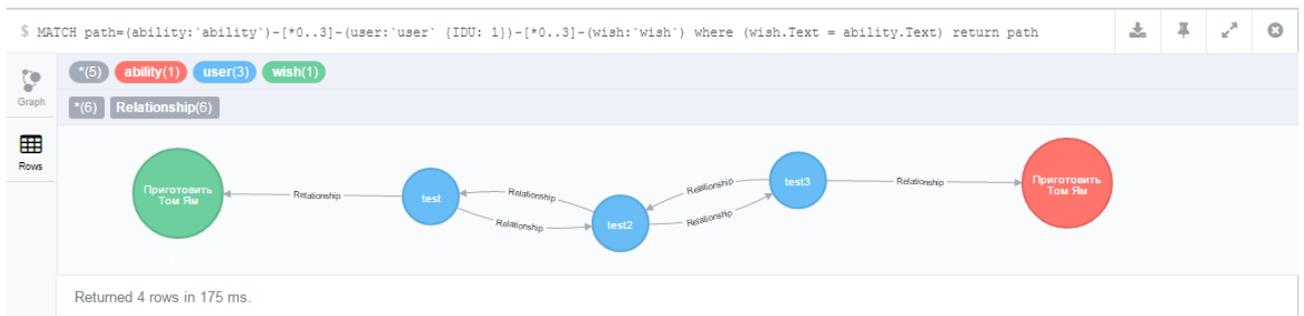


Рисунок 2 - Запрос с показом времени поиска.

3.4. Обоснование выбора платформы и инструмента разработки

3.4.1 Среда .NET Framework

.NET Framework – программная среда для разработки принципиально новых приложений и веб-служб. Разработчиками были заложены в платформу следующие функциональные возможности:

- Согласованная объективно-ориентированная среда программирования. Она подразумевает возможность сохранения и внедрения объективного кода, как для локального, так и удаленного выполнения;
- Минимализация возможности возникновения конфликтов при развертывании ПО. Это также касается управления различными версиями;
- Возможность безопасного выполнения кода, даже если его источник – малоизвестный или производитель, который не является проверенным;
- Обеспечение выполнения кода, исключающее возможности возникновения проблемных ситуаций в части выполнения сценариев, интерпретируемого кода и производительности;
- Единые принципы работы для разработки различных видов ПО, таких как Windows и веб-приложения;
- Создание среды, которая обеспечивает интеграцию внутреннего кода NET Framework с другими.

Основа платформы NET Framework – одноименная библиотека классов и среда общезыкового выполнения CLR. Ее можно рассматривать, как агента, который осуществляет управление кодом во время его выполнения и одновременно предоставляет, такие сервисы как:

- Управление потоками;
- Управление памятью;
- Удаленное взаимодействие.

На эти процессы накладывается строгая типизация, а также другие способы проверки точности кода. Это необходимо для обеспечения безопасности. Основной функцией среды выполнения является управление кодом.

Код, обращаемый к среде выполнения является управляемым. Соответственно код, который не обращается к среде выполнения, называется неуправляемым. Библиотека классов – комплексная сборка, допускающая многократное использования типов, применяемых для создания приложений, которые запускаются через командную строку с графическим интерфейсом GUI. Также эта сборка ориентирована на разработку приложений с использованием последних технологических наработок, таких как службы XML и Web Forms.

Программная среда .NET Framework размещается посредством неуправляемых компонентов, которые загружают среду CLR в собственные процессы и активируют процессы выполнения управляемого кода. Так создается программная среда, которая использует возможности управляемого и неуправляемого выполнения. Это означает, что платформа .NET Framework предоставляет несколько базовых сред выполнения, поддерживает создание ПО независимыми разработчиками, производителями.

Пример – ASP.NET размещает среду выполнения и обеспечивает масштабируемую среду для управляемого кода на стороне сервера. ASP.NET осуществляет работу со средой выполнения для обеспечения выполнения приложений служб XML и ASP.NET.

Пример неуправляемого приложения – Internet Explorer. Оно размещает среду выполнения, которая имеет вид расширений типов MIME. Такая концепция размещения среды позволяет интегрировать управляемые компоненты или инструменты управления Windows Forms в разметку HTML.

Это позволяет выполнять мобильный управляемый код, а также использовать его плюсы в рамках выполнения условий частичного доверия и изолированным хранением файлов.

Диаграмма, расположенная ниже показывает взаимосвязь среды CLR и библиотеки классов с приложениями и системой в целом. Там также отображен алгоритм работы управляемого кода в условиях расширенной архитектуры.



Рисунок 3 - .NET Framework внутри системы

Далее будут рассмотрены основные возможности среды разработки .NET Framework.

3.4.2 Особенности CLR

CLR – среда, выполняющая управление памятью, отвечающая за выполнение кода, контроль его безопасности, а также компиляцию. Среда CLR осуществляет управление системными службами. Эти инструменты – внутренние средства, ориентированные на управляемый код, который выполняется в среде CLR.

Для обеспечения высокого уровня безопасности управляемые компоненты обладают разными степенями доверия. Они зависят от ряда условий и факторов, таких как происхождение компонентов:

- Всемирная паутина – интернет;
- Локальная сеть, работающая в рамках одного предприятия;
- Отдельный ПК.

Это позволяет регулировать степень доступа управляемого компонента к файлам, реестру и другим основополагающим функциям, даже если они используются одним активным приложением.

Управление доступом для кода обеспечивается средой выполнения. К примеру, можно поручить, или правильней будет сказать, доверить приложению, которое исполняется и является внедренным в веб-страницу исполнение каких-либо процессов (воспроизведение видео, звука, анимации) без получения доступа к личной информации, данным системы или сети. Из всего вышесказанного следует, что возможности безопасности CLR предоставляют ПО, развернутому в интернете, множество функциональных возможностей.

CLR, помимо прочего, гарантирует надежность кода путем реализации широких возможностей в части проверки и типизации кода. Эти возможности объединены в систему, которая называется CTS или система общих типов. Она реализует возможность самоописания управляемого кода. Компиляторы Microsoft генерируют управляемый код, который соответствует требованиям системе CTS, что означает – код может воспринимать иные экземпляры и управляемые типы, обеспечивая их строгую типизацию и правильность.

В функциональные возможности CLR (управляемой среды) входит предотвращение ошибок, которые могут возникать в ПО. примером такой деятельности является автоматическое размещение объектов, а также ссылок на них. При этом они прекращают работать, освобождаются, если объект не используется. Автоматизированный контроль памяти решает 2 наиболее распространенные проблемы, возникающие у приложений:

- Утекающая память;
- Неактуальные ссылки на память.

CLR предоставляет практические преимущества разработчикам приложений. Для их создания может использоваться удобный способ – язык, движок без потери преимуществ, которые предлагает среда исполнения, включая классы и компоненты. Которые были созданы сторонними мастерами с использованием других возможностей программирования. Это доступно любому производителю компиляторов, обращающихся к среде выполнения.

Языковые компиляторы, ориентированные на среду разработки NET Framework, делают его средства доступными для существующего кода, сгенерированного на соответствующих языках программирования. Это в значительной степени упрощает процедуру переноса существующих приложений.

Несмотря на то, что CLR создавалась с ориентацией на перспективные разработки в части ПО, система поддерживает современные, а также ретро программы. Взаимодействие управляемого и неуправляемого кода дает возможность применять компоненты COM и библиотеки DLL.

Разработка среды выполнения осуществлялась с целью увеличения уровня производительности. При этом общезыковая среда выполнения предоставляет множество стандартных служб времени выполнения, которые коду совершенно не интересны. Компиляция обеспечивает выполнение управляемого кода через машинный язык компьютера. Одновременно с этим диспетчер памяти блокирует фрагментацию памяти, увеличивая объем памяти, которая ориентирована на увеличение производительности.

Еще одним преимуществом CLR является возможность размещения в серверных приложениях Microsoft SQL Server, службы IIS (Internet Information Services), которые являются высокопроизводительными. Эта концептуальная особенность дает возможность создавать индивидуальные программные алгоритмы, используя управляемый код, а также высокой производительностью

передовых производственных серверов. Они позволяют размещать среду выполнения CLR.

3.4.3 .NET Framework – Библиотека классов

Библиотека классов платформы .NET Framework – сборка типов, которые активно интегрируются в среду выполнения и наоборот. Эта сборка объектно-ориентирована на предоставление типов, которые дают возможность управляемому коду наследовать функции. Такая возможность значительно упрощает работу с типами и сокращает время необходимое для освоения новых возможностей среды NET Framework. Библиотека классов позволяет интегрировать компоненты сторонних разработчиков.

Сборка классов .NET Framework реализует набор интерфейсов, они могут использоваться для создания пользовательских классов коллекций. При этом они без проблем будут объединяться с библиотекой среды .NET Framework.

Объектно-ориентированные библиотеки классов, типов .NET Framework дают возможность решать типовые задачи программирования:

- Работать со строками;
- Собирать данные;
- Подключаться к базам данных;
- Получать доступ к файлам.

Дополнением к таким стандартным задачам в библиотеке классов являются типы, поддерживающие множество специализированных сценариев разработки. Это позволяет применять платформу .NET Framework для разработки таких типов приложений и служб:

- Консольные приложения.
- Приложения с графическим интерфейсом пользователя Windows (Windows Forms).
- Приложения Windows Presentation Foundation (WPF).
- Приложения ASP.NET.
- Службы Windows.

