

БАЗА ДАННЫХ КОНТЕНТОВ С ПРИЗНАКАМИ ДЕСТРУКТИВНОСТИ**Е.Ю. Чапурин, Н.М. Лантюхов, П.Д. Федоров, А.О. Феоктистов, А.Ю. Егоров**

Статья посвящена созданию базы данных для хранения информации о выявленных деструктивных контентах, созданию математического обеспечения для поиска, аналитики и визуализации результатов работы с созданной базой данных. При помощи разработанного математического аппарата для созданной базы данных существует возможность поиска и анализа контентов с признаками деструктивности на любых информационных ресурсах. Структура базы данных позволяет хранить информацию о контенте любого типа. Разработано программное обеспечение, позволяющее автоматизировать работу с синтезированной базой данных. Разработанный программный комплекс применим для организации процесса риск-анализа и позволяет минимизировать затрачиваемое время, автоматизировать некоторые части данного процесса, а также организовать простое и понятное взаимодействие пользователя с защищенной базой данных контентов деструктивного характера. Полученные результаты могут быть использованы для проведения подробного процесса риск-анализа контентов и пользовательских комментариев, что поможет создать более детальные механизмы обеспечения безопасности конкретного пользователя и профилактики эпидемий в социальных сетях. На основе предложенного программного комплекса существует возможность реализовать механизм модерации контента до его фактической публикации на информационных ресурсах.

Ключевые слова: база данных, деструктивный контент, социальная сеть.

Введение

В современном мире все более остро выраженной становится задача управления киберпространством. Ярким примером последнего использования «армии троллей» является 72-й Центр информационно-психологических операций украинской армии, который был уничтожен в ходе спецоперации [1]. Резкий рост количества и качества интернет-технологий [2-10] стал основой для создания большого числа социальных сетей, предназначенных для общения пользователей и комфортного существования человека в современном мире, причем социальные сети зачастую являются плацдармом для размещения фейковой и деструктивной информации.

Объем и вредоносность контента, распространяемого в социальных сетях, нарастает с каждым днем: все больше и больше людей подвержены влиянию деструктивного контента за счет роста популярности социальных сетей. Финансовые объемы средств, вложенных в информационное противоборство, геометрически увеличиваются с каждым

годом, что создает дополнительные информационные угрозы. Отсюда актуальность их технического и правового регулирования не просто необходима, а нарастает с каждым днем.

К основным причинам привлекательности использования социальных сетей как среды для распространения деструктивного контента среди злоумышленников можно отнести следующие [11,12]:

- быстрое получение пользователем новостной информации, включая фейковые вбросы,

- общение без учета физического местоположения позволяет злоумышленникам избегать наказания за счёт трудностей, связанных с их поимкой,

- большой объём возможностей передачи медиаконтента и прочих типов информационного влияния,

- игровая среда, как источник дохода и средство мошенничества и «отмывания» денег,

- бегство от одиночества и реальной жизни пользователей, как катализатор восприимчивости индивида к деструктивному контенту.

При этом способы обхода защитных фильтров информационных ресурсов в настоящее время позволяют эффективно маскировать деструктивный контент под видом новостей, рекламных видео и прочих вариантов размещения информации, предназначенной для большого числа пользователей [13,14].

Одним из первых шагов на пути к защите пользователей от подобных источников деструктивного контента может стать система обнаружения, классификации и хранения деструктивного контента, работа которой основывается на анализе конкретного файла или информационного ресурса путём сравнения с имеющимися в базе данных видами деструктивного контента по характерным для них признакам. В результате подобного анализа система в автоматическом режиме определяет, является ли рассматриваемый объект деструктивным или нет.

Методическое обеспечение защищённой базы данных контентов с признаками деструктивности

Стандартные средства работы серверов баз данных являются фундаментом для построения всех баз данных в распределённых системах. Отсюда возникает необходимость использовать альтернативу данному методу, который заключается в создании системы, привязанной к системе управления базами данных (далее – СУБД). Работа с многоплатформенными СУБД в описанных вариантах никак не поддерживается. Из этого следует применение дорогостоящих серверов особенно там, где в этом нет такой необходимости. Проблема решается посредством создания программного комплекса данных СУБД, не имеющего программной привязки к самой базе данных (далее – БД). Основным отличием от встроенных СУБД является репликация данных серверов «MS SQL» на сервера «Interbase», «Sybase» или «Oracle». Причем подобное изменение всей сущности СУБД никак не изменяет логику самой СУБД. Поэтому разработчикам распределённых систем не придется каким-либо образом

изменять код, так как функционирование данной системы не нарушается.

Обоснование выбора средств и способов, используемых для создания защищённой базы данных деструктивных контентов

База данных о контенте с признаками деструктивности в существующих условиях работы должна соответствовать следующим требованиям к структуре и способам работы с ней:

- работа должна осуществляться с использованием минимального числа дополнительных модулей, необходимых для установки с целью обеспечения простоты работы и доступности,

- необходимо создать пользовательский интерфейс с понятным и простым набором элементов, который реализует необходимые функции для работы с базой данных: добавление записей, их редактирование, удаление, поиск и обновление данных, отображаемых в активном окне,

- для исключения возможности реализации атак на сервер, на котором расположена база данных, база данных должна располагаться локально,

- с целью исключения возможности реализации атак на программное обеспечение, предназначенное для работы с базой данных в контексте использования web-страниц для работы приложения, необходимо исключить использование web-сегмента для разработки,

- требуется использовать программные средства, позволяющие работать с базой данных на устройствах с различной архитектурой и установленной операционной системой.

