Lockdown

by

Andrew Mueller, Gunner Speros, and Jace Cheeseman

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

Andrew Mueller	4/26/2021
Andrew Mueller	Date
Gunner Speros	4/26/2021
Gunner Speros	Date
	4/26/2021
Jace Cheeseman	Date
	4/26/2021
Professor Ryan Moore	Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

Spring 2021

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ABSTRACT

According to research done by Statista out of the 2,000+ respondents Social Media, Binge-watching, and Gaming were three out of the top five leading distractions while on a computer. Lockdown is the ultimate tool to help keep users productive while on their computer whether at home or work. Using solely a C# application and PowerShell scripts we can help minimize the daily distractions of the internet. While there are many ways users can block websites, those are only a portion of the problem with few solutions for the rest. This paper discusses the work, validation, and development of the desktop application as well as our hopes for future features and results.

1. INTRODUCTION

1.1 Introduction

With technology being at the forefront of our everyday lives, we find ourselves becoming increasingly distracted by it when we need to be focused on something more important. Our simple and easy to use desktop application will allow you to block those distractions and get you back focused on what really matters. Lockdown application will allow you to customize when and what is being blocked, making your life easier by reducing distractions.

1.2 Problem

In the wake of today's current pandemic of COVID-19 millions of workers have been forced out of their workspace. Only to find themselves working these same jobs from their own homes. According to Kate Lester, President of Workspace Analytics, "Our best estimate is that we will see 25-30% of the workforce working at home on a multiple-days-a-week basis by the end of 2021." Without the threat of bosses and coworkers looking over your shoulder to keep you on track, it can be hard to stay productive constantly. Based on our research there are few solutions to keep workers productive while at home and most of these solutions are created for Macs (SelfControl) or web browsers (StayFocusd) with limitations.

1.3 Solution

Lockdown is a desktop application that allows users to block access to specific programs, websites, and files on their computer. Users will have access to creating timers and profiles to keep them personally productive as well as reminders that will ensure a healthy work-life balance with premade checks (posture check, water check, timed breaks, etc.) and custom-made checks.

1.4 Project Goals

The main goals for Lockdown are to allow any user to block distractions on their computer and keep them focused on what is more important. This will include building a desktop application for the Windows Operating Systems that has features such as blocking applications and websites, reminders, profiles and timers. We are creating this solution because we believe everyone should have access to reducing distractions and creating a healthier mindset.

1.5 Overview

This report will cover our group's progress this far in designing and building our C# desktop application. We will discuss our goals for this application and what action items need to be completed to accomplish them. A Project Timeline will also be provided to present a visual representation of our plan.

2. DISCUSSION

2.1 Project Concept

The purpose of this application is to reduce or even eliminate the most time-wasting distractions you encounter while using your computer. Lockdown strives to offer everyone the opportunity to become less distracted and build a healthy mindset. Lockdown will help you stay focused on what matters most by allowing any user the capabilities to block distracting applications and website at a touch of a button.

2.2 Design Objectives

Our team's design objectives stayed relatively consistent throughout development. Initially the plan was to use a C# library called CefSharp where we would build a webapp to integrate into our C# application for better UI design, but quickly ran into issues. We decided to scrap CefSharp and use the C# programing language because we wanted an all-in-one solution for the Windows Operating system unlike the other options that required multiple installs such as browser extensions and desktop applications. We wanted to use as many solutions as possible we could that are naturally built into windows already to accomplish our features.

2.3 Technical Elements

Methodology and Technical Approach

Throughout development we used an Agile development approach. Focusing on shorter one-week sprints to chip away at the larger scale project over time. With our short

time span for the project, we have used these shorter sprints to prioritize what needs to get done. We setup a static code analysis tool, Embold, in order to keep a constant analysis of our program during our development and keep our code clean and secure. We met twice a week for group work sessions and to confirm we are not running into any blockers.

