Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Российский государственный профессионально-педагогический

университет»

Институт инженерно-педагогического образования

Кафедра информационных систем и технологий

Курсовой проект

по дисциплине «Инженерия программного обеспечения»

тема «Программа для ограничения трафика социальных сетей

на 2 часа»

Работу выполнили:

студенты группы ИЭ-402п И.С. Кильмяшкина

К. А. Неустроева

Н.К. Бельская

К.И. Пшеничников

С.Д. Валеев

Руководитель: М. Ю. Черноскутов

# Оглавление

[Введение 3](#_Toc532629576)

[1 Концентрация 5](#_Toc532629577)

[1.1 Концентрация внимания, виды и следствия 5](#_Toc532629578)

[1.2 Способы концентрации 7](#_Toc532629579)

[2 Техническое задание 8](#_Toc532629580)

[3 Теоретические основы 14](#_Toc532629581)

[3.1 Основные понятия 14](#_Toc532629582)

[3.2 Методы и переменные для WPF 15](#_Toc532629583)

[3.3 Методы для задания счетчика 16](#_Toc532629584)

[3.4 Блокировка доступа к социальным сетям 17](#_Toc532629585)

[4 Описание продукта и выполненные работы 21](#_Toc532629586)

[4.1 Перечень выполненных работ 21](#_Toc532629587)

[4.2 Описание продукта 26](#_Toc532629588)

[Заключение 34](#_Toc532629589)

[Список использованных источников 35](#_Toc532629590)

# Введение

В современном мире сложно сохранить концентрацию и внимание на выполнении одной задачи. Стоит взяться за дело, как из социальных сетей начинает приходить поток уведомлений и сообщений. Как показывает практика, большинство людей не могут игнорировать все внешние раздражители. При выполнении определенных задач они отвлекаются на любые звуки, особенно если этот звук - уведомление из социальной сети. Человек, который не обладает повышенным вниманием к выполнению поставленной задачи, обязательно отвлечётся и зайдет в интернет-сообщество, что займет много времени.

Пара минут на проверку ленты в Facebook, еще пара минут на просмотр видео и фото в Instagram. Из таких «пары минут» собирается огромное количество времени проведенное бесполезно. Именно для концентрации внимания было разработано программное обеспечение, чтобы помочь пользователям сформировать привычки к неотрывной работе на внешние раздражители.

Данный курсовой проект был разработан в среде Microsoft Visual Studio 2017. Интегрированная среда разработки Microsoft Visual Studio – это среда запуска, которая позволяет создавать, редактировать и отлаживать код приложения, а затем его публиковать. Помимо стандартного редактора и отладчика, Visual Studio включает в себя средства выполнения кода, графические конструкторы и многие другие функции для упрощенного процесса разработки программного обеспечения [1].

Приложение реализовано в виде простого, интуитивно-понятного интерфейса при помощи WPF (Windows Presentation Foundation) в Visual Studio. WPF предоставляет разработчикам возможность создавать клиентские приложения для настольных Windows.В возможности разработки приложений с помощью модуля визуализации в WPF можно отнести элементы управления, привязку данных, шаблоны, стили, графику, анимацию, текст и многие другие полезные доя разработки инструменты. Многие возможности WPF применены в рамках разработанного курсового проекта.

Реализованное программное обеспечение подходит, как для домашнего использования, так и для профильного в организациях.

Конечным пользователем программы может быть любое лицо, ознакомившееся с руководство пользователя и использующее операционную систему Windows, браузер Internet Explorer.

В ходе разработки курсового проекта были выявлены наиболее часто посещаемые социальные сети пользователей браузера Internet Explorer, ими оказались Instagram, Facebook, Twitter, Одноклассники и сеть Вконтакте. Именно к ним было реализовано автоматическое ограничение доступа.

Актуальность данной работы обусловлена тем, что возникла потребность в разработке программного обеспечения, которое позволило бы пользователям Internet Explorer сфокусироваться на выполнении одной задачи, не отвлекаясь на социальные сети.

Объектом исследования являются процесс работы пользователей в браузере Internet Explorer.

Предметом исследования является – анализ взаимодействия пользователей с социальными сетями через Internet Explorer.

Цель работы – разработка программного продукта для концентрации внимания пользователей социальных сетей в Internet Explorer.

В соответствии с поставленной целью в работе определены следующие задачи:

* исследовать предметную область, провести анализ работы пользователей браузера Internet Explorer;
* проанализировать Internet-источники, посвященные аналогичным программным продуктам;
* разработать проект программного продукта;
* реализовать программный продукт в среде Microsoft Visual Studio.

# 1 Концентрация

1.1 Концентрация внимания, виды и следствия

Концентрация внимания на важных вещах повышает эффективность и работоспособность, а также отличается хорошей наблюдательностью и организацией. Повышенная концентрация позволяет точно, быстро и аккуратно выполнить работу.

В среднем, учеными зафиксировано, что максимальна концентрация внимания возможна от 30 до 90 секунд, после этого срока, способность «собраться» возможна лишь благодаря повышенному вовлечению и определенными усилиям: волевыми и физическим.

Плохая концентрация и потеря внимательности может быть спровоцирована следующими факторами [5]:

* недостаточный сон;
* неполноценное питание;
* включенный телевизор;
* шум транспорта;
* посторонние разговоры и действия;
* усталость;
* стресс и депрессия;
* лекарственное или алкогольное воздействие.

Чтобы исключить нарушение концентрации внимания во время работы, и обеспечить развитие концентрации необходимо огородить себя от других воздействий.

Внимание – это сосредоточенность и направленность психической деятельности на определенный объект, что предполагает повышение уровня сенсорной, интеллектуальной и двигательной активности. Внимание является базовым психическим процессом, «питающим» все другие психические функции и виды деятельности. Оно обеспечивает организованный и целенаправленный отбор поступающей информации, избирательную и длительную сосредоточенность психической активности на объекте или деятельности, а также направленность и избирательность познавательных процессов. Вниманием определяется точность и детализация восприятия, прочность и избирательность памяти, направленность и продуктивность мышления и воображения.

Внимание может иметь следующие разновидности [5]:

* произвольное – это сознательная и целенаправленная концентрация внимания на каком-либо объекте или действии, которые связаны с интересом, профессиональной деятельностью, обучением или прочей необходимостью;
* непроизвольное – возникает неосознанно, в связи с каким-либо нестандартным событием или попаданием в новую среду;
* постпроизвольное – возникает автоматически в том случае, когда концентрация на каком-либо объекте происходит с регулярной периодичностью (работа, учеба и прочее).

