**PERFORMANCE OPTIMIZER PROGRAM BASED ON BATCH SCRIPTING FOR OUTDATED COMPUTERS**

**WITH WINDOWS OS**

JUDE MENDOZA DUQUEZA

ISMAEL D. GARCIA

RONWELL CARL B. LANGIT

LUIS GABRIEL PALACIOS

MARK BENEDICT I. QUINTO

EDRICK VINCE C. SINSUAN

SECOND PHILIPPINE INTERNATIONAL SCHOOL

OLAYA STREET RIYADH,

KINGDOM OF SAUDI ARABIA

TITLE PAGE

INQUIRIES AND INVESTIGATION

NOVEMBER 7, 2019

APPROVAL SHEET

This investigatory project, **PERFORMANCE OPTIMIZER PROGRAM BASED ON BATCH SCRIPTING FOR OUTDATED COMPUTERS WITH WINDOWS OS**, in partial fulfillment of the requirements of the subject Inquiries and Investigation was examined and passed on.

**Charlotte de Ramos**

Adviser

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date Signed

ACKNOWLEDGEMENTS

First of all, we are grateful to our Almighty God for guiding us throughout this investigatory project. We thank Him for the knowledge He bestowed on us in order to complete this research.

We are thankful for our parents for their unwavering support and protection during the experimentation phase of the investigatory project.

To our research adviser, Mrs. Charlotte de Ramos, who greatly helped us with the formation of the research paper, as well as monitoring the progress of our project which helped us manage our time as they always remind us of the deadlines we need to comply to.

We also want to acknowledge the various authors of websites that voluntarily shared their knowledge to the public which greatly helped with the formation of the program as well as our research paper.

Lastly, we would like to thank our fellow researchers for their help regarding the research paper.

**- The Researchers**

ABSTRACT

In this research paper, the aim is to improve the performance of outdated models of computers through a program we created based on batch scripting called *performance optimizer.* In this program, it commands the computer through the command prompt.

This study conducted experiments to test if the various modifications that are applied by the program improves app launching times, boot time, and internet speed and responsiveness. There were three subjects that had gone three trials. The data gathered from these trials were used in the three-phased calculation that averaged the difference between before and after the computer was patched, as well as the difference if the computer was tested after a week it was patched.

The results of the experiments varied, and were ultimately dependent on each of the subject’s inherent performance. Subject 1 received the most negative results initially but improved after becoming more accustomed to the modifications for a week except its internet speed. Subject 2 had the most beneficial results, the optimizer drastically improved its overall performance. Subject 3 was the most functional out of the three before the optimizer was applied. As a result, it only received minor improvements, but we attribute this mostly to the already excellent performance of the laptop. Simply put, it cannot be further improved upon in a significant manner.

The study gathered information on the effects of an optimizer upon a varied selection of laptops, each possessing its own level of performance. The optimizer is most effective when used on laptops which have a low performance, improving them with different degrees of effectiveness. In terms of its capability of improving the laptops access to the internet, the optimizer was successful. Therefore, the recommendation drawn upon the results of the study is that the optimizer can be applied on outdated laptops to improve their functionality and avoid their obsoletion.

**Keywords:** *command prompt, app launching time, boot time, internet speed and responsiveness, modifications, laptops, program, optimize, batch script*

TABLE OF CONTENTS

[TITLE PAGE I](#_Toc23967852)

[APPROVAL SHEET II](#_Toc23967853)

[ACKNOWLEDGEMENTS III](#_Toc23967854)

[ABSTRACT IV](#_Toc23967855)

[CHAPTER I 1](#_Toc23967856)

[Introduction 1](#_Toc23967857)

[Background of the Study 1](#_Toc23967858)

[Statement of The Problem 2](#_Toc23967859)

[Hypothesis 3](#_Toc23967860)

[Significance of The Study 4](#_Toc23967861)

[Scope and Limitations 4](#_Toc23967862)

[Definition of Terms 5](#_Toc23967863)

[Conceptual Framework 7](#_Toc23967864)

[CHAPTER II 8](#_Toc23967865)

[Review of Related Literature and Studies Related Literature 8](#_Toc23967866)

[The rapid development and change of technology 8](#_Toc23967867)

[The waterfall software development methodology 10](#_Toc23967868)

[Improving PC performance 11](#_Toc23967869)

[DNS configuration 12](#_Toc23967870)

[CMD and batch file 13](#_Toc23967871)