Application

Our application is coded using C# and utilizes options that are built into Windows operating system. We take use of user permissions to allow users to block applications that are currently available on their PC. By making use of the host file that is built into Windows as well we can redirect any domain name to any I.P. address that we would like, this gives us room to redirecting the user to a simple website that allows us to inform them why they cannot connect.

2.4 User Profile

Lockdown

Team: Drew Mueller, Jace Cheeseman, Gunner Speros

There will be two types of users who will interact with Lockdown. The first user group described is the development team who will update and work on the program. The second user group consists of the work/learn from home users who want to stay productive.

Table 1: User Profile Form 1

User Profile Form 1
Application: GitHub, Visual Studio, C#, PowerShell
Potential Users: Developer
Software and Interface Experience: The user should be able to open the project solution and add features based on the next tasks.
Experience with Similar Applications: Should be familiar with GitHub
Task Experience: Using the applications to create, maintain, and expand on the application.
Frequency of Use: Once the program is created the user will use this to maintain and expand on what is currently implemented.
Key Interface Design Requirements that the Profile Suggests: This user will need to be able to work with the programming languages and interfaces to update and maintain the program.

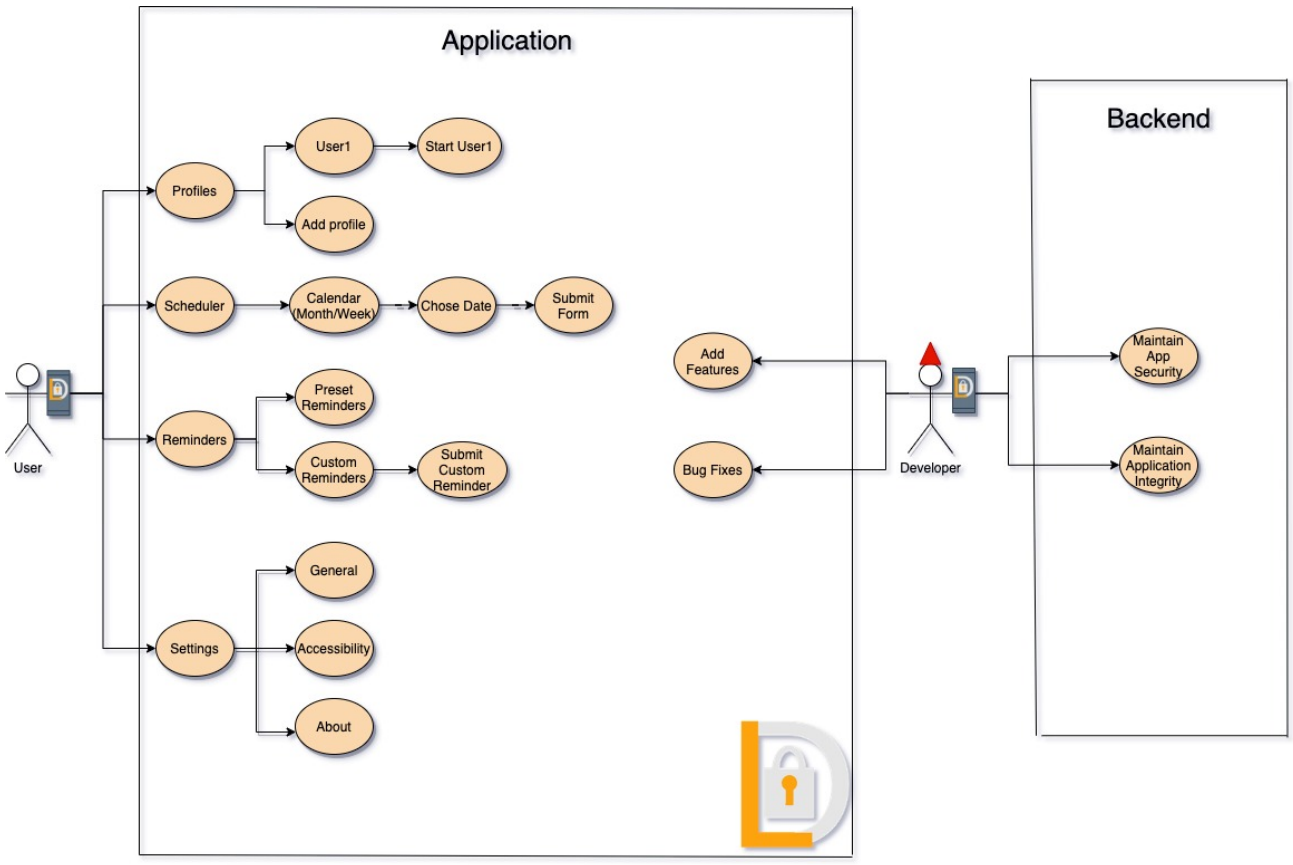
Table 2: User Profile Form 2

User Profile Form 2
Application: Lockdown Application
Potential Users: Work/Learn from home user
Software and Interface Experience: Users should have a basic understanding of using the internet and what desktop programs are.
Experience with Similar Applications: Calendar, Google Chrome, Microsoft Edge
Task Experience: Using the mouse to select through the options users will select and start any profile they have created.
Frequency of Use: When it is time for the user to go to class/work, usually weekdays for multiple hours.
Key Interface Design Requirements that the Profile Suggests: The program needs to be easily accessible and navigational by the user.

2.5 User Case Diagram

Our User Case Diagram lays out the interaction that users will have within the desktop application as well as how the Developer will use it.

Figure 1: Use Case Diagram



2.6 Testing

Testing Methodology

We wanted to get the product into the hands of actual users. Each of us chose individuals to share the program with and for them to test. We wanted to focus on getting a diverse group in order to see how different people from different walks of life use it. Our individuals consist of both home and work users.

