

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
РОСТОВСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ
УНИВЕРСИТЕТ (РИНХ)

**Факультет Компьютерных технологий и
информационной безопасности**

**Кафедра Информационных технологий и защиты
информации**

ДОПУСТИТЬ К ЗАЩИТЕ
Зав.кафедрой ИТиЗИ
к.э.н., доцент
Ефимова Е.В.
« ____ » _____ 2020
г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
на тему:
«РАЗРАБОТКА МОДУЛЯ СБОРА И АНАЛИЗА
ПУБЛИЧНОЙ ИНФОРМАЦИИ ДЛЯ МАРКЕТИНГОВОГО
АГЕНТСТВА»**

Выполнил(а)

студент(ка) группы ПРИ-

341

Направление

подпись

М.В. Ремез

09.03.04 Программная инженерия

Руководитель выпускной
квалификационной
работы
д.э.н., проф.

подпись

Е.Н. Тищенко

Ростов-на-Дону, 2020
ФГБОУ ВО «РОСТОВСКИЙ ГОСУДАРСТВЕННЫЙ
ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ (РИНХ)»

Факультет Компьютерных технологий и информационной
безопасности
Кафедра Информационных технологий и защиты информации

«УТВЕРЖДАЮ»
Зав.кафедрой ИТиЗИ
к.э.н., доцент
Ефимова Е.В.
«___» _____ 20__ г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

Обучающегося Ремеза Максима Валерьевича
группы ПРИ-341

Ф. И. О.

1. Тема выпускной квалификационной работы:

Разработка модуля сбора и анализа публичной
информации для маркетингового агентства _____

2. Срок сдачи студентом законченной ВКР на кафедру «___»
_____ 20__ г.

3. Исходные данные для ВКР

_____ ООО
«Фейском»

*указать название и местонахождение организации, на материалах которой подготовлена
работа*

4. Структура ВКР

- 1) Анализ предметной области
- 2) Разработка веб-сервиса

3) Экономическое обоснование программного
продукта

Дата выдачи задания « ____ » _____ 20__ г.

Руководитель ВКР _____ Тищенко Е.Н.
подпись Ф. И. О.

Задание к исполнению принял _____ Ремез
М.В. _____
подпись Ф.И.О. обучающегося

РЕФЕРАТ

Выпускная квалификационная работа 61с., 27 рис., 27 табл., 39 источников, 1 прил.

ВЕБ-СЕРВИСЫ, СЕРВЕРНОЕ ПРОГРАММИРОВАНИЕ, PYTHON, DJANGO, API, ОТКРЫТЫЕ ИСТОЧНИКИ ДАННЫХ, SMM.

Объектом разработки являются алгоритмы сбора и анализа информации из открытых источников для сотрудников маркетингового агентства ООО «Фейском».

Предмет разработки – модуль сбора и анализа информации из открытых источников для ООО «Фейском».

Методы проведения работы:

- исследование предметной области;
- функциональное и объектно – ориентированное программирование;
- тест-кейсы программного кода;
- нагрузочное тестирование веб-сервиса;
- отладка и оптимизация существующих алгоритмов.

Целью выпускной квалификационной работы является анализ существующих методов сбора информации из открытых источников, разработка и внедрение найденного оптимального метода в ООО «Фейском» для сотрудников smm-отдела агентства.

Результатом выполнения бакалаврской работы является веб-сервис позволяющий автоматически формировать отчетную информацию из открытых источников.

Результаты проекта подготовлены и внедрены в агентстве ООО «Фейском».

Экономическая эффективность проекта имеет положительное значение, согласно допустимым критериям оценивания.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

5

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

7

1.1 Исходные данные

7

1.2 Проблематика и актуальность разработки

8

1.3 Техническое задание

10

1.4 Выбор CASE-средств

13

1.5 Выбор инструментария разработки

14

1.6 Проектирование USE-CASE диаграмм

18

1.6.1 Разработка UML – диаграммы

18

1.6.2 Разработка диаграммы активности

20

1.6.3 Разработка диаграммы классов

24

1.6.4 Разработка диаграммы последовательностей

25

1.6.5 Разработка диаграммы компонентов

.....
28

1.7 Проектирование прототипа интерфейса
.....

29

1.8 Проектирование базы данных
.....

33

1.9 Календарный план разработки
.....

35

2 РАЗРАБОТКА ВЕБ-СЕРВИСА
.....

37

2.1 Разработка модуля сбора и анализа информации
.....

38

2.2 Разработка модуля личного кабинета пользователя
.....

40

2.3 Разработка модуля администрирования
.....

41

2.4 Тестирование программного продукта
.....

42

2.4.1 Выбор метода тестирования программного продукта
.....

42

2.4.2 Тест-кейсы проверки качества программного продукта
.....

42

2.4.3 Баг репорты программного продукта
.....

48

2.4.4 Нагрузочное тестирование
.....

53

3 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

.....

54

3.1 Расчет прямых и накладных расходов

.....

54

3.2 Себестоимость работ и расчет прибыли

.....

56

3.3 Оценка эффективности внедрения ПП

.....

57

ЗАКЛЮЧЕНИЕ

.....

58

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

.....

59

Приложение

.....

62

Приложение А Исходный код веб-сервиса

.....

62

ВВЕДЕНИЕ

Успешная деятельность современных digital - агентств зависит от множества факторов. Ключевыми из них являются эффективность оптимизации рабочего процесса и гибкость в выборе инструментов, которые они используют в своей работе.

Данная выпускная квалификационная работа выполнялась на основе исходных статистических данных маркетингового агентства Facесom. Агентство специализируется на области стратегических исследований рынков, планирования и разработки маркетинговой концепций, а также всестороннего управления маркетингом. В частности, занимается SMM - продвижением клиентов.

SMM-продвижение (Social Media Marketing) — это наиболее современный и эффективный способ привлечения аудитории и увеличения лояльности к бренду клиента посредством социальных сетей, блогов, форумов, сообществ. Social Media Marketing является наиболее перспективным методом продвижения. Он позволяет общаться с аудиторией напрямую социальных сетей. Задачи, которые можно решать с помощью SMM:

- продвижением бренда, брендингом;
- увеличением числа посетителей сайта;
- ростом известности марки;
- повышением лояльности аудитории по отношению к бренду.

В процессе своей работы SMM - специалисты ежедневно обращаются к различным сторонним средствам сбора и анализа информации о социальных группах их клиентов, а также анализируют успешность введения социальных сетей конкурентов клиентов. Под сбором информации следует понимать использование платных сервисов анализа социальных сетей (Popsters).

Помимо этого, львиную долю времени от сбора и анализа информации занимает ручное копирование статистических данных из платных сервисов и последующие составление отчетности с помощью этих данных средствами Microsoft Excel. Количество анализируемых проектов прямо пропорционально затраченным времени сотрудников и финансовых ресурсов компании.

От того насколько оптимизирована будет работа SMM-специалиста зависит очень многое:

- сколько проектов одновременно он сможет вести;
- качество предоставляемого контента;
- эффективность введения социальных групп.

Анализируя опыт работы схожих агентств, становится очевидным, что программное обеспечение для анализа социальных сетей, разрабатываемое в основном зарубежными компаниями, не в полной мере соответствует специфике, финансовым возможностям, принципам функционирования и нуждам рассматриваемой агентства. Поэтому актуальным становится разработка сервиса, который удовлетворит потребности агентства в оптимизации времени сотрудников и снижении финансовых издержек агентства.

Для разрешения проблемы, связанной с анализом больших объемов данных в социальных сетях, необходима разработка веб-сервиса, который обеспечивает автоматизацию процесса сбора статистических данных и составление отчетности по SMM - продвижению.

Объектом разработки являются алгоритмы сбора и анализа информации из открытых источников, интегрированный в разрабатываемый веб-сервис.

В процессе проектирования и разработки следует решить задачи:

- провести анализ предметной области;
- спроектировать USE-CASE диаграммы, базу данных, прототип интерфейса программного продукта, блок-схемы и разработать алгоритмы работы программного продукта;
- провести тестирование программного продукта по средствам тест-кейсов и нагрузочного тестирования;
- определить эффективность работы smm-специалистов после внедрения программного продукта.

Целью выпускной квалификационной работы является анализ существующих методов сбора информации из открытых источников, разработка и внедрение найденного оптимального метода в ООО «Фейском».

Внедрение разрабатываемого программного продукта позволит повысить продуктивность работы SMM-специалистов и уменьшит финансовые издержки компании.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Исходные данные

Предметом исследования является маркетинговое агентство Facesom. Компания осуществляет свою деятельность с 2011 года. Facesom Marketing & Branding - одна из ведущих консалтинговых компаний Ростовской области, специализирующаяся в области стратегических исследований рынков, планирования и разработки маркетинговой концепций, а также всестороннего управления маркетингом.

Конкурентным преимуществом компании Facesom Marketing & Branding считается многократный успешный опыт внешнего управления маркетингом в производственных и торгующих компаниях.

Среди клиентов — крупнейшие производители, торговые центры, строительные и медицинские компании, банки и рестораны, государственные корпорации и др.

Уникальными услугами компании стали маркетинговые исследования, стратегическое планирование маркетинга, обеспечивающее рост рыночного потенциала, и аудит маркетинга на предприятиях.

За годы своей деятельности команда специалистов Facesom разработала уникальные методики и технологии, стабильно обеспечивающие долгосрочный рост продаж и стоимости нематериальных активов — репутация, торговые марки, бренды.

Помимо этого, ООО «Фейском» занимается SMM - продвижением бизнеса. SMM-продвижение (Social Media Marketing) — это наиболее современный и эффективный способ привлечения аудитории и увеличения лояльности к бренду клиента посредством социальных сетей, блогов, форумов, сообществ. Social Media Marketing является наиболее перспективным методом продвижения. Он позволяет общаться с аудиторией напрямую социальных сетей. Задачи, которые можно решать с помощью SMM:

- продвижением бренда, брендированием;
- увеличением числа посетителей сайта;

- ростом известности марки;
- повышением лояльности аудитории по отношению к бренду.

1.2 Проблематика и актуальность разработки

В процессе своей работы SMM - специалисты агентства регулярно обращаются к различным сторонним средствам сбора и анализа информации о социальных группах их клиентов, а также анализируют успешность введения социальных сетей конкурентов клиентов.

Под сбором информации следует понимать использование как платных сервисов анализа социальных сетей (Popsters - сервис аналитики популярности и актуальности контента и интересов аудитории в необходимых сообществах социальных сетей), так и общедоступных сервисов (встроенная статистика сообщества vk.com).

Собранная статистика необходима SMM - специалистам для составления ежемесячного отчета о проделанной работе, а также еженедельном отслеживании эффективности проведения маркетинговой компании. В следствии чего львиная часть времени сотрудника отнимается ручным копирование статистических данных из платных сервисов и последующие составление отчетности с помощью этих данных средствами Microsoft Excel.

Количество анализируемых проектов прямо пропорционально затраченным времени сотрудников и финансовых ресурсов компании. Количество проектов агентства достаточно велико, и им становится не выгодным использовать сторонний продукт анализа социальных сетей (Popsters). На таблице 1 продемонстрированы инструменты анализа социальных сетей, их стоимость и основные преимущества.

Таблица 1.1 - Инструменты анализа социальных сетей.

Продукт	Стоимость	Возможности

1	2	3
Popsters	от 399 руб./месяц за одну социальную сеть	1. Отчёты и аналитика (ВКонтакте, Одноклассники, Facebook, Instagram, Google+, Twitter, Pinterest) 2. Анализ конкурентов
Locowise	от 695 \$ в месяц	1. Отчёты и аналитика (Facebook, Instagram, Twitter, LinkedIn) 2. Планирование публикаций 3. Анализ конкурентов

Окончание таблицы 1.1

1	2	3
StarComm ent	Тарифы: • Старт (2 задания) — 290 рублей. • Стартап (10 заданий) — 690 рублей. • Компания (30 заданий) — 1900 рублей. Свой Инстаграм (10 аккаунтов) — 550 рублей.	1. Отслеживание упоминаний бренда (ВКонтакте, Одноклассники, Facebook, Instagram, Google+, Twitter) 2. Фильтры по ключевым словам
LiveDune	Тарифы: Блогер - 300 руб. в месяц Бизнес - 2000 руб. в месяц Агентство - 9900 руб. в месяц	1. Отчёты и аналитика (ВКонтакте, Одноклассники, Facebook, Instagram, Google+, Twitter, Pinterest) 2. Отслеживание упоминаний бренда
FeedSpy	От 4.95 \$ в месяц за одну социальную сеть	1. Отчёты и аналитика (ВКонтакте, Одноклассники, Facebook, Instagram, Google+,

		Twitter, Pinterest) 2. Отслеживание упоминаний бренда 3. Фильтры по ключевым словам
--	--	-------------------------------------------------------------------------------------------

От того насколько оптимизирована будет работа SMM-специалиста зависит очень многое:

- сколько проектов одновременно он сможет вести;
- качество предоставляемого контента;
- эффективность введения социальных групп.

Проведя анализ работы агентств с схожей спецификой работы, становится очевидным, что программное обеспечение для анализа социальных сетей, разрабатываемое в основном зарубежными компаниями, не в полной мере соответствует специфике, финансовым возможностям, принципам функционирования и нуждам рассматриваемой агентства.

Вследствие чего становится актуальным разработка сервиса, который удовлетворит потребности SMM-специалиста в оптимизации его рабочего процесса и позволит агентству снизить финансовые затраты на оплату платных сервисов анализа социальных групп.

Разрешением проблемы, связанной с анализом больших объемов данных по сбору статистических данных в социальных сетях, необходима разработка веб-сервиса, который обеспечивает автоматизацию процесса сбора статистических данных и составление отчетности по SMM - продвижению.

В процессе проектирования и разработки следует решить задачи:

- исследовать предметную область, сформулировать задачу автоматизации рабочего процесса агентства и провести обзор существующих программных средств в области анализа социальных сетей;
- составить техническое задание на разработку сервиса;
- спроектировать алгоритмы работы модулей веб-сервиса;
- спроектировать интерфейс модулей веб-сервиса;
- разработать сервис согласно собранным требованиям.

1.3 Техническое задание

Перед началом формирования технического задания необходимо собрать общие сведения по работе агентства: проанализировать деятельность сотрудников и выявить моменты в их работе, которые можно автоматизировать.

Далее необходимо сформировать общие требования к разрабатываемому программному обеспечению, а именно:

- обозначить назначение и цель разработки;
- определить в каком виде будет существовать программное обеспечение (сайт, мобильное приложение, настольное программное обеспечение и пр.);
- описать характеристики объекта автоматизации;
- описать требования к программному обеспечению;
- описать требования к квалификации сотрудников;
- выделить требования к надежности работы программного обеспечения;
- описать требования к безопасности;
- описать требования к дизайну и удобству использования программного продукта;
- описать требования к техническому обслуживанию программного продукта;
- выделить требования к документации;
- обозначить сроки проектирования, разработки, тестирования и внедрения программного обеспечения.

Сбор требований к разрабатываемому программному продукту было решено провести посредством интервью с сотрудниками агентства ООО «Фейском». Ниже описаны итоговые требования заказчика (генерального директора ООО «Фейском») к программному продукту, высказанные им в ходе беседы:

1) Предмет автоматизации – автоматизация сбора статистических данных для отчетных документов.

2) Цель разработки - оптимизация рабочего процесса SMM-специалиста агентства.

3) Программный продукт должен быть реализован в виде веб-сервиса с возможностью доступа к его функционалу 24/7 посредством веб-браузера.

4) Программный продукт должен осуществлять:

- Сбор статистических данных из открытых источников (vk.com) за временной период;
- Расчет статистических данных (количество подписчиков, лайков, комментариев, репостов, просмотров, средние показатели, ER, ERday, ERpost, до 5 популярных публикаций по категориям);
- Формировать отчетность и выводить её на экран пользователю;
- Сохранять в базу данных веб-сервиса отчетность по требованию пользователя;
- Предоставлять пользователю сохранённые им отчеты и возможность их удаления;
- Экспортировать отчетный документ в файл в формате xlsx.

5) Требования к пользователям - сервис должен быть рассчитан на использование пользователем без специальных технических навыков, знания технологий или программных продуктов, за исключением общих навыков работы с персональным компьютером под управлением операционной системы (ОС) Windows и стандартными веб-браузерами (Mozilla Firefox, Chrome, IE).

6) Требования к дизайну: дизайн сервиса следует делать минималистичным, в светлых тонах. Помимо этого, необходимо использовать адаптивную верстку для корректного отображения на персональных компьютерах (ПК), смартфонах и планшетах.

7) Требования к безопасности - доступ к функционалу веб-сервиса должен осуществляться по средством авторизации пользователя (логин и пароль). Пароль в базе данных не должен храниться в открытом виде (должен быть зашифрован). На этапе создания пользователей предъявлять следующие требования к паролю: использовать парольную фразу длиной не менее 8 символов, пароль должны состоять из латинских букв нижнего и верхнего регистра с

вкраплениями специальных символов. Осуществить защиту от SQL-инъекций.

8) Для минимизации расходов при разработке и в дальнейшем сопровождении программного обеспечения необходимо использовать только свободное программное обеспечение.

9) Веб-сервис должен состоять из публичной и не публичной части. К публичной части относится:

- Страница «Вход» - форма входа в не публичную часть сервиса по средством ввода комбинации логина и пароля.

- Страница «Выход» - страница, информирующая пользователя об успешном выходе из не публичной части веб-сервиса.

- К не публичной части веб-сервиса относится:

- Страница «Анализ» - форма для поиска информации по сообществу. По вводу исходных данных (ссылка на сообщество, временной диапазон поиска) и нажатию на кнопку «поиск» осуществляется поиск информации, расчет данных и вывод отчетности пользователю с возможностью сохранения отчета или его экспорта в формате xlsx. Данная страница доступна только smm-специалистам.

- Страница «Отчёты» - страница, на которой выгружаются сохранённые пользователем отчеты по сообществам с возможностью их удаления и группировки по сообществам. Данная страница доступна только smm-специалистам.

- Страница «Пользователи» - страница, на которой выводится список пользователей веб-сервиса, с возможностью их удаления и группировки по ролям. Данная страница доступна только администраторам веб-сервиса.

- Страница «Добавить нового пользователя» - страница, на которой можно создать нового пользователя. по средством ввода в форму данных о новом пользователе (логин, пароль, электронная почта). Данная страница доступна только администраторам веб-сервиса.

- Страница «Профиль» страница, на которой выводится основные данные пользователя, логин, почта, тип доступа

(администратор или smm-специалист). Так же имеется возможность редактировать данные профиля (основные данные и пароль). Данная страница доступна всем авторизованным пользователям.