Данные требования обусловлены спецификой работы и имеющимися условиями, связанными с техническими возможностями работы с базой данных. На основании сформированных требований можно представить следующие решения, удовлетворяющие данным требованиям:

- для работы с базой данных необходимо использовать два модуля: «tkinter» и «pymysql», которые отвечают за создание графического интерфейса и работу с базой данных соответственно. Их установка, как

дополнительных модулей, не требуется поскольку они интегрированы в основной модуль языка программирования «Python»,

- с помощью модуля «tkinter» создается графический интерфейс, реализующий возможности добавления, удаления, редактирования, поиска и обновления активного окна программы в виде соответствующих интерактивных элементов и дополнительных диалоговых окон, используемых для работы с базой данных;

- модуль «pymysql» позволяет работать с базой данных без web-страниц, используя только локально установленное приложение,

- использование встроенных криптографических средств языка программирования «Python» обеспечивает конфиденциальность передаваемых данных между приложением и базой данных, расположенной на выделенном сервере,

- создаваемое программное обеспечение реализует локальную работу с базой данных, тем самым исключая использование web-сегмента и обеспечивая безопасность,

- язык программирования «Python», используемый при реализации программного обеспечения, является мультиплатформенным и позволяет не привязываться к используемой операционной системе.

Архитектурное решение по управлению базами данных

При разработке архитектуры по управлению данными используются три основные компонента, которые связаны друг с другом:

- публикатор (издатель) - подписчик системы по управлению данными,
- интерфейс для доступа к системе управления базами данных,
- репликатор транзакций.

Актуальность такого подхода к выбору архитектуры заключается во внедрении дополнительного элемента – интерфейса управления системой баз данных. Особенностью репликатора транзакций является непрерывная отправка транзакционных сообщений. Таким образом реализуется синхронизация всех имеющихся в системе компонентов. Архитектура

системы управления базой данных представлена на рис. 1.



Рис. 1. Архитектура управления базой данных

Обмен транзакционными пакетами осуществляется с помощью стека сетевых протоколов TCP/IP. В данном случае издатель является серверной частью системы, а клиент может получить доступ с помощью механизма соединения, используемого в вышеуказанном стеке протоколов, путем указания требуемых параметров подключения. Таким образом, любое устройство, при наличии соответствующих сетевых интерфейсов, может осуществлять работу с имеющейся системой. Для функционирования системы может использоваться распределенная компьютерная сеть, в которой параметры работы зависят, в первую очередь, от задержки обработки на сетевых устройствах и времени ответа узлов компьютерной сети. Схема связи публикатора и подписчика представлена на рис. 2.

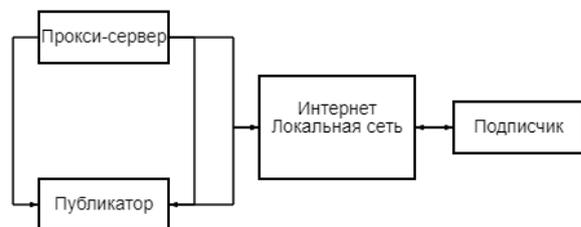


Рис. 2. Схема связи между публикатором и подписчиком

Для обеспечения надежности и отказоустойчивости системы управления базами данных используется программное обеспечение, такое как прокси-сервера,

проксирующие через себя трафик для обеспечения безопасности при взаимодействии с системой (рис. 3).

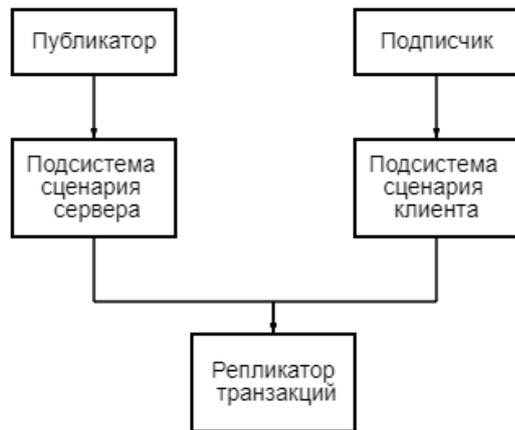


Рис. 3. Структурная схема механизма Публикатор – Подписчик

При обработке запросов от клиента и издателя используются соответствующие сценарии для подсистем. Запрос, отправленный клиентом или публикатором, передается сначала на подсистему сценария клиента или сервера, а после взаимодействие происходит по сетевым протоколам через репликатор транзакций. После окончания работы, соединение между публикатором и подписчиком разрывается. Работа публикатора основана на пуле соединений, которые дают возможность нескольким клиентам использовать одно и то же соединение из пула для построения взаимодействия. Механизмы работы сценариев позволяют им взаимодействовать с репликатором транзакций, далее рассмотрим его структуру.

Структура репликатора транзакций

Репликатор транзакций является главным компонентом системы и используется для реализации сценариев подсистем для публикатора и подписчика. Его основной задачей является управление и рассылка транзакций. Особенностью данного компонента является инкапсуляция данных и работа с ними на уровне сетевых взаимодействий, в соответствии с реализацией стека протоколов TCP/IP. Аутентификация и авторизация подписчика происходит посредством получения

цифрового ключа, его обработки и дешифровки. После успешного соединения на сервере базы данных, возвращается код положительного соединения, исполнение сценариев публикатора и подписчика продолжается. Далее идет проверка на редактирование данных. При повреждении структуры данных, подсистемой будет сформирован новый пакет, который через механизм упаковки будет передан соответствующей подсистеме сценария.

Подписчик получает сообщение и в своей реплике обновляет соответствующие данные. Подсистема управления транзакциями определяет эти изменения начинает оптимизацию с помощью механизма оптимизации транзакций и передает полученный эшелон данных ПФПД. После получения клиентом пакетов, они отправляются в подсистему управления транзакциями, и, в случае расхождения, управление передается подсистеме по управлению временем транзакций. Этот процесс работает до тех пор, пока не закончится список проверяемых пакетов. После окончания работы в базу данных записывается информация о прекращении сессии издателя.

Структура модуля доступа к СУБД

С помощью механизмов авторизации в СУБД другим подсистемам предоставляется доступ к работе с данными через соответствующий интерфейс (рис. 4).