Снижение концентрации внимания приводит к рассеянности, которая бывает нескольких видов:

* истинная – внимание постоянно переключается с одного объекта на другой, не задерживаясь при этом на долгое время (случай, когда это состояние вызвано нервным или физическим истощением, а также сильным стрессом, называют прострацией);
* мнимая – возникает в результате сосредоточения на каких-то личных мыслях, в результате чего концентрация на внешних объектах не представляется возможной;
* ученическая – быстрая переключаемость с одного процесса на другой (наиболее характерна для школьников и студентов, откуда и произошло ее название);
* старческая – медленная переключаемость (вызвана нарушениями умственных процессов, связанными с возрастом);
* мотивационно-обусловленная – речь идет о сознательном отключении внимания от того или иного объекта или процесса, который вызывает неприятные или нежелательные ассоциации;
* избирательная – со временем привычные вещи перестают обращать на себя внимание человека (речь может идти о процессах в организме или же бытовых явлениях).

1.2 Способы концентрации

Если человек хочет стать более внимательным и сконцентрированным в своих действиях как можно дольше, ему необходимо избегать отвлекающих факторов и четко выработать определенную стратегию, чтобы выполнить поставленную задачу. Для начала необходимо минимизировать, а лучше вообще избежать все возможные источники отвлечения внимания, в первую очередь – интернет, который перенасыщен интересной и ценной информацией, особенно когда приходит время поработать. Все социальные сети могут сильно отвлекать и в конечном итоге отложить ваше важное дело на второй план [4].

Избежать окружающих воздействий самостоятельно можно при помощи программы, которая позволяет ограничить доступ к социальным сетям на 2 часа.

# 2 Техническое задание

Наименование программы – «Bizy flower».

Краткая характеристика области применения: программа предназначена для личного применения, возможно применение в организациях с целью контроля трафика.

Основанием для проведения разработки является задание преподавателя. Задание утверждено самим заказчиком (преподавателем) и организацией – РГППУ. Дата утверждение документа – 01.11.2018г.

Основанием для проведения разработки являются документы, разработанные и утверждённые Российским государственным профессионально-педагогическим университетом. В список документов входит:

1. Учебный план;
2. Устав Университета;
3. Положение о курсовой работе.

Наименование темы разработки – «Программа для ограничения трафика социальных сетей на 2 часа».

Условное обозначение темы разработки (шифр темы) – «BF».

Представление разработанного приложения, отражено на рисунке 1.



Рисунок 1 – Окно приложения

Функциональное назначение программы – автоматическое ограничение социальных сетей (vk.com, instagram.com, facebook.com, youtube.com, ok.ru, twitter.com) на 2 часа, а также интерактивное взаимодействие с пользователем (досрочное снятие ограничений).

Приложение реализовано в виде простого, интуитивно-понятного интерфейса при помощи WPF (Windows Presentation Foundation).

Программа подходит, как для домашнего использования, так и для профильного в организациях.

Конечным пользователем программы может быть любое лицо, ознакомившееся с руководство пользователя и использующее операционную систему Windows, браузер Internet Explorer.

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

1. Блокировку и разблокировку сайтов, заданных разработчиком (редактирование списков сайтов не предоставляется возможном на этапе первой версии).
2. Очистка журнала браузеров (в частности IE), для корректной блокировки сайтов.
3. Интерактивное оформление приложение, где результатом взаимодействия ПО и пользователя является соответствующая форма. Если пользователь выполняет условие и не останавливает работу блокировщика в течении 2-х часов – программа хвалит клиента, иначе – ругает. Все манипуляции происходят двумя кнопками.
4. Возможность открыть руководство пользователя (в любом виде) и ознакомиться, до запуска программы.

Входные данные программы – клик мыши на соответствующую кнопку.

Выходные данные программы – это форма, как результат взаимодействия с конечным пользователем.

Требования к временным характеристикам программы не предъявляются.

Требования к надежности программы не предъявляются.

Требования к обеспечению надежного (устойчивого) функционирования программы не предъявляются на этапе внедрения.

Далее надежное (устойчивое) функционирование программы должно быть обеспечено выполнением заказчиком, конечным пользователем совокупности организационно-технических мероприятий, перечень которых приведен ниже:

* организацией бесперебойного питания технических средств;
* использованием лицензионного программного обеспечения.

Требований к времени для восстановления работы программы нет. Программу необходимо запускать вновь для восстановлений функций приложения. Данные об работе пользователя с приложением не хранятся в системе.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем операционной системы, не должно превышать времени повторного запуска ТС и ОС при условии соблюдения условий эксплуатации технических и программных средств. Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы возможны вследствие некорректных действий пользователя при взаимодействии с операционной системой и программы. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя лишь после изучения функционала.

Количество персонала, требуемого для работы программы, должно составлять в количестве одного человека (конечный пользователь программы – клиент).

Конечный пользователь программы должен обладать практическими навыками работы с экранными формами для корректного взаимодействия.

Требования к составу и параметрам технических средств

В состав технических средств должен входить IBM-совместимый персональный компьютер (ПЭВМ), включающий в себя:

1. Windows 7 (x86 и x64), Windows 8 (x86 и x64), Windows 10 (x86 и x64).
2. Требования к оборудованию: процессор мощностью 1,6 ГГц или выше.
3. 1 ГБ ОЗУ (1,5 ГБ для виртуальной машины).
4. 30 мб свободного дискового пространства.
5. Мышь.
6. Монитор.
7. Видеоадаптер, совместимый с DirectX 9 и поддерживающий разрешение экрана 1024 x 768 точек и выше.
8. Microsoft .NET Framework Версия 4.7.03056.
9. Бразуер Internet Explorer, версия любая удобная для конечного пользователя.

Исходные коды программы должны быть реализованы на языке С#. В качестве интегрированной среды разработки программы должна быть использована среда Microsoft Visual Studio 2017.

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы.

Бразуер Internet Explorer, версия любая удобная для конечного пользователя.

Программа должна обеспечивать взаимодействие с пользователем (оператором) посредством графического пользовательского интерфейса, разработанного согласно рекомендациям компании-производителя операционной системы.

Требования к программной документации

Предварительный состав программной документации

Состав программной документации должен включать в себя:

1. Техническое задание;
2. Спецификацию программы (указано в пояснительной записке);
3. Руководство системного программиста;
4. Руководство оператора;
5. Описание работы программы (руководство пользователя);
6. Курсовой проект.

Разработка должна быть проведена в несколько этапов:

1. Макет вида приложения;
2. Модель внутренней обработки информации;
3. Тестирование модели.
4. Совместить пользовательский интерфейс и построенную модель обработки данных;
5. Интерактивное тестирование;
6. Пуск и наладка;
7. Тестирование развертки приложения;
8. Внедрение.

Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. Разработка программы;
2. Разработка программной документации;
3. Испытания программы.

На стадии внедрения должен быть выполнен этап разработки - подготовка и передача программы.