- Страница «Редактировать профиль» - страница с формой для редактирования основных данных профиля. Данная страница доступна всем авторизованным пользователям.

- Страница «Сменить пароль» - страница с формой изменения пароля профиля. Данная страница доступна всем авторизованным пользователям.

1.4 Выбор CASE-средств

CASE-средства используются на начальном этапе процесса разработки программного обеспечения. Данный инструмент позволяет автоматизировать процесс разработки программного обеспечения (ПО), а именно:

- выделить все необходимые бизнес-процессы, протекающие внутри агентства;
- проанализировать их;
- определив взаимосвязанные процессы;
- оптимизировать их инфраструктуру.

Основной польза использования CASE - средств при разработки программного обеспечения – это значительное сокращение временных и финансовых издержек, а также значительное повышение качества программного обеспечения на выходе.

Выбор CASE-средства будет для каждого конкретного случая будет индивидуален и зависеть от множества факторов:

- цель моделирования и анализа процессов, исходя из целей определится необходимый функционал CASE-средства. Помимо этого определяется уровень детализации моделей и отчетности;
- финансовые возможности предприятия. Стоимость использования CASE-средства должна быть экономически выгодна при разработки ПО;

- квалификация проектировщика. При выборе CASE-средства стоит учитывать сколько времени потребуется на обучение персонала;
- удобство использования. CASE-средства должны быть визуально и интуитивно понятны пользователям, для того чтобы разработчику приходилось как можно меньше затрачивать усилий а разработку;
- использование стандартных методологий. Стандартизация упрощает взаимодействие CASE-средств с другими ИС.

Для выбора наиболее лучшего CASE-средства для разработки сервиса было выбрано 4 популярных ПО. Сравнительная таблица их характеристик представлена в таблице 1.

Таблица 1.2 - CASE-средства

Средства	Удобство	Порог вхождения	Стоимость
1	2	3	4
MS Project	+	Средний	650 рублей в месяц
IBM Rational Architect	-	Сложный	2 580 рублей
draw.io	+	Легкий	Бесплатно
Umbrello	-	Средний	Бесплатно

Исходя из всех критериев мною был выбран онлайн сервис для разработки UML-диаграмм draw.io в пользу удобства использования сервиса, доступа онлайн на любом носителе и отсутствием платы за использование.

1.5 Выбор инструментария разработки

Так как развитие и поддержку сервиса после его передачи клиенту могут выполнять сторонние разработчики, необходимо понять какой язык программирования актуален в данный момент, степень его поддержки в сообществах, наличие сторонних библиотек и множество других параметров.

Для выбора языка программирования обратимся к статистике компании TIOBE Software, которая каждый месяц изучает популярность языков программирования в мире и составляет рейтинг — индекс TIOBE. На рисунке 1.1 представлен рейтинг 10 языков программирования.

May 2020	May 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	17.07%	+2.82%
2	1	▼	Java	16.28%	+0.28%
3	4	▲	Python	9.12%	+1.29%
4	3	▼	C++	6.13%	-1.97%
5	6	▲	C#	4.29%	+0.30%
6	5	▼	Visual Basic	4.18%	-1.01%
7	7		JavaScript	2.68%	-0.01%
8	9	▲	PHP	2.49%	-0.00%
9	8	▼	SQL	2.09%	-0.47%
10	21	▲▲	R	1.85%	+0.90%
11	18	▲▲	Swift	1.79%	+0.64%
12	19	▲▲	Go	1.27%	+0.15%
13	14	▲	MATLAB	1.17%	-0.20%
14	10	▼▼	Assembly language	1.12%	-0.69%
15	15		Ruby	1.02%	-0.32%

Рисунок 1.1 - Рейтинг языков программирования.

Так как сервис будет реализован виде веб-сервиса то для его разработки необходим серверный язык программирования. Серверный язык программирования позволяет динамично отображать различные данные на сайте по мере востребованности конкретного пользователя. Схема получения информации динамических сайтов продемонстрирована на рисунке 1.2.

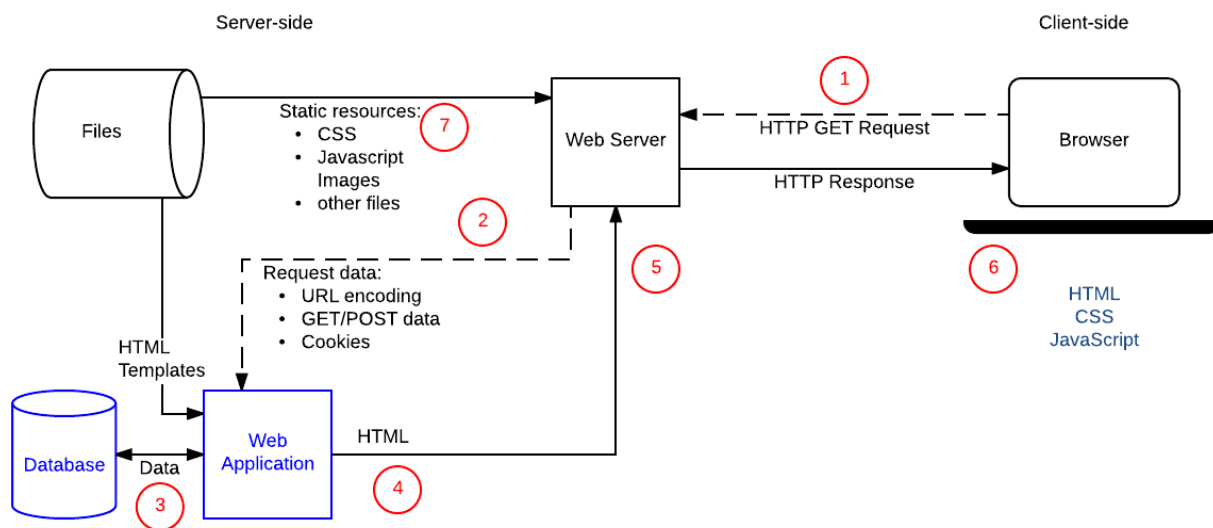


Рисунок 1.2 - Схема запросов динамических сайтов

Данные, для отображения информации, в основном подгружаются из базы данных, располагающейся на сервере и отправляемые клиенту для отображения через некоторый код (например, HTML и JavaScript). Помимо этого, в функционал серверных языков программирования входит:

- обработка информации происходит на стороне сервера;
- все сведения сохраняются в базе данных сервера и удобно сортируются в ней;
- результат обработки не зависит от того, каким браузером пользуется пользователь;
- повышается безопасность обработки данных, так как у пользователя или возможного нарушителя нет прямого доступа к обработчику информации.

Из представленного списка подходит 4 языка программирования. Анализ выбранных языков программирования представлен в таблице 3.

Таблица 1.3 - Анализ языков программирования

Язык	Быстродейст	Порог	Кроссплатформен	Распространенност
------	-------------	-------	-----------------	-------------------

и	вие	вхожден	ность	ь сторонних
1	2	3	4	5
C#	+	-	-	+
php	-	+	+	+
pyth on	+	+	+	+
java	-	-	+	+

В техническом плане для реализации проекта нет каких-либо ограничений по выбору языка программирования, так как практически любой функционал сервиса может быть реализован по средством выбранных 4 языков программирования. Однако с точки зрения экономики наиболее предпочтительным оказывается язык программирования - Python.

Python - это высокоуровневый интерпретируемый язык программирования с широкими возможностями использования. Язык ориентирован на читабельности компьютерного кода и на повышение производительности разработчика. Язык имеет большую поддержку в профессиональном сообществе, огромное количество готовых сторонних библиотек и низкий порог вхождения.

Так же для повышения скорости написания сервиса необходим веб-фреймворк. Фреймворк - это готовая веб-оболочка, облегчающая разработку и объединение разных компонентов большого программного проекта.

Таблица 1.4 - Анализ фреймворков для языка программирования Python

Фреймворки	Порог вхождения	Поддержка в профессионально м сообществе	Распространенность сторонних библиотек
1	2	3	4
Django	+	+	+
Flask	+	-	+

Web2py	+	+	-
--------	---	---	---

Проанализировав существующие решения, выбор пал на фреймворк Django. Django самый популярный фреймворк Python, что в свою очередь гарантирует поддержка в профессиональном сообществе и огромное количество готовых сторонних библиотек для него.

Помимо этого, для проектирования интерфейса веб-сервиса необходимо выбрать инструмент прототипирования. Существующие решения на рынке представлены в таблице 1.5.

Таблица 1.5 - Анализ инструментов прототипирования интерфейсов

Наименование	Платформа	Стоимость
1	2	3
InVision	Браузер	1 проект бесплатно Тарифы от 15 \$
Marvelapp	Браузер	2 проекта бесплатно Тарифы от 12 \$
Framer	Mac OS, Браузер (beta)	Тарифы от 12 \$
Principle	Mac OS	129 \$
Figma	Mac OS, Windows, Браузер	3 проекта бесплатно Тарифы от 12 \$

Функционал представленных в таблице 4 инструментов в общей своей массе схож и подходит для решения поставленной задачи. Проанализировав все инструменты, выбор пал на инструмент «Figma» так как он кроссплатформенный, бесплатен и обладает всем необходимым функционалом.

В качестве IDE был выбран PyCharm. Данная интегрированная среда разработки кроссплатформенна, имеет в своем функционале графический отладчик кода, средства анализа, инструменты для юнит-тестов, поддерживает разработку в веб и Django. Имеет бесплатный тарифный план для студентов.

1.6 Проектирование USE-CASE диаграмм

1.6.1 Разработка UML - диаграммы

Аббревиатура UML (Unified Modeling Language) – переводится как унифицированный язык моделирования. Эта система диаграмм используется для объектно-ориентированного анализа и проектирования в области разработки программного обеспечения. С успехом используется для:

- составления спецификации;
- визуализации работы ПО;
- моделирование бизнес-процессов;
- отображение организационных структур;
- конструирования и документирования программных систем.

С помощью языка графического описания (UML) мы построим диаграмму использования разрабатываемого сервиса.

Диаграмма использования необходима для описания функционала и поведения разрабатываемого продукта. Помимо этого, позволяет разработчику в доступной форме обсудить разрабатываемую систему с заказчиком (конечным пользователем сервиса).

На диаграмме использования отображаются:

- акторы — стилизованный человечек, обозначающий набор ролей пользователя, взаимодействующих с разрабатываемым сервисом;
- прецеденты — эллипс с надписью, обозначающий различные варианты использования сервиса, доступные акторам;
- комментарии;
- отношения между элементами диаграммы (бывают расширения (extend), либо включения (include)).

Перед началом построения диаграммы использования необходимо выполнить следующие:

- выделим круг лиц, взаимодействующих с сервисом;
- выделить как можно больше процессов, которые могут выполнять пользователи, но не следует слишком дробить процессы, а выделить лишь те, от которых пользователь получит значимый результат;
- добавить к прецедентам словесное описание.

Согласно составленному техническому заданию, была спроектирована диаграммы использования (рисунок 1.3-1.5).

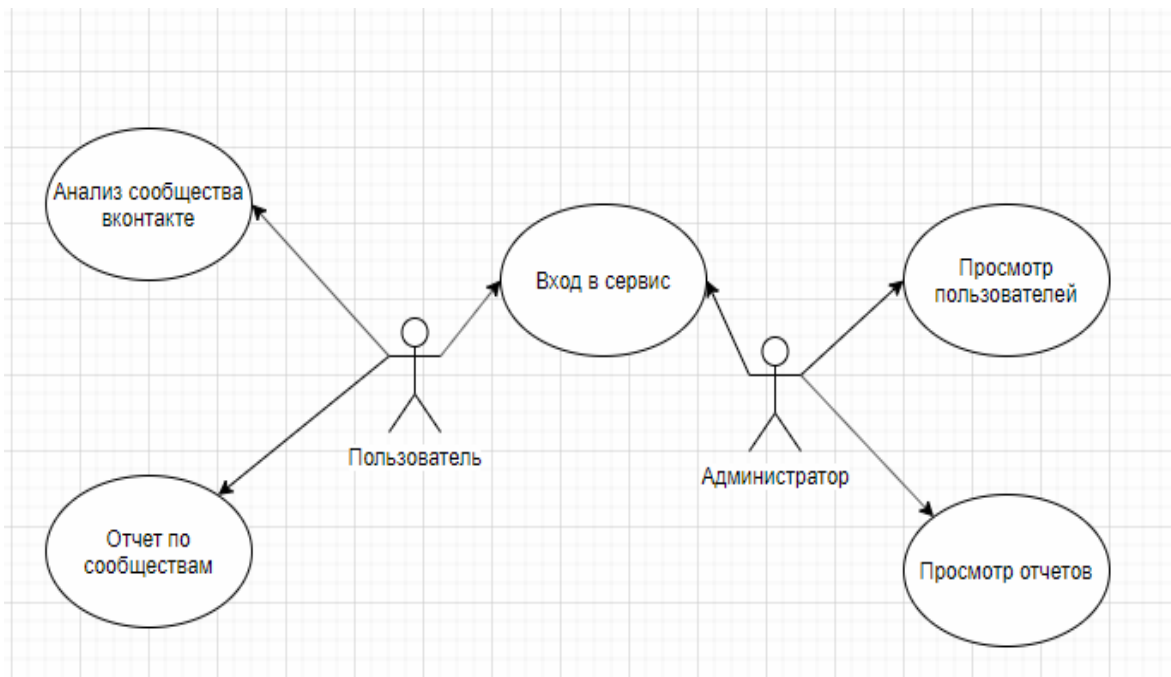


Рисунок 1.3 – общая диаграмма использования

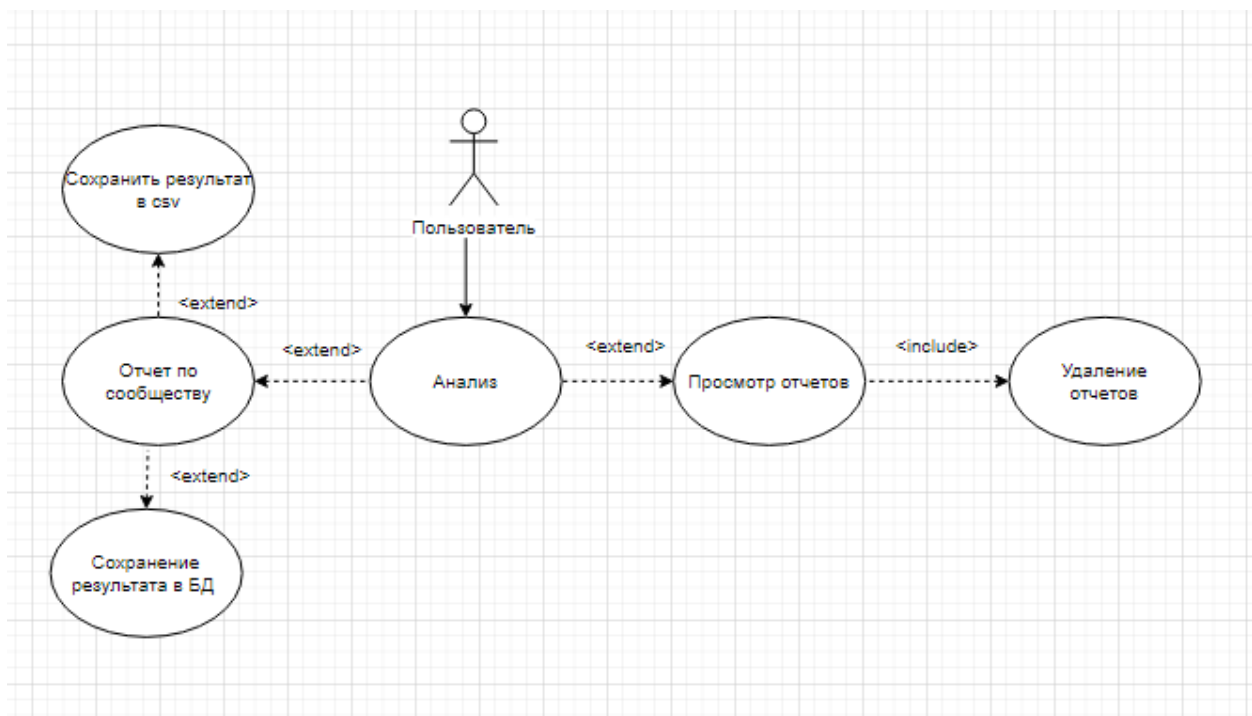


Рисунок 1.4 – диаграмма использования для роли «Пользователь»

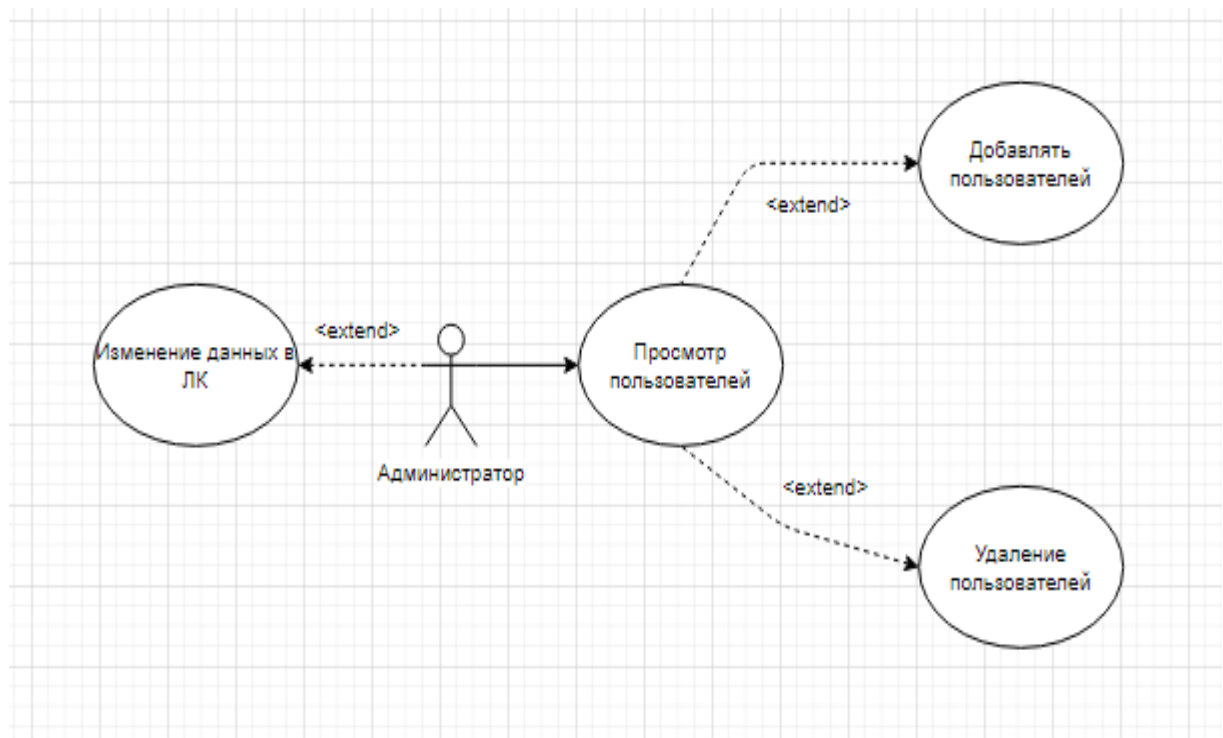


Рисунок 1.5 – диаграмма использования для роли «Администратор»

1.6.2 Разработка диаграммы деятельности

Следующим шагом проектирования является разработка диаграммы деятельности. Диаграмма деятельности наглядно отражает аспекты поведения системы.