[CHAPTER III 15](#_Toc23967872)

[Methodology 15](#_Toc23967873)

[Research Design 15](#_Toc23967874)

[Requirement Gathering & Analysis 15](#_Toc23967875)

[System Design 16](#_Toc23967876)

[Implementation and Internal Testing 17](#_Toc23967877)

[Deployment of System or Experimental Stage 23](#_Toc23967878)

[CHAPTER IV 36](#_Toc23967879)

[Data Results and Analysis 36](#_Toc23967880)

[After PO 36](#_Toc23967881)

[After a week with PO 48](#_Toc23967882)

[REFERENCES 58](#_Toc23967883)

[APPENDICES 60](#_Toc23967884)

[Tables of Data 60](#_Toc23967885)

[First Phase 60](#_Toc23967886)

[Second Phase 68](#_Toc23967887)

[Final Phase 76](#_Toc23967888)

[Short Experiment Gallery 80](#_Toc23967889)

[Subject 1 80](#_Toc23967890)

[Subject 2 80](#_Toc23967891)

[Subject 3 81](#_Toc23967892)

[Researchers 81](#_Toc23967893)

# CHAPTER I

## Introduction

Ever since technologies were invented, our progress towards improvement has been increasing exponentially as time passes wherein various technologies get deprecated or discarded faster than before, this is specially the case for computers. Due to this, the necessity of buying new computers is also ever increasing, but not all could satisfy such need because of the also increasing price of these new technologies. With the dilemma to choose between using old and cheap computer or new and expensive one, an in-between option must be created. The project entitled “Performance Optimizer Program Based on Batch Scripting for Outdated Computers with Windows OS” was made to provide another option for this problem. This project aims to provide a snappier and smoother experience for users who are using computers that are old and/or slow due to the setup of the computer being outdated or deprecated.

## Background of the Study

Outdated computers limit the productiveness of those who uses them because it means having a slow and unresponsive system, it requires a lot of hassle to maintain the performance of the computer, and it is a situation that is inevitable for computers as time passes. In addition to this, newer computers are as well getting more expensive as they get more advanced. Therefore, a solution must be created that is cost-free, hassle-less and effective. A solution that various researchers formulated is a program that uses the system’s tools that are not visible to a casual user of a computer. The project entitled “Performance Optimizer Program Based on Batch Scripting for Outdated Computers with Windows OS” is a program based on the language the command-line interface or CMD.EXE uses which is found only in Windows OS. Through research, a collection of scripts was compiled and made into a single program that is ran by a system application which ensures the functionality and security of the program. It has the following features: Files Cleaner, Registry Patcher, DNS Configurator, and more. The functionality of each feature is further discussed in the methodology of the research paper. All of which were tested for their effectiveness by various methods discussed in the experimental stage of the methodology.

## Statement of The Problem

The following questions are to be answered by the study:

1. How fast will the computer be after patching?
   1. App Launching
   2. Boot Time
2. How about its internet speed and responsiveness?
3. Will the performance be immediate or will it take time?

## Hypothesis

Based on the preliminary study conducted and visualization, the following are our hypotheses:

1. Based on manual time measurements and testing,
   1. App Launching
      1. Alternative Hypothesis: there should be a decrease of at least 1 second to the computer’s app launching time.
      2. Null Hypothesis: there is no significant decrease with the computer’s app launching time.
   2. Boot Time
      1. Alternative Hypothesis: there should be a decrease of at least 1 second to the computer’s boot time.
      2. Null Hypothesis: there is no significant decrease with the computer’s app launching boot time.
2. For the internet speed, speed test will be conducted through the website - <https://www.speedtest.net/> - by Ookla whereas an improvement in ping, download speed, and/or upload speed should be measured and ranked.
   1. Alternative Hypothesis: There is at least one server that has a higher rank than having the internet provider’s DNS server set.
   2. Null Hypothesis: The internet provider’s DNS server is ranked the highest or has tied at the top with a server and none of the servers are more responsive than the given.
3. Based on manual time measurements and testing,
   1. Alternative Hypothesis: The computer showed an increase in performance compared with its previous records after a week or more of normal usage.
   2. Null Hypothesis: The computer did not show an increase in performance compared with its previous records after a week or more of normal usage.