- Ориентированные на сервисное направление приложения с помощью Windows Communication Foundation (WCF).
- Приложения, которые поддерживают бизнес-процессы Windows Workflow Foundation (WF).

3.4.4 Инструмент Visual Studio

Microsoft Visual Studio - линия продуктов гиганта Microsoft, содержащих интегрированную среду разработки ПО и множество иных средств, инструментов для осуществления разработки. Такие функциональные особенности позволяют создавать консольные приложения и приложения с графическим интерфейсом, включая поддержку технологии Windows Forms, а также: [6]

- Веб-сайты;
- Веб-приложения;
- Веб-службы.

Разработка может осуществляться в родном и управляемом коде для всех платформ, поддерживаемых: [6]

- Microsoft Windows;
- Windows Mobile;
- Windows CE;
- NET Framework;
- Xbox;
- Windows Phone;
- NET Compact Framework;
- Microsoft Silverlight.

Visual Studio содержит редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как настройщик уровня исходного кода, так и в качестве отладчика машинного уровня. Другие встраиваемые инструменты включают в себя: [6]

- Редактор форм для упрощения создания графического интерфейса приложения;
- Веб-редактор, дизайнер классов и дизайнер схемы базы данных.

Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне. Сюда входит поддержка систем контроля версий исходного кода (Subversion и Visual SourceSafe), добавление новых наборов инструментов. Например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования или инструментов для прочих аспектов процесса разработки программного обеспечения.

Примеры – клиент Team Explorer для работы с Team Foundation Server.

Основными причинами выбора Visual Studio в этом проекте являются:

1. Интегрированный язык C#;
2. Упрощенная реализация общих задач;
3. Технология IntelliSense, позволяющая ускорить кодирование путем сокращения текстового объема, который создается путем набора на клавиатуре и сокращения необходимости обращаться к внешним документам;
4. Расширение Resharper, предусматривающее синтаксический анализ кода, дополнительные средства автоматического заполнения и подсветки синтаксиса;
5. Простота подключения библиотеки для работы с графовой БД к проекту написанному на языке C#.

3.4.5 Язык программирования C# – отличия и преимущества

Основным функциональным преимуществом языка программирования C# является его ориентированность на платформу Microsoft .NET. Разработчики C# стремились создать для программистов свободный, естественный доступ ко всем функциональным возможностям платформы .NET. Такой подход можно считать вынужденным, так как .NET обладала большими возможностями, чем любой из существовавших в то время языков программирования.

Разработчики C# стремились скрыть от программистов большую часть малозначительных технических особенностей, такие как операции распаковки и

упаковки типов, сбор мусора, инициализация переменных. Это позволило при написании программы с использованием С# концентрировать внимание на решении содержательной части задачи. Для чего была предпринята попытка учесть опыт реализации VisualBasic из-за возможностей успешного сокрытия деталей реализации. Одновременно с этим VisualBasic малоэффективен при разработке масштабных промышленных систем.

Разработчики С# позиционируют язык, который совмещает мощьность С++ и простоту VisualBasic. Еще одним преимуществом нового языка программирования являются возможности расширения. При работе с С# отсутствует необходимость заботиться об обратной совместимости, которая в значительной степени усложняет исправление проблем, а также интегрирование новых возможностей в язык программирования. Более предметное описание сложностей, возникающих при расширении старого языка программирования, можно найти в книге Б. Страуструпа "Дизайн и эволюция языка С++", М.: ДМК, 2000.

Учитывая все вышеизложенное, можно сделать вывод, что С# является принципиально новым языком программирования. Он заточен под разработки для среды .NET, предоставляет возможности для оперативного прототипирования приложений и позволяет крупные проекты.

3.4.6 Технология ASP .NET MVC 4.0

Для разработки программного продукта была выбрана технология ASP.NET MVC 4 и язык С#, являющийся самым популярным языком для написания бизнес логики на платформе.NET, основные преимущества технологии ASP.NET MVC 4:

1. Расширяемость

MVC Framework также построен как ряд независимых компонентов, удовлетворяющих .NET интерфейс или построенных на абстрактном базовом классе, что позволяет легко заменять одни компоненты другими. ASP.NET MVC дизайнеры построили его таким образом, чтобы дать разработчикам три варианта выбора для каждого компонента MVC Framework:

- I. использовать реализацию по умолчанию компонента в его нынешнем виде (чего должно быть достаточно для большинства приложений);
- II. вывести подкласс реализации по умолчанию для настройки ее поведения;
- III. заменить компонент полностью при помощи новой реализации интерфейса или абстрактного базового класса.

2. Жесткий контроль над HTML и HTTP

ASP.NET MVC признает важность получения чистой, соответствующей стандартам разметки. Его встроенные методы HTML помощника предоставляют соответствующие стандартам выходные данные.

Страницы, сгенерированные ASP.NET MVC, не содержат никаких данных ViewState, поэтому они могут быть в сотни килобайт меньше, чем обычные страницы, созданные при помощи ASP.NET WebForms. Несмотря на современную широкополосную связь и быстрые подключения, эта экономия пропускной способности до сих пор чрезвычайно притягательна для конечных пользователей.

3. Тестируемость

Архитектура MVC дает разработчикам отличную возможность создавать веб-приложение таким, чтобы его можно было легко сопровождать и тестировать. Тестируемость – это не только вопрос модульного тестирования. ASP.NET MVC приложения также хорошо работают с инструментами автоматического тестирования. Разработчик может написать тестовые скрипты, которые имитируют взаимодействие с пользователем, без необходимости гадать, какие структуры HTML элементов, CSS классы или ID будет генерировать фреймворк, поэтому при разработке не придется беспокоиться о структуре, если она вдруг неожиданно изменится.

4. Мощная система маршрутизации (роутинга)

Стиль ссылок изменился, поскольку технология веб приложений улучшилась. Такие ссылки, как:

`/App_v3/User/Index.aspx?action=show%20prop&prop_id=82742`

можно встретить довольно редко.

Теперь они заменены более простым и чистым форматом:

`/for-rent/Tomsk/ulica-Uchebnaya-40`

Есть несколько веских причин для заботы о структуре URL.

Во-первых, поисковые системы придают значительный вес ключевым словам, находящимся в URL. Поиск "аренда в Томске" (rent in Tomsk) имеет гораздо больше шансов с простым URL.

Во-вторых, многим пользователям Интернета теперь хватит навыков и знаний, чтобы понять URL, и оценить возможности навигации, набрав его в адресной строке своего браузера.

В-третьих, когда кто-то понимает структуру URL, он, скорее всего, будет ссылаться именно на него, поделится этой ссылкой с другом или даже продиктует ее вслух по телефону.

В-четвертых, такая ссылка не предоставляет технические подробности, папки, имена файлов и структуру приложения на весь общественный Интернет, так что вы можете изменить внутреннюю реализацию, не нарушая ссылки.

Теперь можно контролировать схему ссылок и ее связь и отношение к приложению, то есть разработчик свободен в создании шаблона URL-адресов, которые являются значимыми и полезными для пользователей, без необходимости соответствовать предопределенному шаблону.

5. Открытый исходный код

В отличие от предыдущих платформ веб-разработки от Microsoft, можно загрузить исходный код для ASP.NET MVC и даже изменить и скомпилировать собственную версию. Это имеет неоценимое значение, когда отладка касается системы компонентов, и необходимо зайти в код (и даже прочитать комментарии программистов-создателей). Это также полезно при создании

коммерческих программ для того, чтобы оценить какие возможности существуют для дальнейшего развития или как действительно работают встроенные компоненты.

Исходя из вышеописанных критериев, был сделан выбор в пользу разработки серверной части с использованием платформы .Net, языка C#, инструмента разработки Visual Studio и технологии ASP.NET MVC.

4. Разработка интерфейса

4.1. Инструменты, используемые для улучшения интерфейса

В реализуемом проекте используется нескольких паттернов программирования, в частности MVC и Dependency injection (слабой зависимости кода), а так же будут реализованы подходы быстрого доступа к данным без постоянной их загрузки из базы данных.

Паттерн MVC - (модель-представление-контроллер) довольно известный паттерн используемый для разделения между уровнями представления, контроллера и модели данных. Контроллер используя модели данных передаёт данные на представление, там самым модели образуют объединение в использовании данных, а представление хранит отображаемую часть.

Синтаксис ASP.NET Razor — это простая модель программирования, позволяющая внедрять серверный код в веб-страницу. Страница также может содержать разметку HTML, данные CSS и клиентские сценарии (JavaScript и jQuery). Синтаксис Razor основан на платформе ASP.NET, которая входит в состав .NET Framework и специально предназначена для создания веб-приложений. Синтаксис Razor предоставляет все возможности ASP.NET, но при этом он проще в изучении для новичков. Если вы эксперт, ваша работа с ним станет продуктивнее. Хотя это простой синтаксис, его связь с ASP.NET означает, что по мере усложнения ваших веб-приложений вы всегда можете воспользоваться возможностями более крупной платформы. [5]

При выполнении страницы с кодом Razor сервер выполняет код перед отправкой страницы в браузер. Выполнение кода на сервере позволяет выполнять более сложные задачи, чем допускает клиент, например работать с серверными базами данных. Более того, серверный код позволяет динамически генерировать клиентский контент. Он может создавать разметку HTML или другой контент, чтобы отправить его в браузер вместе со статическим HTML на странице. [5]

Паттерн Dependency injection, DI это абстрактный принцип, как набор рекомендаций для написания слабо связанного кода. Суть, которого в том, что

каждый компонент системы должен быть как можно более изолированным от других, не полагаясь в своей работе на детали конкретной реализации других компонентов. Так как проект разделён на две основные части, их было решено сделать максимально изолированными, так контроллеры подключаются с использованием вызова и хранения в памяти подключения, к веб сервисам для работы с графовой БД.