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. Постановка задачи;
2. Определение и уточнение требований к техническим средствам;
3. Определение требований к программе;
4. Определение стадий, этапов и сроков разработки программы и документации на неё;
5. Выбор языков программирования;
6. Согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. Разработка, согласование и утверждение программы и методики испытаний.
2. Проведение приемо-сдаточных испытаний.
3. Корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика.

# 3 Теоретические основы

3.1 Основные понятия

Согласно техническому заданию, необходимо разработать программу для ограничения трафика социальных сетей на 2 часа.

Для разработки приложения использовались такие понятия как свойство, метод, класс и объект, файл, блокировка и разблокировка.

Свойство – это член класса, который предоставляет доступ к полям класса, чтобы узнать их значение или выполнить их установку. При использовании свойства, обращение происходит не к полю класса, а к вызову соответствующего неявного метода. Такой метод называется аксессор. Аксессор имеет два метода [8]:

* get – используется для получения данных;
* set –используется для записи.

В разработанном приложении данный метод осуществлен в качестве таймера обратного отсчёта.

Метод – это именованный блок кода, который выполняет какую – либо функцию. Методы вместе с полями являются основными членами класса.

Класс – это набор данных (полей, атрибутов, членов класса) и функций для работы с методами. Класс представляет собой шаблон, на основе которого будут создаваться его экземпляры – объекты.

Поле – это переменная, объявленная внутри класса.

Объект – это отдельный представитель класса, имеющий конкретное состояние и поведение.

Файл – это организованная структура данных, распознаваемая компьютером как единое целое.

Блокировка – это ограничение доступа к сетевому ресурсу клиента.

Разблокировка – это восстановление доступа к заблокированному ресурсу.

В рамках курсового проекта разработанная программа является приложением Windows и разработана в среде Visual Studio на языке C#. Данная программа реализована с использованием графического интерфейса.

Входными данными программы, является клик мыши на соответствующую кнопку. При нажатии на кнопку происходит событие, при котором операционная система посылает приложению сообщение, и запускается соответствующий обработчик именно для этого события. Для приложения можно выделить следующие функции:

* блокировку и разблокировку сайтов, заданных разработчиком (редактирование списков сайтов не предоставляется возможном на этапе первой версии);
* очистка журнала браузеров (в частности IE), для корректной блокировки сайтов;
* интерактивное оформление приложения, где результатом взаимодействия ПО и пользователя является соответствующая форма. Если пользователь выполняет условие и не останавливает работу блокировщика в течении 2-х часов – программа хвалит клиента, иначе – ругает. Все манипуляции происходят двумя кнопками;
* возможность открыть руководство пользователя (в любом виде) и ознакомиться, до запуска программы.

3.2 Методы и переменные для WPF

Для корректного отражения программы на разных расширениях экранов компьютеров созданы переменные:

* screeHeight - высота экрана, которая имеет свойство FullPrimaryScreenHeight. Данное свойство получает высоту в пикселях клиентской области для полноэкранного окна на мониторе основного дисплея 9[];
* screeWidth - ширина экрана, которая имеет свойство FullPrimaryScreenWidth. Данное свойство получает ширину в пикселях клиентской области для полноэкранного окна на мониторе основного дисплея [10].

Ключевым элементом при создании пользовательского элемента WPF используется DependencyProperty. Данный экземпляр используется для анимации, установки значений по умолчанию и другие возможности WPF. Одно из самых главных преимуществ WPF –создание насыщенного графического интерфейса, используя объявления интерфейса и код на управляемых языках C# [7].

Для управления действиями в приложении служит функция System.Diagnostics.Process.Start, она применяется для показа текстового файла в блокноте, запуска браузера по умолчанию, создания и отправки писем, используя технологии веб-страниц. В разработанном приложении функция управления действиями применяется для очистки журнала браузера Internet Explorer и работы с текстовым файлом [2].

Очистка истории браузера происходит при запуске приложения, после проведения данной процедуры произойдет изменение файла hosts и блокировка социальных сетей.

3.3 Методы для задания счетчика

Блокировка доступа к социальным сетям осуществляется через метод play\_Click\_1, который обрабатывает клик мышки о кнопке Play, по щелчку мыши осуществляется блокировка и задание счетчика.

Класс DispatcherTimer предоставляет таймер, интегрированный в очередь, которая предоставляет службы для управления очередью рабочих элементов для потока. Очередь обрабатывается в заданном интервале времени с заданным приоритетом. В разработанном приложении этот интервал равен двум часам [12].

TimeSpan.FromSeconds(1d)- возвращает указанный временной интервал, представляющий количество секунд

Для работы таймера созданы следующие методы:

* аксессор TickCounter для получения и записи данных о тике;
* метод Timer\_Tick, для осуществления тика и проверки окончания таймера. В данном методе осуществляется вызов разблокировки и закрытия открытой формы.

Для обработки событий используется EventHandler – делегат, не имеющий данных. Делегат представляет собой объект, который может ссылаться на метод. При создании делегата получается объект, содержащий ссылку на метод, делегат позволяет вызвать метод, на который он ссылается.

3.4 Блокировка доступа к социальным сетям

Hosts файл используется для блокировки доступа в социальные сети.

Файл hosts содержит в себе адреса тех сайтов, которые будут ограничены для доступа с компьютера с использование браузера Internet Explorer. Если для простого пользователя информация о нем почти неизвестна, то профессиональные системные администраторы постоянно используют его для настройки как внутренних, так и внешних сетей.

Полный путь к файлу hosts выглядит следующим образом C:\Windows\System32\drivers\etc. Именно в «etc» лежит файл hosts. Для блокировки сайта необходимо указать его адрес в файле hosts.

Файл hosts связывает IP-адреса с именами. Без уникального IP-адреса, которым в сети TCP/IP идентифицируются компьютеры, нужный компьютер найти невозможно. IP-адреса представлены в виде групп цифр – NNN.NNN.NNN.NNN, где NNN меняется от 0 до 255 и в таком виде трудны для запоминания и работы. Чтобы облегчить запоминание и обращение к IP-адресам компьютеров, каждому IP-адресу ставится в соответствие доменное имя. Система преобразует доменное имя, по которому пользователь обращается к определенному компьютеру, в соответствующий IP-адрес, и далее он используется для установления соединения с нужным компьютером [11].

Вначале ведение списка хост-имен с их IP-адресами было обязанностью всех компьютеров сети. Этот список до сих пор хранится в файле hosts. Получив от пользователя доменное имя, система ищет в файле hosts соответствующий адрес. За ведение этого списка отвечает системный администратор. Вследствие стремительного роста Internet и появления все новых и новых очень больших сетей функции преобразования доменных имен в IP-адреса были переданы серверам доменных имени DNS (Domain Name Server). Тем не менее, файл hosts продолжает использоваться для хранения доменных имен и IP-адрес хост-компьютеров, соединения с которыми устанавливаются наиболее часто. Этот файл обрабатывается раньше запроса к DNS и потому имеет приоритет, которым можно воспользоваться не только для того чтобы ускорить обращения к нужным сайтам, но и предотвратить посещение не нужных. Делается это просто добавлением в файл hosts (открываем блокнотом) строчки.