Данный вид диаграмм схож с блок-схемами и наглядно демонстрирует как поток управления переходит от одного процесса к другому.

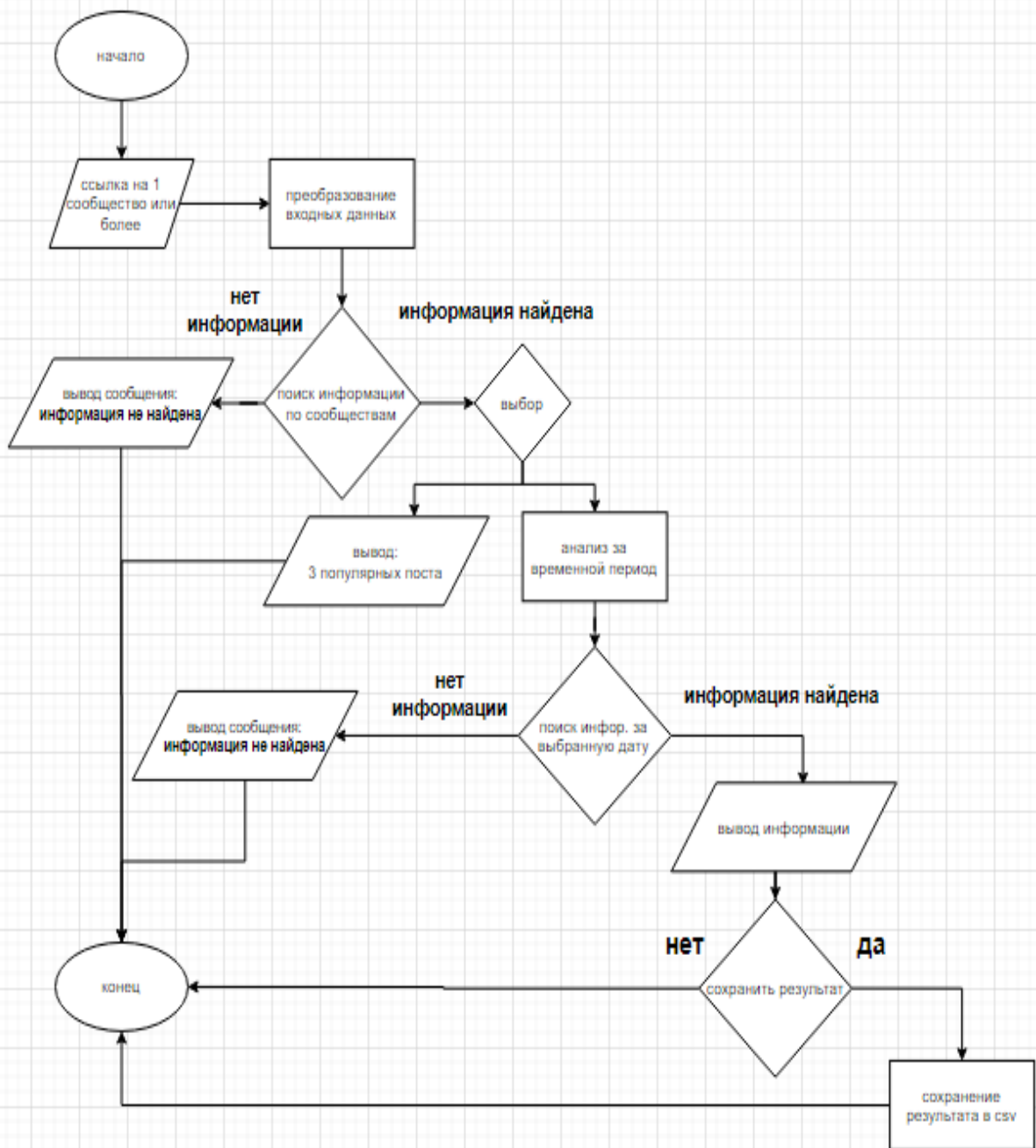


Рисунок 1.6 - Диаграмма деятельности пользователя «Подробный анализ сообщества»

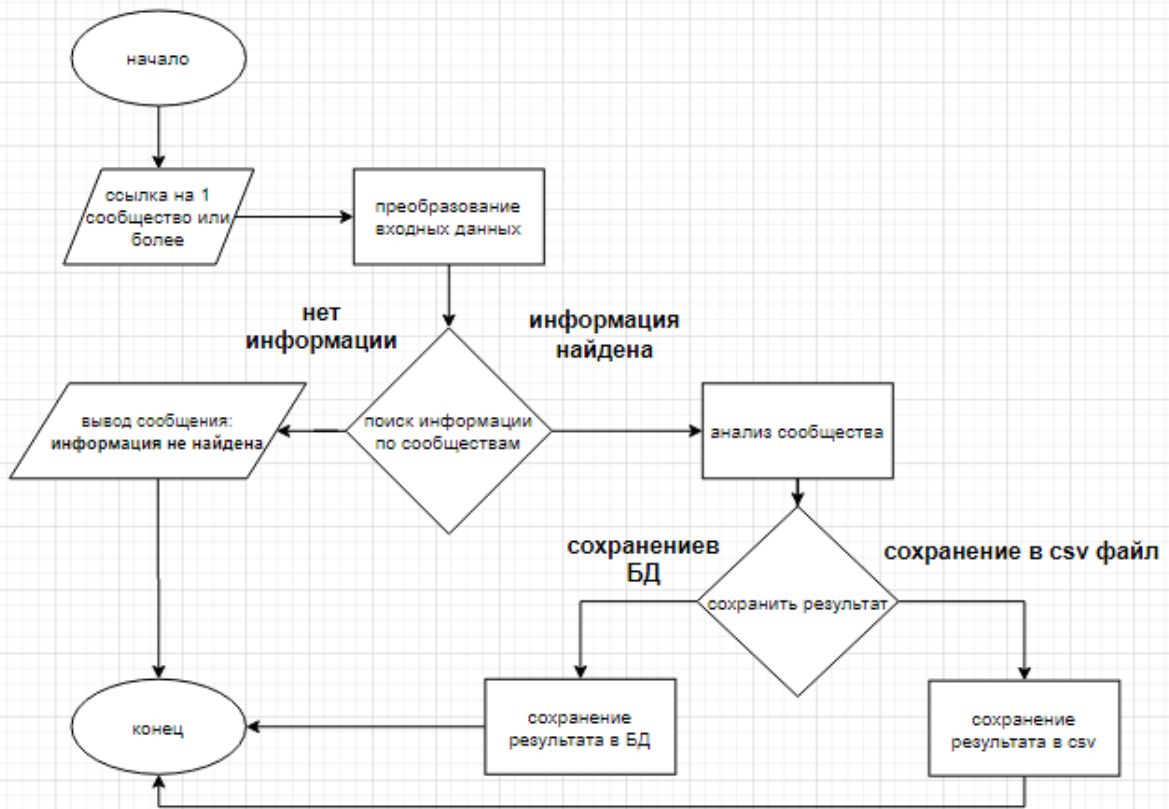


Рисунок 1.7 - Диаграмма деятельности пользователя «Общий анализ сообщества»

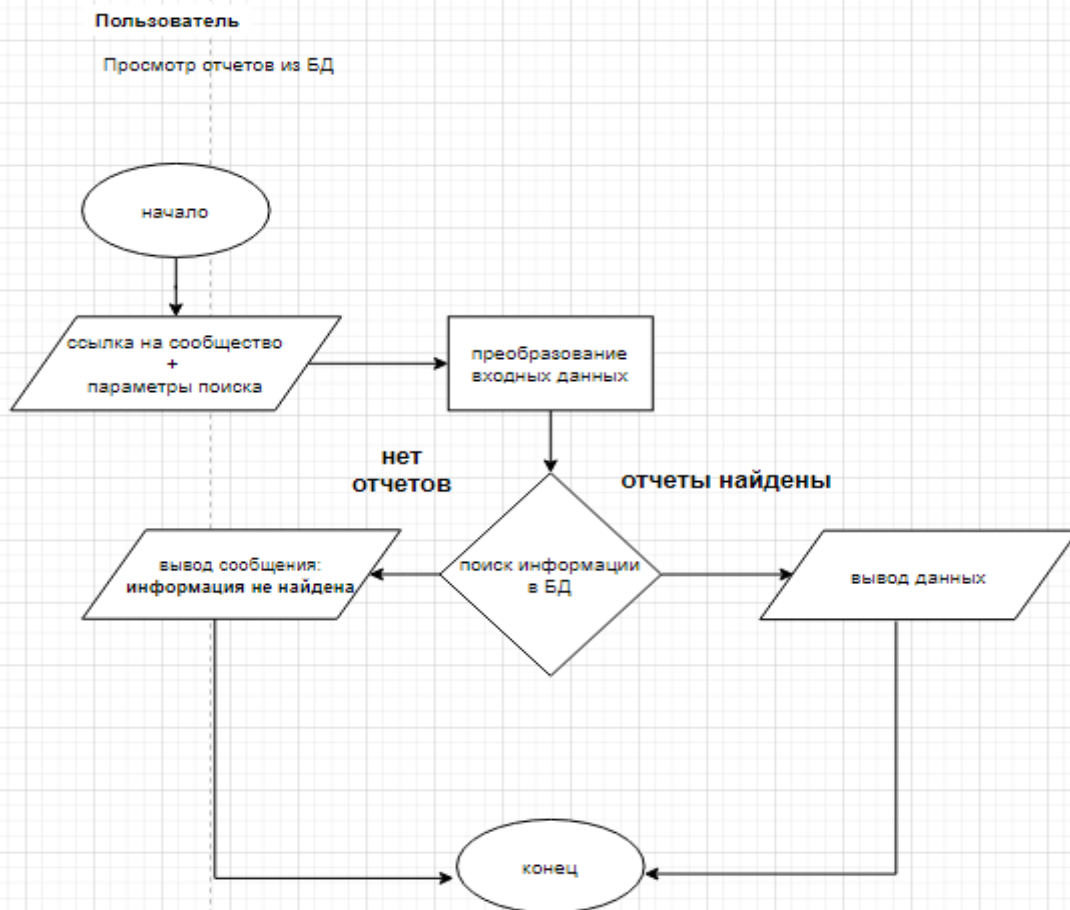


Рисунок 1.9 - Диаграмма деятельности пользователя «Просмотр отчетов из БД»

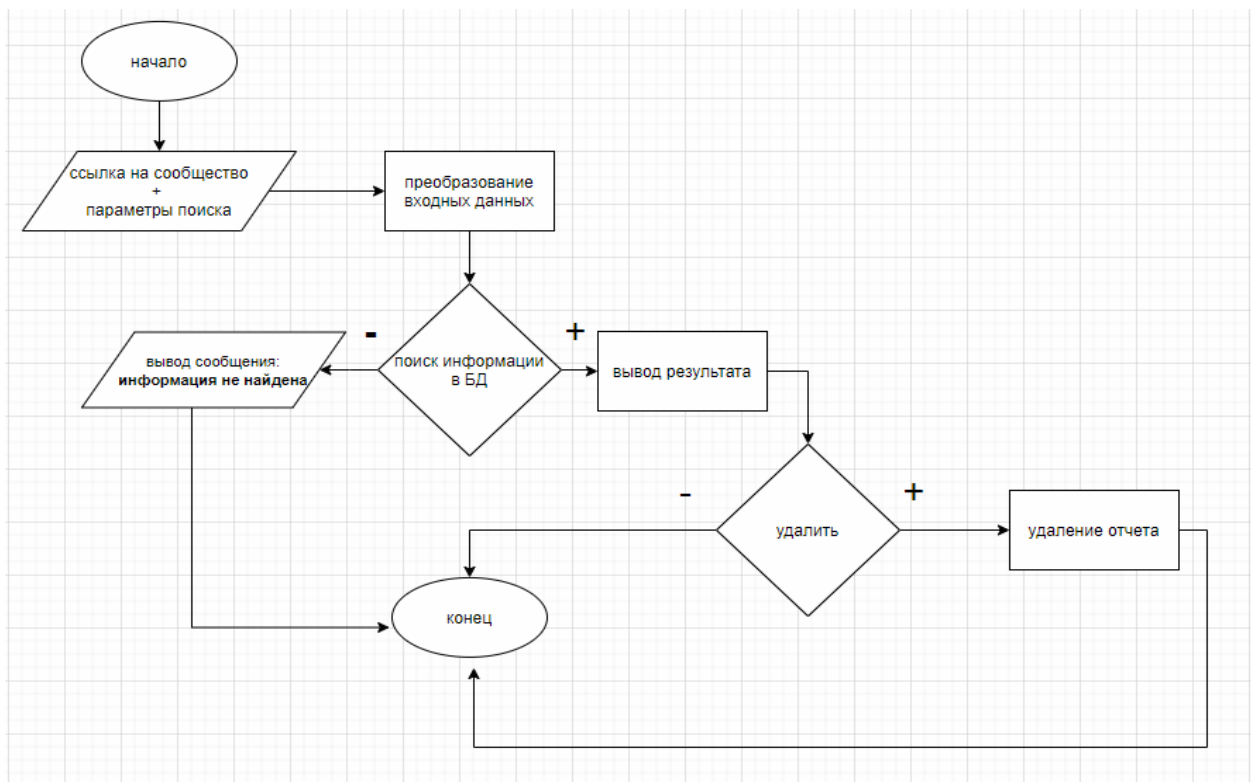


Рисунок 1.10 - Диаграмма деятельности администратора «Просмотр и удаление отчетов из БД»

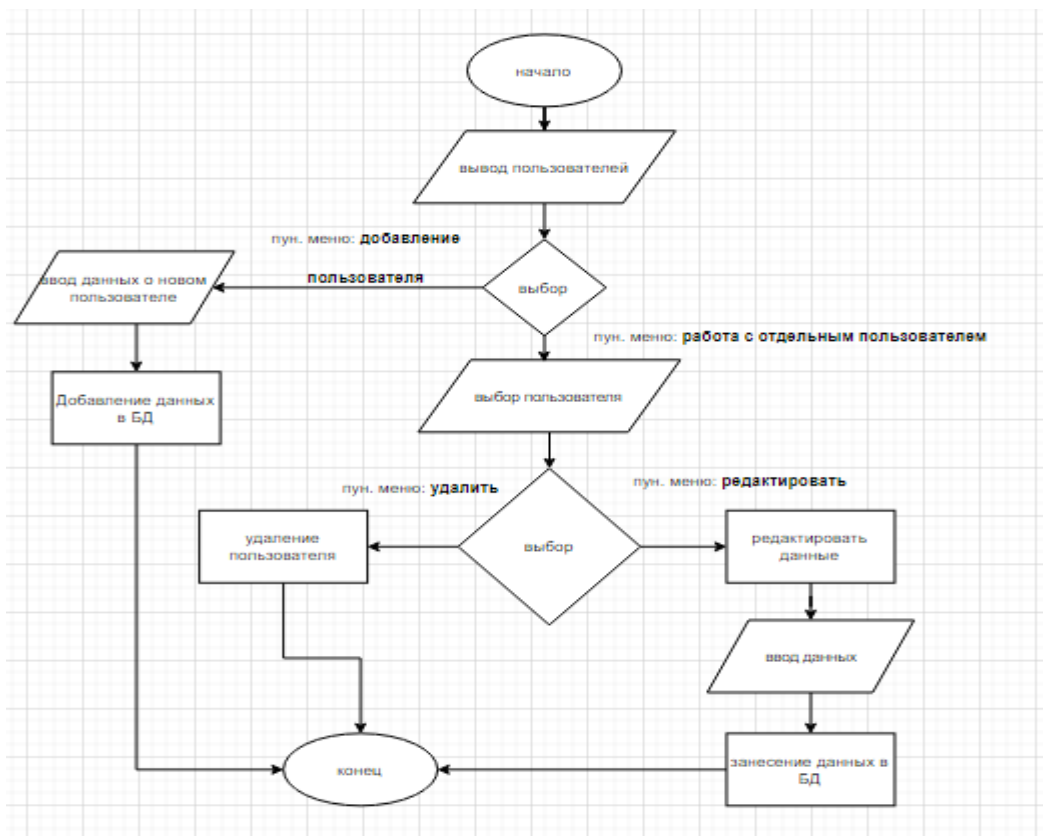


Рисунок 1.11 - Диаграмма деятельности администратора «Добавление новых пользователей, редактирование данных и удаление пользователей»

1.6.3 Разработка диаграммы классов

Диаграммы классов – это структурная диаграмма, показывают отношения между классами, набор классов и интерфейсов взаимодействия. Диаграммы классов проектируются на логическом уровне и служат для:

- демонстрации архитектуры разрабатываемого программного продукта;
- моделирования логики работы и обработки данных;
- проектирования логики навигации экранов, отображая пограничные классы и их логические взаимосвязи.

Диаграммы классов для разрабатываемого веб-сервиса продемонстрированы на рисунках 1.12-1.13.