Bootstrap это популярный свободный графический набор инструментов для сайтов, помогает сделать веб страницы более гибкими к разрешению экрана. В разрабатываемом проекте используются следующие компоненты:

Координационная сетка для удобного расположения веб элементов на странице и гибкости блоков при изменении размеров окна.

Кнопки, информационные и навигационные панели, постраничный просмотр в таблицах с использованием DataTables для улучшенного визуального восприятия. DataTables дополнительная библиотека написанная на Javascript, для быстрого вывода таблиц, с расширенными возможностями.

4.2. Проработка механизмов взаимодействия пользователя с системой

Для удобства разработки используется паттерн модель-представление-контроллер, который при вызове страницы, вызывается метод контроллера с передачей данных, который был определён через роутинг входящих данных. Далее контроллер реализует логику в вызванном методе, может вызвать подключенный интерфейс из модуля работы с БД. После получения данных от вызванного в интерфейсе метода, обрабатывает данные собирая модели, и предаёт полученную модель для отображения в представлении. Представление имеет HTML код с вставлением ASP и C# кода в страницу для распределения данных и отображения. Клиенту передаётся уже готовая HTML страница.

4.2.1 Информация о пользователе

При регистрации пользователь может внести информацию о себе, в последствии видеть на своей основной странице или отредактировать на странице редактирования профиля. Пользователь может загрузить собственную

фотографию для того что бы другие пользователи смогли найти его в системе. Так же он может внести общую информацию о имени и фамилии, возрасте, местоположении и другие данные. Показано на рисунке 4.

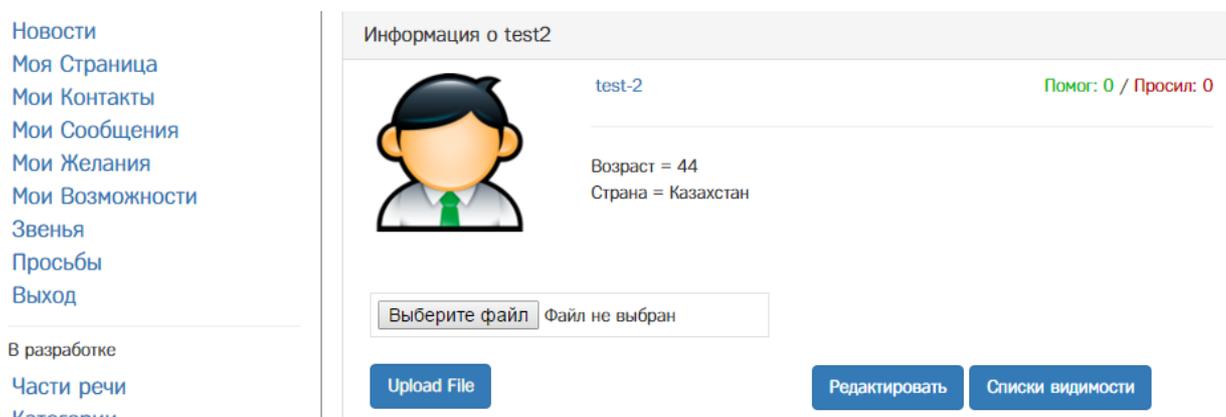


Рисунок 4 - Информация о пользователе

4.2.2 Список контактов первого круга

Место для отображения и редактирования связей с другими пользователями. Кнопка "Поиск" для перехода на страницу поиска пользователей в системе, для того что бы найти новые контакты, указав в строке поиска часть имени, фамилии или логина другого пользователя. Увидев нужного пользователя в списке можно выслать ему приглашение, установив сложность и тип отношений. После подтверждения приглашённым пользователем появляются связи в БД. В дополнение пользователям доступен переход в чат. Показано на рисунке 5.

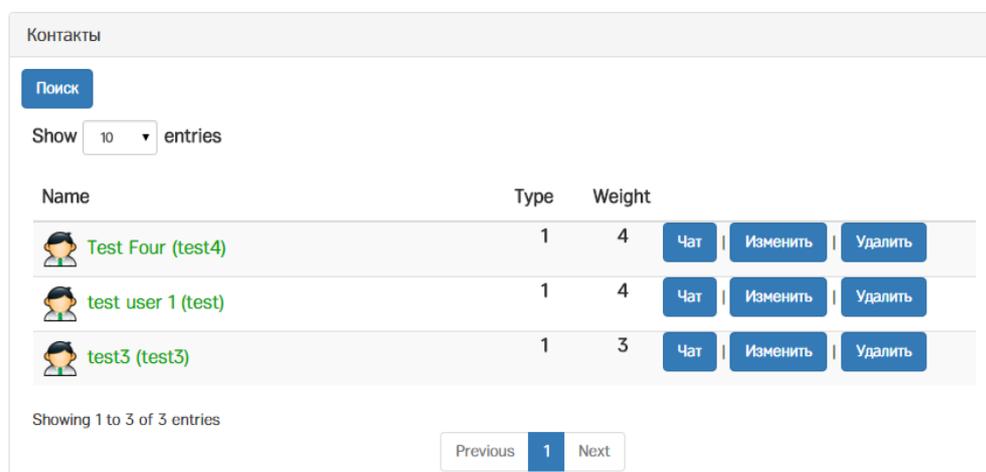


Рисунок 5 - Список контактов первого круга

4.2.3 Список желаний, страницы для редактирования

После создания профиля пользователю доступен раздел для добавления потребностей (желаний) и просмотра списка. В связи с тем что у пользователя может быть множество желаний на странице со списком реализован постраничный просмотр с шагом в 10 желаний. Со списка пользователь может войти на страницу редактирования или удалить желание. На странице редактирования пользователь может ввести текст желания, определить его категорию, а так же выбрать группу, для которой доступен просмотр этого желания.

Группирование необходимо для того, что бы часть желаний или возможностей пользователь не показывал всем, а только определённом кругу лиц, которых он может добавить в группу. Показано на рисунке 6.

Моя Страница
Мои Контакты
Мои Сообщения
Мои Желания
Мои Возможности
Начать поиск
Выход

В разработке
Части речи
Категории

Изменить желание

Изменить желание

Описание

Покушать мороженное с арбузом

Категория

Кушать

Выберите группу видимости:

Public

Изменить желание

Рисунок 6 - Страница для редактирования желания

4.2.4 Список возможностей.

Вместе с разделом желаний есть раздел возможностей, который по функционалу очень схож с разделом желаний, но логически отличается от него противоположностью. Так же для каждой возможности имеется страницы создания, просмотра, редактирования и удаления. В объект возможность входит текст, категория, группа видимости и сложность выполнения.

Сложность выполнения это величина, которую устанавливает пользователь, в связи с тем, что некоторые возможности пользователь может выполнить, но с трудом и система будет учитывать сложность и искать сначала

пользователей, которым легче выполнить желание, а затем тех кому сложнее в порядке убывания числа сложности. Число сложности может колебаться от 1 до 5, чем сложнее выполнить тем число больше.

4.2.5 Пользовательский чат

Место обмена пользовательскими сообщениями, в котором можно быстро найти все сообщения относящиеся к определённому желанию, для более удобного перенаправления или добавления в конференцию пользователе. В связи с тем что пользователям нужно иметь быстрый доступ к сообщениям а так же поточное обновление данных, на странице было реализовано постоянная подгрузка сообщений с перерывом в 10 секунд, при разрыве связи время перерыва увеличивается до 30 секунд до его восстановления. После запроса новых сообщений у пользователя обновляется частичное представление места где отображаются все сообщения. Так как метод по отдаче сообщений для оппонентов может вызываться довольно часто, в класс содержащий методы была добавлена статическая переменная со словарём быстрого доступа ConcurrentDictionary. Визуальное представление чата показано на рисунке 7.

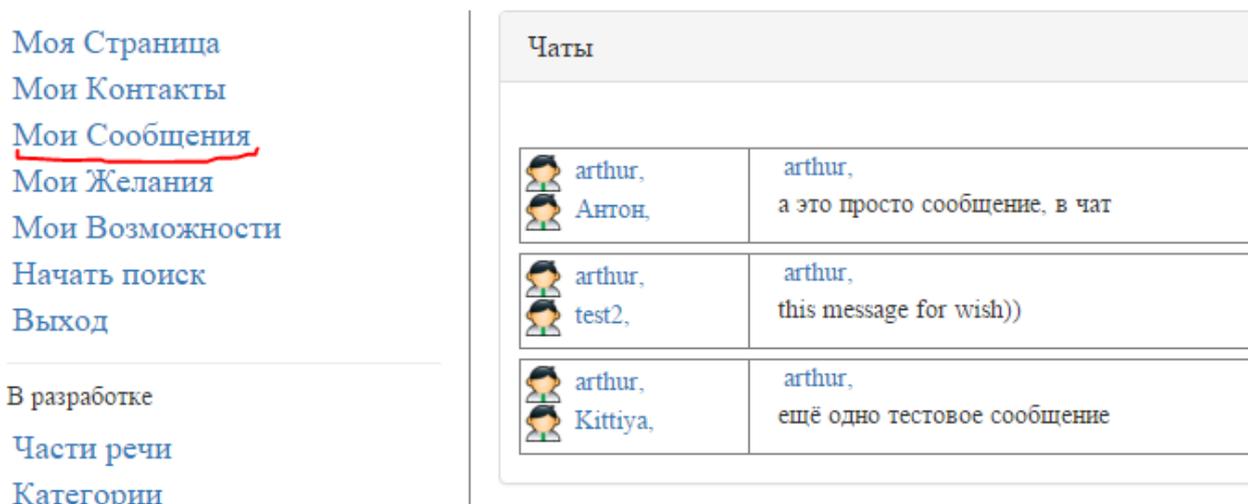


Рисунок 7 - Пользовательский чат

4.2.6 Раздел новостей о потребностях друзей

Область для отображения цепочек, в которых участвует данный пользователь и потребности первого круга знакомых, которые он может

передать одному из своих знакомых, если алгоритмы поиска не выявили совпадений для создания цепи пользователей. Показано на рисунке 8.