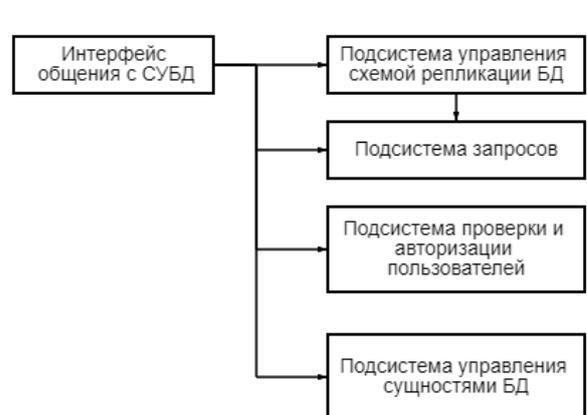


Рис. 4. Структура реализации механизма доступа к данным в системе

Интерфейс взаимодействия с базой данных – это набор программных решений, необходимых для реализации возможности

клиентов управления данными в базе данных. Подсистема управления работой базой данных используется для создания и редактирования ее физической и логической структуры, объектов репликации, таких как:

- отношения,
- атрибут отношения,
- виртуальные таблицы,
- хранимые функции,
- индексы,
- ключи: основные, внешние.

Этой информации достаточно для схемы реляционной базы данных. Подсистема управления сущностями дает возможность делать триггеры для реплицируемых таблиц.

Модель обслуживания работы с базой данных

Описанными выше моделями обслуживается система обработки данных, которые, в свою очередь, создаются и изменяются в процессе работы подсистемы обслуживания. Выявление и однозначная идентификация участвующих в работе данных сопровождается необходимостью настройки реплицируемых сущностей. Для однозначной идентификации изменяемой, удаляемой или вставляемой записи требуется создание для этой таблицы поля, значения которого уникальны во всех распределённых узлах системы работы с базой данных.

Для каждой таблицы необходимо создание триггеров для выявления данных, которые участвуют в работе, и последующего сохранения их в журнал транзакций. Для триггера вставки и удаления необходимо выявление идентификатора таблицы и записи. Для триггера изменения требуется также идентификатор поля. Общий параметр, который используется при срабатывании триггеров, есть ни что иное, как идентификатор пользователя БД. После определения параметров, необходимых для создания триггеров выявления данных, участвующих в работе, система создаёт триггеры для таблиц, которые описаны в схеме работы. Приведённое структурное изменение произойдёт для всех таблиц, составляющих схему работы с базой данных.

При реализации методического обеспечения используются следующие программные модули:

- «tkinter»;
- «pymysql».

База данных в данном случае будет иметь высокую степень реляционности с необходимым и достаточным количеством таблиц, каждая из которых имеет свой внешний ключ, по которому будет производится выборка для тех или иных запросов на поиск данных.

Таблицы базы данных построены исходя из полей основной таблицы в базе данных, а именно:

- ID – primary key, значение типа integer. Необходим для внутреннего подсчёта записей в таблицах и обращения по данному атрибуту к ним,

- description – текстовый атрибут для занесения необходимой информации о контенте, содержащий данные в строковом виде,

- activity – атрибут, хранящий данные о том, какое количество пользователей просмотрело данный пост, данные содержатся в числовом виде,

- link – атрибут содержит ссылку на сам пост, по которой к нему можно перейти. Имеет строковый тип данных для возможности хранения различных символов, встречающихся в ссылках,

- author – атрибут содержащий данные об авторе поста в случае, если его возможно установить. Атрибут имеет текстовый тип данных, поскольку зачастую пользователи используют не настоящие имена, а ники, в которых могут содержаться различные символы,

- comments – количество комментариев. Атрибут имеет целочисленный тип и предназначен для хранения числа комментариев.

Описанные атрибуты будут использованы для хранения и обработки информации о контенте с признаками деструктивности, содержащихся в базе данных. В дальнейшем данная таблица будет являться набором внешних ключей для дочерних таблиц с целью повышения реляционности и упрощения поиска информации по различным атрибутам.

Так, каждый из атрибутов описанной выше таблицы, используется для взаимодействия с таблицами, характеризующими каждый из атрибутов более детально. Для базы данных предлагается следующая архитектура:

– *description* – атрибут является внешним ключом на таблицу, в которой более подробно рассматривается контент и его первичные признаки:

1) наличие ключевых слов, по использованию которых можно судить о его деструктивности. Проверка вхождений слов должна производиться из словаря,

2) наличие ссылок, с призывом перехода на сторонние ресурсы,

3) краткая характеристика контента;

– *activity* – атрибут является внешним ключом на таблицу, в которой более подробно рассматривается контент и его первичные признаки:

1) количество лайков,

2) количество дизлайков,

3) активность от зарегистрированных на ресурсе пользователей,

4) активность от незарегистрированных на ресурсе пользователей,

5) средняя длительность просмотра поста;

– *link* – атрибут является внешним ключом на таблицу, в которой более подробно рассматривается контент и его первичные признаки:

1) ресурс размещения,

2) доменная зона,

3) доступность поста (переход возможен только по ссылке/ доступен только определённой группы пользователей/ доступен в поисковой выдаче);

– *author* – атрибут является внешним ключом на таблицу, в которой более подробно рассматривается контент и его первичные признаки:

1) зарегистрирован ли он на сайте,

2) наиболее посещаемые разделы сайта,

3) активность на сайте (какие разделы просматриваются, реакция на них);

– *comments* – атрибут является внешним ключом на таблицу, в которой более подробно рассматривается контент и его первичные признаки:

1) количество положительных комментариев,

2) количество негативных комментариев,

3) наличие призывов к противоправным действиям.

Говоря о взаимодействии с базой данных, стоит обратить внимание на использование десктопного приложения, позволяющего работать с базой данных, размещенной на локальном сервере с использованием шифрования.

Это упрощает работу с базой данных и позволяет избежать реализации атак на веб-сервер в случае размещения базы данных на нём.

Данное взаимодействие основывается на использовании пользовательского интерфейса, позволяющего выполнять необходимые операции с записями в базе данных, созданного с помощью модуля «tkinter» языка программирования «Python».

Сама база данных реализовывается при использовании модуля « pymysql », предназначенного для работы с базой данных.