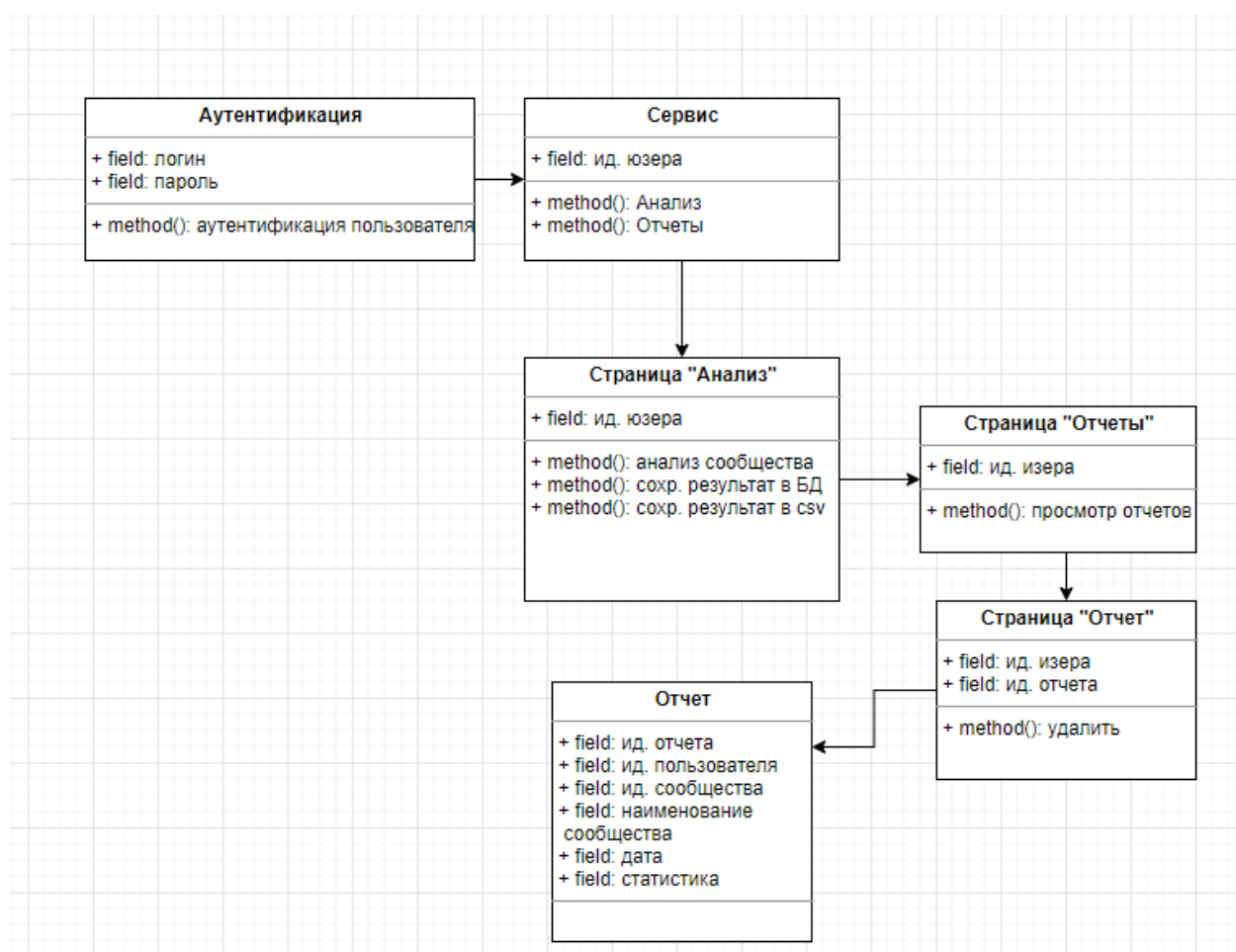


Рисунок 1.12 - Диаграмма классов «Пользователь»

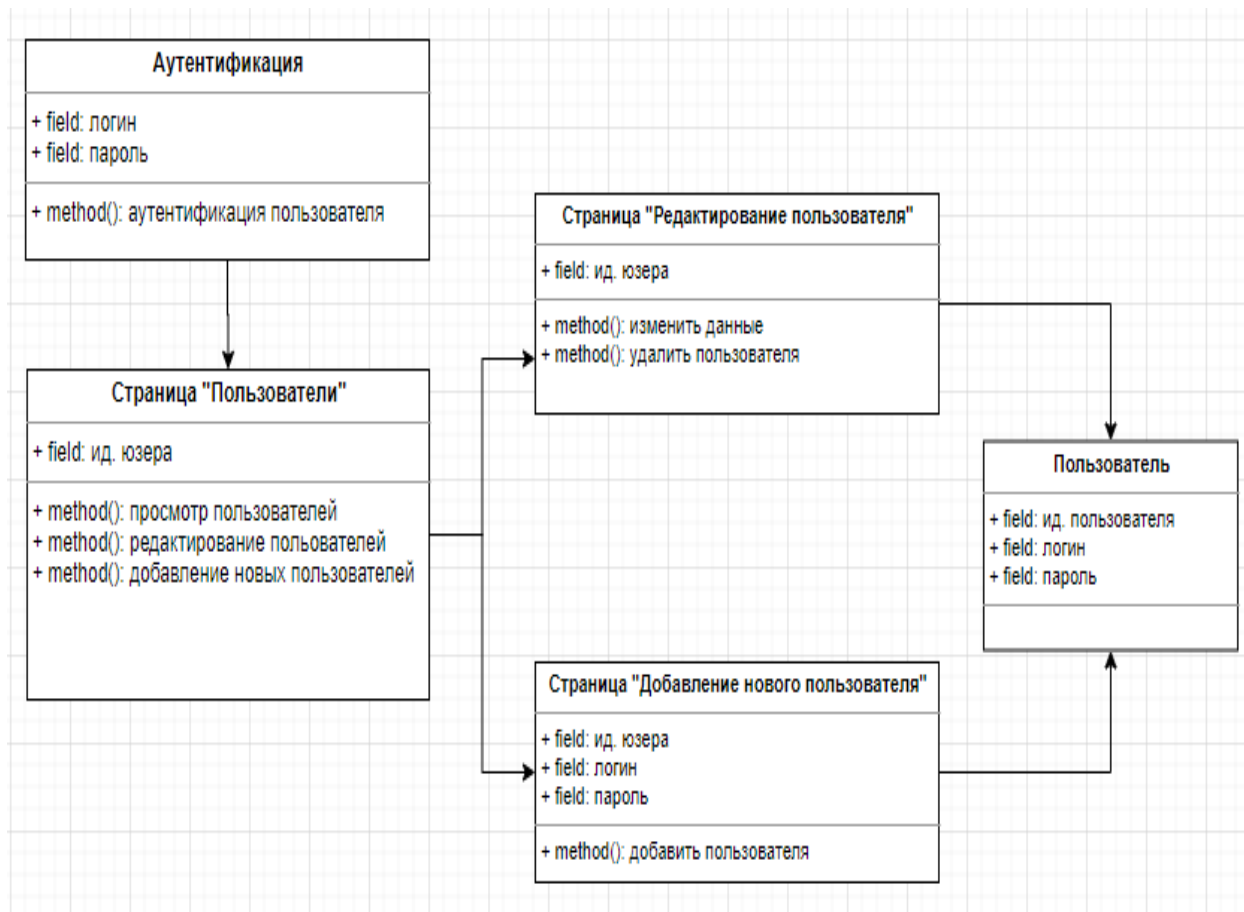


Рисунок 1.13 - Диаграмма классов «Администратор»

1.6.4 Разработка диаграммы последовательностей

Для уточнения диаграммы классов построим диаграмму последовательностей. Данный вид диаграмм представляют собой более детальное описание логики сценариев использования веб-сервиса. Иными словами, отражают поток событий, происходящих в рамках варианта использования.

Диаграмма последовательностей для пользователя веб-сервиса продемонстрирован на рисунке 1.14.

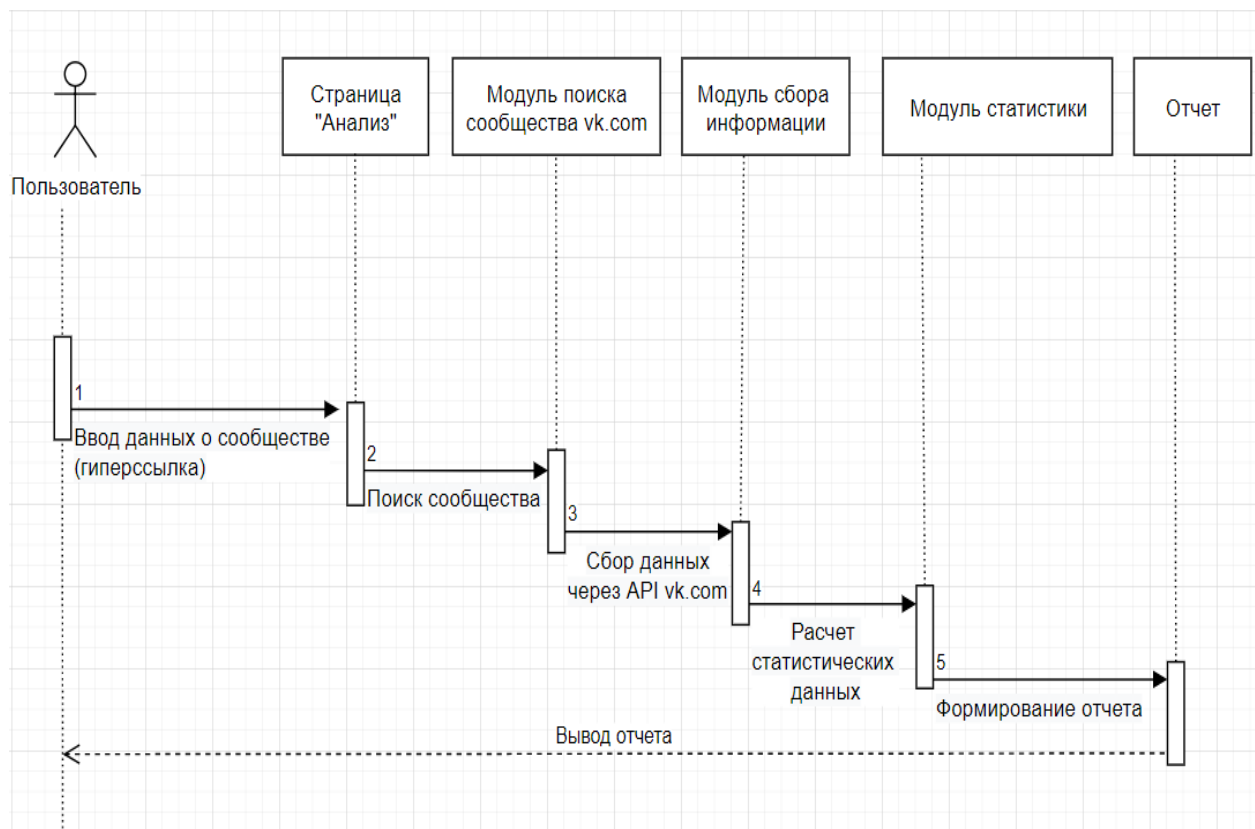


Рисунок 1.14 - Диаграмма последовательности «Пользователь»

На данной диаграмме отображен процесс анализа сообщества в vk.com. Описание по шагам:

1. Пользователь сервиса посещает страницу «Анализ», далее заполняет поля формы, где указывает ссылку на сообщество и временной диапазон поиска и посылает запрос на поиск.

2. Далее модуль поиска сообществ анализирует входные данные и проверяет существует ли сообщество. Если сообщество не найдено, пользователь получает уведомление о этом.

3. Если сообщество найдено, то модуль поиска информации проверяет существуют ли публикации за указанный период времени. Если публикаций не найдены, пользователь получает уведомление о этом.

4. Далее модуль статистики собирает статистические данные по сообществу и просчитывает маркетинговые данные.

5. На последнем этапе формируется отчет по сообществу.

Диаграмма последовательностей для администратора веб-сервиса продемонстрирован на рисунке 1.15.

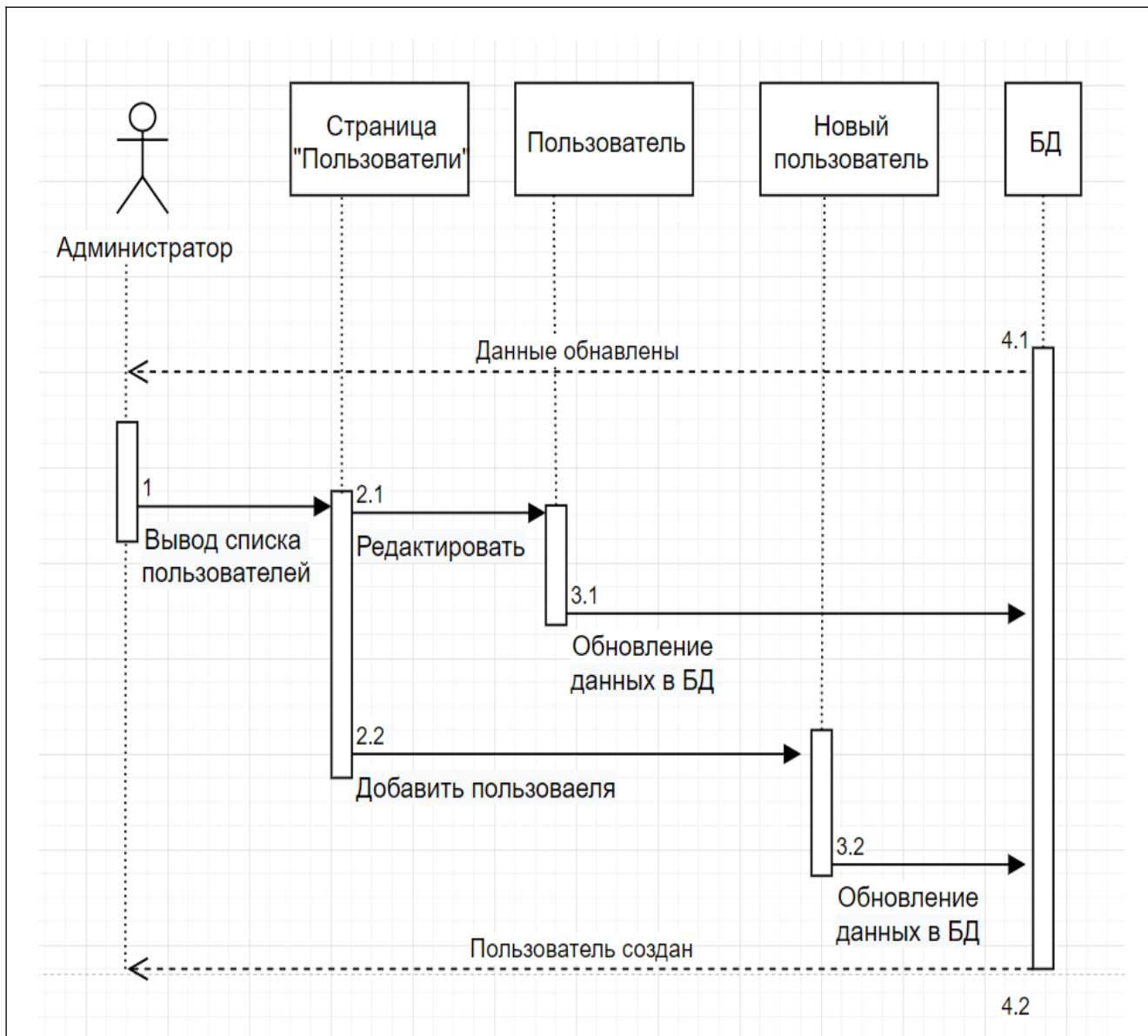


Рисунок 1.15 - Диаграмма последовательности «Администратор»

На данной диаграмме отображен процесс администрирования веб-сервиса. Описание по шагам:

1. Администратор посещает страницу пользователя. На странице выводится список имеющихся пользователей сервиса и возможность редактировать их.

2. По нажатию на кнопку «редактировать», рядом с выбранным пользователем, администратор перенаправляется на страницу с формой редактирования данных.

3. После изменения данных данные обновляются в БД и администратор перенаправляется на страницу со списком пользователей.

4. По нажатию кнопки «добавить нового пользователя», администратор перенаправляется на страницу с формой заполнения

данных о новом пользователе. По нажатию кнопки «добавить» данные обновляются в БД.

1.6.5 Разработка диаграммы компонентов

Далее перейдем к финальному этапу проектирования веб-сервиса: разработке диаграммы компонентов.

Спроектированные ранее диаграммы были концептуальны, отражали общее понимание физической системы веб-сервиса и относились к логическому уровню построения.

Для физического отражения веб сервиса используются диаграммы компонентов. Основная функция данного типа диаграмм состоит в определении зависимости между программными компонентами, в роли которых выступают: базы данных, фреймворки, API и исходные коды программ.

Диаграмма компонентов разрабатывается для:

- отображения физических и концептуальных схем БД;
- визуальное отображение структуры веб-сервиса;
- многократное использование фрагментов программного кода.

Диаграмма компонентов для веб-сервиса представлена на рисунке 1.16.

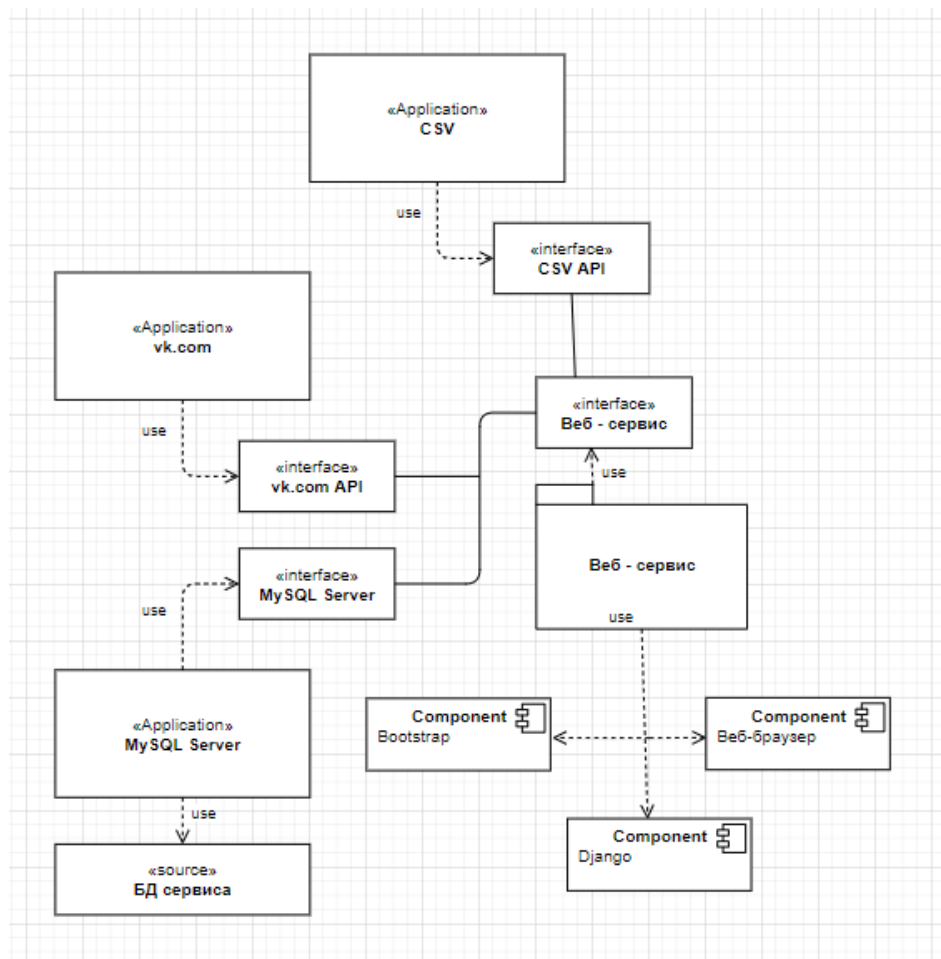


Рисунок 1.16 - Диаграмма компонентов

Веб-сервис в процессе своей работы использует следующие компоненты:

- фреймворк Django - веб-оболочка сервиса, в которой описана логику работы с базой данных (model), прописаны обработчики запросов (control), генерирует шаблоны веб-страниц (view). По мимо этого Django осуществляет маршрутизацию, т.е. связывает веб странички.