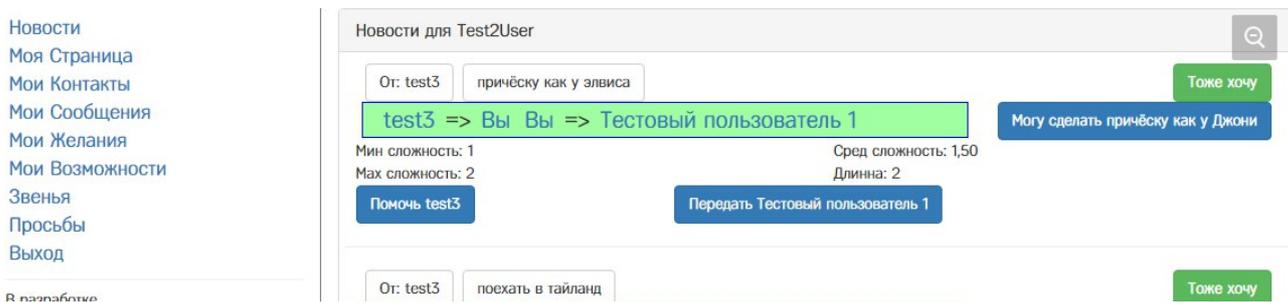


Рисунок 8 Раздел новостей о потребностях друзей

4.2.7 Раздел переданных просьб

Текущий раздел предназначен для просмотра списка просьб, в которых данный пользователь участвует или является исполнителем. Пользователь может нажать кнопку помочь и перейти в чат для общения по данной потребности, отказаться от выполнения, оценить взаимодействие с пользователем или завершить после выполнения. Показано на рисунке 9.

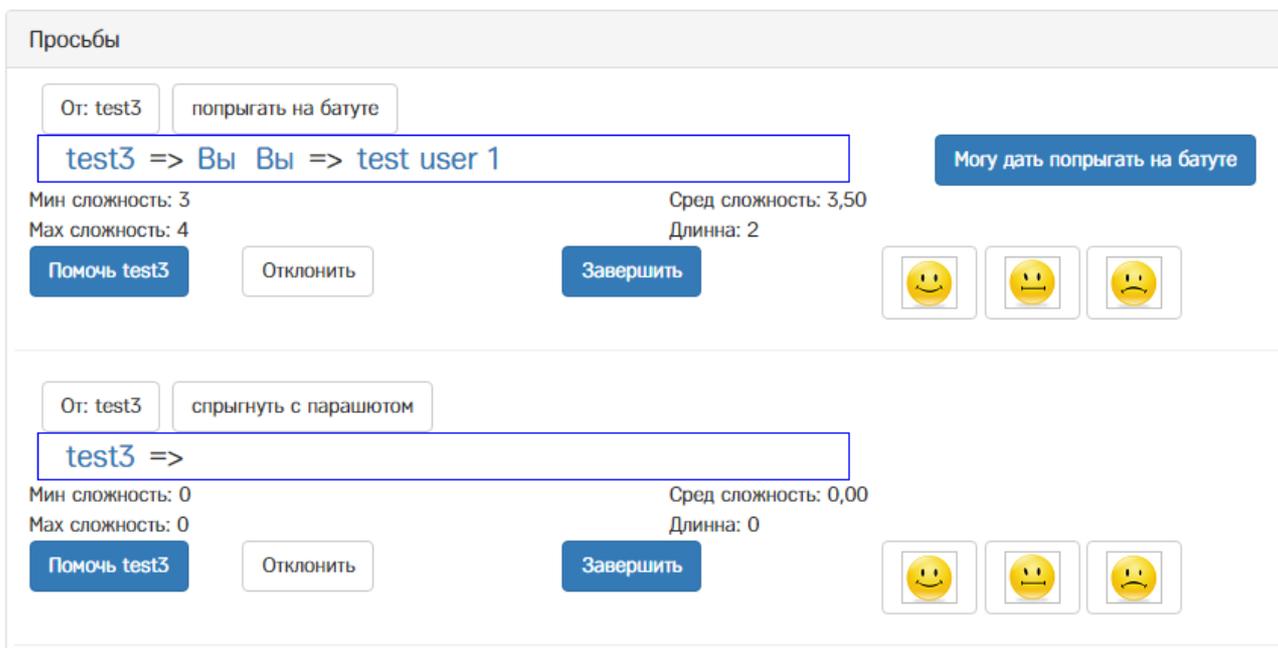


Рисунок 9 - Список просьб для пользователя

5. Разработка бизнес-логики

Реализация бизнес логики будет сделана в контроллерах, вызов которых происходит через роутинг и запускается клиентом. Контроллер используя модели будет передавать данные для представления пользователю.

5.1. Модели

В ходе проектирования было выявлено что в системе будут три действующих лица, показанные на рисунке 10.

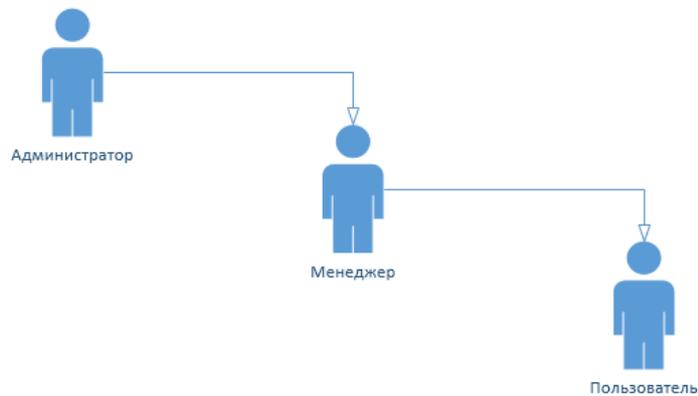


Рисунок 10 - Диаграмма действующих лиц

Администратор имеет полный доступ к базе данных и может изменять систему поиска удовлетворения желаний. Так же назначает Менеджеров, которые будут модерировать данные и следить за правильностью работы поисковой системы.

Менеджер может модерировать данные и следить за правильностью работы поисковой системы исполнения желаний, находить нежелательный контент, проводить предупредительные и блокирующие нежелательных пользователей работы, составлять отчёты.

Пользователю будут доступны следующие варианты действий: Регистрация, Авторизация, управление потребностями (желаниями), управление возможностями, управление списком знакомых первого круга и их группирование, переписка в чате, просмотр списка потребностей круга знакомых или связанных с ними пользователей и участие в реализации желаний.

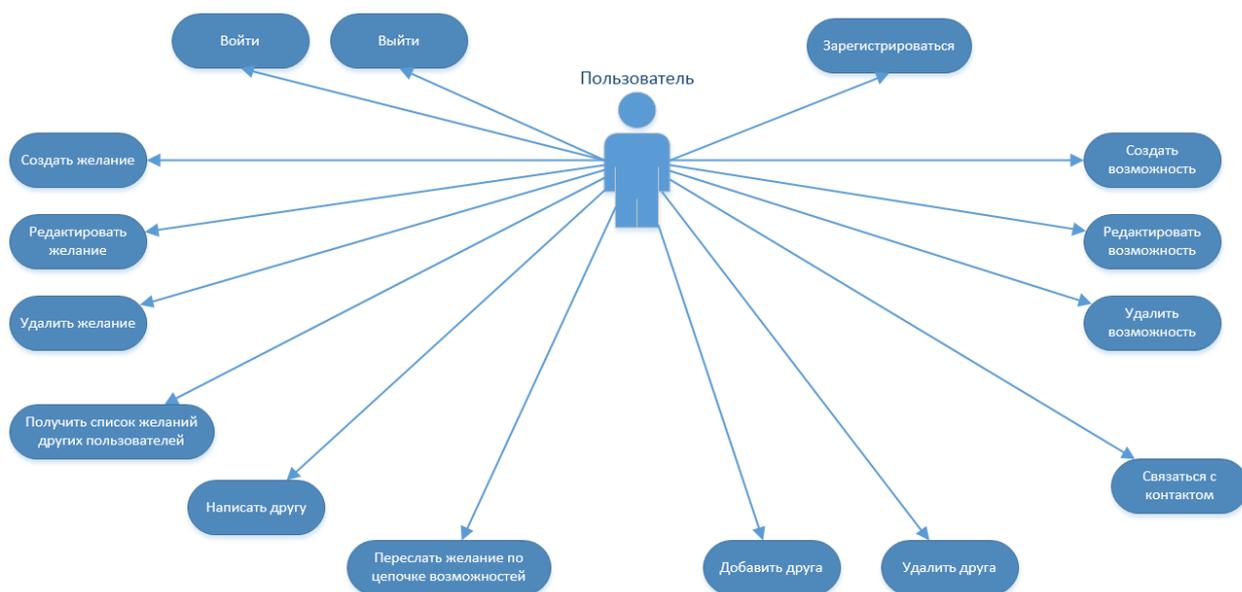


Рисунок 11 Диаграмма вариантов использования для пользователя

5.2. Алгоритмы поиска

В процессе проектирования ИС была заложена идея об использовании нескольких алгоритмов, которые будут искать связи между потребностями и возможностями пользователей, исходя из данных в графовой базе данных и получать данные в виде цепей из пользователей.

