**“SiteStall”  
Software Requirements Specification**

Claire Kolln, Jimmy Lam, Noah Tigner, Lucas Hyatt, Sean Wilson

Version 1

**Table of Contents**

[**1. SRS Revision History**](#_heading=h.tvy62pxlgwph) **1**

[**2. The Concept of Operations (ConOps)**](#_heading=h.bcwj2x8jvskl) **2**

[2.1. Current System or Situation](#_heading=h.3znysh7) 2

[2.2. Justification for a New System](#_heading=h.2et92p0) 2

[2.3. Operational Features of the Proposed System](#_heading=h.tyjcwt) 3

[2.4. User Classes](#_heading=h.3dy6vkm) 4

[2.5. Operational Scenarios](#_heading=h.4d34og8) 4

[**3. Specific Requirements**](#_heading=h.u2kh4puz76rn) **7**

[3.1. External Interfaces (Inputs and Outputs)](#_heading=h.hlam56fipw4u) 8

[3.2. Functions](#_heading=h.26in1rg) 8

[3.3. Usability Requirements](#_heading=h.lnxbz9) 8

[3.4. Performance Requirements](#_heading=h.35nkun2) 9

[3.5. Software System Attributes](#_heading=h.49hkenwt6hgl) 9

[**4. References**](#_heading=h.44sinio) **9**

[**5. Acknowledgements**](#_heading=h.2jxsxqh) **9**

# 

# 1. SRS Revision History

**Date Author Description**

**————— ——— ————————————————————————————**

2-13-2020 ck Created the initial document.

2-13-2020 ck Created first draft of Concepts of Operations

2-19-2020 JL Created Specific Requirements

2-20-2020 ck Revised ConOps sections 2.1-2.5

2-20-2020 ck Added reference to section 4 and cited SRS Template in section 5

2-20-2020 JL Revised Specific Requirements sections 3.1 - 3.5

# 

# 2. The Concept of Operations (ConOps)

## 2.1. Current System or Situation

This system is intended to assist a user with focusing on their digital tasks. The internet, while a useful tool in many respects, can also be a distraction for users hoping to accomplish a designated task. With social media websites, streaming services, and seemingly limitless browsing content just a click away, simple pieces of work can eat up more time and hinder absorption of materials.

Simply blocking the internet is not a solution to the problem of digital distractions because the internet is woven into how the modern student or professional accomplishes their tasks. Internet use is almost a requirement in most educational institutions and workplaces. Students use digital tools both in and outside of the classroom. Assignments, grades, and files are made available online via websites such as Canvas by Instructure or course webpages. Instant messaging services such as Slack, Skype, or Microsoft Teams are commonly used in workplaces to share information among teams.

Currently there are several systems that each handle a subset of SiteStall’s proposed functionality. SelfControl, a free open-source application for macOS, allows users to define a “blacklist” of sites to block for a defined period of time and will not allow access until the timer has run out. Forest, a mobile application, allows the user to start “focus time” for a predefined amount of time, and then simulates a tree growing during the time period. Leaving the Forest application and opening any other application on the user’s mobile device will cause the tree to start dying, motivating the user to not leave the application until the time is up.

## 2.2. Justification for a New System

Current systems are too narrow in scope, and do not provide data to inform the user about their time spent focusing. There are already sufficient systems in place to help users track how much time they spend not focusing, such as Screen Time on the iPhone and RescueTime for desktop. The new system intends to focus on helping the user regain control over their time by limiting their access to distracting websites and by letting them define what is most valuable to them to be blocked from.

SiteStall intends to expand upon the functionality already offered by SelfControl by tracking and displaying data pertaining to the user’s focused time and adding in a motivational component to offer rewards to the user for time spent focused. Forest, which is very effective in motivating users from using their smartphone during focus time, is limited to mobile. SiteStall will focus on the myriad of ways that a user can distract themselves using the internet on their laptop or desktop computer, where a majority of work is done for both school and in the workplace.

The scheduling aspect of the proposed system is in part motivated by the McGregor (2015) “The Secret to Fighting Digital Distractions?” article which appeared in the *Washington Post*. The article points to a study led by Richard Patterson from Cornell University, which sought to apply behavioral economics research to the context of students taking an online class. Patterson employed three customized software tools developed by the creators of RescueTime for the study. The most effective tool out of the three was the tool that allowed students to plan in advance when their distraction-blocking would occur, followed by the tool that allowed users to just block distracting websites for a period of time. He concluded that "Increasing the cost of procrastination now can help nudge you to getting back on track”.

The reward aspect of SiteStall is inspired by the Forest application. (<https://www.forestapp.cc>). As described above, it incentivizes users to stay focused by allowing the user to grow a tree while they stay focused on their task. The tree will die if the user gives in to distraction. It offers a motivational component in allowing the user to grow their own forest of trees, which can be customized as they earn new types of trees by focusing.

## 2.3. Operational Features of the Proposed System

At its core, SiteStall will block the user from accessing distracting websites, allow them to schedule focus time in advance, motivate them to keep focusing, and display reports of their time spent focusing. At a high level, system usage will follow the cycle illustrated below in Figure 1.