## Significance of The Study

Society: The improvement of outdated hardware will provide a cost-free solution with the same efficiency and effectiveness of its priced counterparts. Its open-source nature may help other software developers in creating projects related to the one conducted. It can prolong the utilization of computer laboratories as the once considered “obsolete” units may find new purpose through the upgrade they will receive will enhance them to a far more competent state. Finally, it will contribute greatly to the reduction of technological pollution due to the disposal of hardware described as “unusable”.

## Scope and Limitations

This project was made using Visual Studio and Github, in which the program we made takes the form of a batch file which can only be run on existing versions of windows as it uses the command prop of the operating system. The effectiveness of the software will only be tested on old or outdated computers and it is not ensured that the same effect will be experienced for new computers.

## Definition of Terms

* **Optimizer -** A person in a large business whose task is to maximize profits and make the business more efficient.
* **Outdated** **–** no longer current: OUTMODED
* **Operating System (OS) –** An operating system is a powerful, and usually large, program that controls and manages the hardware and other software on a computer.
* **Program** – is a specific of ordered operations for a computer to perform.
* **Ping** – is a signal sent to a host that requests a response.
* **GitHub** – a web-based version-control and collaboration platform for software developers.
* **Batch file** – is a text file that contains a sequence of commands for a computer operating system.
* **Microsoft Visual Studio -** is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.
* **Command line interface** (**CLI**) - is a text-based user **interface** (UI) used to view and manage computer files.
* **Modular** - employing or involving a module or modules as the basis of design or construction. Relating to an educational course designed as a series of independent units of study that can be combined in a number of ways.
* **Open Sourced -** denoting software for which the original source code is made freely available and may be redistributed and modified**.**
* **RAM (pronounced ramm) -** is an acronym for random access memory, a type of computer memory that can be accessed randomly; that is, any byte of memory can be accessed without touching the preceding bytes.
* **Registry -** The Windows Registry is a database of settings used by Microsoft Windows. It stores configurations for hardware devices, installed applications, and the Windows operating system.
* **Bloatware** - software whose usefulness is reduced because of the excessive disk space and memory it requires; unwanted software included on a new computer or mobile device by the manufacturer.
* **User experience (UX) -** design is the process design teams use to create products that provide meaningful and relevant experiences to users.
* .**vbs file** - contains lines of codes in the Visual Basic programming language and may be associated with other programs.
* **MB** - The **megabyte** is a multiple of the unit byte for digital information. Its recommended unit symbol is **MB**. A single **megabyte** is one million bytes of information. This definition has been incorporated into the International System of Quantities.
* **WMIC -** is the abbreviation of Windows Management Interface Command, is a simple command prompt tool that returns information about the system you are running it on. **WMIC** extends WMI for operation from several command-line interfaces and through batch scripts.
* **DNS resolver -** the client side of the **DNS** is called a **DNS resolver**. A **resolver** is responsible for initiating and sequencing the queries that ultimately lead to a full resolution (translation) of the resource sought, e.g., translation of a domain name into an IP address.
* **Boot Time-** The time it takes for a device to be ready to operate after the power has been turned on.

## Conceptual Framework

The following diagram below illustrates the conceptual model of the study.

# CHAPTER II

## Review of Related Literature and Studies Related Literature

This chapter includes the ideas, concepts, insights, and conclusions that will aid in the familiarization of information significant to the study.

### The rapid development and change of technology

In an article by the un.org, President of the 73rd Session of the UN General Assembly Garcés (2019) stated that, “Technological change is rapidly changing our present and our perspectives for the future.” In this session, she discussed about how this rapid change will affect the plans of achieving SDGs or Sustainable Development Goals for 2030. It is clear how our technologies are improving in a rate that we have not anticipated, that because of this, the UN is structuring plans in order for this technological to be monitored and be mitigated as there are risks involved with this rapid development.

To further prove that technological changes and development are happening rapidly, Gershenfeld (2003) director of MIT’s Center for Bits and Atoms, stated:

We've already had a digital revolution; we don't need to keep having it. The next big thing in computers will be literally outside the box, as we bring the programmability of the digital world to the rest of the world. With the benefit of hindsight, there's a tremendous historical parallel between the transition from mainframes to PCs and now from machine tools to personal fabrication. By personal fabrication I mean not just making mechanical structures, but fully functioning systems including sensing, logic, actuation, and displays.

Here, he said that in the future, in respect to the time he has stated this, that people would eventually be able to fabricate or create their own machines without needing industrial manufacturing. And just 6 years later, a company named Makerbot released Cupcake CNC that allowed normal consumers with limited technical knowledge to print their own designs and machines which caused a change to the 3D printer market (Gift of Gavin Smith, 2017).