Помимо описанных выше модулей, также используется дополнительный модуль « encryption », предназначенный для шифрования данных, передаваемых между сервером и устройством. Данный подход используется для обеспечения безопасности работы и стабильности соединения с сервером.

Взаимодействие с распределенной базой данных построено на основе реализации sql запросов непосредственно к базе данных через библиотеку, необходимую для подключения к системе управления базой данных, а также вызовов функций работы с ней при помощи элементов интерфейса приложения.

Сами запросы передаются в json формате от приложения к серверу в зашифрованном виде, что обеспечивает дополнительную безопасность обмена информацией.

Доступ для работы с базой данных осуществляется после аутентификации и авторизации на сервере приложения, что исключает возможность обращения к базе

данных неавторизованных пользователей. Подход имеет следующие преимущества:

- простота развертывания,
- простота реализации,
- скорость работы,
- защищенность от атак.

Механизмы работы пользовательского интерфейса

Механизм добавления записей в базу данных реализован с помощью: класса, описывающего инициализацию функций добавления данных, элементов интерфейса, связанных с данными функциями и методов, реализующих работу с БД.

При запуске пользователем программы происходит инициализация главного окна и вывод всех элементов управления с

заданными параметрами, в частности, функция добавления данных реализована при помощи кнопки «Добавить запись», которая размещена на главной странице программы и работает с соответствующим экземпляром класса, реализующим метод «Child_add», который объявляется и инициализируется при нажатии на соответствующую кнопку. При нажатии пользователем на кнопку «Закрыть», программа будет завершена, а сессия с БД закрыта. Имеется возможность обновить информацию, получив ее из БД с помощью соответствующей кнопки в меню управления. Интерфейсное окно ввода данных представлено на рис. 5.

Рис. 5. Диалоговое окно добавления данных

После этого вызывается дополнительное диалоговое окно, в котором производится ввод данных, с помощью введения их в соответствующие поля и нажатия кнопки «Добавить». Нажатие данной кнопки, в свою очередь, вызывает методы класса,

предназначенного для работы с БД, и с их помощью записывает введенные данные в базу. Результат добавления данных выводится в главном окне. Результат работы представлен на рис. 6.

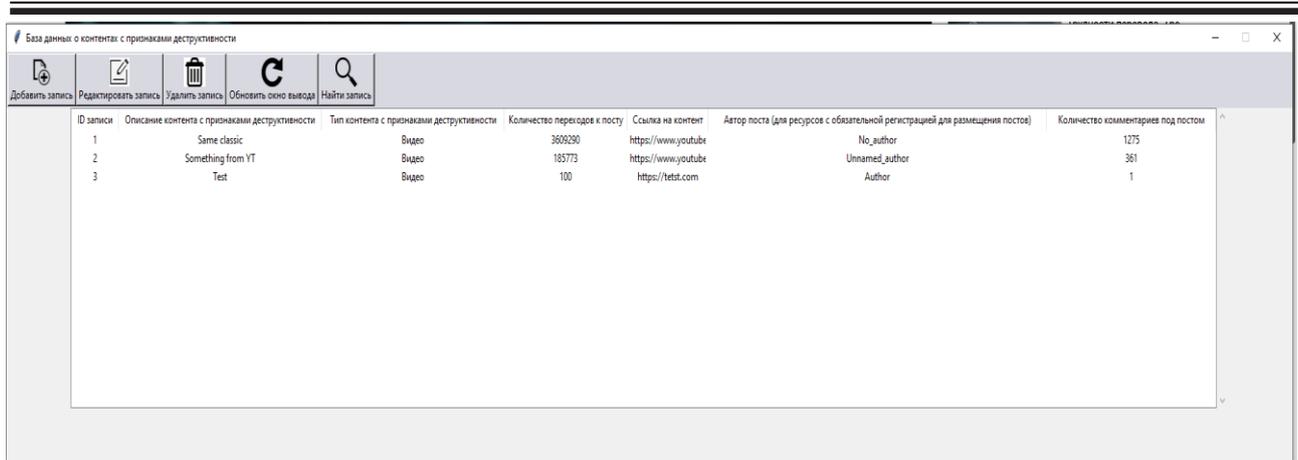


Рис. 6. Диалоговое окно после добавления данных

При редактировании записей происходит инициализация дополнительного окна и вывод элементов управления программы с заданными параметрами. В частности, функция редактирования данных реализована при помощи кнопки «Редактировать запись», которая размещается на главном окне и вызывает метод экземпляра класса «Child_update»,

который объявляется и инициализируется программой при нажатии на соответствующую кнопку. В этом окне имеется возможность сортировки выбранных данных по заданному пользователем полю. Интерфейсное окно редактирования данных представлено на рис. 7.