По умолчанию в чистой системе в файле hosts не указываются никакие значения, а присутствуют исключительно закомментированные строки (начинаются с символа «#») с описанием файла.

Синтаксис записи очень простой – сначала указывается IP адрес, а затем через любое количество пробелов (символов табуляции) прописывается название хоста (компьютера, узла или домена). Для каждой записи подобного рода используется отдельная строка [3].

Пример файла hosts:

102.54.94.97 – исходный сервер.

38.25.63.10 – узел клиента х.

127.0.0.1 localhost #адрес вашего компьютера .

127.0.0.1– это внутренний адрес вашего компьютера.

Интернет браузер на компьютере с модифицированным файлом Hosts может:

* не открывать сайты (адреса) – блокировка IP;
* открывать не те сайты (адреса) – подмена IP;
* открывать не все сайты (адреса) сети – фильтрация IP.

Некоторое программное обеспечение может вносить свои изменения в файл hosts, например, для ограничения доступа к сайту.

Для того чтобы заблокировать какой-либо сайт, достаточно добавить в данный файл простую строчку: «127.0.0.1 ссылка на сайт».

Файл hosts до внесения данных об ограничении доступа к сайтам отражен на рисунке 2.

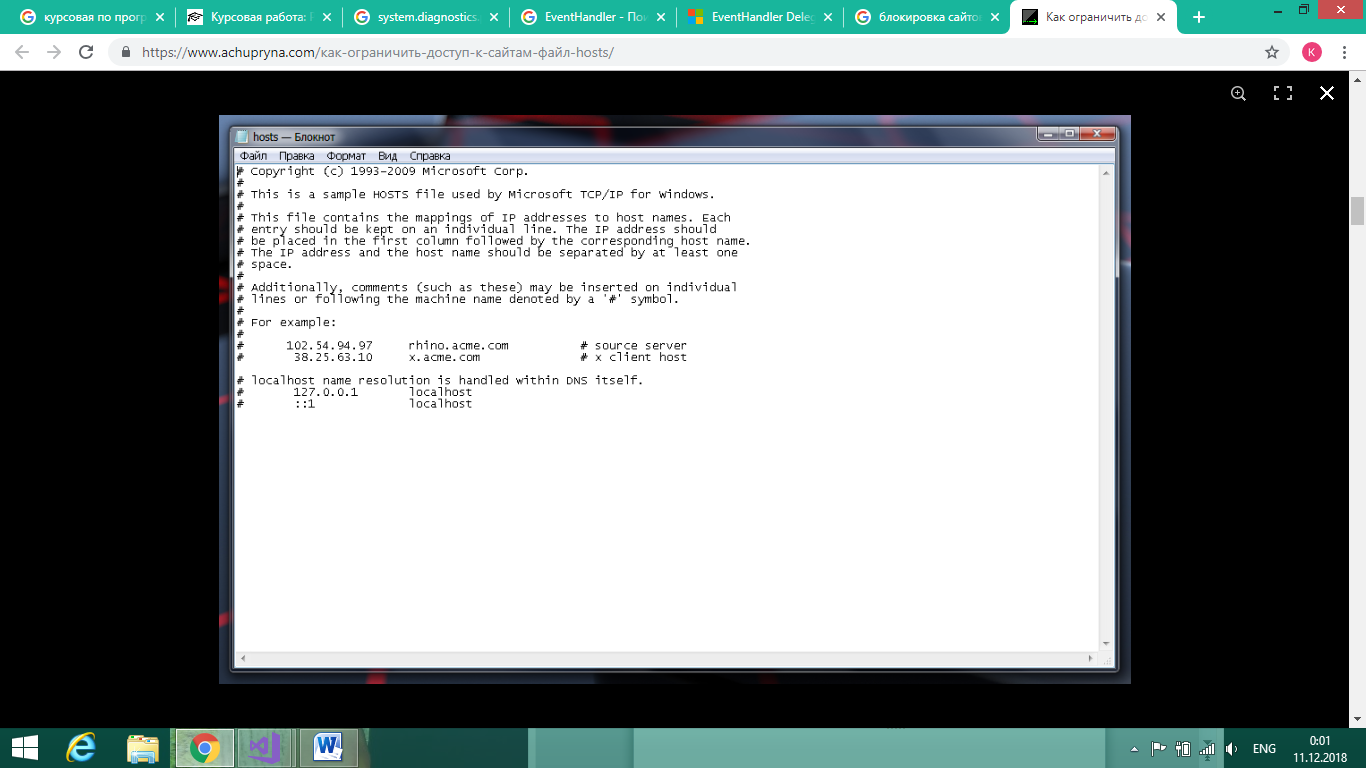


Рисунок 2 – Файл hosts до изменений

Файл hosts после изменений отражен на рисунке 3. В файл добавлена блокировка сайта vk.com.