Figure 1: Use Cycle for Proposed SystemCycle of use for SiteStall


Users will be able to acknowledge distractions by customizing the sites that will be blocked when they start their focus time. This can be done by creating a user defined group of sites to be blocked as a unit, or by adding singular sites to the block list. They can also schedule future instances where certain sites will be blocked, and commit to distraction-free periods in the future.

While the sites are being blocked, they will not be able to access the websites that they have defined. As time passes, they will receive motivational messages at certain milestones in their focus time and be able to view their progress towards their reward. If the user decides to end focus time before time is up, they have to input a reason why they are ending early. This feature helps them to stay accountable. If they end on time, they are asked to input what they were able to accomplish during the period of time.

After the user is done focusing, they can view the results of their time spent using SiteStall. Information such as how long they spent focusing on certain subjects or tasks, or for which reasons they have most commonly quit the sessions early, will be displayed. Armed with this information and motivated by what they will be able to get done, they can modify their customizations for their next period of focus.

## 2.4. User Classes

The user classes for the proposed system are as follows:

1. Students - To define groups of distracting websites to block while they study at the secondary, undergraduate, or graduate school level. Would use reports to refine their study habits.
2. Professionals - To define groups of distracting websites to block while they work either in the workplace or from home.
3. Test Proctors - While administering exams which allow computers to block websites that students might attempt to access to cheat or communicate with one another or an outside source while taking the exam.
4. Parents - To monitor and control the sites that their children are allowed to access during “screen time”. Could be used to block inappropriate websites or to block distracting sites while their child is using the computer for school.

## 2.5. Operational Scenarios

1. **Use Case: Block websites before a period of focus**

***Preconditions:*** A student to focus more on their schoolwork. They know there are websites which often distract them from focusing fully on a task and desire to remove those distractions. They use a computer as a part of their study routine.

1. The student sits down to start focusing and pulls out their study materials, including their personal computer.
2. The student starts their computer and navigates to their course web page and look up which homework assignment is due the following day.
3. They find themselves thinking about the funny cat video their friend at school told them about and quickly go to YouTube to look up the video.
4. Four cat videos later, they realize how much study time they have lost and remember they have SiteStall installed on their computer. They open the application and quickly define a list of websites that they know they will be tempted to access and name it “Homework Distractions!”
5. They set the timer for one hour and start their focus time.
6. After thirty minutes, a notification pops up on the corner of their screen telling them they are halfway to their focus goal.
7. After another thirty minutes, they are notified that time is up. They enter the name of the homework assignment that they were working on.
8. Having completed their assignment, they look up more cat videos on YouTube.

***Postconditions:*** The user has focused for their desired amount of time and were able to complete their homework assignment. The system is ready to be used again for the next focus session.

1. **Use Case: Schedule SiteStall for a future period**

***Preconditions:*** A professional desires to spend less time at work getting distracted from their tasks. They use a computer with internet access for a majority of their day.

1. At the beginning of their work week, the professional goes into work and sits down in their cubicle.
2. They look at their calendar for the week and realize that they have a lot of work to get done before Friday, and are beginning to feel stressed.
3. They remember that they have SiteStall installed on their work computer and open the application.
4. They click on the scheduling feature of the main window and schedule focus time to be between their working hours. They select two lists of websites that they have previously defined from the dropdown list. The first is Streaming Services, and they schedule that list to be blocked from 9 until noon, and then from noon until 5 because they occasionally like to watch one episode of their favorite show on Netflix during their lunch break. The second category is Social media, and they set it up to be blocked from 9 to 5 every day. The scheduling process takes less than five minutes to set up for the whole week.
5. Having their websites blocked, they resume working.
6. At the end of the day, they get a notification to enter in what they were able to accomplish that day.
7. At the end of the week, they can view a display of everything they were able to get done during the week when they were in focus time.

***Postconditions:*** The user successfully blocked 2 groups of websites for the entire week and has

viewed their accomplishments.

1. **Use Case: At the end of the week, a student views their reports**

***Preconditions:*** A student has spent a lot of time focusing during the past week and wants to view a report of their time spent focusing over the course of the week.

1. They start their computer and open the SiteStall application by clicking on the icon.
2. They navigate to and press the reports button on the application’s main window
3. A new window opens and they are prompted to enter a period of time. The student selects “From the past week” from the dropdown menu. The other options are “from the past month”, “from the past year”, and Custom.
4. The window displays the student’s data from the past week.
5. They are quickly able to distinguish what they spent the most time working on while using SiteStall and how many rewards they have earned.
6. They also see that they have ended three sessions of focusing early because they needed to look at Twitter.
7. They see the amount of minutes wasted by ending the session early make a goal to not go on twitter next week.
8. They close SiteStall by pressing the exit button.

***Postconditions:*** The student has viewed their focusing data from the past week and have

renewed motivation for focusing in the weeks to come.

1. **Use Case: Block websites for a proctored exam**

***Preconditions:*** A computer science professor has decided to give an exam which requires students to write and compile code on their machines. As a prerequisite for the course, he had every student install SiteStall on their computers.