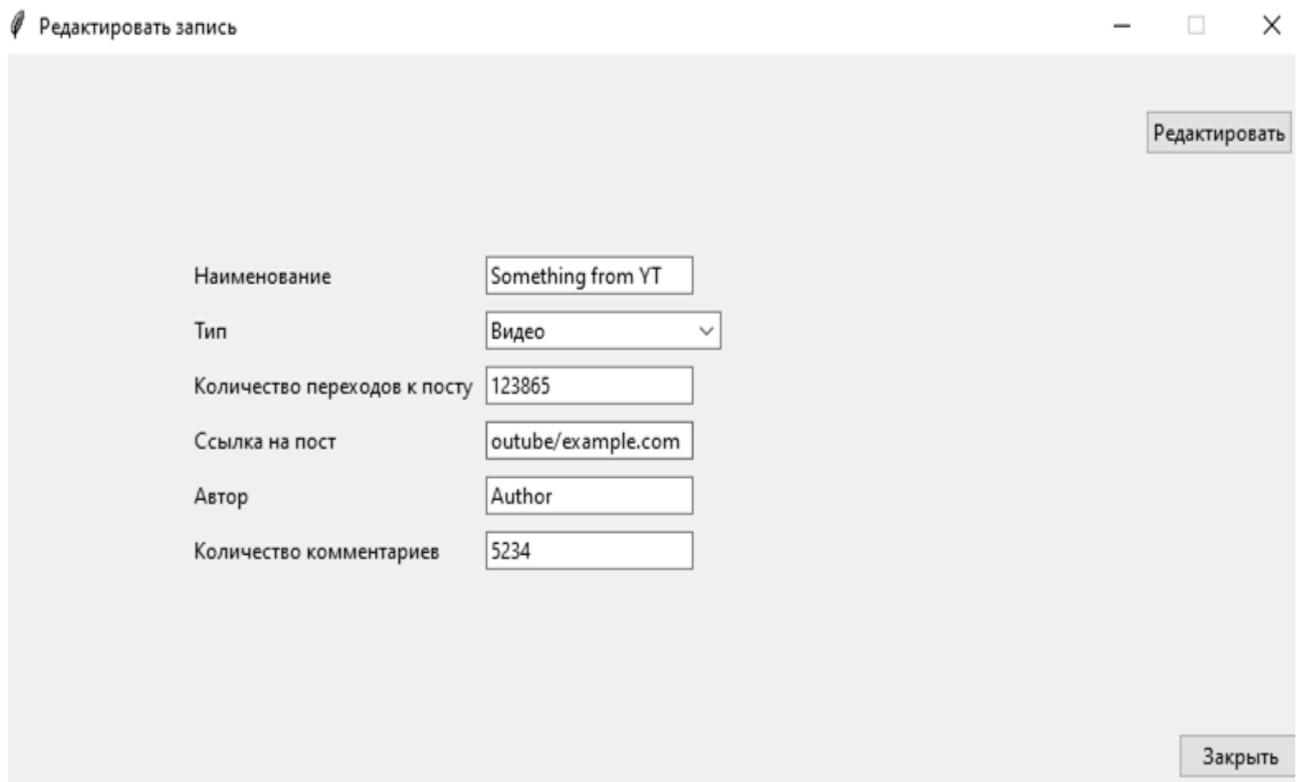


Рис. 7. Диалоговое окно редактирования данных

После этого вызывается дополнительное диалоговое окно, в котором производится ввод данных, с помощью введения их в соответствующие поля и нажатия кнопки «Редактировать». Нажатие данной кнопки вызывает методы класса, предназначенного

для работы с БД, и с их помощью записывает введенные данные в базу. Результат редактирования данных выводится в главном окне. Результат работы представлен на рис. 8.

ID записи	Описание контента	Тип контента	Количество просмотров	Ссылка на контент	Автор поста	Количество комментариев	Риск распространения контента
5062	[club9884911]Овеч, 12 апреля	Картинка	133478	vk.com/public8254824	624016224	176	0.00046107179167
5063	#allsigns@theastro	Текстовый комментарий	201370	vk.com/public8253814	539780795	148	0.001128479759844
5064	#allsigns@theastro	Ссылка	297835	vk.com/public8253362	353126491	334	0.001146253492167
5065	#advice@theastro	Текстовый пост	164871	vk.com/public8252876	502261088	222	0.000599940326638
5066	#allsigns@theastro	Картинка	318749	vk.com/public8252665	486054537	300	0.001215662461186
5067	#allsigns@theastro	Видео	185724	vk.com/public8252381	636334334	318	0.000627602699172
5068	#allsigns@theastro	Текстовый комментарий	150217	vk.com/public8250748	513890284	49	0.0004499898959227
5069	#allsigns@theastro	Ссылка	319969	vk.com/public8250275	524544289	157	0.001493750869685
5070	[club9884911]Рыбы, 11 апреля	Картинка	170785	vk.com/public8250179	442306615	84	0.000462218230317
5071	[club9884911]Весы, 11 апреля	Текстовый комментарий	138636	vk.com/public8249874	90500534	58	0.000348010393771
5072	#allsigns@theastro	Видео	161187	vk.com/public8249800	192490190	75	0.12089025
5073	[club9884911]Девя, 11 апреля	Картинка	137798	vk.com/public8249709	464210474	52	0.077165496
5074	[club9884911]Лев, 11 апреля	Картинка	139147	vk.com/public8249626	615041943	104	0.14471288
5075	стартуем гороскоп! ???	Текстовый пост	91026	vk.com/public8249261	409933343	20	0.0182052
5078	Something from YT	Видео	123865	https://youtube.com	Author	5234	64.830941

Рис. 8. Диалоговое окно после редактирования данных

Поиск записей происходит при помощи дополнительного окна, в который вводится текст, на основании которого формируется запрос в базу данных с заданными параметрами, в частности, функция поиска данных реализована при помощи кнопки

«Найти запись». Она размещается на главном окне и вызывает экземпляр класса «Search», который инициализируется при нажатии на соответствующую кнопку. Окно редактирования данных представлено на рис. 9.

Рис. 9. Диалоговое окно поиска данных

После этого вызывается дополнительное диалоговое окно, в котором производится ввод данных, с помощью введения их в соответствующее поле и нажатия кнопки «Поиск». Нажатие данной кнопки вызывает методы класса, предназначенного для

работы с базой данных, и с их помощью записывает введенные данные в базу. Результат поиска данных выводится в главном окне. Результат работы представлен на рис. 10.

ID записи	Описание контента	Тип контента	Количество просмотров	Ссылка на контент	Автор поста	Количество комментариев	Риск распространения контента
4811	#allsigns@theastro	Видео	585056	vk.com/public8331499	278435913	2642	0.002802405333028
4812	#allsigns@theastro	Видео	406892	vk.com/public8331250	515744602	181	0.001744188014963
4813	#allsigns@theastro	Ссылка	461469	vk.com/public8330906	515744602	137	0.001651437740508
4814	#allsigns@theastro	Текстовый комментарий	205186	vk.com/public8330747	505179651	70	0.000628757124196
4816	#allsigns@theastro	Картинка	197524	vk.com/public8329877	184951732	96	0.0006463960505429
4823	#allsigns@theastro	Текстовый комментарий	252388	vk.com/public8327986	515744602	214	0.000882355928258
4824	#allsigns@theastro	Ссылка	171662	vk.com/public8327763	515744602	126	0.000813541840217
4827	#allsigns@theastro	Видео	155152	vk.com/public8327299	515744602	65	0.00050879469265
4829	#allsigns@theastro	Текстовый пост	196607	vk.com/public8326348	186638475	169	0.001075995228347
4848	#allsigns@theastro	Текстовый пост	232438	vk.com/public8321838	316774539	268	0.000754948280705
4849	#allsigns@theastro	Ссылка	322108	vk.com/public8321471	460891154	287	0.00130493223047
4850	#allsigns@theastro	Ссылка	408048	vk.com/public8321327	7761246	190	0.00139071385328
4852	#allsigns@theastro	Текстовый пост	234529	vk.com/public8319585	515744602	155	0.001233807974096
4853	#allsigns@theastro	Ссылка	169967	vk.com/public8318754	325610209	148	0.000954949215566
4855	#allsigns@theastro	Видео	317720	vk.com/public8318412	515744602	203	0.001672000388963