In order to understand how rapid this growth is, an example of this growth is the exponential growth of transistors which is described by the Moore’s Law by Intel co-founder Gordon E. Moore. According to Roser and Ritchie (2019), “Moore’s Law is the observation that the number of transistors on integrated circuits doubles approximately every two years.” In this law, Moore predicted the continuing growth of the number of transistors that would be present until 1975 based on the data gathered 1959 to 1965. Moore (1965) stated “There is no reason to believe it will not remain nearly constant for at least 10 years.” This was then proved to be true not only until 1975 but also until this present day. In addition to this, this law also applies to other technological developments like computational power, pixels per dollar of digital cameras, storage capacity, computing efficiency, and microprocessors (Roser and Ritchie, 2019).

### The waterfall software development methodology

There are various methodologies for a software project to develop. According to Rachiele (2018), a full stack developer and a member in Medium.com, these methodologies are called Software development methodologies which corresponds to the overall process of a software project. He stated 13 examples: Agile Software Development, Crystal Methods, Dynamic Systems Development Model (DSDM), Extreme Programming (XP), Joint Application Development (JAD), Lean Development (LD), Rapid Application Development (RAD), Rational Unified Process (RUP), Scrum, Spiral, Systems Development Life Cycle (SDLC), and Waterfall (a.k.a. Traditional). Out of these examples, he said that the Agile Software Development and Scrum Methodology were mostly mentioned in his job interviews. However, these methodologies are time intensive and are not fitting to be used for projects that has a defined end. Thus, there was a need to explore other methodologies.

Jamsheer K (2018), the tech lead at Acodez.in, also discussed in a blog the software development methodologies wherein he gave the pros and cons of 12 methodologies (excluding SDLC). Most of the methodologies needed expertise, a large amount of time, and/or funding. However, there was one method which didn’t need any. The Waterfall model, “the traditional method of explaining the software development process in software engineering” (K, 2018). This statement by K proves that this method is one of the basic methods in this field. In addition to this, he also mentioned that it is linear, easy, simple, time-saving, and allows easy testing and analysis. Thus, it was clear that this methodology is fitting for beginners and/or small projects that focuses on a specific goal.

### Improving PC performance

When a person asks about how to improve performance, most people would answer to improve the hardware which is true. But, according to Phillips (2014), a Microsoft Certified IT Professional (MCITP), there is another way to speed up Windows by modifying the registry. In his blog he mentioned various tweaks that are said to make the desktop, startup, explorer (file manager of windows), and the visual effects faster, all of which could overall improve the user experience. However, all of these tweaks require great cautiousness as an example from bleepingcomputer.com shows a user that accidentally deleted a registry entry which turned his or her computer into an unbootable state.

Similarly, registryrecycler.com (2009), owned by Developer Tribe Private Ltd., a company that focuses on software development in Pakistan, also claims to increase the speed of a computer by memory modifications through the windows registry. In their blog post, they stated possible modifications that could be done that could improve a computers performance. Generally, their modifications are aimed at improving the memory management of a computer which they said is directly proportional to the speed of the computer.

Meskauskas (2017), an internet security expert, professional malware researcher and founder of RCS Software, also suggests another way to make a computer fast, by removing bloatware. He stated that these are software that are pre-installed that are not useful and are only a burden to a computer’s performance. Thus, with this said, he gave an instruction on how to uninstall or remove these applications through various ways; uninstalling manualy, using “Refresh Windows Tool,” and using Powershell. He also provided a list which he considers as bloats: 3D Builder, Alarms and Clock, Calculator, Calendar and Mail, Camera, Get Office, Get Skype, Groove Music, Get Started, Microsoft Solitaire Collection, Maps, Money, Movies & TV, OneNote, News, Phone Companion, People, Voice Recorder, Photos, Store, Sports, Weather, and Xbox.

These modifications could be combined as each does not conflict with each other. Thus, through a program that has access to the system, the patches could be done simultaneously to a computer.