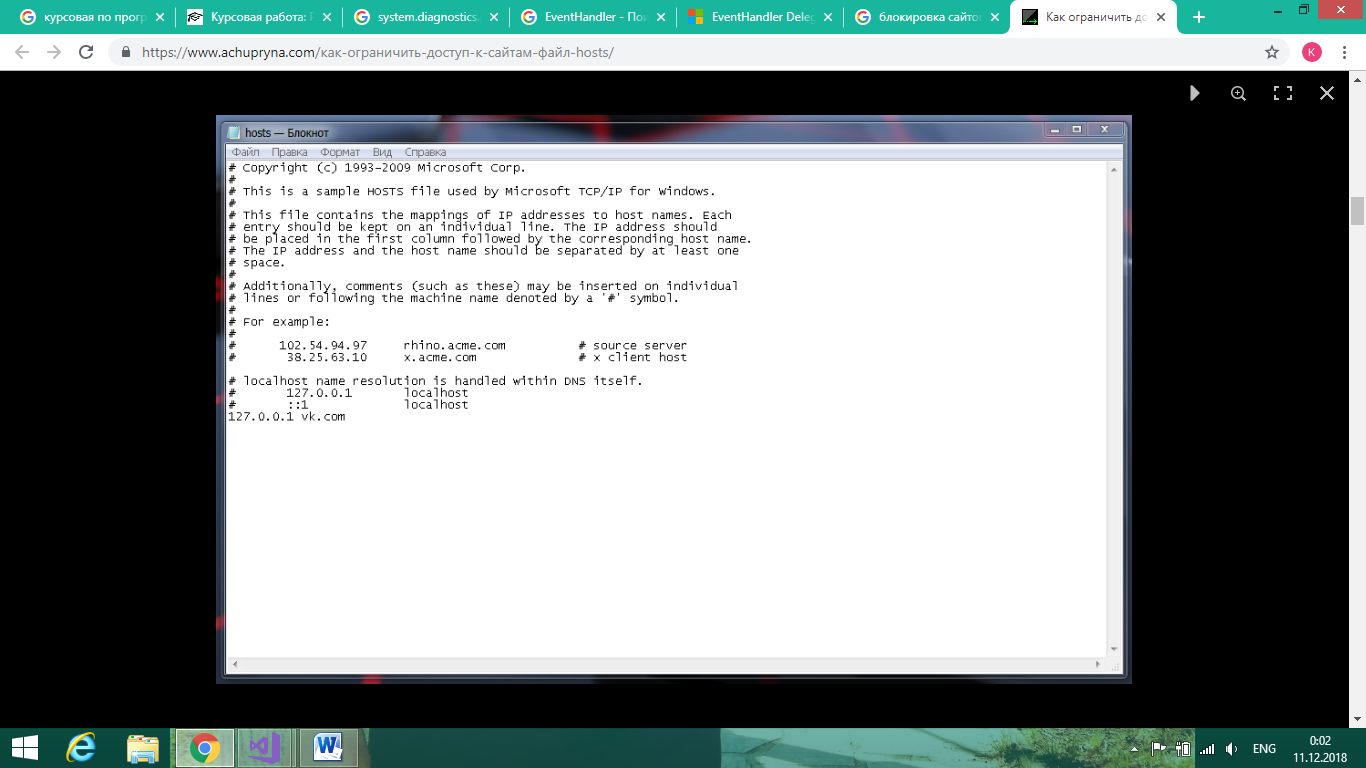


Рисунок 3 – Файл hosts после изменений

По окончанию работы метода Play\_Click\_1 вызывается метод Block\_Timer редактирующий данные в hosts файле. Файл Hosts редактируется до исходного состояния

Перемещение мыши отлавливаем метод Grid\_MouseMove. После того, как пользователь сделал малейшее перемещение, программное обеспечение отсчитывает 7 секунд от старта движения и завершает приложение.

# 4 Описание продукта и выполненные работы

4.1 Перечень выполненных работ

Этапы разработки проекта:

1. Составлено техническое задание.

Первым делом было составлено техническое задание для разработки программного обеспечения. В техническом задании было выбрано имя проекта, краткая характеристика области применения, основания для разработки, назначение разработки, были определенны требования к программе и к функциональным характеристикам, а так еж условия эксплуатации, стадии и этапы разработки, и порядок контроля и приемки программного обеспечения.

1. Разработан первичный функционал программы.

Открытие хоста в безопасном режиме

string path = @"c:\windows\system32\drivers\etc\hosts";

string block = @"block.txt";

string hosts\_true = "hoststrue.txt";

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

string block\_str = string.Empty;

using (System.IO.StreamReader reader = System.IO.File.OpenText(block))

{

block\_str = reader.ReadToEnd();

}

using (System.IO.StreamWriter file = new System.IO.StreamWriter(path, false))

{

file.WriteLine(block\_str);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message + Environment.NewLine + "Зайдите в безопасном режиме");

}

}

private void button2\_Click(object sender, EventArgs e)

{

string hosts\_true\_str = string.Empty;

using (System.IO.StreamReader reader = System.IO.File.OpenText(hosts\_true))

{

hosts\_true\_str = reader.ReadToEnd();

}

using (System.IO.StreamWriter file = new System.IO.StreamWriter(path, false))

{

file.WriteLine(hosts\_true\_str);

}

}

}

}

1. Создание красочной формы интерфейса, как на рисунке 4 и интерактивного взаимодействия с пользователем.

В этом этапе разработки мы описали интерфейс программы, в виде цветка, а также добавлены кнопки play и stop, для запуска и остановки программы.

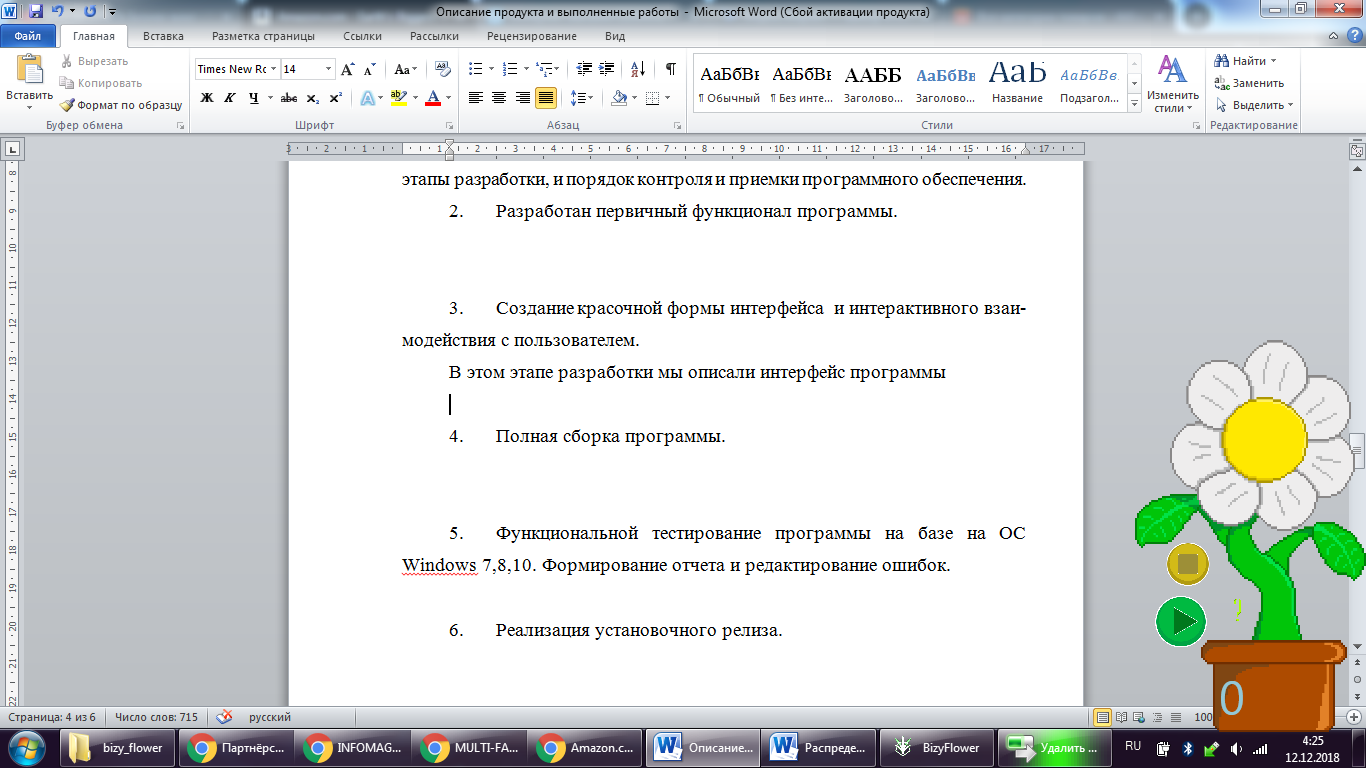


Рисунок 4 - Интерфейс программы

1. Полная сборка программы.
2. Функциональной тестирование программы на базе на ОС Windows 7,8,10. Формирование отчета и редактирование ошибок.