Рис. 10. Диалоговое окно после поиска данных

В программе так же реализован поиск по ссылке на пост или комментарий для

сохранения принципов вложенности и реляционности сущностей БД. Для поиска по

ссылке имеется отдельный элемент интерфейса, который инициализирует поиск по атрибуту ссылки и позволяет найти как сам пост, так и все ответы на него по

ключевому слову. Результат работы функции поиска представлен на рис. 11.

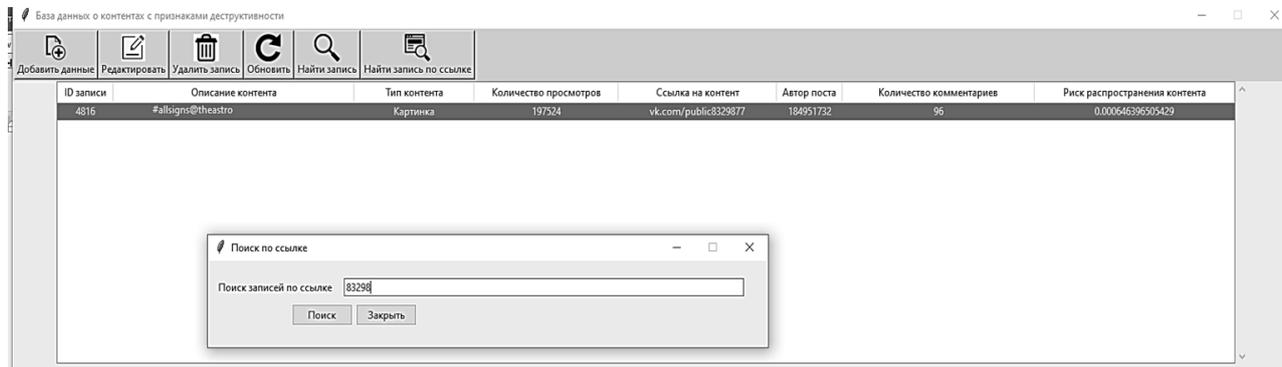


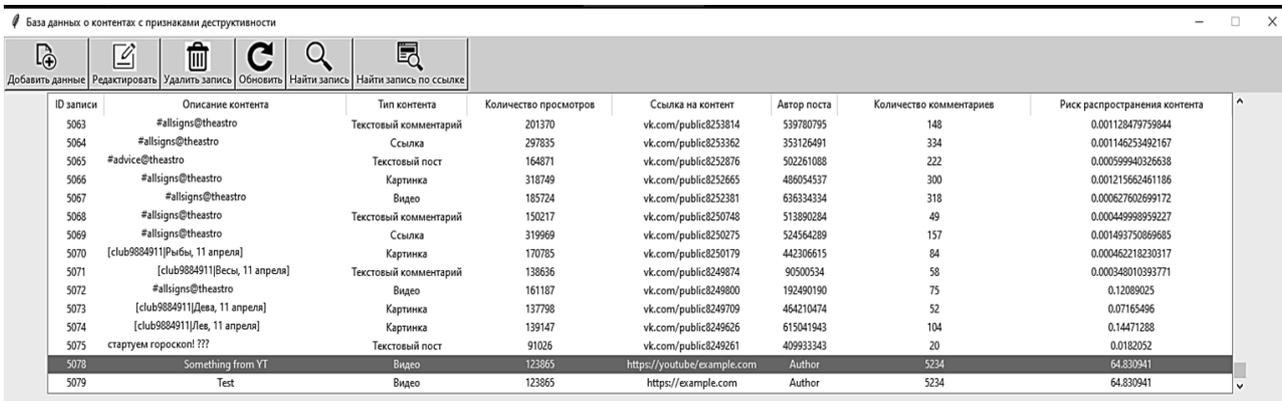
Рис. 11. Диалоговое окно поиска по ссылке и результат выполнения данной операции

Кнопка обновления главного стола при помощи функций модуля «tkinter» реализует сброс фильтров отображения данных из базы после поиска для отображения всех сведений, содержащихся в базе данных

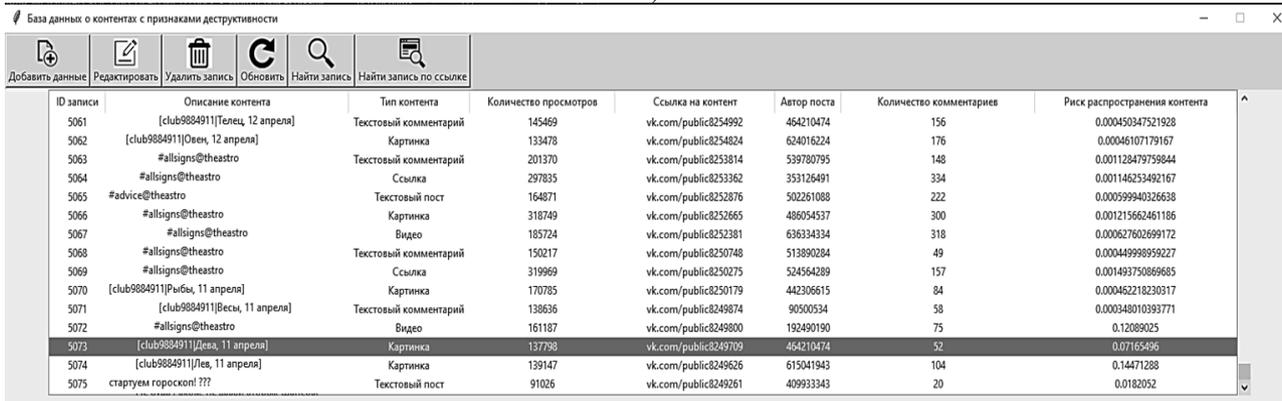
Удаление записей происходит при помощи инициализации дополнительного окна, в который вводится искомый текст и на основании которого формируется запрос в БД с заданными параметрами. В частности,