### DNS configuration

Domain Name System (DNS) is “the digital equivalent of a phonebook, providing a number (IP address) for a given name (URL)” (Frew, 2016). Elaboratively, as the user enters a website in his or her browser, the computer asks a server through the internet called a recursive resolver which calls other servers in the internet to answer the inquiry. After reaching the recursive resolver, the resolver will then contact the first in a series of DNS servers or the root server which has DNS information on top level domains (.com, .fr, .co.uk). The root server will then give the recursive resolver to the TLD name server which contains second level domains (name of the website). Finally, the process ends as the TLD server gives the full IP address of the full domain to the recursive resolver that will then send the data of the website to the user’s browser for it to appear (McCallion and Hopping, 2019).

According to Rubenking (2019), there are various reasons that one should change his or her DNS server: protection, privacy, speed, and stability. ISPs or internet service providers are not known to provide secure, private, and/or fast DNS servers as they don’t give priority to such. Thus, he suggests switching to a service that gives priority to DNS security and privacy. He mentioned three well-known DNS servers: Google Public DNS, OpenDNS, and Cloudflare. Each of these DNS servers ensure privacy and protection, however, they differentiate with each other when it comes to offered special services and unique characteristics like; FamilyShield servers of OpenDNS which filters out inappropriate content, Google Public DNS’s memorable IP address, and the special security and protection against DDoS attacks or Distributed Denial of Service attacks.

### CMD and batch file

According to computerhope.com (2019), a website by Nathan Emberton, a former network support engineer and level IV senior technician, CMD or windows command line is “a user interface that is navigated by typing commands at prompts, instead of using the mouse.” It provides an alternative way of doing various tasks much faster than navigating the GUI or the graphic user interface. One way of doing so is through the use of scripts, mainly a batch file or a .bat file.

According to the same website, a batch file or a batch job is a list of commands stored as .bat in a windows PC that is executed sequentially by the computer which enables the user to accomplish multiple tasks mostly without the need of his or her input. Thus, with the use of commands available in the command prompt, it is possible to create a program that could modify the computer without interfering with the GUI nor the need of any technical input from the user as these can be automated.

# CHAPTER III

## Methodology

### Research Design

This project used quantitative experimental research design to test the effectivity of various scripts and programs used in the program. Wherein, through various researches, as a software-based research, the Waterfall Model had been used as a basis of methodology model for its direct approach being fitting to the project’s aim and purpose.

### Requirement Gathering & Analysis

With the assessment of experience, ability, and time, the following list shows the requirements that each of researchers did satisfy or comply to:

* Basic knowledge and logical understanding of how *cmd* codes works
* A laptop installed with Visual Studio Code and is connected to the Internet
* Patience and dedication
* Interest about the project

These were the following conditions of in which greatly affected the project’s success. There was only one material that was required as it did not make use of any physical materials other than what has been mentioned.

### System Design

The system of the program was designed based on the language provided by the command-line interface and was only available on windows which was a known limitation itself. The following list characterizes the program’s structure and design:

* Minimal file size
* Modular structure
* Supported by all currently existing windows version
* Open-sourced
* Easy-to-use with descriptions for each of its functions

As for its features, each of the following functions are tested and assessed through different laptops with different versions of windows in order to ensure proper execution to avoid unwanted effects by errors due to lack of tests.

* Files Cleaner
* Registry Patcher
* Disk Checker (and fixer)
* DNS Configurator
* Removing bloatware

These features were expected to affect the following aspects of the computer:

* App launching time
* RAM Management
* Internet Speed or Responsiveness (Ping)
* Overall UX Experience

### Implementation and Internal Testing

With the design given, the developers started the implementation of the system into a program. The development of the program split into various stages that reduced the chances of bugs occurring with the final revision of the program.

The following are the stages:

* Establishing the main template
* Implementation of individual features
* Polishing and final internal testing

#### Establishing the main templates

The main template refers to the structure of the code which is said to be modular and minimal. Thus, the code was separated into parts whereas each functionality in their own space with their own scripts do their own tasks. The following are the major parts of the code:

* BatchGotAdmin – A set of codes that prompts the user to grant administrative access for the program for it to do its functions properly (A set of codes provided by a superuser.com member)
* Start – Section of the code that set the variables that are assigned to a specific directory or file for the modules to use as the program initiates.
* Menu – The main interface of the program which provides the user the ability to navigate through the various features or modules that the program provides.
* Redirectors – These serve as the mediators of the main interface to the modules wherein each redirector of a module changes the directory of the command prompt to the module’s designated directory in the computer. This section also holds the exit prompt.
* Modules – Section of the code wherein the functionalities of the program are encoded. Each module in this section are isolated from other modules to ensure the proper execution of commands that will be done by the module. The following list shows the shortened or aliased names with their original name:
  + FCLEAN – File Cleaner
  + REGEDIT – Registry Editor
  + DSKCHK – Disk Checker and Fixer
  + DNS – DNS Configurator
  + REMBLT – Remove Bloatware