В ходе развертывания дефектов и зависаний не выявлено. Система работает корректно.

Тестирование установки производилось на Windows 8, 10 – без нареканий, приложение запустилось после установки и сформировало ярлык на рабочем столе.

Тестирование установки на Windows 7 – без нареканий, но при запуске приложения наблюдались рамки на окнах.

1. Реализация установочного релиза.
2. Функциональное тестирование и формирование отчет.

Выявленные ошибки при тестировании:

* в процессе тестирования была выявлена необходимость обладать правами администратора для редактирования файла hosts;
* при несоблюдении рекомендаций блокировка сайтов не происходит, а также наблюдаются визуальные проблемы с интерфейсом. После чего приложение аварийно завершается;
* при нажатии на play появляется ошибка. Необходимо править пути к текстовым файлам, на которые ссылается приложение;

1. Подготовка документации: руководство разработчика, руководство установки программы, руководство пользователя.

Руководство установки программы содержит необходимые указания для установки программы на компьютерах пользователей. В руководстве указано месторасположения файла установщика и как его скачать, возможные варианты установки программы, для пользователей операционной системы Windows 7 программу устанавливать не нужно, она запекается нажатием на файл запуска приложения. Для пользователей операционных систем Windows 8 и 10, программу нужно установить, нажав на файл установщика и выполнить наглядно показанные шаги установки программы.

Руководство разработчика содержит листинг кода с необходимыми комментариями для дальнейшей работы с приложением.

Руководство пользователя содержит информацию о пошаговых действиях при использовании продукта, о возможных ошибках и их устранениях.

Задание выполняли командой. Состав команды:

* Бельская Надежда Константиновна;
* Кильмяшкина Ирина Сергеевна;
* Пшеничников Константин Игоревич;
* Валеев Сергей Денисович;
* Неустроева Кристина Андреевна.

Распределение ролей в команде представлено в таблице

| Вид деятельности | Студент |
| --- | --- |
| Постановка задачи.  Выбор языков программирования. | Кильмяшкина Ирина Сергеевна |
| Определение и уточнение требований к техническим средствам.  Определение требований к программе.  Определение стадий, этапов и сроков разработки программы и документации на неё. | Бельская Надежда Константиновна  Пшеничников Константин Игоревич  Валеев Сергей Денисович  Неустроева Кристина Андреевна |
| Составление технического задания для ПО. | Неустроева Кристина Андреевна |
| Макет вида приложения.  Модель внутренней обработки информации. | Кильмяшкина Ирина Сергеевна |
| Функциональное тестирование модуля. | Пшеничников Константин Игоревич |
| Совмещение пользовательского интерфейса и построенной модели обработки данных.  Создания установочника. | Кильмяшкина Ирина Сергеевна |
| Функциональное тестирование приложения.  Формирование отчета по функциональному тестированию.  Отладка приложения исходя из отчета. | Неустроева Кристина Андреевна  Кильмяшкина Ирина Сергеевна |
| Функциональное тестирование установщика.  Формирование отчета о тестировании. | Бельская Надежда Константиновна |
| Создание документации разработчика. | Кильмяшкина Ирина Сергеевна |
| Создание руководства пользователя. | Пшеничников Константин Игоревич |
| Создание руководства по установке. | Бельская Надежда Константиновна |
| Формирование главы о введении, целях, задачах, актуальности работы. | Неустроева Кристина Андреевна |
| Формирование главы о техническом задании. | Пшеничников Константин Игоревич |
| Формирование главы о теоретических основах, которые были использованы в проекте. | Неустроева Кристина Андреевна |
| Формирование главы с информацией о продукте, перечне выполненных работ.  Создание сводной таблицы об этапах выполненных работ. | Валеев Сергей Денисович |
| Написание заключения к курсовому проекту. | Бельская Надежда Константиновна |
| Объедение глав по курсовому проекту и коррекция оформления. | Валеев Сергей Денисович |
| Создание презентации к защите проекта. | Валеев Сергей Денисович |

4.2 Описание продукта

Согласно целям и задачам нашего проекта мы определили, что нам необходимо разработать программу, предназначенную для ограничения трафика социальных сетей, таких как Instagram, Facebook, Twitter, Одноклассники, Вконтакте, для браузера Internet Explorer.

Согласно проведенному анализу предметной области, интернет источников, аналогичных проектов, а также исходя их наших возможностей, мы приняли реализовать проект в среде Microsoft Visual Studio 2017.

Данная программа реализована с использованием графического интерфейса.

Входными данными программы, является клик мыши на соответствующую кнопку. При нажатии на кнопку происходит событие, при котором операционная система посылает приложению сообщение, и запускается соответствующий обработчик именно для этого события. Для приложения можно выделить следующие функции:

* блокировку и разблокировку сайтов, заданных разработчиком (редактирование списков сайтов не предоставляется возможном на этапе первой версии);
* очистка журнала браузеров (в частности IE), для корректной блокировки сайтов;
* интерактивное оформление приложения, где результатом взаимодействия ПО и пользователя является соответствующая форма. Если пользователь выполняет условие и не останавливает работу блокировщика в течении 2-х часов – программа хвалит клиента, иначе – ругает. Все манипуляции происходят двумя кнопками;
* возможность открыть руководство пользователя и ознакомиться, до запуска программы.

В Microsoft Visual Studio мы реализовали следующий код [6]:

namespace WpfApp2

{

public partial class MainWindow : Window

{ /// <summary>

/// Замеры экрана, переменный вынесены, как глобальный, чтобы в дальнейшем было удобно работать

/// screeHeight - высота экрана

/// screeHeight - ширина экрана

/// </summary>

double screeHeight = SystemParameters.FullPrimaryScreenHeight;

double screeWidth = SystemParameters.FullPrimaryScreenWidth;

/// <summary>

/// Счетчик и взаимодействие get и set методов

/// MainWindow.TickCounterProperty реализация взаимодействия счетчика, формы и обработки значений

/// Значение задаются только в случае вызов readomly

/// </summary>

private DispatcherTimer \_timer;

private static readonly DependencyProperty tickCounterProperty = DependencyProperty.Register(

"TickCounter", typeof(int), typeof(MainWindow), new PropertyMetadata(default(int)));

/// <summary>

/// Обозначение ресурсов в качестве переменных для удобства использования, так же путь к hosts файлу