функция удаления данных реализована при помощи кнопки «Удалить запись», которая размещается на главном окне и вызывает функцию «delete_records», инициализирующуюся при нажатии на соответствующую кнопку.

После этого происходит удаление данных из базы при помощи выполнения SQL скриптов. Результат удаления данных выводится в главном окне (рис. 12).



а)



б)

Рис. 12. Диалоговое окно а) – до, б) – после

Говоря о методах работы интерфейса с базой данных, можно дать характеристику их работе как набору классов с внутренними методами, инициализация которых происходит в момент срабатывания триггера.

После активации триггера происходит инициализация экземпляров классов, отвечающих за реализацию определенной части функционала, которая выполняется при вызове описанных в классе методов.

Все классы реализуются в методах работы с БД при помощи модуля «sqlite3», который позволяет использовать конструкции языка sql при взаимодействии с элементами интерфейса для работы с БД.

Пользователю доступны следующие возможности:

- получение любого количества записей из выбранной БД в виде таблицы, в столбцах которой указаны необходимые метрики, а в строках – значения,
- поиск записи в БД по выбранному критерию или метрике,
- фильтрация данных по одному или нескольким выражениям,
- сортировка записей по выбранной метрике, а также выбор способа сортировки,
- сохранение выбранных данных в виде текстового документа или таблицы,
- возможность вручную добавлять, удалять, изменять записи в выбранной БД.

Обслуживание баз данных

Обслуживание БД ресурсов и БД деструктивных контентов происходит в автоматическом режиме с помощью скриптов, которые включают в себя:

- обновление и актуализацию данных,
- удаление старых и недостоверных данных,
- проверка корректности добавляемых данных.

Обновление и актуализация данных происходит по следующему алгоритму:

- проверка метрик деструктивного контента или ресурса на изменение данных,
- новые значения метрик записываются в БД,

- для БД деструктивных контентов пересчитываются значения вовлеченности и величины риска по известным формулам.

Удаление старых и недостоверных данных предоставляет возможность убрать из БД данные, дата записи которых превышает указанную (задается в параметрах скрипта). Это позволяет в автоматическом режиме актуализировать список используемых ресурсов или деструктивных контентов. Также данные проверяются на логическое соответствие полям и недостоверные записи удаляются.

Данный скрипт предполагается запускать вручную, либо по расписанию, которое будет выбрано исходя из интенсивности получаемых данных.

Проверка корректности добавляемых данных. Запуск скрипта происходит при добавлении новой записи в БД. Он проверяет каждую метрику на соответствие типу данных, значению и так далее. На рис. 13 приведена блок-схема работы запрограммированного алгоритма работы с БД.

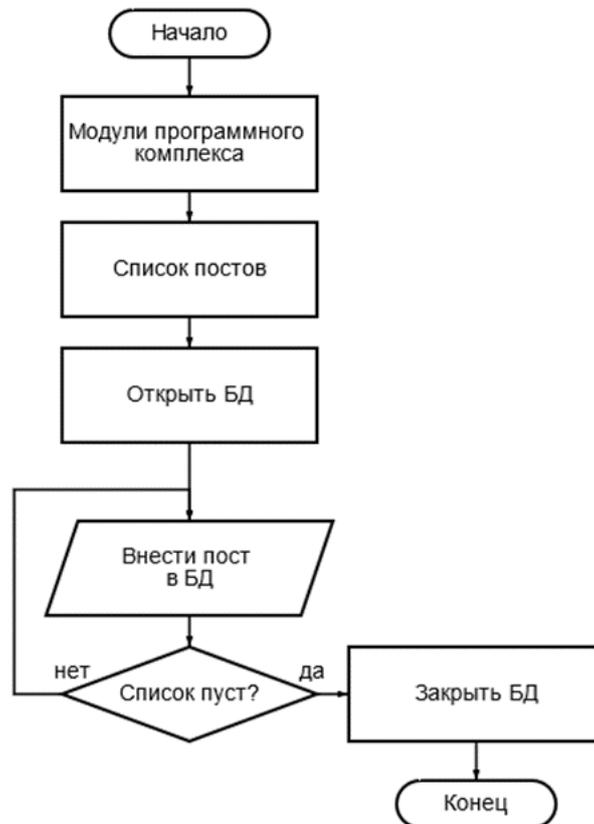


Рис. 13. Алгоритм внесения контента, признанного деструктивным, в базу данных

Заключение

Для более эффективной борьбы с деструктивным контентом была разработана база данных, в которой хранится информация о всех выявленных случаях такого контента. Для работы с базой данных было создано математическое обеспечение, позволяющее проводить поиск, анализ и визуализацию результатов. Разработанный программный комплекс позволяет автоматизировать процесс риск-анализа и упрощает взаимодействие пользователя с базой данных. Результаты работы с базой данных могут использоваться как для проведения детального анализа контента, так и предотвращения его публикации на информационных ресурсах. Возможность модерации контента до публикации помогает дополнительно обезопасить пользователей и предотвратить эпидемии в социальных сетях.

Список литературы

1. Без «Байрактаров» и Центра информационно-психологических операций: что уничтожено на Украине. Интернет-

ресурс.