В процессе получения данных от пользователя о новых желаниях или потребностях, контроллер создаёт отдельный поток с созданием отложенной задачи по запуску алгоритмов поиска и добавляет его в список задач. Когда метод для поиска освобождается, он берёт новую задачу из списка и начинает выполнение. В качестве алгоритмов были выбраны: семантический поиск, расстояние Левенштейна и категоризация данных. Метод последовательно выполняя поиск с использованием разных алгоритмов и данных из графовой СУБД, сравнивая тексты и категории желаний и возможностей трёх ближайших кругов контактов. По окончании поиска формируется список возможных цепей из пользователей, который сохраняется во временное хранилище с быстрым доступом, имеющее выгрузку и загрузку данных в файл.

Используя семантический поиск метод анализирует входящие данные, разбивает предложения на слова и составляет список часто встречающихся

ближайших слов. Используя эти знания система старается подобрать слово похожие по ближайшим словам, для замены одно смысловых слов. Это позволяет искать похожие по смыслу слова в данных о возможностях других пользователей. Сбор данных о схожести слов, через использование ближайших слов, приводит к обучению системы и улучшению поиска, создавая тем самым базу для будущих анализов. Модератор имеет возможность редактировать данные в структуре схожести слов.

Вместе с этим, используется алгоритм расстояния Левенштейна, который ищет минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую, и предоставляет данные насколько слова похожи. Данный алгоритм используется для поиска похожих слов с небольшими изменениями, в приставках, суффиксах, окончаниях и однокоренных словах с изменением букв в разных падежах русского языка. Используя и тестируя алгоритм расстояния Левенштейна было выявлено, что изменение 2-3 букв в словах с длиной более 5-6 букв, является достаточным для выявления однокоренных похожих слов.

Анализируя слова в предложении, опытным путём было выявлено, что 60 и более % достаточно для утверждения о схожести двух предложений о потребности и возможности пользователей и вероятном их соединении. Пример работы алгоритма расстояния Левенштейна показан на рисунке 12.

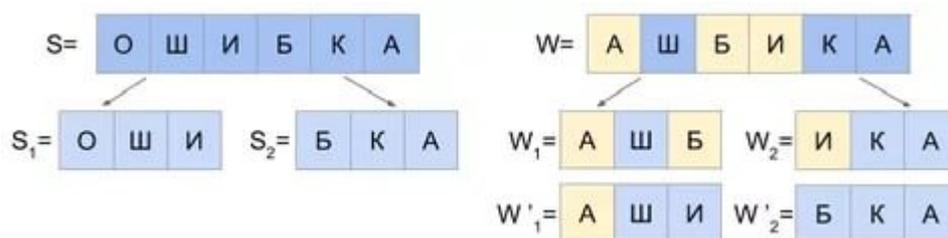


Рисунок 12 - схема работы алгоритма расстояния Левенштейна

Для ускорения поиска используется категоризация, которая помогает провести анализ данных ближайших кругов в данной категории, тем самым ускоряя процесс. Так как данные предложений одинаковых категорий, чаще

всего приводят к поиску цепей с небольшим количеством звеньев, поэтому данный алгоритм был добавлен и успешно используется вместе с остальными.

Так как поиск с использованием системы алгоритмов не всегда приводит к выявлению цепей из пользователей, то была реализована система информирования контактов о потребностях пользователей с возможностью ручной передачи потребностей между пользователями, создавая тем самым цепи, которые приводят желающего к исполнителю. На рисунке 13 показана общая последовательность действия и запуск алгоритмов поиска цепей.



Рисунок 13 - Диаграмма последовательности действий

6. Тестирование

Для тестирования создадим небольшой набор данных из 20 пользователей. У каждого из них будет по 10 контактов, потребностей и возможностей. Для этого будет сделан один тестовый метод, при вызове которого будет выполняться разработанный тестовый план, включающий следующие пункты:

1. Создание и подготовка данных пользователя;
2. Запись созданного класса данных в БД;
3. Цикл создания списка потребностей, заполнение тестовыми данными;
4. Сохранение в БД, списка потребностей с созданием связи между пользователем и потребностью;
5. Цикл создания списка возможностей пользователя, с заполнение тестовыми данными;
6. Сохранение списка возможностей пользователя в БД, с созданием связи между пользователем и возможностью;
7. Цикл поиска случайных контактов, с созданием списка выбранных контактов;
8. Цикл соединения выбранных контактов с данным пользователем, с созданием связи между пользователями и сохранением в БД.

В результате вызова метода, в БД добавлялись новые данные для тестирования. В дальнейшем запускался метод, для поиска цепей между пользовательскими потребностями и возможностями, в ближайших трёх кругах. В результате получались цепи длиной от 2х до 7 пользователей с определённым процентом совпадения. В качестве результатов выбирались данные, где процент совпадения текстов потребности и возможности разных пользователей, был больше 60%.

Проанализировав данные было решено, что на данном этапе система поиска взаимосвязей работает хорошо. В будущем будет производиться тестирование на небольшом количестве реальных пользователей.

7. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.

7.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.

7.1.1 Потенциальные потребители результатов исследования

Для анализа потребителей результатов исследования необходимо рассмотреть целевой рынок и провести его сегментирование.

Объект исследование: информационная система для исполнения потребностей одних пользователей с помощью возможностей других пользователей.

Целевой рынок: социальные сети, поисковые системы, форумы, активные пользователи интернета.

Таблица 7 - Сегментирование целевого рынка

Вид услуги	Вид интернет-ресурса		
	Информационная система	Социальная сеть	Поисковая система
Поиск	+	+	+
Общение	+	+	-
Выполнение потребностей	+	-	-

- 1) Поиск - получение информации, удовлетворяющей критериям поиска.
- 2) Общение - коммуникация пользователей посредством чатов.
- 3) Выполнение потребностей - создание цепочек потребитель-исполнитель, с учётом пользовательских связей и информации о потребности потребителя и возможности исполнителя.

Таким образом, можно сделать вывод, что информационный портал, обладает расширенным функционалом и более ресурсоэкономный по сравнению со другими видами интернет-ресурсов. Привлекательным сегментом и целью будущего развития является внедрение разрабатываемого информационного портала в различные интернет-ресурсы.

7.1.2. Анализ конкурентных технических решений

Таблица 7.1 - Оценочная карта для сравнения конкурентных технических решений.

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б _ф	Б _{к1}	Б _{к2}	К _ф	К _{к1}	К _{к2}
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Точность поиска для выполнения потребностей	0,3	5	2	3	1,5	0,6	0,9
2. Скорость поиска	0,1	4	3	5	0,4	0,3	0,5
3. Вовлечённость пользователей	0,05	5	5	5	0,25	0,25	0,25
4. Простота использования	0,12	4	4	5	0,48	0,48	0,6
5. Лёгкость в разработке	0,08	2	4	5	0,16	0,32	0,4
6. Безопасность	0,08	4	4	1	0,32	0,32	0,08
Экономические критерии оценки эффективности							
1. Конкурентоспособность продукта	0,1	5	3	1	0,5	0,3	0,1
2. Цена	0,08	5	3	5	0,4	0,24	0,4
3. Финансирование научной разработки	0,09	4	5	5	0,36	0,45	0,45
Итого	1	38	33	35	4,37	3,26	3,68

К₁ – социальная сеть или форум.

К₂ – поисковая система.

Как видно из анализа, наш метод имеет высокий балл по сравнению с другими видами интернет-решений, что делает его конкурентоспособным на рынке. Преимущества нашего интернет-решения по сравнению с остальными в том что для решения поставленной задачи он подходит больше остальных, вместе с тем имеет хорошие показатели в лёгкости использования, вовлечённости пользователей в процесс и низкую стоимость использования.

7.1.3. Технология QuaD

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

Таблица 7.2 - Оценочная карта для сравнения конкурентных технических решений (разработок).

Критерии оценки	Все критерии	Баллы	Максимальный балл	Относительное значение (3/4)	Средневзвешенное значение (5x2)
1	2	3	4	5	
Показатели оценки качества разработки					
1. Точность поиска для выполнения потребностей	0,3	95	100	0,95	0,285
2. Скорость поиска	0,1	80	100	0,8	0,08
3. Безопасность	0,1	80	100	0,8	0,08
Показатели оценки коммерческого потенциала разработки					
4. Конкурентоспособность продукта	0,2	80	100	0,8	0,16
5. Цена	0,1	70	100	0,7	0,07
6. Финансовая эффективность научной разработки	0,2	85	100	0,85	0,17
Итого	1	500	600	4,9	0,845

Оценка качества и перспективности по технологии QuaD определяется по формуле:

$$P_{cp} = \sum V_i \cdot B_i$$

где P_{cp} – средневзвешенное значение показателя качества и перспективности научной разработки; V_i – вес показателя (в долях единицы); B_i – средневзвешенное значение i -го показателя.

Полученное средневзвешенное значение (84,5) позволяет считать перспективность данной разработки.