#### Implementation of individual features

The features of the program were implemented after the main templates have been established. With the features, specific codes were cherry-picked through various research in order to have the optimal results. The following provides a more comprehensive system design on how the features perform their functions:

* Files Cleaner
  + This script uses the commands “cleanmgr” and “schtasks” to instruct the computer to delete manually or automatically (though task scheduler) various directories of the computer that are safe to remove. With this command the following folders could be erased:
    - Temporary Setup Files
    - Old Chkdsk files
    - Setup Log Files
    - Windows Update Cleanup
    - Windows Defender Antivirus
    - Windows upgrade log files
    - Diagnostic data viewer database files
    - Downloaded Program Files
    - Temporary Internet Files
    - System error memory dump files
    - System error minidump files
    - Files discarded by Windows upgrade
    - System created Windows Error Reporting Files
    - Windows ESD installation files
    - DirectX Shader Cache
    - Mixed Reality
    - Previous Windows installation(s)
    - Recycle Bin
    - RetailDemo Offline Content
    - Update package Backup Files
    - Deletes all System Restore points
    - Temporary files
    - Temporary Windows installation files
    - Thumbnails
    - User file history
* Registry Patcher
  + This will do various modification to the system’s registry with the use of “regedit” command, whereas it is aimed to do the following:
    - Improve launching of applications by disabling explorer background functions
    - Decrease the time of system start delay by disabling start-up apps.
    - Increase the system’s responsiveness by shortening the timeout for programs to respond.
    - Improve RAM Management by enabling and disabling various behaviors of the RAM.
  + This may also bring some side-effects to the computer you are using depending on the age whereas newer computers may not benefit from this. This MUST be used with caution.
* Disk Checker and Fixer
  + This script makes use of the command “chkdsk” that will do what has been stated. The system will provide various options that are ‘normalized’ for the understanding of normal or casual users. The following options are:
    - Check system disk (C:)
      * At boot and/or Now
      * With or without fixing errors found by chkdsk
  + Checking with fix requires the files in the disk that will be scanned to be closed so it is recommended to have the ‘at boot’ enabled.
* DNS Configurator
  + This script will only modify IPv4 preferred and alternative DNS as other internet settings will not be necessary to change. It will make use of the commands “wmic” and “ipconfig” in order to perform its functions. The developers had research about the three-top performing DNS servers (including one that is new) that we used in the project and had found the following:
    - Cloudflare
    - Google DNS
    - Quad9
  + After applying the DNS, the script will then delete or remove DNS resolver cache for it to reconfigure.
* Remove bloatware
  + This script will make use of “wmic” command in order to uninstall programs that is considered useless or unnecessary. There is a list of programs that we had created based on research and our assessment through different websites:
    - Cyberlink Media Suite
    - PowerDVD
    - Nero
    - Candy Crush Saga
    - Ask Toolbar
    - WildTangent
    - 3D viewer
    - Messaging
    - Microsoft News
    - Mixed Reality Portal
    - Snip & Sketch
    - Cyberlink Power Director
    - Cyberlink Youcam
    - Mobile Plans
    - People
    - Web Media Extensions
    - Web Image Extensions
    - QuickTime
    - MyCleanPC
    - Shockwave Player
    - Microsoft Silverlight
    - Coupon Printer for Windows
  + There will be a list of removed bloatware created by the program at the desktop before the program proceeds to uninstall the applications.

#### Polishing and final internal testing

This stage is the final stage before proceeding to the experimental stage or the measurement of its effectiveness to the computer by various means. With this stage, the developers minimized the codes and fixed bugs or errors that occurred as the code was being refined. Repeated testing was done through different computers and laptops to assess its universality.

### Deployment of System or Experimental Stage

This last stage will assess the effectiveness of the program’s script with specific testing of the following aspects of the computer:

* App launching and initializing
* Boot time
* Internet responsiveness and speed

Whereas, the final testing was a test conducted a week after the computer was patched by the program in order to compare the results with the data collected directly after being patched. It followed the same procedure; however, the computer was used normally without PO for a week.