<https://www.mk.ru/politics/2022/02/24/bez-bayraktarov-i-centra-informacionnopsikhologicheskikh-operaciy-chto-unichtozheno-na-ukraine.html> (дата обращения: 04.03.2022)

2. Остапенко А.Г. «Инфодемия» и социальные сети: индуцированные риски и шансы / А.Г. Остапенко, Е.А. Шварцкопф, А.А. Остапенко, Т.Ю. Мирошниченко, С.Д. Трубицын, Н.М. Лантюхов, А.Н. Бартенев. // *Информация и безопасность*. 2020. Т. 23. Вып. 2. С. 235-244.

3. Остапенко А.Г. «Инфодемия» и социальные сети: модели эпидемического процесса. / А.Г. Остапенко, Е.А. Шварцкопф, А.А. Остапенко, Д.А. Нархов, П.Д. Федоров, Р.В. Сорокин. // *Информация и безопасность*. 2020. Т. 23. Вып. 2. С. 285-290.

4. Остапенко А.Г. «Инфодемия» и социальные сети: актуальные объекты и задачи исследования. / А.Г. Остапенко, Р.В. Сорокин, С.В. Лихобабин, А.О. Ткаченко, А.Н. Бартенев, Ю.Г. Пастернак. //

URL:

Информация и безопасность. 2020. Т. 23. Вып. 4. С. 535-544.

5. Остапенко А.Г. Основы метрологии контентов для мониторинга социальных сетей на предмет обеспечения информационной безопасности. Часть 1. / А.Г. Остапенко, Е. С. Соколова, А. В. Ещенко, А. А. Остапенко, Т.Ю. Чапурина. // Информация и безопасность. 2019. Т. 22. Вып. 2. С. 170-180.

6. Остапенко А.Г. Основы метрологии контентов для мониторинга социальных сетей на предмет обеспечения информационной безопасности. Часть 2. / А.Г. Остапенко, Е. С. Соколова, А.В. Ещенко, А. А. Остапенко, Т.Ю. Чапурина. // Информация и безопасность. 2019. Т. 22. Вып. 2. С. 181-187.

7. Остапенко А.Г. Программное обеспечение для мониторинга процессов восприятия и распространения деструктивных контентов в социальных сетях. / А.Г. Остапенко, Е.Ю. Чапурин, Е. С. Соколова, А.Г. Зимницкий, И.А. Боков, С.В. Лихобабин, А.О. Ткаченко, А.И. Дегтярев. // Информация и безопасность. 2019. Т. 22. Вып. 2. С. 188-205.

8. Остапенко А.Г. Мониторинг и управление рисками социо-информационного пространства в целях обеспечения региональной и национальной безопасности. / А.Г. Остапенко, В.П. Железняк, Е.Ю. Чапурин, А.А. Остапенко, С.Д. Трубицын, О.А. Остапенко, Т.Ю. Мирошниченко. // // Информация и безопасность. 2019. Т. 22. Вып. 4. С. 557-572.

9. Остапенко А.Г. Метрики деструктивного контента на видеохостинге Youtube. / А.Г. Остапенко, В.Е. Кунавин, В.С. Сидельникова, О.А. Остапенко // Информация и безопасность. 2018. Т. 21. Вып. 3. С. 284-289

10. Остапенко А.Г. Организация мониторинга постов социальной сети вконтакте с помощью интерфейса VKAPI / А.Г. Остапенко, Е.Р. Нежелский, М.Н. Степанов, Е. Ружицкий, А.В. Заряев // Информация и безопасность. 2018. Т. 21. Вып. 3. С. 408-415.

11. Официальный сайт Российского фонда фундаментальных исследований. Интернет-ресурс. URL: <https://www.rfbr.ru/rffi/ru/>.

12. Шматова Ю.Е. Динамика статистических и социологических показателей состояния психического здоровья населения России // Проблемы развития территории, №3 (101), 2019, С. 76-96.

13. Капитанаки В.Е., Чермянин С.В., Иванов О.С. Направления психологической помощи населению в ситуации COVID-crisis // Вестник Тверского государственного университета. Серия: Педагогика и психология. 2020. № 2 (51). С. 72- 83.

14. Дюк В.А., Сенкевич Ю.И., Цветков О.В. Исследование влияния фазоконтрастных визуальных стимулов на спектральные характеристики ЭЭГ // Биотехносфера, №5-6 (17-18), 2011, С. 11-17

Правительство Воронежской области
Government of the Voronezh Region

Воронежский государственный технический университет
Voronezh State Technical University

Поступила в редакцию 15.03.2023

Информация об авторах

Чапурин Евгений Юрьевич – главный советник Правительства Воронежской области; ассистент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Лантюхов Никита Михайлович – студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Федоров Павел Денисович – студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Феоктистов Артем Олегович – студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Егоров Анатолий Юрьевич – студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

DATABASE OF CONTENT WITH SIGNS OF DESTRUCTIVENESS

E.Yu. Chapurin, N.M. Lantyukhov, P.D. Fedorov, A.O. Feoktistov, A.Yu. Egorov

The article is devoted to the creation of a database for storing information about the identified destructive content, the creation of mathematical software for search, analytics and visualization of the results of work with the created database. With the help of the developed mathematical apparatus for the created database, it is possible to search and analyze content with signs of destructiveness on any information resources. The database structure allows you to store information about any type of content. The software allowing to automate work with the synthesized database is developed. The developed software package is applicable to the organization of the risk analysis process and allows you to minimize the time spent, automate some parts of this process, as well as organize a simple and understandable user interaction with a secure database of destructive content. The results obtained can be used to conduct a detailed process of risk analysis of content and user comments, which will help create more detailed mechanisms to ensure the safety of a particular user and prevent epidemics in social networks. Based on the proposed software package, it is possible to implement a mechanism for content moderation before its actual publication on information resources.

Keywords: database, destructive content, social networks.

Submitted 15.03.2023

Information about the authors

Evgeny Yu. Chapurin – Assistant, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

Nikita M. Lantyukhov – Student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

Pavel D. Fedorov – Student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

Artem O. Feoktistov – Student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

Anatoly Yu. Egorov – Student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com