7.1.4. SWOT – анализ

Данный раздел - комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта.

Таблица 7.3 - Матрица SWOT.

	Сильные стороны	Слабые стороны
	<p>С1. Квалифицированный персонал.</p> <p>С2. Более низкая стоимость разработки по сравнению с другими интернет-решениями.</p> <p>С3. Экономичность</p> <p>С4. Инновационная идея</p>	<p>Сл1. Дефицит программистов исследователей</p> <p>Сл2. Большой объём данных для обработки</p> <p>Сл3. Необходимость быстрой вычислительной мощности.</p>
Возможности	СиВ	СЛиВ
<p>В1. Использование собственных вычислительных мощностей</p> <p>В2. Появление дополнительного спроса на продукт.</p> <p>В3. Расширение областей применения данного продукта.</p> <p>В4. Сотрудничество с иностранными компаниями, ВУЗами.</p> <p>В5. Снижение стоимости разработки интернет-решения.</p>	<p>1. Проведение исследований с целью повышения качества получаемых поисковых решений.</p> <p>2. Возможность взрывного роста с быстрым набором пользователей из социальных сетей.</p>	<p>1. Увеличение областей применения данного метода привлечет новых инвесторов.</p> <p>2. Обработка больших объёмов данных, позволяет найти новые области применения.</p>
Угрозы	СиУ	СЛиУ
<p>У1. Быстрое масштабирование проекта</p> <p>У2. Отсутствие финансирования</p> <p>У3. Отсутствие спроса на результаты исследования</p>	<p>1. Повышение конкурентоспособности за счет расширения областей применения интернет-решения.</p>	<p>1. Плохая компания по продвижению продукта</p> <p>2. Быстрый рост объёмов данных</p> <p>3. Дефицит новых программистов.</p>

В результате SWOT-анализа целью научно-исследовательской работы является реализация интернет-решения с поиском путей для реализации потребностей за счёт возможностей пользователей данного интернет-решения и проведение исследований с целью повышения качества получаемых поисковых решений. Для увеличения спроса и количества пользователей будут добавлены взаимосвязи с разными социальными сетями. Вместе с тем интересность проекта и отсутствие аналогичных решений может быть полезно в продвижении.

7.2. Определение возможных альтернатив проведения научных исследований.

Таблица 7.4 - Морфологическая матрица для интернет-решения

	1	2	3
А. Хранение данных	Графовая БД	Реляционная БД	Документо-ориентированная
Б. Поиск	Полнотекстовой	Семантический	Комбинированный
В. Общение	Чат	Лента новостей	Отсутствует

Оценивание вариантов с позиции их функционального содержания и ресурсосбережения наиболее оптимальными считаю следующие варианты использования:

Использование 1: А1Б3В2

Использование 2: А2Б2В1

Использование 3: А3Б1В3

7.3. Планирование научно-исследовательских работ.

7.3.1. Структура работ в рамках научного исследования.

Для выполнения научных исследований необходимо формирование рабочей группы, в состав которой могут входить научные сотрудники и преподаватели, инженеры, техники и лаборанты, численность групп может варьироваться.

Планирование комплекса предполагаемых работ осуществляется в следующем порядке:

1. определение структуры работ в рамках научного исследования;
2. определение участников каждой работы;
3. установление продолжительности работ;
4. построение графика проведения научных исследований

Таблица 7.5 - Перечень этапов, работ и распределение исполнителей.

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания	Руководитель, дипломник
Выбор направления Исследований	2	Поиск и изучение материалов по теме	Студент-Дипломник
	3	Выбор направления исследований	Руководитель, дипломник
	4	Календарное планирование работ по теме	Руководитель, дипломник
Теоретические и экспериментальные исследования	5	Разработка интернет- решения	Студент-Дипломник
	6	Тестирование	Студент-Дипломник
	7	Анализ экспериментальных данных	Студент-Дипломник
Обобщение и оценка результатов	8	Оценка эффективности полученных результатов	Руководитель, дипломник
	9	Определение целесообразности проведения ВКР	Руководитель
Проведение ОКР			
Разработка технической документации и проектирование	10	Разработка блок-схемы, диаграмм UML	Студент-Дипломник
	11	Оформление результатов по охране труда	Дипломник, консультант по охране труда
	12	Подбор данных для выполнения экономической части работы	Студент-Дипломник
	13	Оформление экономической части работы	Студент-дипломник и консультант по экономике
Оформление отчета по НИР (комплекта документации по ОКР)	14	Составление пояснительной записки	Студент-Дипломник
	15	Процесс рецензирования работы	Консультанты по охране труда и экономической части
	16	Защита бакалаврской работы	Руководитель, дипломник

7.3.2. Определение трудоемкости выполнения работ

Для определения ожидаемого (среднего) значения трудоемкости $t_{ожі}$ используется следующая формула:

$$t_{ожі} = \frac{3t_{\min i} + 2t_{\max i}}{5}$$

$t_{ожі}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

$t_{\min i}$ – минимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.;

$t_{\max i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях T_p , учитывающая параллельность выполнения работ несколькими исполнителями:

$$T_{p_i} = \frac{t_{ожі}}{Ч_i}$$

T_{p_i} – продолжительность одной работы, раб. дн.;

$t_{ожі}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.

$Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел

7.3.3 Разработка графика проведения проектирования

При выполнении дипломных работ студенты в основном становятся участниками сравнительно небольших по объему научных тем. Поэтому наиболее удобным и наглядным является построение ленточного графика проведения научных работ в форме диаграммы Ганта.

Для перевода длительности каждого этапа из рабочих в календарные дни, необходимо воспользоваться формулой:

$$T_{ki} = T_{p_i} \cdot k_{\text{кал}}$$

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

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

$k_{\text{кал}}$ – коэффициент календарности.

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}},$$

Где $T_{\text{кал}}$ – количество календарных дней в году, 365

$T_{\text{вых}}$ – количество выходных дней в году, 59

$T_{\text{пр}}$ – количество праздничных дней в году, 14

$$k_{\text{кал}} = \frac{365}{365 - 59 - 14} = 1,25$$

Таблица 7.6 - Временные показатели проведения научного исследования.

№	Наименование работ	Трудоёмкость работ									Исполнители
		t_{min} , чел-дни			t_{max} , чел-дни			$t_{\text{ож}}$, чел-дни			
		Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	
1	Составление и утверждение технического задания	1	1	1	3	3	3	1.8	1.8	1.8	Р
2	Выбор направления исследований	1	1	1	3	3	3	1.8	1.8	1.8	Р Д
3	Поиск и изучение материалов по теме	3	3	3	7	7	7	4.6	4.6	4.6	Д
4	Календарное планирование работ по теме	1	1	1	3	3	3	1.8	1.8	1.8	Р Д
5	Разработка интернет- решения	15	15	15	200	200	200	89	89	89	Д
6	Тестирование	10	10	10	100	100	100	46	46	46	Д

7	Анализ экспериментальных данных	3	3	3	15	15	15	7.8	7.8	7.8	Р, Д
8	Оценка эффективности полученных результатов	3	3	3	7	7	7	4.6	4.6	4.6	Д Р
9	Разработка блок-схемы, диаграмм UML	5	5	5	14	14	14	8.6	8.6	8.6	Д
10	Оформление результатов по охране труда	5	5	5	8	8	8	6.2	6.2	6.2	Д К ₁
11	Оформление экономической части работы	7	7	7	14	14	14	9.8	9.8	9.8	Д К ₂
12	Составление пояснительной записки	15	15	15	25	25	25	19	19	19	Д
13	Процесс рецензирования работы	4	4	4	8	8	8	5.6	5.6	5.6	Д К ₁ К ₂
14	Защита бакалаврской работы	2	2	2	5	5	5	3.2	3.2	3.2	Д Р

Таблица 7.7 - Временные показатели проведения научного исследования.

№	Наименование работ	Длительность работ в рабочих днях, T_{pi}			Длительность работ в календарных днях, T_{ki}			Исполнители
		Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	
1	Составление и утверждение технического задания	1.8	1.8	1.8	2,25	2,25	2,25	Р
2	Выбор направления исследований	1.8	1.8	1.8	2,25	2,25	2,25	Р Д
3	Поиск и изучение материалов по теме	4.6	4.6	4.6	5,75	5,75	5,75	Д

4	Календарное планирование работ по теме	1.8	1.8	1.8	2,25	2,25	2,25	Р Д
5	Разработка интернет-решения	89	89	89	111,25	111,25	111,25	Д
6	Тестирование	46	46	46	57,5	57,5	57,5	Д
7	Анализ экспериментальных данных	7.8	7.8	7.8	9,75	9,75	9,75	Р, Д
8	Оценка эффективности полученных результатов	4.6	4.6	4.6	5,75	5,75	5,75	Д Р
9	Разработка блок-схемы, диаграмм UML	8.6	8.6	8.6	10,75	10,75	10,75	Д
10	Оформление результатов по охране труда	6.2	6.2	6.2	7,75	7,75	7,75	Д, К ₁
11	Оформление экономической части работы	9.8	9.8	9.8	12,25	12,25	12,25	Д К ₂
12	Составление пояснительной записки	19	19	19	23,75	23,75	23,75	Д
13	Процесс рецензирования работы	5.6	5.6	5.6	7	7	7	Д К ₁ К ₂
14	Защита бакалаврской работы	3.2	3.2	3.2	4	4	4	Д Р

• Р – руководитель

• Д – дипломник

• К₁ – консультант по охране труда

• К₂ – консультант по экономике

На основе табл. 7.7. строится календарный план-график. При этом работы на графике следует выделить различной штриховкой в зависимости от исполнителей, ответственных за разные виды работ (Таблица 7.8.).