- фреймворк Bootstrap - необходим для быстрой вёрстки адаптивного дизайна веб-страниц.

- веб-браузер - инструмент-посредник, осуществляющий доступ к сервису.

- Источником информации для анализа социальных сообществ является веб-ресурс vk.com. Доступ к информации осуществляется через API веб-ресурса.

Для хранения информации об пользователях и отчетах веб-сервис использует СУБД MySQL.

Генерация отчетов в формат `xlsx` происходит с использованием модуля `xlsxwrite`.

1.7 Разработка прототипа интерфейса

Далее в инструменте прототипирования интерфейсов «Figma» спроектируем основные страницы веб-сервиса.

Страница анализа сообществ представлена на рисунке 1.17.

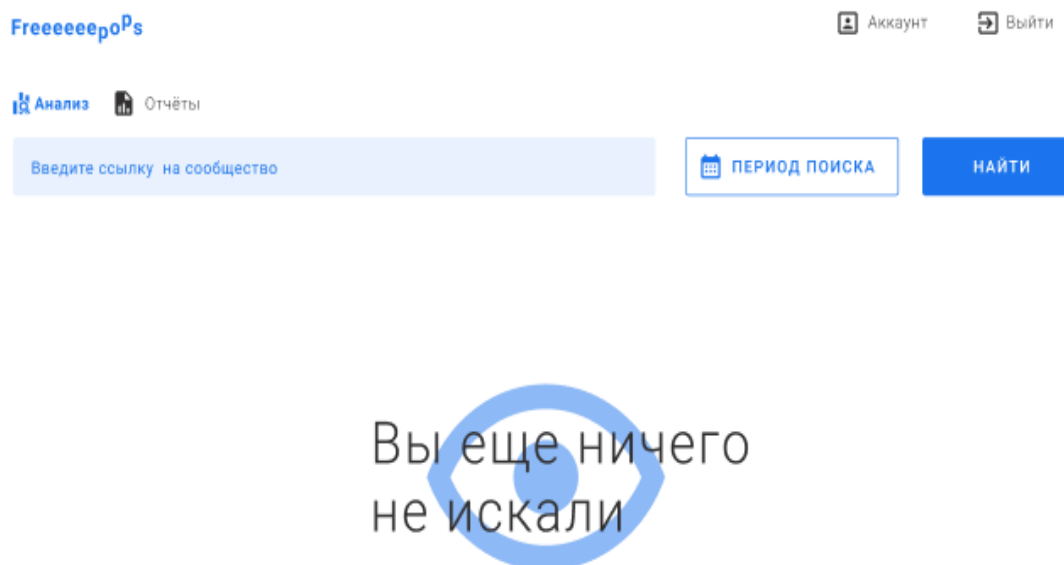


Рисунок 1.17 – прототип страницы «Анализ»

На данной странице имеется форма для ввода ссылки на сообщество и выбора временного диапазона поиска информации.

Страница с отчетом по сообществу представлена на рисунке 1.18.

https://vk.com/puplic1244423

ПЕРИОД ПОИСКА

НАЙТИ

Сообщество: МСО Панацея

Период: с 2020-05-01 по 2020-05-06

3 48 112

52 1205 10

ERPOST 9.444%

ERDAY 9.444%

ERVIEW 9.444%

Сохранить отчет

Скачать отчет

ЛАЙКИ

КОММЕНТАРИИ

ПРОСМОТРЫ

ДАТА

РЕПОСТЫ

ERPOST

МСО Панацея
1 мая 2020 в 10:30

10 0 108 1

ERPOST 9.444% ERVIEW 9.444%

Дорогие друзья, с праздником Первомай вас! 🌸

Этот весенний праздник наполнен солнечным светом и теплом. Желаем, чтобы работа всегда приносила вам удовольствие и возможности для самореализации. 🌸

Пусть в каждом доме царят счастье, мир и благополучие! 🌸
Ваша МСО «Панацея»!
#Панацея
#полимОМС #страховойполис
#медицинскийполис #1мая #ростов #май

С Праздником Весны и Труда!

1 МАЯ

ПЕРЕЙТИК ЗАПИСЕ

МСО Панацея
1 мая 2020 в 10:30

С Праздником

Рисунок 1.18 – прототип страницы «Отчет по сообществу»

На данной странице отображается статистическая информация по сообществу, а именно:

- количество участников, лайков, комментариев, репостов и т.д.;
- маркетинговые расчеты (ERday, ERpost, ERview);
- до пяти лучших постов за временной период по категориям (лайки, комментарии, просмотры, репосты, ERpost).

Помимо этого, пользователь может сохранить или скачать отчет.

Сообщества

Ростов Главный

МСО Панацея

Ростов Главный

Дата поиска							ERPOST	ERDAY	ERVIEW	Диапазон	Удалить
4 июня 2020 г.	52	23	12	5433	8	341	12.4	12.4	12.4	2020-06-01 - 2020-06-04	X
4 июня 2020 г.	52	23	12	5433	8	341	12.4	12.4	12.4	2020-06-01 - 2020-06-04	X
4 июня 2020 г.	52	23	12	5433	8	341	12.4	12.4	12.4	2020-06-01 - 2020-06-04	X
4 июня 2020 г.	52	23	12	5433	8	341	12.4	12.4	12.4	2020-06-01 - 2020-06-04	X
4 июня 2020 г.	52	23	12	5433	8	341	12.4	12.4	12.4	2020-06-01 - 2020-06-04	X
4 июня 2020 г.	52	23	12	5433	8	341	12.4	12.4	12.4	2020-06-01 - 2020-06-04	X
4 июня 2020 г.	52	23	12	5433	8	341	12.4	12.4	12.4	2020-06-01 - 2020-06-04	X
4 июня 2020 г.	52	23	12	5433	8	341	12.4	12.4	12.4	2020-06-01 - 2020-06-04	X
4 июня 2020 г.	52	23	12	5433	8	341	12.4	12.4	12.4	2020-06-01 - 2020-06-04	X
4 июня 2020 г.	52	23	12	5433	8	341	12.4	12.4	12.4	2020-06-01 - 2020-06-04	X
4 июня 2020 г.	52	23	12	5433	8	341	12.4	12.4	12.4	2020-06-01 - 2020-06-04	X
4 июня 2020 г.	52	23	12	5433	8	341	12.4	12.4	12.4	2020-06-01 - 2020-06-04	X

Рисунок 1.19 – прототип страницы «Отчеты пользователя»

На данной странице выводится список сохранённых пользователем отчетов. Их можно удалить и отфильтровать по группам.

Мои данные

логин: admin

почта: mail@mail.com

Имя: Иван

Фамилия: Иванов

Редактировать

Сменить пароль

Рисунок 1.20 – прототип страницы «Личный кабинет пользователя»

Страница «Личный кабинет» содержит информацию о пользователе. На данной странице можно отредактировать данные и сменить пароль.

Пользователи [Добавить](#)

Пользователи

Администраторы

	Логин	Последних вход	Удалить
Администраторы			
SMM-специалисты	admin	05-05-20 08:46	X
	alex11	15-04-20 18:46	X
	admin	05-05-20 08:46	X
	alex11	15-04-20 18:46	X

Рисунок 1.21 - прототип страницы «Пользователи сервиса»

На данной странице выводится список пользователей сервиса. Имеются кнопки для фильтрации пользователей по ролям, а также их удаление.

Пользователи [Добавить](#)

Добавление нового пользователя

Логин:

Имя:

Фамилия:

Почта:

Пароль:

Повторите пароль:

 Администратор[Добавить](#)

Рисунок 1.21 - прототип страницы «Добавить нового пользователя»

Данная страница имеет форму добавления нового пользователя. К полю почта осуществляется проверка на ввод данных типа почта. К полю пароль идет проверка данных на ввод:

- длина пароля не менее 8 символов;
- парольная фраза состоит из цифр и символов.

По нажатию на чек-бокс «Администратор» пользователю назначается роль администратора сервиса. Если чек-бокс не активен, то пользователь будет иметь роль smm-специалиста.

1.8 Проектирование базы данных

Исходя из требований заказчика, минимизировать расходы по средством использования свободного программного обеспечения, было решено в качестве системы управления базой данных для веб-сервиса выбрать СУБД MariaDB. Данная СУБД распространяется на условиях общей лицензии GNU (GPL, GNU Public License), это существенно снизит затраты заказчика на приобретение дополнительного ПО.

Данная система - одна из самых популярных и самых распространенных, с открытым исходным кодом. Отличается хорошей скоростью работы, гибкостью и надежностью. MariaDB не предназначена для работы с большими объемами информации, но ее применение отлично подходит для интернет сайтов, что является одним из ключевых моментов в выборе данной СУБД.

На рисунке 1.22 представлена структура базы данных веб-сервиса.

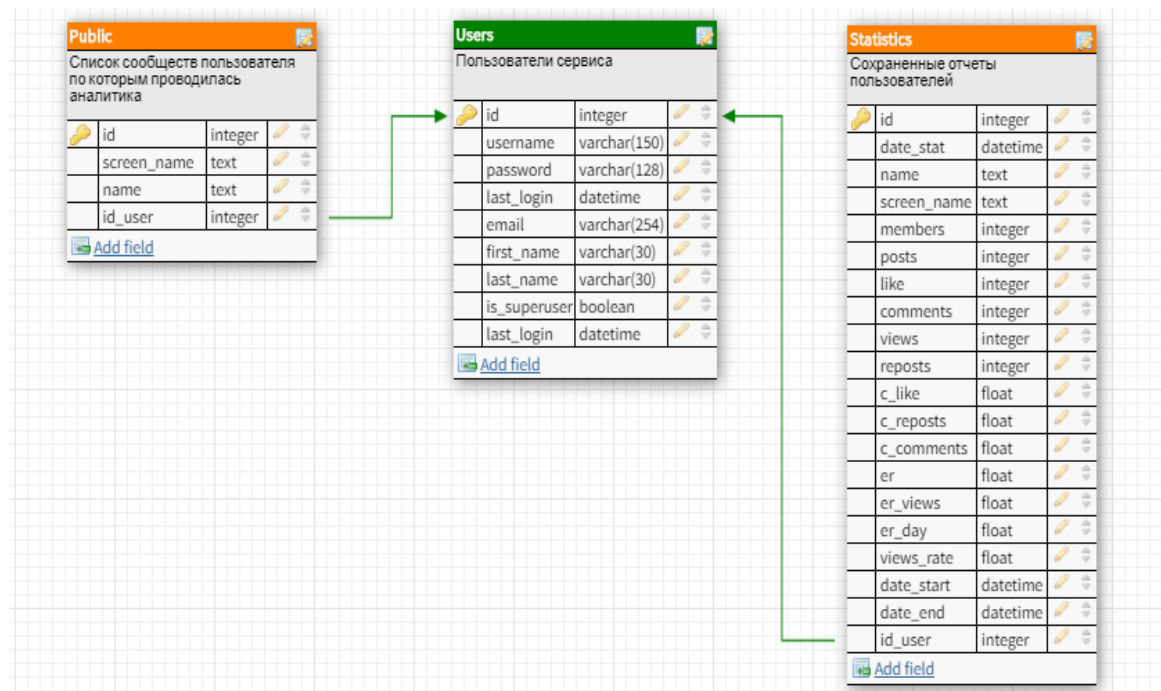


Рисунок 1.22 – структура базы данных

Таблица «Users» содержит информацию о пользователях веб-сервиса и имеет следующие поля:

- `id` - идентификационный номер пользователя; поле имеет тип `int`, первичный ключ;
- `username` - логин пользователя по которому осуществляется вход в систему, тип `varchar`;
- `password` - пароль пользователя, тип `varchar`;
- `last_login` - время и дата последнего входа в веб-сервис, тип `datetime`;
- `email` - электронная почта пользователя, тип `varchar`;
- `first_name` - имя пользователя, тип `varchar`;
- `last_name` - фамилия пользователя, тип `varchar`;
- `is_superuser` - флаг определения роли пользователя, тип `boolean`.

- Таблица «Public» содержит информацию о сообществах, по которым пользователь осуществлял сбор статистических данных. Таблица имеет следующие поля:

- `id` - идентификационный номер записи; поле имеет тип `int`, первичный ключ;
- `screen_name` - доменное имя сообщества, поле имеет тип `text`;
- `name` - наименование сообщества, поле имеет тип `text`;

- id_user - идентификационный номер пользователя, внешний ключ. Поле имеет тип int.

- Таблица «Statistics» содержит отчетную информацию о сообществах, по которым пользователь осуществлял сбор статистических данных и сохранил их в базе данных веб-сервиса. Таблица имеет следующие поля:

- id - идентификационный номер записи; поле имеет тип int, первичный ключ;

- date_stat - дата и время сохранения отчета, поле имеет тип datetime;

- screen_name - доменное имя сообщества, поле имеет тип text;

- name - наименование сообщества, поле имеет тип text;

- members - количество подписчиков сообщества, поле имеет тип int;

- posts - количество публикаций сообщества, поле имеет тип int;

- like - количество лайков публикации сообщества, поле имеет тип int;

- comments - количество комментариев публикации сообщества, поле имеет тип int;

- views - количество просмотров публикации сообщества, поле имеет тип int;

- reposts - количество репостов публикации сообщества, поле имеет тип int;

- c_like - среднее количество лайков публикации сообщества за период, поле имеет тип float;

- c_comments - среднее количество комментариев публикации сообщества за период, поле имеет тип float;

- c_reposts - среднее количество репостов публикации сообщества за период, поле имеет тип float;

- c_views - среднее количество просмотров публикации сообщества за период, поле имеет тип float;

- c_like - среднее количество публикаций сообщества за период, поле имеет тип float;

- er - показатель вовлеченности, поле имеет тип float;

- er_views - показатель вовлеченности по просмотрам, поле имеет тип float;

- er_day - показатель вовлеченности по дню, поле имеет тип float;
- views_rate - показатель вовлеченности по просмотрам за период, поле имеет тип float;
- date_start - дата начала поиска информации, поле имеет тип datetime;
- date_end - дата окончания поиска информации, поле имеет тип datetime;
- id_user - идентификационный номер пользователя, внешний ключ. Поле имеет тип int.

1.9 Календарный план разработки

Разработка календарного плана является неотъемлемой частью продуктивной разработки программного продукта. Календарное планирование заключается в разработке и последующем контроле исполнения расписания с учетом выделенных сроков и ресурсов. Работа по составлению календарного плана состоит из:

- 1) Определения перечня предстоящих работ;
- 2) Анализ ресурсной потребностей (время, знания, навыки);
- 3) Составление плана сроков выполнения работ;
- 4) Расчёт трудозатрат.

Календарный план разработки программного продукта представлен на рисунке 1.23.

	Цели	Начало	Конец	Интервал	Статус
1	Сбор исходных данных	11 05 20	11 05 20	0	Достигнута
	Анализ деятельности				
2	предприятия	12 05 20	13 05 20	1	Достигнута
3	Сбор требований для ТЗ	14 05 20	14 05 20	0	Достигнута
4	Составление ТЗ	15 05 20	15 05 20	0	Достигнута
	Выбор инструментариев				
5	разработки	16 05 20	17 05 20	1	Достигнута
6	Проектирование базы данных	18 05 20	18 05 20	0	Достигнута
	Проектирование use-case				
7	диаграмм и алгоритмов	19 05 20	20 05 20	1	Достигнута
	Прототипирование				
8	пользовательского интерфейса	21 05 20	21 05 20	0	Достигнута
9	Разработка ПП	24 05 20	11 06 20	18	Достигнута
10	Тестирование и отладка ПП	05 06 20	11 06 20	6	Достигнута
	Внедрение программного				
11	продукта	12 06 20	12 06 20	0	Достигнута

Рисунок 1.23 - календарный план разработки ПО

Диаграмма Ганта предназначена для иллюстрации разных этапов разработки программного продукта. Визуально представляет собой простой набор полосок, состоящих из двух главных осей: дел и времени. Каждому временному промежутку приписывается определённая задача, которая должна быть выполнена. Диаграмма Ганта работы над проектом представлена на рисунке 1.24.