/// path - путь к хост файлу

/// block - файл формата txt, где хранятся листинг для блокировки сайтов

/// hosts\_true - файл того же формата, где хранится чистый файл хост, для восстановления

/// </summary>

string path = @"c:\windows\system32\drivers\etc\hosts";

string block = Properties.Resources.block;

string hosts\_true = Properties.Resources.hoststrue;

/// <summary>

/// MainWindow - где идет подгрузка всех объектов, замеры экрана пользователя, расположение WPF и очистка журнала бразуера IE

/// Left - ширина экрана

/// Top - длина экрана

/// </summary>

public MainWindow()

{

InitializeComponent();

Unlock();

this.Top = (screeHeight - 380.625);

this.Left = (screeWidth - 312.028);

System.Diagnostics.Process.Start("rundll32.exe", "InetCpl.cpl,ClearMyTracksByProcess 255");

}

/// <summary>

/// TickCounter - аксессор для получение и записи данных о тике

/// </summary>

public int TickCounter

{

get { return (int)GetValue(TickCounterProperty); }

set { SetValue(TickCounterProperty, value); }

}

/// <summary>

/// Связка аксессора, тика и формы в формате public

/// </summary>

public static DependencyProperty TickCounterProperty => tickCounterProperty;

/// <summary>

/// Timer\_Tick метод осуществляющий тик и проверку окончания тайминга

/// Так же идет вызов разблокировки, закрытие существующей формы и вызов формы windows1,

/// когда конечный пользователь не нарушил условия тайминга

/// </summary>

/// <param name="sender"></param> переменная для связки таймера

/// <param name="e"></param> параметр не содержит данных

private void Timer\_Tick(object sender, EventArgs e)

{

if (--TickCounter <= 0)

{

var timer = (DispatcherTimer)sender;

timer.Stop();

Unlock();

Window2 window2 = new Window2();

window2.Show();

this.Close();

}

}

/// <summary>

/// play\_Click\_1 метод обработывающий клик мыши по кнопки play

/// По щелчку запускает блокировщик и задается счетчик

/// </summary>

private void play\_Click\_1(object sender, RoutedEventArgs e)

{

//System.Diagnostics.Process.Start("rundll32.exe", "InetCpl.cpl,ClearMyTracksByProcess 1");

TickCounter = 60;

\_timer = new DispatcherTimer();

\_timer.Interval = TimeSpan.FromSeconds(1d);

\_timer.Tick += new EventHandler(Timer\_Tick);

\_timer.Start();

Block\_Timer();

}

/// <summary>

/// Block\_Timer метод редактирующий данные в hosts файле

/// </summary>

/// <exception>

/// Обработчик, на случай, если пользователь использует приложение без надлежащих прав (администратора)

/// </exception>

private void Block\_Timer()

{

try

{

using (System.IO.StreamWriter file = new System.IO.StreamWriter(path, false))

{

file.WriteLine(block);

file.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message + Environment.NewLine + "Зайдите в безопасном режиме");

}

}

/// <summary>

/// stop\_Click метод отвечающий за клик по внопке stop

/// Реализация разблокировки при помощи вызов метода Unlock, закрытия текущей формы и вызов формы windows1

/// </summary>

/// <param name="sender"></param> переменная для связки таймера

/// <param name="e"></param> параметр не содержит данных

private void stop\_Click(object sender, RoutedEventArgs e)

{

try

{

\_timer.Stop();

Unlock();

Window1 window1 = new Window1();

window1.Show();

this.Close();

}

catch (Exception)

{

Application.Current.Shutdown();

}

}

/// <summary>

/// Unlock метод возврата файла hosts к прежнему состоянию

/// </summary>

private void Unlock()

{

using (System.IO.StreamWriter file = new System.IO.StreamWriter(path, false))

{

file.WriteLine(hosts\_true);

file.Close();

}

}

}

}

Здесь представлено код главного окна, листинг других форм может увидеть в проекте.

Далее для того, чтобы программа работала корректно и нас при блокировке сайта, не перенаправило на сайт сохранившийся в памяти, необходимо очистить журнал браузера.

Очистка истории браузера происходит при запуске приложения, после проведения данной процедуры произойдет изменение файла hosts и блокировка социальных сетей.

Hosts файл используется для блокировки доступа в социальные сети.

Файл hosts содержит в себе адреса тех сайтов, которые будут ограничены для доступа с компьютера с использование браузера Internet Explorer.

Hosts изменяют для того, чтобы заблокировать доступ к определенному ресурсу в интернете, или для того, чтобы перенаправить пользователя на другой сайт.

Обычно, первоначально вредоносный код выполняется после запуска скачанной из интернета программы. В этот момент, автоматически вносятся изменения в свойства ярлыка браузера, и довольно часто добавляются дополнительные строки в файл hosts.

Для блокирования какого-либо сайта (например, сайт ВКонтакте) вносятся строки подобного вида:

127.0.0.1 vk.com

Если для простого пользователя информация о нем почти неизвестна, то профессиональные системные администраторы постоянно используют его для настройки как внутренних, так и внешних сетей.

Изменения файла hosts в нашей программе реализовано при помощи пула адресов, найденных в интернете, и при помощи стандартно маски, описанной выше.

Интерфейс программы реализован при помощи WPF (Windows Presentation Foundation) в Visual Studio.

Платформа Windows Presentation Foundation (WPF) позволяет создавать клиентские приложения для настольных систем Windows с привлекательным пользовательским интерфейсом.

В основе WPF лежит независимый от разрешения векторный модуль визуализации, использующий возможности современного графического оборудования. Возможности этого модуля расширяются с помощью комплексного набора функций разработки приложений, которые включают в себя язык XAML, элементы управления, привязку к данным, макет, двухмерную и трехмерную графику, анимацию, стили, шаблоны, документы, мультимедиа, текст и типографические функции. WPF входит в состав .NET Framework, поэтому есть возможность создавать приложения, включающие другие элементы библиотеки классов .NET Framework.

Одной из важных особенностей является использование языка декларативной разметки интерфейса XAML, основанного на XML: вы можете создавать насыщенный графический интерфейс, используя или декларативное объявление интерфейса, или код на управляемых языках C# и VB.NET, либо совмещать и то, и другое.