Scope of Testing

We had the individuals test each part of the program that is currently functional and asked them to rate the program based on its benefits and ease of use. We had some users block a specific application or website and had other users block both or multiple at the same time.

Plan

Table 3: Testing Table

Record #	Test case #	Input	Expected output	Actual output	Pass/ Fail	Reason for failure/success	Date
1	1A	Ryan Kelly	Blocked Steam	Blocked the Application	P		2/5/21
2	2A	Ben Kollar	Blocked YouTube	Blocked the Website	P		2/7/21
3	3A	Blake Jones	Blocked Reddit	Blocked the Website	P		2/10/21
4	4A	Tony Mueller	Blocked facebook.com and League of Legends	Blocked application and website	P		2/15/21

2.7 Budget

Our budget for the project is based on primarily the work hours that was/will be put into the development and maintenance of the project. We used free tools at our disposal to keep the budget as cheap as possible.

Table 4: Project Budget

Project Asset Type						**Funding Source (if applicable)**	
Select One	Comments:					Select One	Comments:
Risk Identification (See Risk Types tab)						**Project Stakeholder(s)**	
	Risk Rating* 1-5 (5 is high)	***Comments***	***Weight***	***Score***		N/A	
Work Effort (days)	3		40%	1.20			
Complexity	2		60%	1.20			
Project Risk Score:				**2.40**			
Estimate of Benefits							
If project will generate revenue, estimate 1 year here:							
Select other benefits the project may bring a customer or user:							
Risk Avoidance	☐						
Improved customer satisfaction	☑						
Increased system availability	☐						
Productivity or process improvement	☑						
Reduced costs	☑						

Estimated Cost Rough Order of Magnitude:							**Comments:**
	Rate Per/Hr	**Work Effort (Hours)**	**1 X Costs**	**Ongoing Annual**			
				Rate Per/Hr	**Work Effort (Hours)**	**1 X Support Cost**	
Labor - IT	20	325	$ 6,500.00	20	20	$ 400.00	
Labor - External			$ -		0	$ -	
Software - External	[redacted]			[redacted]			
Hardware - External							
Misc.							
TOTAL			$ 6,500.00			$ 400.00	
5-Year ROI Analysis							
Description		**5- Year Expected**	**Conservative (1.5)**				
Total Costs		**$ 8,500.00**	**$ 12,750.00**				
Total Benefit		**$ -**	$0				
Total Costs/Benefit Differential		**$ (8,500.00)**					
Conservative Costs/Benefit Differential		**$ (12,750.00)**					

2.8 Project Timeline

The project timeline shows deadlines that we have set upon ourselves for Lockdown. Using a timeline between August of 2020 to April of 2021 we will complete these tasks according to the deadlines set.