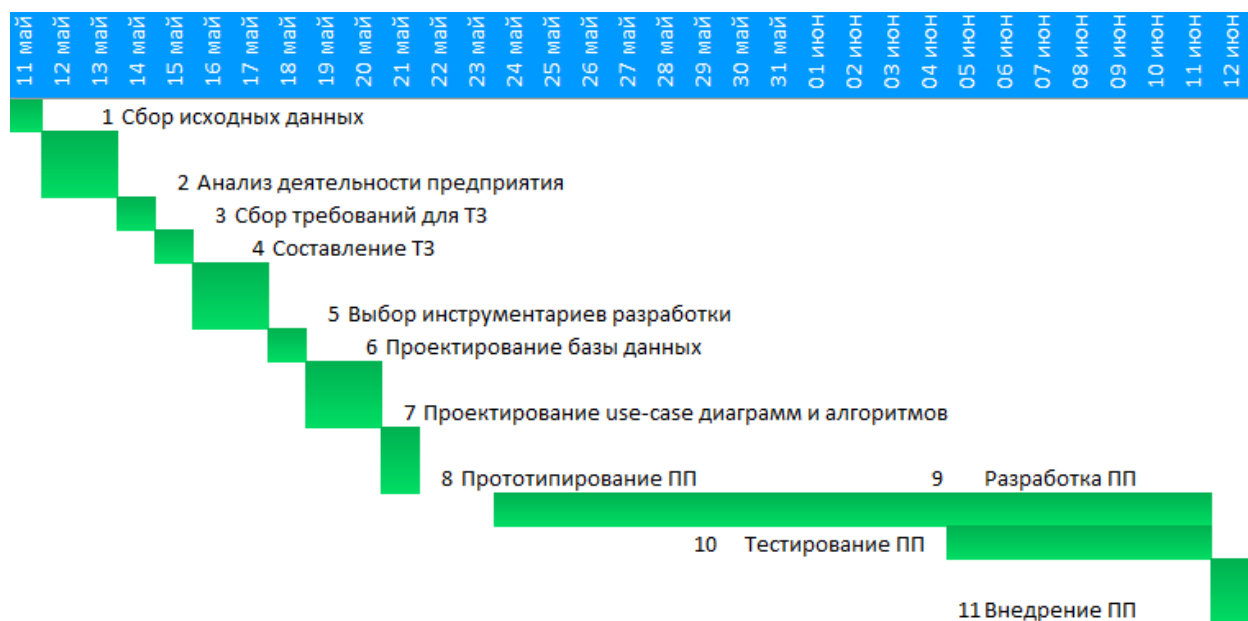


Рисунок 1.24 - Диаграмма Ганта

2 РАЗРАБОТКА ВЕБ-СЕРВИСА

Программный продукт (веб-сервис) состоит из следующих программных модулей:

1) Страница «Анализ» - осуществляет анализ публикаций сообщества vk.com, сохранение и экспорт результата. На данной странице имеется форма для ввода адреса сообщества и даты поиска информации. После успешного анализа сообщества пользователю выводится статистическая информация с возможностью сохранения и экспорта отчета.

2) Страница «Отчеты» - вывод отчетов пользователя, группировка по сообществам и возможность удаление отчетов.

3) Страница «Пользователи» - вывод пользователей сообщества, группировка по ролям (администраторы и smm-менеджеры) и возможность удаление пользователей;

4) Страница «Добавить пользователя» - добавление нового пользователя в веб-сервисе;

5) Страница «Профиль» - вывод личных данных пользователя с возможность их редактирования и изменения пароля;

6) Страница «Сменить пароль» - форма для смены парольной фразы;

7) Страница «Редактировать данные» - форма для редактирования данных профиля пользователя;

8) Страница «Вход» - осуществляет вход непубличную часть функционала веб-сервиса;

9) Страница «Выход» - осуществляет выход из непубличной части функционала веб-сервиса.

В веб-сервисе присутствуют 3 типа пользователей:

1) Неавторизованный пользователь - имеет доступ только к странице «Вход»;

2) SMM - менеджеры - пользователи имеют доступ к функционалу анализа сообществ и работе с сохранёнными отчетами;

3) Администраторы - пользователи имеют доступ к функционалу работы с пользователями (просмотр, создание и удаление пользователей).

2.1 Разработка модуля сбора и анализа информации

Данный модуль состоит из трех функциональных частей:

- 1) Сбор данных из открытых источников – VK API;
- 2) Расчёт статистических данных на основе собранных данных;
- 3) Вывод результата пользователю.

Алгоритм сбора информации будет собирать информацию об сообществе из открытых источников – VK API. Алгоритм работы по шагам:

1) Данные из формы проверяются на корректность. Ссылка, переданная через форму, проверяется на наличие в ней адреса на сообщество. Если ссылка не распознана, пользователю выведется сообщение с ошибкой.

2) Далее отправляется запрос на поиск сообщества. Если сообщество не найдено, пользователю выведется сообщение с ошибкой.

3) Если сообщество найдено, производится запрос на получение публикаций за временной период, заданный пользователем. Если за данный период не найдено данных, пользователю выведется сообщение с ошибкой.

4) Данные компонуется и передаются в работу обработчику подсчета статистики.

Алгоритм расчета статистической информации будет рассчитывать статистические данные по публикациям, полученные обработчика сбора данных. Алгоритм работы по шагам:

1) Массив данных передается из обработчика сбора информации.

2) Переменная $i = 0$, это счетчик цикла, len – количество элементов переданного массива.

3) Далее работает цикл пока len больше счетчика i . На каждой итерации работы цикл производится расчет статистических данных.

4) Данные передаются на вывод их пользователю.

Блок-схема работы алгоритмов продемонстрирована на рисунке 2.1-2.2.

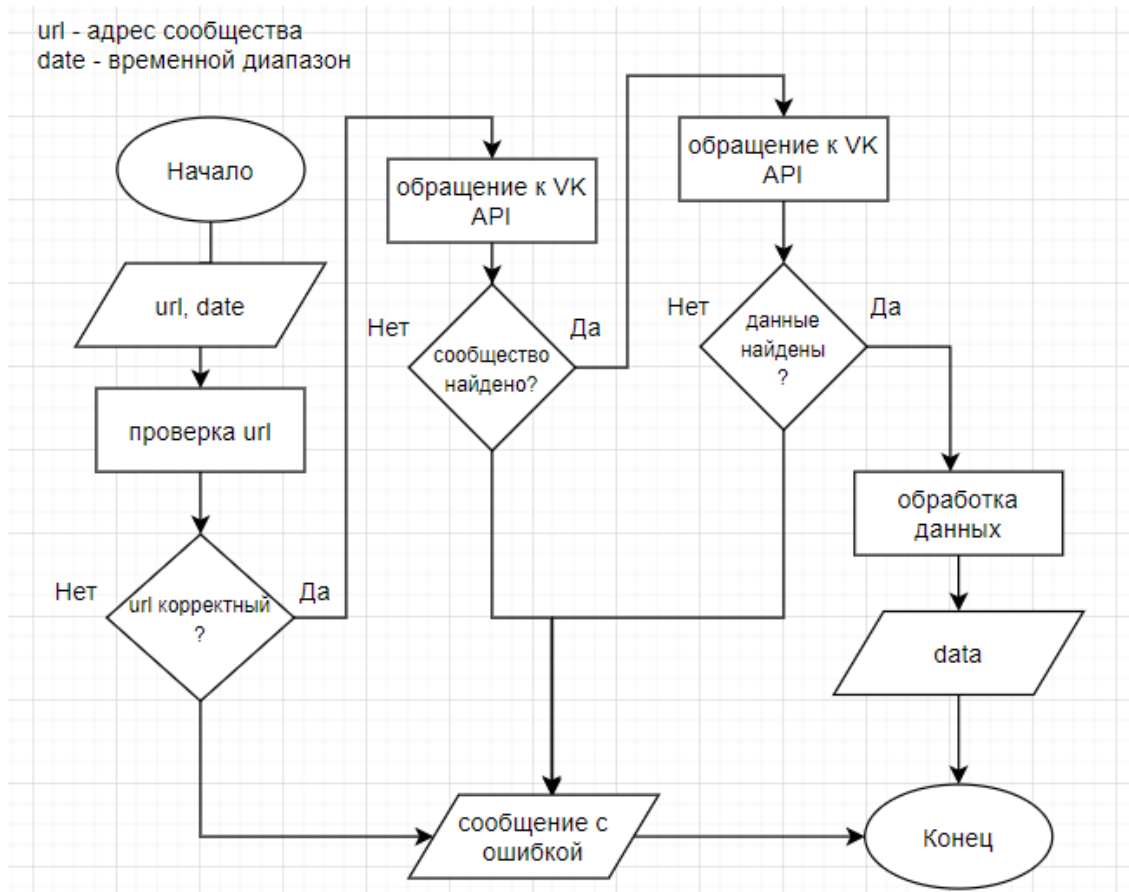


Рисунок 2.1 - блок-схема алгоритма сбора информации

posts - массив данных публикаций
len - количество элементов в массиве
data - массив статистических данных

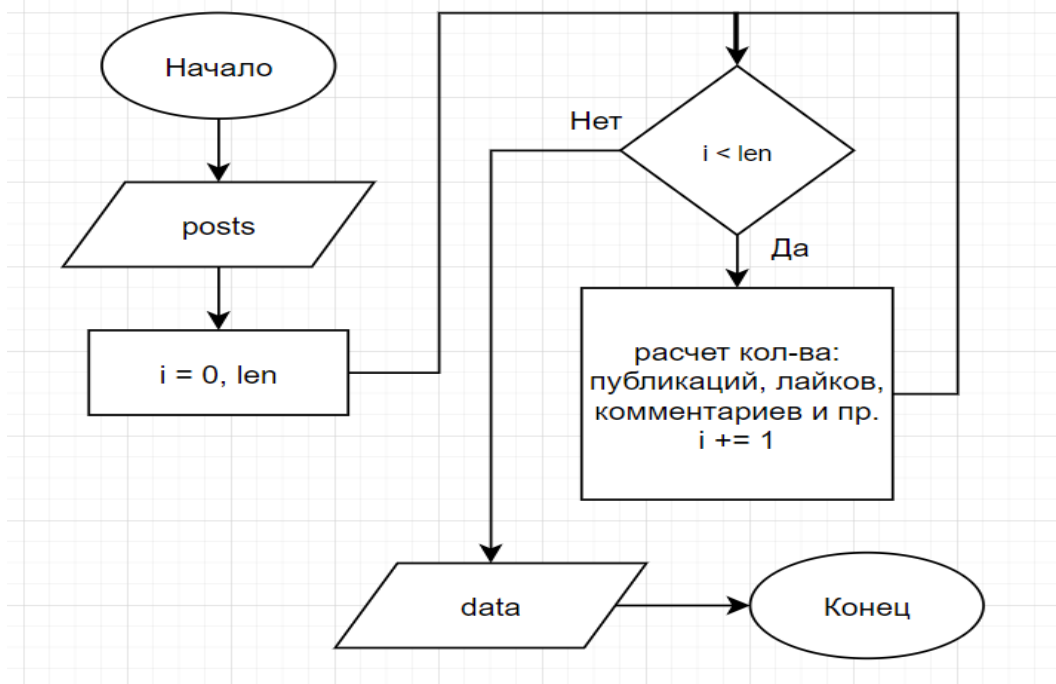


Рисунок 2.2 - блок схема алгоритма расчета статистики

Программный код представлен в приложении 1. За работу алгоритмов сбора, анализа и вывода результатов пользователю отвечают следующие функции:

1) `get_urls(urls)` - функция анализа url-адреса, в случае успешной работы функция возвращает доменное имя или идентификатор сообщества;

2) `search_public(urls)` - поиск сообщества, в качестве аргументов функция принимает идентификатор или доменное имя сообщества, в случае успешной работы функция возвращает краткую информацию по сообществу;

3) `get_posts(urls, date_start, date_end)` - функция осуществляет поиск публикаций в сообществе за временной период. В качестве аргументов принимает идентификатор или доменное имя сообщества, даты диапазона поиска. В случае успешной работы функция возвращает информацию по публикациям сообщества;

4) `count_data(date_start, date_end, posts, id)` - функция осуществляет расчет статистики на основе полученных данных публикаций сообщества. В качестве аргументов принимает идентификатор пользователя, даты диапазона поиска, информацию по публикациям. В случае успешной работы функция возвращает рассчитанную статистику;

5) `analysis(request)` - функция отвечающая за представление данных анализа пользователю. В качестве аргументов принимает системную переменную объекта ответа и запроса. В случае успешной работы функция выводит на страницу результат анализа;

6) `save_reports(posts, stat, date_start, date_end, id)` - функция отвечает за сохранения отчёта в базе данных веб-сервиса. В качестве аргументов принимает идентификатор пользователя, даты диапазона поиска, информацию по публикациям. В случае успешной работы функция возвращает сообщение об успешном сохранении отчета;

7) `create_reports(data)` - функция отвечает за формирование отчетного файла в формате excel. В качестве аргументов принимает данные рассчитанной статистики. В случае успешной работы функция возвращает файл статистики в формате `xlsx`.

2.2 Разработка модуля личного кабинета пользователя

Модуль личного кабинета состоит из функциональных частей:

- 1) Вывод данных пользователя;
- 2) Редактирование данных пользователя;
- 3) Смена пароля.

За работу этих частей отвечают следующие функции:

1) `profile(request)` - функция, отвечающая за вывод информации о пользователе на экран. В качестве аргументов принимает системную переменную объекта ответа и запроса. В случае успеха пользователю выводится информация о нем.

2) `edit(request)` - функция, отвечающая за редактирование данных пользователя. В качестве аргументов принимает системную переменную объекта ответа и запроса. В случае успеха новые данные о пользователе добавляются в БД;

3) `password_change(request)` - функция, отвечающая за изменение парольной фразы пользователем. В качестве аргументов принимает системную переменную объекта ответа и запроса. В случае успеха изменяет заданную ранее парольную фразу;

4) `login (request)` - функция, отвечающая за авторизацию пользователя в веб-сервисе. В качестве аргументов принимает системную переменную объекта ответа и запроса. В случае успеха пользователь получает доступ к функционалу согласно его роли;

5) `logout(request)` - функция, отвечающая за деавторизация пользователя веб-сервиса. В качестве аргументов принимает системную переменную объекта ответа и запроса. В случае успеха пользователь перенаправляется на страницу входа в веб-сервис.

Помимо этого, в работе вышеописанных функция используются функции входящие в состав фреймворка Django:

- 1) `login_required()` - проверка пользователя на предмет аутентификации в веб-сервисе;
- 2) `render()` - создает представление(рендер страниц) на стороне клиента;

3) `redirect()` - перенаправление пользователя на заданную страницу.

2.4 Разработка модуля администрирования

Данный модуль осуществляет управление пользователями веб-сервиса, позволяет просматривать, удалять и создавать новых пользователей.

1) `get_users(request)` - функция, отвечающая за вывод пользователей на экран администратору веб-сервиса. В качестве аргументов принимает системную переменную объекта ответа и запроса. В случае успеха выводится список пользователей с возможностью их фильтрации по ролям и удалением пользователей;

2) `new_user(request)` - функция, отвечающая за добавление нового пользователя веб-сервиса администратором. В качестве аргументов принимает системную переменную объекта ответа и запроса. В случае успеха добавляется новый пользователь в БД веб-сервиса, (администратор или smm-специалист);

3) `del_user(request, username)` - функция, отвечающая за удаление пользователя веб-сервиса администратором. В качестве аргументов принимает системную переменную объекта ответа и запроса. В случае успеха удалится пользователь из БД веб-сервиса.

2.5 Тестирование программного продукта

2.5.1 Выбор метода тестирования программного продукта

Для тестирования программного продукта были выбраны следующие методы:

1) функциональное тестирование; проверяется соответствие реализованных функций заданным требованиям, проверяется каждая из функций web-приложения и все они в комплексе; исследуются все сценарии использования; проверяется адекватность хранимых и

выходных данных, методы их обработки, обработка вводимых данных, методы хранения данных;

2) конфигурационное тестирование; проверяется поведение приложения при различных разрешениях экрана и в различных браузерах;

3) тестирование безопасности; данный метод позволяет убедиться, что данные хранятся надежно, доступ к ним блокирован для посторонних лиц; проверяется защищенность БД, каналов связи, интерфейсов ввода и транспорта данных;

4) нагрузочное тестирование проверяет, какую нагрузку на данном этапе может выдержать сайт, т.е. проверяется, сколько посетителей одновременно могут зайти на сайт, просмотреть страницы и заказать товар; для тестирования используются специальные web-сервисы или программы; для тестирования сайта я воспользовался web-сервисом loaddy.com и loadimpact.com;

5) тестирование пользовательского интерфейса; проверяется удобство использования программного продукта, сколько времени затрачивается на то или иное действие, какие возникают проблемы у пользователя и в состоянии ли он их решить.

2.5.2 Тест-кейсы проверки качества программного продукта

Тест-кейс – это последовательность действий, направленная на проверку какого-либо функционала, описывающая как прийти к фактическому результату.

Тест-кейс включает в себя:

1) название — основная тема, или идея тест-кейса; краткое описание его сути;

2) предусловия — описание условий, которые не имеют прямого отношения к проверяемому функционалу, но должны быть выполнены (например, для того чтобы просмотреть список заказов, необходимо авторизоваться);

3) шаги – описание последовательности действий, которая должна привести нас к ожидаемому результату;

4) ожидаемый результат – результат: что мы ожидаем увидеть после выполнения шагов;

5) результат теста – каждый выполненный тест-кейс дает один из трех результатов: положительный, отрицательный результат или выполнение заблокировано, если после одного из шагов продолжение теста невозможно.

В таблицах 2.1 – 2.12 представлены пройденные тест-кейсы веб-сервиса.