В Microsoft Visual Studio мы реализовали следующий код:

<Window

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WpfApp2"

x:Class="WpfApp2.MainWindow"

mc:Ignorable="d"

Title="BizyFlower"

WindowStartupLocation="Manual"

AllowsTransparency="True" WindowStyle="None" Background="Transparent" Height="437.625" Width="412.028" Topmost="True"

Left="1220" Top="420">

<Window.Foreground>

<ImageBrush/>

</Window.Foreground>

<Window.BorderBrush>

<ImageBrush/>

</Window.BorderBrush>

<Window.Template>

<ControlTemplate TargetType="{x:Type Window}">

<Border x:Name="newBorder" Cursor="Arrow" >

<Border.Background>

<ImageBrush ImageSource="Properties/цветок1.png"/>

</Border.Background>

<Grid>

<TextBlock x:Name="textBlock" Height="Auto" TextWrapping="Wrap" Text="{Binding TickCounter,

RelativeSource={RelativeSource FindAncestor,

AncestorType={x:Type local:MainWindow}}}" Width="Auto" Margin="165,347,138,36.6" Foreground="#FF93C8D7" FontFamily="a\_LCDNova" FontSize="48"/>

<Button x:Name="play" Margin="94,266,252,110.6" Click="play\_Click\_1" >

<Button.OpacityMask>

<ImageBrush ImageSource="Properties/плей.png"/>

</Button.OpacityMask>

<Button.BorderBrush>

<ImageBrush/>

</Button.BorderBrush>

<Button.Foreground>

<ImageBrush/>

</Button.Foreground>

<Button.Background>

<ImageBrush ImageSource="Properties/плей.png" Stretch="Uniform"/>

</Button.Background>

</Button>

<Button x:Name="stop" Margin="108,212,252,171.6" Click="stop\_Click">

<Button.OpacityMask>

<ImageBrush ImageSource="Properties/стоп.png"/>

</Button.OpacityMask>

<Button.BorderBrush>

<ImageBrush/>

</Button.BorderBrush>

<Button.Foreground>

<ImageBrush/>

</Button.Foreground>

<Button.Background>

<ImageBrush ImageSource="Properties/стоп.png" Stretch="Uniform"/>

</Button.Background>

</Button>

<Button x:Name="qa1" Margin="165,271,210,138.6" >

<Button.OpacityMask>

<ImageBrush ImageSource="Properties/вопрос.png"/>

</Button.OpacityMask>

<Button.BorderBrush>

<ImageBrush/>

</Button.BorderBrush>

<Button.Foreground>

<ImageBrush/>

</Button.Foreground>

<Button.Background>

<ImageBrush ImageSource="Properties/вопрос.png" Stretch="Uniform"/>

</Button.Background>

</Button>

<!--Элемент захвата и изменения размера - работает только для прямоугольных окон-->

</Grid>

</Border>

</ControlTemplate>

</Window.Template>

</Window>

Следующие формы внешне по заполнению xml-файла очень похожи, ознакомиться подробнее можно изучив наш проект.

Итак, у нас получилась действующая программа для ограничения трафика социальных сетей на 2 часа. Программа работает на ОС Windows 7,8,10 и предназначена для браузера Internet Explorer.

Реализованная программа подходит, как для домашнего использования, так и для профильного в организациях. Пользователем программы может быть любое лицо, ознакомившееся с руководство пользователя и использующее операционную систему Windows, браузер Internet Explorer.

# Заключение

Актуальность данной работы обусловлена тем, что возникла потребность в разработке программного обеспечения, которое позволило бы пользователям сфокусироваться на выполнении конкретной задачи, не отвлекаясь на сторонние онлайн-сервисы.

Целью данной работы была разработка программного продукта для концентрации внимания пользователей социальных сетей. В ходе создания курсового проекта было написано техническое задание [16] и согласовано с преподавателем, реализован продукт, который является приложением Windows и разработан в среде Visual Studio на языке C#, а также составлена вся необходимая документация для дальнейшего сопровождение ПО (документация разработчика [13], руководство по установке [14] и руководство пользователя [15]). Данная программа осуществлена с использованием графического интерфейса.

Для корректной работы приложения необходимо очищать журнал истории посещений в браузере. Очистка происходит автоматически при запуске продукта, после чего происходит изменение файла hosts и блокируются веб-сайты, выбранные нами заранее.

Было реализовано функциональное тестирование программного продукта и установочника, по результатам которого были внесены изменения в структуру кода и в документацию для дальнейшего сопровождения.

В итоге получилась действующая программа для ограничения трафика социальных сетей на 2 часа. Программа работает на ОС Windows 7,8,10 и предназначена для браузера Internet Explorer. Пользователем программы может быть любое лицо, ознакомившееся с руководство пользователя и использующее операционную систему Windows, браузер Internet Explorer.

Поставленные цели достигнуты в срок, нареканий по работе программе на этапе проектного внедрения нет.

# Список использованных источников

1. Добро пожаловать в интегрированную среду разработки Visual Studio[Электронный ресурс]. – Режим доступа:https://docs.microsoft.com/ru-ru/visualstudio/ide/visual-studio-ide?view=vs-2017 (дата обращения: 10.11.18).
2. Как использовать класс Process в .NET Framework [Электронный ресурс]. – Режим доступа: http://developer.alexanderklimov.ru/articles/process

.php (дата обращения: 01.12.18).

1. Как ограничить доступ к сайтам, файл hosts [Электронный ресурс]. – Режим доступа: https://www.achupryna.com/hosts/ (дата обращения: 22.11.18).
2. Как сконцентрироваться [Электронный ресурс]. – Режим доступа: https://ru.wikihow.com/сконцентрироваться (10.12.18).
3. Концентрация внимания. Тренировка и развитие концентрации внимания (упражнения) [Электронный ресурс]. – Режим доступа: https://businessman.ru/new-koncentraciya-vnimaniya-trenirovka-i-razvitie-koncentracii-vnimaniya-uprazhneniya.html (10.12.18).
4. Общие сведения о WPF [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/visualstudio/designers/introduction-to-wpf?view=vs-2017 (дата обращения: 10.12.18).
5. Определение и регистрация свойств зависимости [Электронный ресурс]. – Режим доступа: https://professorweb.ru/my/WPF/base\_WPF/level4

/4\_3.php (дата обращения: 10.12.18).

1. Свойства [Электронный ресурс]. – Режим доступа:https://professorweb.ru/my/csharp/charp\_theory/level5/5\_14.php (дата обращения: 10.12.18).
2. Системные параметры. Свойство полной высоты основного экрана [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/en-us/dotnet/api/system.windows.systemparameters.fullprimaryscreenheight?view=netframework-4.7.2 (дата обращения: 11.12.18).
3. Системные параметры. Свойство полной ширины основного экрана [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/en-us/dotnet/api/system.windows.systemparameters.fullprimaryscreenwidth?view=netframework-4.7.2 (дата обращения: 11.12.18).
4. Файл Hosts [Электронный ресурс]. – Режим доступа: http://wiki.compowiki.info/FajjlHosts (дата обращения: 25.12.18).
5. DispatcherTimer Class [Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru/library/windows/desktop/windows.ui.xaml.dispatchertime (дата обращения: 10.12.18).
6. [Документация разработчика.lnk](Документация%20разработчика.lnk)
7. [руководство по установке.docx](руководство%20по%20установке.docx)
8. [руководство пользователя.docx](руководство%20пользователя.docx)
9. [тех задание.docx](тех%20задание.docx)