1. At the beginning of the testing period, the instructor writes down a list of websites on the whiteboard: Stack Overflow, Geeks for Geeks, and a several other websites are on the list.
2. They ask the students to start SiteStall on their computers and block the listed websites and then asks them to attempt to pull them up in their web browsers.
3. They deliver the instructions for the exam and then occasionally walk around the classroom to ensure that no students have turned SiteStall off.
4. After the exam, the students turn in their exam online and then turn SiteStall off.

***Postconditions:*** The professor has successfully proctored an exam during which all students

could use nothing but their IDE and their brains.

# 

# 3. Specific Requirements

The specific requirements for the software system are listed below.

Must have requirements:

1. The ability to block access to user specified websites.
2. The user created blocklist will be saved session to session.
3. The user can set a timer such that websites are blocked until the timer ends.
4. The user can schedule a time range where the websites specified in the blocklist will be inaccessible until the scheduled time expires.
5. A countdown timer will display the amount of time left until all the websites are unblocked.
6. The software must be run on macOS 10.14 (Mojave).
7. The software must be run using Python 3.7.
8. The software must be able to be run from the terminal.

Should have requirements:

1. The software should be able to be executed by clicking on an app icon.
2. The user should be able to group websites together into a category, such as social media, entertainment, etc.
3. A rewards system would encourage the user to remain distraction free when using the Site Stall software.
4. The user should be able to see the history of their performance from the past.
5. The system should be able to remind the user on how much time is left until the websites are unblocked.
6. There should be a notification whenever the time has expired.

Could have requirements:

1. If the user exits early, the system could prompt the user for a reason as to why they need to exit.
2. The application could be accessible from the Mac menu bar.
3. An email address could be linked with the software in order to send information such as user history or reminders.
4. The Site Stall software could have different modes of use. For example, parent/proctor mode would only allow the parent or proctor to enable or disable the blocking.

Won’t have requirements:

1. The Site Stall software will not measure the time spent on different websites.

## 

## 

## 3.1. External Interfaces (Inputs and Outputs)

*Input: select which websites to block.*

This allows the user to specify the websites that they wish to block. A graphical user interface (GUI) will assist the user in keeping track of what websites are on the list. In order to add a website to the block list, there will be an option on the GUI where the user can enter a new website by typing the URL, such as facebook.com or youtube.com. The block list will be saved whenever a new website is included, and the list will be saved between sessions. This allows the user to block the same websites whenever the Site Stall software is launched.

*Output: website blocker.*

This is the main functionality responsible for blocking websites. Once the list of websites to be blocked is created, the program will write to the hosts file located in the /private/etc/ folder. Each website will be mapped to the localhost IP address. Whenever a browser is opened and the user wishes to access a blocked website, the request will be rerouted to the computer’s localhost which effectively blocks access to the website.

## 3.2. Functions

Once the Site Stall software is started, the user will have the option of entering a new website into the block list.

1. The program will verify the entered website is valid by pinging the address.
2. Once the address has been pinged, the system will know whether or not the address was valid.
3. If the address was valid, it will be written to the output, as described below. Otherwise, a message will appear to notify the user the entered website is invalid, and the invalid website will not be written to the output file.
4. The output goes to the hosts file located on the user’s computer in the /private/etc folder. Each validated website will be mapped to the localhost of the computer.

## 3.3. Usability Requirements

1. The Site Stall software should be able to open and operate without any errors.
2. A GUI will allow the user to control the software:
   1. The user will be able to edit the website block list.
   2. Edit the scheduled time(s) at which the system should block.
   3. Start a timer such that websites are blocked when the timer has not ended.
3. All websites on the block list should not open if the user enters the website into the search bar of a browser.

## 3.4. Performance Requirements

1. The software should start up within 1 second.
2. Verifying a user-entered website should not take more than 1 second.

## 

## 3.5. Software System Attributes

*Usability*

The system should be straightforward to use. The user should not have to perform a lot of steps from launching the program to either add a website to the block list or to start the website blocking feature. To make it as straightforward as possible, a GUI will allow the user to visually see what websites are being or going to be blocked.

*Reliability*

The system should perform as expected without any errors. When the user is attempting to add a website to the block list, there is always a chance the user may unintentionally enter something wrong. Therefore, the system will first validate the website to make sure it is valid, by pinging the entered website, before adding it to the block list.

*Portability*

Since the code for the Site Stall software will be written in Python, it should be portable to any machine capable of running Python 3.7. In particular, running the software should be independent of the version of an operating system, such as macOS 10.14 (Mojave) or macOS 10.15 (Catalina).

# 4. References

McGregor, Jena. "The secret to fighting digital distractions may be all about commitments, not

willpower." *Washington Post*, 21 May 2015. *Gale Academic OneFile*,

<https://link.gale.com/apps/doc/A414540723/AONE?u=euge94201&sid=AONE&xid=b20e127c>. Accessed 17 Feb. 2020.

# 5. Acknowledgements

This SRS has been built from the SRS template provided by Anthony Hornof.