Таблица 2.1 – Вход в веб-сервис

1	2	
Название:	Тест «Вход в веб-сервис»	
Функция:	login ()	
Предусловие:	Открыть страницу «Вход»	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован
Заполнить поля формы (логин и пароль)	На экране отображается введенная информация.	«пройден»
Нажать кнопку «Войти»	Пользователь получает доступ к закрытому функционалу.	«пройден»

Таблица 2.2 – Выход из веб-сервиса

1	2	
Название:	Тест «Выход из веб-сервиса»	
Функция:	logout ()	
Предусловие:	Быть авторизованным пользователем	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован
Нажать кнопку «Выйти»	Пользователь перенаправляется на страницу входа в веб-сервис	«провален»

Таблица 2.3 – Анализ сообщества вк

1	2	
Название:	Тест «Анализ сообщества вк»	
Функция:	analysis ()	
Предусловие:	Открыть страницу «Анализ»	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован
Заполнить поля формы (ссылка на сообщество, диапазон даты)	На экране отображается введенная информация.	«пройден»
Нажать кнопку «Поиск»	Выводится найденная информация.	«пройден»

Таблица 2.4 - Сохранение отчета

1	2	
Название:	Тест «Сохранение отчета»	
Функция:	save_reports ()	
Предусловие:	1) Открыть страницу «Анализ»; 2) Провести анализ сообщества.	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован
Нажать кнопку «Сохранить»	Появляется сообщение об успешном сохранении отчета.	«провален»

Таблица 2.5 - Скачивание отчета

1	2	
Название:	Тест «Скачивание отчета»	
Функция:	get_reports ()	
Предусловие:	1) Открыть страницу «Анализ»; 2) Провести анализ сообщества.	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован
Нажать кнопку «Скачать»	Отчет скачивается на компьютер пользователя.	«провален»

Таблица 2.6 - Просмотр отчетов

1	2	
Название:	Тест «Просмотр отчетов»	
Функция:	reports ()	
Предусловие:	Открыть страницу «Отчеты»	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован
Нажать кнопку «Отчеты»	Выводится список отчетов пользователя.	«пройден»

Таблица 2.7 - Удаление отчета

1	2	
Название:	Тест «Удаление отчета»	
Функция:	del_report ()	
Предусловие:	Открыть страницу «Отчеты»	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован
Нажать кнопку «Удалить»	Появляется сообщение об успешном удалении отчета.	«провален»

Таблица 2.8 - Просмотр пользователей

1	2	
Название:	Тест «Просмотр пользователей»	
Функция:	get_users ()	
Предусловие:	Открыть страницу «Пользователи»	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован
Нажать кнопку «Пользователи»	На экран выводятся пользователи веб-сервиса.	«пройден»

Таблица 2.9 – Удаление пользователя

1	2	
Название:	Тест «Удаление пользователя»	
Функция:	del_user ()	
Предусловие:	Открыть страницу «Пользователи»	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован
Нажать кнопку «Удалить»	Появляется сообщение об успешном удалении пользователя.	«провален»

Таблица 2.10 – Добавление пользователя

1	2	
Название:	Тест «Добавление пользователя»	
Функция:	new_user ()	
Предусловие:	Открыть страницу «Добавить»	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован
Заполнить данные формы	На экране отображается введенная информация.	«пройден»
Нажать кнопку «Добавить»	Появляется сообщение об успешном добавлении пользователя.	«пройден»

Таблица 2.11 – Изменение данных профиля

1	2	
Название:	Тест «Изменение данных профиля»	
Функция:	edit ()	
Предусловие:	Открыть страницу «Редактировать»	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован

Окончание таблицы 2.11

1	2	3
Заполнить данные формы	На экране отображается введенная информация.	«пройден»
Нажать кнопку «Сохранить»	Появляется сообщение об успешном изменении данных пользователя.	«пройден»

Таблица 2.12 - Сменить пароль

1	2	
Название:	Тест «Сменить пароль»	
Функция:	password_change ()	
Предусловие:	Открыть страницу «Сменить пароль»	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован
Заполнить данные формы	На экране отображается введенная информация.	«пройден»
Нажать кнопку «Изменить»	Появляется сообщение об успешном изменении пароля.	«пройден»

Таблица 2.13 - Личный кабинет

1	2	
Название:	Тест «Личный кабинет»	
Функция:	profile ()	
Предусловие:	Открыть страницу «Профиль»	
Действие	Ожидаемый результат	Результат теста: - пройден - провален - заблокирован
Нажать кнопку «Профиль»	Выводится информация о пользователе	«пройден»

2.5.3 Баг репорты программного продукта

Баг репорт – это документ, в котором ведутся описываются сведения и шаги действий во время проведения тестирования программного обеспечения, приводящие его в нерабочие состояние или вызывающие ошибки в логике работы.

Структура документа, следующая:

- 1) Краевое описание события;
- 2) Название тестируемого программного продукта;
- 3) Компонент приложения, в котором произошла ошибка;
- 4) Номер версии программного продукта;
- 5) Серьёзность ошибки, подразделяется на:
 - S1 Блокирующий (Blocker) – ошибка приводящие программный продукт в нерабочие состояние, из-за чего дальнейшая работа с ним не возможна.
 - S2 Критический (Critical) – неполадки в работе сервера, проблемы с безопасностью или частью функционала программного продукта;
 - S3 Значительный (Major) – некоторые части бизнес-логики программного продукта работают неправильно;
 - S4 Незначительный (Minor) – неточности в работе пользовательского интерфейса;
 - S5 Тривиальный (Trivial) – проблемы связанные с сторонними приложениями мало влияющие на основную бизнес логику приложения.
- 6) Приоритет устранения ошибки подразделяется на:
 - P1 Высокий (High) – ошибка должна быть исправлена немедленно;
 - P2 Средний (Medium) – ошибка должна быть исправлена в ближайшее время;
 - P3 Низкий (Low) – ошибка должна быть исправлена по возможности.
- 7) Описание шагов воспроизведения, приводящих к ошибке;
- 8) Фактический результат, то что произошло не запланировано;

9) Ожидаемый результат, результат запланированный бизнес-логикой программного продукта.

Ошибки, полученные в процессе тестирования, изложены ниже:

1) Ошибка при нажатии на кнопку «Выход», пользователя не перенаправило на страницу «Вход»;

2) Ошибка на странице «Анализ» при нажатии на кнопку «Сохранить», появляется сообщение об успешном сохранении отчета, но сам отчет не сохранился в базу данных;

3) Ошибка на странице «Анализ» при нажатии на кнопку «Скачать», появляется ошибка интерпритатора «Файл не найден»;

4) Ошибка на странице «Отчеты» при нажатии на кнопку «Удалить», появляется сообщение об успешном удалении отчета, но сам отчет не удаляется в базе данных;

5) Ошибка на странице «Пользователи» при нажатии на кнопку «Удалить», появляется ошибка интерпритатора «Пользователь с таким id не найден».

Таблица 2.14 - Баг Репорт 1

1	2
Короткое описание	Ошибка при нажатии на кнопку «Выход», пользователя не перенаправило на страницу «Вход»
Проект	Веб-сервис
Компонент приложения	Модуль личного кабинета пользователя
Номер версии	1
Важность: – S1 Блокирующая (Blocker) – S2 Критическая (Critical) – S3 Значительная (Major) – S4 Незначительная (Minor) – S5 Тривиальная (Trivial)	S2 Критическая (Critical)
Приоритет:	P1 Высокий (High)

– P1 Высокий (High) – P2 Средний (Medium) – P3 Низкий (Low)	
Статус	Новая
Автор	Ремез М.В.
Назначен на	Ремез М.В.
Шаги воспроизведения	1 Нажать в правом верхнем углу кнопку «Выход»
Фактический результат	Пользователь выходит из веб-сервиса, но не перенаправляется на страницу «Вход»
Ожидаемый результат	Пользователь перенаправляется на страницу «Вход»

Таблица 2.15 - Баг Репорт 2

1	2
Короткое описание	Ошибка на странице «Анализ» при нажатии на кнопку «Сохранить», появляется сообщение об успешном сохранении отчета, но сам отчет не сохранился в базу данных
Проект	Веб-сервис
Компонент приложения	Модуль сбора и анализа информации
Номер версии	1.2
Важность: – S1 Блокирующая (Blocker) – S2 Критическая (Critical) – S3 Значительная (Major) – S4 Незначительная (Minor) – S5 Тривиальная (Trivial)	S3 Значительная (Major)
Приоритет: – P1 Высокий (High) – P2 Средний (Medium) – P3 Низкий (Low)	P1 Высокий (High)
Статус	Новая
Автор	Ремез М.В.
Назначен на	Ремез М.В.
Шаги воспроизведения	1 Перейти на страницу «Анализ»

	2 Провести анализ 3 Нажать кнопку «Сохранить»
Фактический результат	Отчет не сохраняется в БД
Ожидаемый результат	Сохранение отчета в БД и появление сообщения об успешном сохранении

Таблица 2.16 - Баг Репорт 3

1	2
Короткое описание	Ошибка на странице «Анализ» при нажатии на кнопку «Скачать», появляется ошибка интерпритатора «Файл не найден»
Проект	Веб-сервис
Компонент приложения	Модуль сбора и анализа информации
Номер версии	1.3

Окончание таблицы 2.16

1	2
Важность: – S1 Блокирующая (Blocker) – S2 Критическая (Critical) – S3 Значительная (Major) – S4 Незначительная (Minor) S5 Тривиальная (Trivial)	S3 Значительная (Major)
Приоритет: – P1 Высокий (High) – P2 Средний (Medium) – P3 Низкий (Low)	P1 Высокий (High)
Статус	Новая
Автор	Ремез М.В.
Назначен на	Ремез М.В.
Шаги воспроизведения	1 Перейти на страницу «Анализ» 2 Провести анализ 3 Нажать кнопку «Скачать»
Фактический результат	Ошибка интерпритатора, файл не найден
Ожидаемый результат	Сохранение отчета на ПК пользователя и появление сообщения об успешном

Таблица 2.17 - Баг Репорт 4

1	2
Короткое описание	Ошибка на странице «Отчеты» при нажатии на кнопку «Удалить», появляется сообщение об успешном удалении отчета, но сам отчет не удаляется в базе данных
Проект	Веб-сервис
Компонент приложения	Модуль сбора и анализа информации
Номер версии	1.4
Важность: – S1 Блокирующая (Blocker) – S2 Критическая (Critical) – S3 Значительная (Major) – S4 Незначительная (Minor) – S5 Тривиальная (Trivial)	S3 Значительная (Major)
Приоритет: – P1 Высокий (High) – P2 Средний (Medium) – P3 Низкий (Low)	P1 Высокий (High)

Окончание таблицы 2.17

1	2
Статус	Новая
Автор	Ремез М.В.
Назначен на	Ремез М.В.
Шаги воспроизведения	1 Перейти на страницу «Отчеты» 2 Нажать кнопку «Удалить»
Фактический результат	Отчет не удаляется в БД
Ожидаемый результат	Удаление отчета в БД и появление сообщения об успешном удалении

Таблица 2.18 - Баг Репорт 5

1	2
Короткое описание	Ошибка на странице «Пользователи» при нажатии на кнопку «Удалить», появляется ошибка интерпритатора «Пользователь с таким id не найден»
Проект	Веб-сервис

Компонент приложения	Модуль администрирования
Номер версии	1.5
Важность: – S1 Блокирующая (Blocker) – S2 Критическая (Critical) – S3 Значительная (Major) – S4 Незначительная (Minor) – S5 Тривиальная (Trivial)	S3 Значительная (Major)
Приоритет: – P1 Высокий (High) – P2 Средний (Medium) – P3 Низкий (Low)	P1 Высокий (High)
Статус	Новая
Автор	Ремез М.В.
Назначен на	Ремез М.В.
Шаги воспроизведения	1 Перейти на страницу «Пользователи» 2 Нажать кнопку «Удалить»
Фактический результат	Ошибка интерпритатора, пользователь с переданным id не найден
Ожидаемый результат	Удаление пользователя и БД и сообщение об успешном удалении

В процессе тестирования программного продукта были выявлены и добавлены в багрепорты ошибки. Для отлова и дальнейшего исправления ошибок, описанных выше, потребовалось две дополнительных итерации тестирования.

2.5.4 Нагрузочное тестирование

Нагрузочное тестирование - позволяет определить, как и при каких условиях будет работать программное обеспечения в стрессовых условия (высокая нагрузка на сервер, большой объём входных данных и пр.).

Для проведения нагрузочного теста был выбран онлайн сервис loaddy.com с бесплатным тарифом тестирования веб-приложений. Результаты тестирования продемонстрированы на рисунке 2.3.

Результат проверки <http://freepops.pythonanywhere.com/> за 5 июня 2020 г. 16:46:31

Выберите локацию ? [Детали теста](#)

100 %
Доступность ?

0.00 s
Скорость загрузки ?

0.25 s
Время ответа ?

0.26 Mb
Получено данных ?

Тип ?	Возрастающая нагрузка	Количество запросов ?	220
Диапазон нагрузки ?	От 5 до 50	Количество ответов ?	220 0
Шаг ?	5	Прямая ссылка ?	http://loaddy.com/result/817377902/

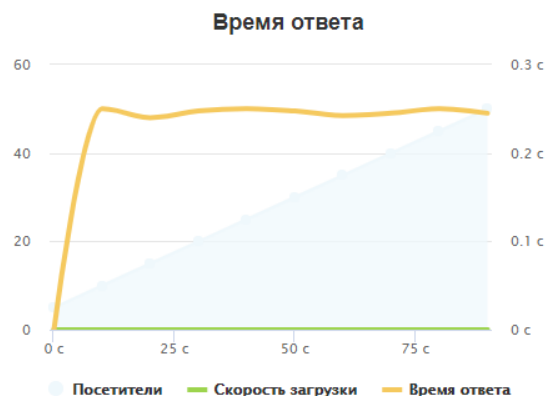


Рисунок 2.3 - результаты нагрузочного тестирования веб-сервиса

Результаты тестов показали, что веб-сервис с успехом выдерживает одновременное её использование 50 пользователями.

3 Экономическое обоснование программного продукта

3.1 Расчет прямых и накладных расходов

Проведем расчет стоимости работ, связанных с разработкой веб-сервиса.

Прямые расходы включают в себя:

- 1) расходы на оплату труда с учетом трудозатрат;
- 2) страховые взносы.

Разработкой веб-сервиса будет заниматься стажер-программист. Средняя заработная плата программиста без опыта работы в Ростове-на-Дону согласно вакансиям hh.ru составляет 25 000 руб. в месяц.

Определим заработную плату за год:

$$\text{З/п за год} = 25\,000 * 12 = 300\,000 \text{ руб.}$$

Определим стоимость трудозатрат за 1 час:

$$\text{Стоимость трудозатрат за час} = \text{ФОТ}/247/8$$

(1)

$$\text{Стоимость трудозатрат за час} = 300\,000 / 247 / 8 = 152 \text{ руб.}$$

Таблица 3.1 - Расчет расходов на оплату труда с учетом трудозатрат

Наименование услуг	Трудозатраты, час	Стоимость трудозатрат в час (руб.)	Стоимость (руб.)
1	2	3	4
Сбор и анализ исходных данных	16	152	2 432
Сбор и составление требований для ТЗ	16	152	2 432
Выбор инструментариев разработки	8	152	1 216
Проектирование базы данных и use-case диаграмм	32	152	4 864
Прототипирование пользовательского интерфейса	8	152	1 216

Окончание таблицы 3.1.

1	2	3	4
Разработка модуля сбора и анализа публичной информации и тестирование	108	152	16 416
Внедрение программного продукта	4		608
Итого:	184		27 968

Рассчитаем временные сроки выполнения проекта.

$$\text{Сроки.пр} = \text{Трудозатраты} / 8 / 22$$

Получится длительность проекта в месяцах.

$$\text{Сроки.пр.} = 184 / 8 / 22 = 1 \text{ мес.}$$

Определим страховые взносы в внебюджетные фонды:

$$\text{Страхов.вз.} = \text{ФОТ}_{\text{Год}} * (20\% + 0,2\%) / 100\% \quad (2)$$

где Страхов.вз. - страховые взносы в пенсионный фонд и фонд социального страхования

ФОТ - фонд оплаты труда;

20% - тариф страховых взносов для компаний в области информационных технологий;

0,2% - отчисления в фонд травматизма и профессионального заболевания, в фонд социального страхования.

$$\text{Страхов.вз.} = 27\,968 * (20\% + 0,2\%) / 100\% = 5\,650 \text{ руб.}$$

Определим накладные расходы. Накладные расходы включают в себя:

1) услуги связи (стоимость интернета - 450 рублей, сотовая связь - 100 рублей)

$$\text{У.с.} = (450 + 100) * 1 = 550 \text{ руб.} \quad (3)$$

2) коммунальные расходы (средняя рыночная ставка на коммунальные услуги за 1 кв. метр составляет 80 рублей)

$$\text{Рк.у.} = \text{нк.у.} * S \quad (4)$$

где Рк.у. - расходы на коммунальные услуги;

нк.у.- средняя рыночная ставка на коммунальные услуги за 1 кв. метр;

S- площадь в кв.метрах (27 кв.м).

$$P_{к.у.} = 80 * 27 = 2\ 160 \text{ руб.}$$

$$P_{ас.нак} = 550 + 2\ 160 = 2\ 710 \text{ руб.}$$

3.2 Себестоимость работ и расчет прибыли

Себестоимость работ включает:

- 1) расходы на оплату труда;
- 2) страховые взносы;
- 3) накладные расходы.

Таблица 3.2 - Себестоимость работ (Ср)

Статьи расходов	Стоимость (руб.)
1	2
1. Расходы на оплату труда	27 968
2. Страховые взносы	5 650
3. Накладные расходы	2 710
Итого (Ср):	36 328

Прибыль определяется, исходя из нормы рентабельности:

$$П = C_p * HR\% / 100\%, \quad (6)$$

где П - прибыль;

Ср - себестоимость работ;

HR - норма рентабельности (12%).

$$П = 36\ 328 * 12\% / 100\% = 4\ 359 \text{ руб.}$$

Расчет налога с выручки по упрощенной системе

Определим налог с выручки по упрощенной системе налогообложения (Нусн), ставка налога при налогообложении «Доход», составляет 6% для ИП.

$$Н_{усн} = (C_p + П) * 6\% / 100\%, \quad (7)$$

где Ср - себестоимость работ;

П - прибыль.

$$Н_{усн} = (36\ 328 + 4\ 359) * 6\% / 100\% = 2\ 441 \text{ руб.}$$

Сводная таблица стоимости работ, связанных с разработкой веб-сервиса.

Таблица 3.3 - Стоимость работ

Статьи расходов	Стоимость, руб.
1	2
Себестоимость работ, Ср	36 328
Прибыль, П	4 359

Налог по УСН, Нусн	2 441
Итого:	43 128

Стоимость владения веб-сервисом равна 0 рублям так как нет затрат на хостинг и приобретение доменного имени (веб сервис развернут на бесплатном хостинге и использует бесплатное доменное имя).

3.3 Оценка эффективности внедрения ПП

Смета затрат на проект представлена в таблице 3.4.

Таблица 3.4 - Смета затрат на проект

№ п/п	Смета затрат на проект	Стоимость, руб.
1	2	3
1	Расходы на оплату труда (ФЗП)	27 968
2	Страховые взносы, (20,2%)	5 650
3	Накладные расходы	2 710
4	Себестоимость работ	36 328
5	Прибыль (... %)	4 359
6	Стоимость работ по разработке ПО	40 687
7	Налог с выручки (6%)	2 441
8	ИТОГО общая стоимость работ	43 128

В среднем до внедрения программного продукта smm-специалист тратил на составление одного отчета 10 минут. Одновременно smm-специалист может вести до 5 проектов и составлять до 20 отчетов в месяц. В агентстве «Фейском» насчитывается 6 smm-специалистов, а оплата их труда за 1 час составляет 218 рублей. Итого smm-специалист тратит до 3 часов 30 минут на составление отчетов, а в денежном эквиваленте затраты на составление отчета 763 рубля. За год затраты на составление отчетов всего отдела smm-специалистов составят 54 936 рубля.

После внедрения программного продукта, время на составление отчета сократилось до 2 минут. Итого затраты за год сократятся до 10 464 рублей.

Рассчитаем срок окупаемости программного продукта:

Срок окупаемости = затраты / экономия от внедрения

Срок окупаемости = 43 128 / (54 936 - 10 464) = 0.96

Срок окупаемости программного продукта 9 месяцев.

ЗАКЛЮЧЕНИЕ

Разработанный программный продукт был разработан и внедрен в агентстве «Fасесom». Созданный программный продукт поможет оптимизировать работу smm-специалистов агентства и уменьшит финансовые издержки компании на использование сторонних средств сбора и анализа информации.