Table 5: Project Timeline

Task #	Task Name	Duration	Start Date	End Date
1	Research and Analysis	21 days	August	September
1.1	Ideas and Brainstorming	7 days	August	August
1.2	Research Solutions	14 days	August	September
2	Deliverables	100 days	August	November
2.1	Team Contract	14 days	August	September
2.2	Project Abstract	9 days	September	October
2.3	Elevator Speech	7 days	October	October
2.4	User Scenarios	28 days	October	November
2.5	Draft Fall Report	28 days	October	November
2.6	Final Fall Report	14 days	November	November
3	Design	21 days	September	October
3.1	Design Database	7 days	September	September
3.2	Design Windows Software	14 days	September	October
4	Development	144 days	September	December
4.1	Setup Database	7 days	January	January
4.2	Develop Windows Application	144 days	September	December
5	Testing	90 days	December	February
5.1	Performance Testing	90 days	December	February
5.2	Feature Testing	28 days	February	March
5.3	Bug Fixes	28 days	February	March
6	Implementation	31 days	March	April
6.1	Create Models	7 days	March	April
6.2	Create Expo Presentation	7 days	March	April
7	Senior Design Expo	1 day	April	April 13th

2.9 Problems Encountered and Analysis of Problems Solved

One problem we ran into this semester was finding a Static Code Analysis Tool that was compatible with our code and user friendly. At first, we found a tool by Synopsys called Coverity. Once we tried implementing Coverity into our SDLC, we discovered that it was not as user friendly as we anticipated and there was very little documentation on how to use it. This is when we discovered a better SCAT by Embold. Embold's SCAT has great documentation on how to use it, making it more user friendly.

Another problem we have encounter was figuring out the best way to save the users' data. We are currently trying to save to text files, but do not think this is the best way to go about it.

We also tried to implement some sort of notification that indicates the application or website is blocked. Such as a pop-up window indicating that the application is blocked by Lockdown. We were never able to accomplish this but got close when trying to redirect a blocked website.

3. CONCLUSION

3.1 Lessons Learned

Throughout this year, we have learned a tremendous amount about what it takes to design and build a C# application as well as project management skills. We have encountered problems that have taken us back a few steps. However, we have also

learned from those mistakes and now have less of a chances of those problems happening in the real world. We learned a lot during our time in senior design and will take everything learn here with us for the rest of our careers.

3.2 Abilities and Skills Developed this Semester

Throughout this development process we have learned many new skills and best practices for our coding. We put a much larger emphasis on commenting our code as we progressed to make our code more readable. Also, as we implemented more features, we found ourselves needing to focus more on the security of our application than we previously thought.

3.3 Future Development Plans

None of us are currently planning on continuing the development of Lockdown after Senior Design. We have many features and ideas that would improve the program, but are not planning to pursue after graduation.

4. BACK MATTER

4.1 References

Department, P., & 31, J. (2020, July 31). Work from home: Leading distractions among workers U.S. 2020. Retrieved November 09, 2020, from https://www.statista.com/statistics/1139757/us-distractions-while-working-from-home-during-coronavirus/

Lister, K. (2020, April 12). Work-at-Home After Covid-19-Our Forecast. Retrieved November 09, 2020, from https://globalworkplaceanalytics.com/work-at-home-after-covid-19-our-forecast

Stanford marshmallow experiment. (2020, September 12). Retrieved November 09, 2020, from https://en.wikipedia.org/wiki/Stanford_marshmallow_experiment