Таблица 7.8. - Календарный план-график проведения проектирования по теме

№	Наименование работ	Исполнители	Т _{кi} , в кал. днях	Продолжительность выполнения работ														
				февр.			март			апрель			май			июнь		
				1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	Составление и утверждение технического задания	Руководитель	2,25	■														
2	Выбор направления исследований	Руководитель, дипломник	2,25	▨														
3	Поиск и изучение материалов по теме	Дипломник	5,75		▨													
4	Календарное планирование работ по теме	Руководитель, дипломник	2,25		■													
5	Разработка интернет-решения	Дипломник	111,25															
6	Тестирование	Дипломник	57,5															
7	Анализ экспериментальных данных	Руководитель, дипломник	9,75															
8	Оценка эффективности полученных результатов	Руководитель, дипломник	5,75															

7.4 Бюджет научно-технического исследования (НТИ).

7.4.1. Материальные затраты.

Расчет материальных затрат осуществляется по следующей формуле:

$$Z_m = (1 + k_T) \cdot \sum_{i=1}^m C_i \cdot N_{расхi}$$

где m – количество видов материальных ресурсов, потребляемых при выполнении научного исследования; $N_{расхi}$ – количество материальных ресурсов i -го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м² и т.д.); C_i – цена приобретения единицы i -го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м² и т.д.); k_T – коэффициент, учитывающий транспортно-заготовительные расходы.

Для реализации интернет-решения не требуются материальные затраты, но материалы используемые для проведения ВКР в процессе разработки и написания документации, были куплены в сети магазинов «Канцелярский мир». Материальные затраты, необходимые для данной разработки занесены в таблицу 7.9.

Таблица 7.9 - Материальные затраты

Наименование	Ед. измерения	Количество			Цена за единицу, руб.			Затраты на материалы, (Зм), руб.		
		Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Бумага	шт	1000	1200	800	0.52	0.52	0.52	560	664	456
Ручка	шт	4	5	3	40	40	40	170	210	130
Итого:								730	874	586

7.4.2. Расчет затрат на оборудование для научно-экспериментальных работ.

В данную статью включают все затраты, связанные с приобретением специального оборудования. Определение стоимости спецоборудования производится по действующим прейскурантам. Стоимость оборудования имеющегося в научно-технической организации, учитываем в калькуляции в виде амортизационных отчислений.

Все расчеты по оборудованию, имеющегося и используемого для выполнения поставленных задач в организации, сводятся в табл. 7.10.

Таблица 7.10 - Расчет бюджета затрат на приобретение спецоборудования для научных работ

№	Наименование оборудования	Цена единицы оборудования, $K_{об\ i}$, руб	Количество	Норма амортизации, $N_{ам\ i}$, %	Мощность прибора, N_i , Вт	Сумма амортизационных отчислений, $E_{ам}$, руб.	Время использования, час.	Сумма затрат на электроэнергию, $E_{э}$, руб
1	Сервер ПК Core i7	32000	1	10	500	1052,05	2880	3240
2	Системный блок Core2Duo	12500	2	10	400	821,92	1080	972
3	Монитор LCD 17"	3500	2	10	200	230,14	1080	486
4	Клавиатура DEXP-400	410	2	20	0	53,92	1080	0
5	Мышь проводная Oklick 115S	230	2	20	0	30,25	1080	0
Итого:						2188,28		4698

Норма амортизации взята с учётом работы сервера, системного блока и монитора 10 лет, а клавиатуры и компьютерной мыши 5 лет.

Расчет амортизационных отчислений.

Затраты определяются в виде амортизации по формуле:

$$E_{\text{ам}} = (\sum K_{\text{обі}} \cdot N_{\text{амі}} \cdot T_{\text{обі}}) / (365 \cdot 100) ,$$

где $K_{\text{обі}}$ – стоимость ед. прибора или оборудования, руб.;

$N_{\text{амі}}$ - норма амортизации прибора или оборудования, %;

$T_{\text{обі}}$ – время использования оборудования, дни.

Расчет затрат на электроэнергию определяется по формуле:

$$E_{\text{э}} = \sum N_i \cdot T_{\text{э}} \cdot C_{\text{э}} ,$$

где N_i - мощность электроприборов по паспорту, кВт

$T_{\text{э}}$ - время использования электрооборудования, час;

$C_{\text{э}}$ - цена одного кВт·ч, руб.

$C_{\text{э}} = 2,25$ руб/ кВт·ч

7.4.3. Основная заработная плата исполнителей темы.

В статью включается основная заработная плата научных работников и непосредственно участвующих в выполнении работ по данной теме. Величина расходов по заработной плате определяется исходя из трудоемкости выполняемых работ и действующей системы окладов и тарифных ставок. В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы в размере 20 –30 % от тарифа или оклада. Расчет основной заработной платы сводится в таблицу 7.15.

Таблица 7.11 - Расчет основной заработной платы.

Исполнители	Исполнители по категориям	Трудоемкость, чел.-дн., Тр			Зарплата, приходящаяся на один чел.-дн., руб., $Z_{дн}$			Всего заработная плата по тарифу (оклам), руб., $Z_{оск}$		
		Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Руководитель	Ассистент	30	30	30	1518,1	1518,1	1518,1	45542,3	45542,3	45542,3
Дипломник		240	240	240	0	0	0	0	0	0
Консультанты:										
По Экономике	Ассистент	15	15	15	1518,1	1518,1	1518,1	22771,1	22771,1	22771,1
По охране труда	Доцент, к.т.н.	11	11	11	2199	2199	2199	24189	24189	24189
Итого:								92502,4	92502,4	92502,4

$$Z_{зп} = Z_{осн} + Z_{доп}$$

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

$Z_{доп}$ – дополнительная заработная плата.

Основная заработная плата работника рассчитывается по следующей формуле :

$$Z_{осн} = Z_{дн} \cdot T_p$$

где $Z_{осн}$ – основная заработная плата одного работника;

$T_{раб}$ – продолжительность работ, выполняемых ассистентом-преподавателем, раб.дн.

$Z_{\text{дн}}$ – 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.дней $M=10,4$ месяца, 6-дневная неделя;

$F_{\text{д}}$ – действительный годовой фонд рабочего времени научно-технического персонала (в рабочих днях).

Таблица 7.12 - Баланс рабочего времени

Показатели рабочего времени	Руководитель	Консультант по экономике	Консультант по БЖД
Календарное число дней	365	365	365
Количество нерабочих дней - выходные дни - праздничные дни	118 дней		
Потери рабочего времени			
- отпуск	48	48	48
- невыходы по болезни	0	0	0
Действительный годовой фонд рабочего времени	244	244	244

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 (т.е. 30% от $Z_{\text{тс}}$);

$k_{\text{д}}$ – коэффициент доплат и надбавок составляет примерно 0,2;

$k_{\text{р}}$ – районный коэффициент, равный 1,3 (для Гомска).

Произведем расчет фонда заработной платы исследователей и занесем результаты в таблицу 7.13.

Таблица 7.13 - Расчет фонда заработной платы

Исполнители	$Z_{тс}, \text{руб.}$	$k_{пр}$	$k_{д}$	$k_{р}$	$Z_{м}, \text{руб.}$	$Z_{дн}, \text{руб.}$	$T_{р}, \text{раб. дн.}$	$Z_{осн}, \text{руб.}$
Руководитель	18264,82	0,3	0,2	1,3	35616,4	1518,1	30	45542,3
Дипломник	0	0,3	0,2	1,3	0	0	240	0
Консультант по Экономике	18264,82	0,3	0,2	1,3	35616,4	1518,1	15	22771,1
Консультант по охране труда	26457,4	0,3	0,2	1,3	51591,93	2199	11	24189
Итого:								92502,4

Отраслевая система оплаты труда в ТПУ предполагает следующий состав заработной платы:

-оклад – определяется предприятием. В ТПУ оклады распределены в соответствии с занимаемыми должностями, например, ассистент, ст. преподаватель, доцент, профессор.

-стимулирующие выплаты – устанавливаются руководителем подразделений за эффективный труд, выполнение дополнительных обязанностей и т.д.

-иные выплаты. В заработной плате следует предусмотреть: надбавку за степень (кандидат наук – 3000 руб., доктор наук – 6000 руб.); надбавку ученого совета (от 2000 руб. до 10000 руб.); районный коэффициент.

Надбавки в заработной плате составляют: 10000 руб. (надбавка учёного совета); 1,3 (районный коэффициент по Томску).

7.4.4 Дополнительная заработная плата исполнителей темы

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 (на стадии проектирования принимается равным 0,12 – 0,15).

$$З_{\text{доп.руково дителя}} = 0,12 \cdot 45542,3 = 5465,08 \text{ руб.}$$

$$З_{\text{доп.дипло мника}} = 0,12 \cdot 0 = 0 \text{ руб.}$$

$$З_{\text{доп.консультанта эк.}} = 0,12 \cdot 22771,1 = 2732,53 \text{ руб.}$$

$$З_{\text{доп.консультанта бжд}} = 0,12 \cdot 24189 = 2902,68 \text{ руб.}$$

7.4.5. Отчисления во внебюджетные фонды (страховые отчисления)

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников. Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

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

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

На 2016 г. в соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30%. На основании пункта 1 ст.58 закона №212-ФЗ для учреждений осуществляющих образовательную и научную деятельность в 2016 году введена пониженная ставка – 27,1%.