В процессе исследования предметной области были проанализированы схожие инструменты сбора и анализа информации и выявлены их сильные и слабые стороны.

На этапе проектирования были спроектированы алгоритмы работы модулей веб-сервиса, use-case диаграммы и спроектирован прототип интерфейса программного продукта.

Разработанный программный продукт соответствует требованиям, описанным в техническом задании на разработку:

- удобный и современный дизайн;
- адаптивная верстка сайта;
- сохранение отчетов в базу данных веб-сервиса;
- экспорт отчетной информации в формат xlsx;
- создание и управление пользователями.

Программный продукт прошел проверку по различным тест-кейсам. Помимо тест-кейсов было проведено нагрузочное тестирование. Результаты тестов показали, что веб-сервис с успехом выдерживает одновременное её использование 50 пользователями.

Был проведен анализ эффективности внедрения веб-сервиса. Срок окупаемости внедряемого программного продукта составляет всего 6 месяцев.

В ближайшее время планируется проведение работ по расширению функционала веб-сервиса, а именно:

- 1) автоматическом сборе ежемесячно отчетной информации;
- 2) составление графиков на основе полученных данных.

Список литературы

1. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — 496 с
2. Скрапинг веб-сайтов с помощью Python. Митчелл Р. - М.: ДМК Пресс, 2016. - 280 с
3. Разработка геоприложений на языке Python / Вестра Э. - М. : ДМК Пресс, 2017. - 446 с.
4. Django: Практика создания Web-сайтов на Python, Дронов В., - М: БХВ-Петербург, 2016 г., 516 стр.
5. Web-дизайн. Удобство использования Web-сайтов, -М: Вильямс, 2009 г., 361 стр.
6. Не заставляйте меня думать. Веб-юзабилити и здравый смысл. 3-е издание, Круг С., - М. :Эксмо, 2017 г., 210 стр.
7. Разработка пользовательских интерфейсов, Тидвелл Д. - М. :O'Reilly, 416 стр.
8. Новая большая книга CSS, Соьер М., - М. :O'Reilly, 2020 г., 720 стр.
9. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5, Никсон Р., - М. :O'Reilly, 2019 г., 496 стр.
10. HTML5, CSS3 и JavaScript. Исчерпывающее руководство. 4-ое издание, Д. Роббинс, 2014 г., 528 стр.
11. Жилина Е.В., Королева А.К. Исследование Web-системы коммерческого предприятия в социальной сети ВКонтакте // SCIENCE TIME. - Казань, 2015. - № 3 (15). - С. 180-194.
12. Жилина Е.В. Функциональный анализ бизнес-процессов рекламной компании при работе с клиентами // Современные проблемы гуманитарных и естественных наук: материалы XXII междунар. научно-практич. конф. 2015 г. - М.: Институт стратегических исследований, 2015. - С. 109-117.
13. Жилина Е.В., Кулебякин Р.Б., Частухина Л.В. Синтаксический анализ текста в Интернете. Парсинг // Эволюция современной науки: материалы XXIV международной конф. г. Уфа, 18 февраля 2017 г. - Уфа, 2017. - С. 28-33.

14. Жилина Е.В., Трипута В.Н., Сергиенко В.Ю. Оптимизация работы веб-приложений // Проблемы информационной безопасности: сб. статей VII Всероссийской заочной Интернет-конференции г. Ростов-на-Дону, 20-21 февраля 2018. - г. Ростов-на-Дону: РГЭУ (РИНХ), 2018.

15. Жилина Е.В., Частухина Л.В., Ганжа А.Е. Моделирование жизненного цикла программного обеспечения // Информационные системы, экономика, управление трудом и производством: Ученые записки. Вып.16. - Ростов-на-Дону: РГЭУ «РИНХ», 2014.

16. Исследование правильности и эффективности средств парсинга информации на веб-ресурсах [Электронный ресурс]. - URL: <https://cyberleninka.ru/article/n/issledovanie-pravilnosti-i-effektivnosti-sredstv-parsinga-informatsii-na-veb-resursah> (дата обращения: 03.03.2020 г.)

17. Разработка парсинг-системы для получения скрытых ссылок со страниц социальных сетей [Электронный ресурс]. - URL: <https://cyberleninka.ru/article/n/razrabotka-parsing-sistemy-dlya-polucheniya-skrytyh-ssylok-so-stranits-sotsialnyh-setey> (дата обращения: 03.03.2020 г.)

18. Разработка автоматизированной системы сбора информации для автомобильного портала [Электронный ресурс]. - URL: <https://cyberleninka.ru/article/n/razrabotka-avtomatizirovannoy-sistemy-sbora-informatsii-dlya-avtomobilnogo-portala> (дата обращения: 03.03.2020 г.)

19. Автоматизация сбора и обработки спортивной статистики для последующего сетевого анализа [Электронный ресурс]. - URL: <https://cyberleninka.ru/article/n/avtomatizatsiya-sbora-i-obrabotki-sportivnoy-statistiki-dlya-posleduyuschego-setevogo-analiza> (дата обращения: 03.03.2020 г.)

20. Методические аспекты изучения парсинга средствами php в курсе "сетевые языки и web-программирование" [Электронный ресурс]. - URL: <https://cyberleninka.ru/article/n/metodicheskie-aspekty-izucheniya-parsinga-sredstvami-php-v-kurse-setevye-yazyki-i-web-programmirovaniye> (дата обращения: 03.03.2020 г.)

21. Разработка программы сбора данных о структуре веб-сайтов [Электронный ресурс]. - URL: <https://cyberleninka.ru/article/n/razrabotka-programmy-sbora-dannykh-o-strukture-veb-saytov>

programmy-sbora-dannyh-o-strukture-veb-saytov (дата обращения: 03.03.2020 г.)

22. Адаптивный краулер для поиска и сбора внешних гиперссылок [Электронный ресурс]. - URL: <https://cyberleninka.ru/article/n/adaptivnyu-krauler-dlya-poiska-i-sbora-vneshnih-giperssylok> (дата обращения: 03.03.2020 г.)

23. Анализ способов сбора социальных данных из сети интернет [Электронный ресурс]. - URL: <https://cyberleninka.ru/article/n/analiz-sposobov-sbora-sotsialnyh-dannyh-iz-seti-internet> (дата обращения: 03.03.2020 г.)

24. Анализ социальных сетей: методы и приложения [Электронный ресурс]. - URL: <https://cyberleninka.ru/article/n/analiz-sotsialnyh-setey-metody-i-prilozheniya> (дата обращения: 03.03.2020 г.)

25. Основные методы анализа, используемые при исследовании социальных сетей [Электронный ресурс]. - URL: <https://cyberleninka.ru/article/n/osnovnye-metody-analiza-ispolzuyemye-pri-issledovanii-sotsialnyh-setey> (дата обращения: 03.03.2020 г.)

26. Методы анализа данных из социальных сетей [Электронный ресурс]. - URL: <https://cyberleninka.ru/article/n/metody-analiza-dannyh-iz-sotsialnyh-setey> (дата обращения: 03.03.2020 г.)

27. Интерфейс. Основы проектирования взаимодействия, Купер А., Рейманн М.- М. :Питер, 2018 г., 632 стр.

28. Большая книга веб-дизайна, Т. Фельке-Моррис, - М. :Эксмо, 2012 г., 608 стр.

29. Web-дизайн. Справочник, Роббинс Д.- М. :КУДИЦ-Пресс, 2008 г., 325 стр.

30. Документация API vk.com, [Электронный ресурс]. - URL: <https://vk.com/dev/manuals> (дата обращения: 03.03.2020 г.)

31. Документация API ok.ru, [Электронный ресурс]. - URL: <https://apiok.ru/> (дата обращения: 06.03.2020 г.)

32. Документация bootstrap 4, [Электронный ресурс]. - URL: <https://getbootstrap.com/docs/4.0/getting-started/introduction/> (дата обращения: 06.03.2020 г.)

33. Документация Django, [Электронный ресурс]. - URL: <https://docs.djangoproject.com/en/3.0/> (дата обращения: 06.03.2020 г.)

34. Документация Django на русском, [Электронный ресурс]. - URL: <https://djbook.ru/> (дата обращения: 06.03.2020 г.)
35. Документация draw.io, [Электронный ресурс]. - URL: <https://drawio-app.com/tutorials/learning/> (дата обращения: 06.03.2020 г.)
36. Статистика популярности языков программирования, [Электронный ресурс]. - URL: <https://www.tiobe.com/tiobe-index/> (дата обращения: 06.03.2020 г.)
37. Разработка Технического задания по ГОСТ 34, [Электронный ресурс]. - URL: <https://habr.com/ru/post/432852/> (дата обращения: 06.05.2020 г.)
38. Онлайн инструмент проектирования баз данных, [Электронный ресурс]. - URL: dbdesigner.net/ (дата обращения: 06.05.2020 г.)
39. Документация модуля Xlsxwriter [Электронный ресурс]. - URL: <https://xlsxwriter.readthedocs.io> (дата обращения: 06.05.2020 г.).

1 Код программного продукта модуля сбора информации

```
# обрабатываем ссылки и получаем id паблика или его домен
def get_urls(urls):
    get_urls = []
    i = 0

    # проверяем есть ли ссылка на вк и в каком формате она
    while i < len(urls):
        print(urls[i].find('public'))
        print(urls[i].find('vk.com'))
        if urls[i].find('public') == -1 and urls[i].find('vk.com') == -1:
            get_urls = 'errors'
        elif urls[i].find('public') == -1:
            get_urls.append(['1', urls[i][urls[i].find('vk.com/') +
7:len(urls[i])]])
        else:
            get_urls.append(['0', ('-' + urls[i][urls[i].find('public') +
6:len(urls[i])])])
        i += 1

    return get_urls

# ищем сообщество
def search_public(urls):
```

```

url_api = 'https://api.vk.com/method/wall.get'

ver = '5.103'

# сервисный ключ вк приложения

token =
'a92ba8d9a92ba8d9a92ba8d9d8a9444243aa92ba92ba8d9f77e
b8bc3a634677be0a63ed'

tmp_data_of_public = []

data_of_public = []

count = 100

# костыль

tmp_m = []

# определяем id или домен сообщества в data

if urls[0] == '0':

    param_domain_or_id = 'owner_id'

else:

    param_domain_or_id = 'domain'

try:

    c = requests.get(url_api,

        params={

            param_domain_or_id: urls[1],

            'access_token': token,

            'v': ver,

            'count': 1,

            'offset': 0

        }

    )

    # получаем основные данные по обществу

    id = str(c.json()['response']['items'][0]['owner_id'])

```

```

    tmp =
requests.get('https://api.vk.com/method/groups.getById?
group_ids=' + id[1:len(

        id)] +
'&access_token=a92ba8d9a92ba8d9a92ba8d9d8a9444243aa92
ba92ba8d9f77eb8bc3a634677be0a63ed&fields=members_coun
t&v=5.103')

    tmp_m.append(tmp.json())

    tmp_data_of_public.append(tmp_m[0]['response'][0])

    # получаем количество постов в сообществе

    count_posts = c.json()['response']['count']

    data_of_public = [tmp_data_of_public[0]['id'],
tmp_data_of_public[0]['name'],

                        tmp_data_of_public[0]['screen_name'],
tmp_data_of_public[0]['members_count'], count_posts]

    tmp_m = []

except:

    return 'errors'

return data_of_public

# получаем посты публика

def get_posts(urls, date_start, date_end):

    url_api = 'https://api.vk.com/method/wall.get'

    ver = '5.103'

    # сервисный ключ вк приложения

    token =
'a92ba8d9a92ba8d9a92ba8d9d8a9444243aa92ba92ba8d9f77e
b8bc3a634677be0a63ed'

```

```

# id или домен сообщества
count = 100
all_posts = []
group_data = []
# костыль
tmp_m = []
i = 0

# перебираем
while i < len(urls):
    # ищем сообщество и осн. данные по нему
    group_data = search_public(urls[i])

    if group_data == 'errors':
        return 'errors'

    # Получаем все посты
    posts = []
    a = []
    b = []
    offset = 0
    u = 0

    # добавляем данные о сообществе
    posts.append(group_data)

    # перебираем все записи в сообществе
    while offset < group_data[4] and u == 0:

```

```

# генерируем запрос
r = requests.get(url_api,
                 params={
                     'domain': group_data[2],
                     'access_token': token,
                     'v': ver,
                     'count': count,
                     'offset': offset
                 })

# отправляем запрос к API vk на получение данных
data = r.json()['response']['items'];

# наполняем поставки: дергаем нужные поля лайки,
текст и прочее
k = 0
while k < len(data) and u == 0:
    img = ""
    text = ""
    # отсекаем посты пользователей
    if data[k]['owner_id'] == data[k]['from_id']:
        # отсекаем рекламные посты
        if data[k]['marked_as_ads'] == 0:
            # отсекаем не нужные даты
            print(str(datetime.datetime.date(
                datetime.datetime.strptime(date_start, "%Y-%m-
%d") + timedelta(days=-1))))

    print(str(datetime.datetime.date(datetime.datetime.fromtimestamp(d

```

```

ata[k]['date'])))

print(str(datetime.datetime.date(datetime.datetime.strptime(date_end, "%Y-%m-%d"))))

print('_____')

if str(datetime.datetime.date(
    datetime.datetime.strptime(date_start, "%Y-%m-%d") + timedelta(days=-1))) <= str(
datetime.datetime.date(datetime.datetime.fromtimestamp(data[k]
['date']))) <= str(
datetime.datetime.date(datetime.datetime.strptime(date_end, "%Y-%m-%d"))):
    # отсекаем посты без вложений
    if 'attachments' in data[k] or 'copy_history' in
data[k]:
        # проверим репост или не репост
        if 'attachments' in data[k]:
            if data[k]['attachments'][0]['type'] ==
'photo':
                img = data[k]['attachments'][0]['photo']
['sizes']
                len(data[k]['attachments'][0]['photo']
['sizes']) - 1)['url']
            else:
                if data[k]['attachments'][0]['type'] ==
'video':
                    img = data[k]['attachments'][0]['video']
['image']
                    len(data[k]['attachments'][0]['video']
['image']) - 1)['url']
            text = data[k]['text']

```



```

else:
    if 'copy_history' in data[k] and 'attachments'
in data[k]['copy_history'][0]:
        if data[k]['copy_history'][0]
['attachments'][0]['type'] == 'photo':
            img = data[k]['copy_history'][0]
['attachments'][0]['photo']['sizes'][
len(data[k]['copy_history'][0]
['attachments'][0]['photo'][
'sizes']) - 1][
'url']
        else:
            if data[k]['copy_history'][0]
['attachments'][0]['type'] == 'video':
                img = \
data[k]['copy_history'][0]
['attachments'][0]['video']['image'][len(
data[k]['copy_history'][0]
['attachments'][0]['video'][
'image']) - 1][
'url']
                text = data[k]['copy_history'][0]['text']
else:
    # print('нет вложений')
    # print(k)
    text = data[k]['text']

# просмотры появились с 2017 года
try:
    views = data[k]['views']['count']
except:

```

```

        views = '0'

        a = [data[k]['owner_id'],

str(datetime.datetime.date(datetime.datetime.fromtimestamp(data[k]
['date']))),

        'https://vk.com/' + group_data[2] + '?w=wall'
+ str(
        data[k]['owner_id']) + '_' + str(data[k]['id']),
img, text,
        data[k]['likes']['count'],
        data[k]['comments']['count'], views, data[k]
['reposts']['count']]
        print(a)

    if a.__len__() != 0:
        b.append(a)
        if a[1] <= str(datetime.datetime.date(
            datetime.datetime.strptime(date_start, "%Y-%m-
%d") + timedelta(days=-1))):
            u = 1

        k += 1
        a = []

    posts.extend(b)

    if b.__len__() != 0 and b[len(b) - 1][1] <= str(
datetime.datetime.date(datetime.datetime.strptime(date_start, "%Y-
%m-%d") + timedelta(days=-1))):
        u = 1

```

```

    b = []

    offset += 100

    group_data = []

    # добавляем собранные посты по паблику в общий список
    all_posts.append(posts)

    # Переход на следующий паблик
    i += 1

return all_posts

```

1 Код программного продукта модуля расчета статистики

```

# рассчитываем статистику
def count_data(request, date_start, date_end, posts, id):
    model = Post
    data = [] # итоговые данные
    post = [] # посты с расчётом статистики
    i = 0
    stat = [0, 0, 0, 0, 0, 0, 0] # статистика

    # получаем данные из бд
    # добавить к условию id пользователя
    q = Post.objects.raw(
        "SELECT * FROM app_post WHERE date_post BETWEEN
" + date_start + " and " + date_end + " AND id_user=" + str(
        id) + ";"
    )
    if not q:
        return 'errors'
    # считаем показатели за период
    while i < (len(q)):
        post.append(
            [q[i].url_post, q[i].url_img, q[i].text_post, q[i].c_like,
q[i].c_comments, q[i].c_views, q[i].c_reposts,
            q[i].date_post, ((q[i].c_like + q[i].c_comments) / posts[0][3]) *
100])
        stat[0] += q[i].c_like
        stat[1] += q[i].c_comments
        stat[2] += q[i].c_views
        stat[3] += q[i].c_reposts

        i += 1

```

```

# считаем средние значение лайки и комментарии
if stat[0] != 0:
    stat[4] = stat[0] / len(q)
else:
    stat[4] = 0
if stat[1] != 0:
    stat[5] = stat[1] / len(q)
else:
    stat[5] = 0

# всего записей за период
stat[6] = len(q)

# сортируем по лайкам, комментам и дате и срезам 5 топ
результатов
# временная переменная для сортировки данных
data_sort = []
data_sort.append(sorted(post, key=itemgetter(3), reverse=True)
[:5]) # like
data_sort.append(sorted(post, key=itemgetter(4), reverse=True)
[:5]) # comm
data_sort.append(sorted(post, key=itemgetter(5), reverse=True)
[:5]) # view
data_sort.append(sorted(post, key=itemgetter(6), reverse=True)
[:5]) # repo
data_sort.append(sorted(post, key=itemgetter(7), reverse=True)
[:5]) # date
# data_sort.append(sorted(post, key=itemgetter(7),
reverse=True)[:5]) # er

form = getUrlsForm(request.POST)
data = {"d_public": posts[0], "stat": stat, "sort": data_sort,
"d_st": date_start, "d_end": date_end, "form": form}

# удалим все данные из таблицы, позже сделать удаление по
заложенному пользователю
Post.objects.filter(id_user=str(id)).delete()

# сохраняем временно (пока не нажмут кнопку Сохранить)
статистику в БД
save_reports(posts[0], stat, date_start, date_end, id)

return data

```