Отчисления во внебюджетные фонды представлены в табличной форме (таблица 7.14).

Таблица 7.14 - Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.			Дополнительная заработная плата, руб.			З _{внеб}		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Дипломник	0	0	0	0	0	0	0	0	0
Руководитель проекта	45542	45542	45542	5465	5465	5465	15302,1	15302,1	15302,1
Консультант по экономике	22771	22771	22771	2732	2732	2732	7650,9	7650,9	7650,9
Консультант по охране труда	24189	24189	24189	2902	2902	2902	8127,3	8127,3	8127,3

7.4.6. Накладные расходы

В данную статью входят расходы на содержание аппарата управления и общехозяйственных (общеуниверситетских) служб, которые в равной степени относятся ко всем выполняемым НИР. По этой статье учитываются оплата труда административно-управленческого персонала, содержание зданий, оргтехники и хозяйственного инвентаря, амортизация имущества, расходы по охране труда и подготовке кадров. Их величина определяется по следующей формуле:

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

где k – коэффициент, учитывающий накладные расходы (можно взять в размере 16%).

7.4.7. Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно-исследовательской работы является основой для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку интернет-решения.

Определение бюджета затрат на научно-исследовательский проект по каждому варианту исполнения приведен в таблице 7.15.

Таблица 7.15 - Расчет бюджета затрат НИИ

Наименование статьи	Сумма, руб.			Примечание
	Исп.1	Исп.2	Исп.3	
1. Материальные затраты	730	874	586	Табл. 7.9
2. Затраты на специальное оборудование для научных (экспериментальных) работ	2188,28	2188,28	2188,28	Табл. 7.10
3. Затраты по основной заработной плате исполнителей темы	92502.4	92502.4	92502.4	Табл.7.11
4. Затраты по дополнительной заработной плате исполнителей темы	11099	11099	11099	Пункт 7.4.4
5. Отчисления во внебюджетные фонды	31080,3	31080,3	31080,3	Пункт.7.4.14
6. Накладные расходы	22016	22039,04	21992,96	16 % от суммы ст. 7.4.6-7.4.14
7. Бюджет затрат НИИ	159615,98	159783,02	159448,94	сумма ст. 7.4.1-7.4.14

7.5. Определение ресурсной, финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности

Интегральный финансовый показатель разработки определяется как:

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

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

Φ_{ri} – стоимость i -го варианта исполнения;

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

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{ri} = \sum a_i \cdot b_i$$

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

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i^a, b_i^p – балльная оценка i -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

n – число параметров сравнения.

Таблица 7.16 - Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Объект исследования	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
1. Точность поиска для выполнения потребностей		0,3	5	2	3
2. Скорость поиска		0,15	4	3	5
3. Вовлечённость пользователей		0,15	5	5	5
4. Простота использования		0,2	4	4	5
5. Лёгкость в разработке		0,1	2	4	5
6. Безопасность		0,1	4	4	1
Итого:		1	4.25	3.4	4.00

$$I_{p-исп1} = 5 \cdot 0.3 + 4 \cdot 0.15 + 5 \cdot 0.15 + 4 \cdot 0.2 + 2 \cdot 0.1 + 4 \cdot 0.1 = 4.25$$

$$I_{p-исп2} = 2 \cdot 0.3 + 3 \cdot 0.15 + 5 \cdot 0.15 + 4 \cdot 0.2 + 4 \cdot 0.1 + 4 \cdot 0.1 = 3.4$$

$$I_{p-исп3} = 3 \cdot 0.3 + 5 \cdot 0.15 + 5 \cdot 0.15 + 5 \cdot 0.2 + 5 \cdot 0.1 + 1 \cdot 0.1 = 4.00$$

Интегральный показатель эффективности вариантов исполнения разработки ($I_{исп.i}$) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп.1} = \frac{I_{p-исп.1}}{I_{финр.1}}, I_{исп.2} = \frac{I_{p-исп.2}}{I_{финр.2}} \text{ и т.д.}$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта (см.табл.7.18) и выбрать наиболее целесообразный вариант из предложенных. Сравнительная эффективность проекта ($\mathcal{E}_{ср}$):

$$\mathcal{E}_{ср} = \frac{I_{исп.1}}{I_{исп.2}}$$

Таблица 7.17 - Сравнительная эффективность разработки

№ п/п	Показатели	Исп.1	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	0.999	1	0.998
2	Интегральный показатель ресурсоэффективности разработки	4.25	3.4	4.00
3	Интегральный показатель эффективности	4.2542	3.4	4.008
4	Сравнительная эффективность вариантов исполнения	1	0.7992	0.9421

Вывод: При сравнении значений интегральных показателей эффективности было выявлено, что более эффективным вариантом решения поставленной технической задачи в бакалаврской работе с позиции финансовой и ресурсной эффективности является реализация 1-го варианта интернет-решения, который является информационной системой для исполнения потребностей одних пользователей с помощью возможностей других пользователей.

Заключение

В будущем проект будет развиваться и перспективой проекта является:

1. Создание на основе разработанной системы, системы для взаимодействия проектов, эта система подразумевает, что в качестве пользователя выступает конкретный бизнес со своими возможностями и потребностями. Например управляющему бизнесом нужен дерево обрабатывающий станок, и этот бизнес имеет возможность делать 3D печать, соответственно по аналогии с обычными людьми может так же взаимодействовать и бизнес. Посредством системы бизнес сообщества становятся плотнее друг к другу.
2. Встраивание системы в структуру "Университета 3.0", в данной концепции, университет является самым ценным ресурсом сейчас для развития сотрудника, бизнеса, проекта. Главным для развития является не ресурсы, не знания, а возможность быстрого доступа к знаниям. Университет становится агрегатором стартапов, бизнесов и мотивированных на воплощение идей студентов.
3. Создание территории доверия - это система, когда один пользователь не станет передавать информацию, помогать в исполнении желания, другому пользователю если он не может за него поручиться, и не отвечает за его адекватность и благонадёжность. Соответственно на основании этой системы, можно создавать территорию доверия которая может лечь в основу концепции "Университета 3.0". Работы над этим ведутся, в институте социально-гуманитарных технологий Томского Политехнического Университета.

Список использованных источников

1. Открытая энциклопедия "Википедия" [Электронный ресурс] / Поисковая система, 2016. URL: https://ru.wikipedia.org/wiki/Поисковая_система, вход свободный. Дата обращения 06.05.2016г.
2. База электронных статей [Электронный ресурс] / Веб-проект "Социальная сеть", 2014. URL: <http://diplomba.ru/work/133638>, вход свободный. Дата обращения 10.05.2016г.
3. Открытая энциклопедия "Википедия" [Электронный ресурс] / Инструмент разработки Microsoft Visual Studio, 2016. URL: https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio, вход свободный. Дата обращения 14.05.2016г.
4. Открытая энциклопедия "Википедия" [Электронный ресурс] / Графовая СУБД Neo4J, 2016. URL: <https://ru.wikipedia.org/wiki/Neo4j>, вход свободный. Дата обращения 12.05.2016г.
5. Библиотека Microsoft [Электронный ресурс] / Технология ASP.NET 2.0, 2015. URL: [https://msdn.microsoft.com/ru-ru/library/hh396384\(v=vs.111\).aspx](https://msdn.microsoft.com/ru-ru/library/hh396384(v=vs.111).aspx), вход свободный. Дата обращения 14.05.2016г.
6. База курсовых работ [Электронный ресурс] / Разработка программного обеспечение для автоматизации формирования учебных программ, 2014. URL: <http://westud.ru/work/214887/Razrabotka-programmnogo-obespecheniya-dlya>, вход свободный. Дата обращения 20.05.2016г.

Приложение А

The General idea of the Information System is receiving, processing, storage and delivery of information to users who receive the data, make decisions, contribute new information.

People have many desires and are looking for an opportunity to implement them in a short time and with minimal effort. It often turns out that someone from contacts or the second circle of contacts of the person, it is possible to implement certain desires. But people may not know that one of the contacts might help. Many well-known proverb "people familiar with the entire world in a few handshakes", which means that users of Information system can be interrelated and you can create the Information system that will unite members to address common goals for couples or group of users.

Task developed is an overview and collection of the needs and capabilities of users, with different access levels of contacts, search through the first round or following rounds of contacts, those who is a possible perpetrator or assistant in the execution of a specific defined user needs, to inform the user contacts new needs, with the aim to involve users in the process of improving search and give the ability to manually specify the path to the Executive.

The system is based on researched and identified by the psychologists of the model, which is aimed at maintaining comfort, ecological relationships-relations between the user model LPSW, i.e for implementing the human need to be satisfied three criteria:

subjectively profitable - the action should be beneficial to the person;

simple - simple in execution;

easy should not cause a person psychological complexity;

The system involves the consideration of the balance of good deeds, as one participant in the relationship is not comfortable regularly unrequited to perform good deeds, respectively, the system keeps track of and provides users with information about where, whom and how many good deeds have been done that ensures your

relationship. This is accomplished through the control of 3-4 good deeds on one side, the other side must make at least 1 good deed, fulfilling the need of the participant.

The system requires users to report on the results of performance requirements, therefore, the system maintains a user rating that would identify how well it fulfilled the need of the user, and to enable feedback to those people who fulfill the needs of the user. Other users seeing the rating can judge the competence of the opponent, and make a decision to transfer or entrust execution of the requirements of the user.