Each testing consists of 3 trials with 3 test subjects all of which are described below. The environment of each laptop was controlled minimally in order to preserve the real performance of the laptop.

Test subjects

The following information were gathered from the system information of the computers.

* #1: HP 15 Notebook PC
  + Name of the user: Ismael Garcia
  + Intel Pentium CPU N3530 @ 2.16Ghz, 4 Cores, 4 Logical Processors
  + 2 GB RAM
  + 500 GB of HDD & 16 GB of External Drive
  + 4 Years (2015)
* #2: Samsung 270E5J
  + Name of the user: Jude Duqueza
  + Intel Core i5-4200U CPU @ 1.60Ghz, 2 Cores, 4 Logical Processors
  + 4 GB RAM
  + 500 GB of HDD
  + 6 Years (2013)
* #3: Toshiba Satellite L850-B757
  + Name of the user: Luis Gabriel Palacios
  + Intel Core i5-3230M CPU @ 2.60Ghz, 2 Cores, 4 Logical Processors
  + 6 GB RAM
  + 100 GB of HDD & 400GB of External Drive
  + 6 Years (2013)

#### App launching and initializing

This aspect will be tested through repeated launching apps whereas we measure the time as we are performing it. The following scripts were initiated or used for this testing:

* Files Cleaner
* Registry Patcher
* Remove bloatware
* DNS Configurator

The following were the conditions for the testing:

* (WITH PO) All patches should be executed before proceeding with the tests.
  + The file cleaner had the following options enabled:
    - Temporary Setup Files
    - Old Chkdsk Files
    - Setup Log Files
    - Temporary Internet Files
    - System Error Memory Dump Files
    - System Error Minidump Files
    - System Created Windows Error Reporting Files
    - Temporary Files
    - Temporary Windows Installation Files
  + All Registry Patches were enabled.
  + All bloatware enlisted were uninstalled.
  + DNS Configurator was enabled with the same internet connection among the test subjects.
* The following apps were tested with the following conditions:
  + Word
    - A document containing a 16MP picture.
  + Chrome
    - The website facebook will be used as the starting website of the application.
    - Each laptop will be connected to a single common source of internet connection.
  + Explorer
    - The windows system directory will be opened.
* There was at least 5 minutes of idle time after the computer boots (timer starts when the windows lock screen gets shown).
* Shortcut of the apps to be tested were shown at desktop.

The procedures of the test were as it follows:

* Start the computer
* After 5 minutes of idle time,
  + Apply the DNS patch on the internet connection.
  + Launch the app to be tested (only one app per boot).
    - The timing starts upon clicking the application 2 times and ends when the application fully loads the content.
    - Input the recorded times in a database.
* Shut down the computer for at least a minute
* Repeat the preceding procedures 3 times with and without applying the PO patches through 3 laptops.

FIRST PHASE

Whereas,

t = number of trials

T = total time (in seconds) of 3 trials

Then,

CA = Chrome average time

WA = Word average time

EA = Explorer average time

or Mi or Mf

SECOND PHASE

After getting both with and without PO performance mean,

Mi = Performance Mean without PO

Mf = Performance Mean with PO

After getting the Performance Mean with and without PO, we subtracted the PM of without PB to the PM with PB, in order to get the PM difference.

FINAL PHASE

Then,

P1 = Performance Difference of Subject 1

P2 = Performance Difference of Subject 2

P3 = Performance Difference of Subject 3

Each PM difference of each test subject were then averaged in order to get the average PM difference in which is denoted by the variable PD.

If PD ≥ -1 then it is generally effective. However, if PD < -1 then it not generally effective.

#### Boot time

This aspect was measured by measuring the time of how long it takes for the computer to boot. The following scripts are initiated or used for this testing:

* Disk Checker and Fixer
* Files Cleaner
* Registry Patcher
* Remove bloatware

The following were the conditions for the testing:

* (WITH PO) All patches should be executed before proceeding with the tests.
  + The file cleaner had the following options enabled:
    - Temporary Setup Files
    - Old Chkdsk Files
    - Setup Log Files
    - Temporary Internet Files
    - System Error Memory Dump Files
    - System Error Minidump Files
    - System Created Windows Error Reporting Files
    - Temporary Files
    - Temporary Windows Installation Files
* Disk Checker ran for checking and fixing errors in drive
* All Registry Patches were enabled.
* All bloatware enlisted was uninstalled.

The procedure of the test is as it follows:

* Start the computer.
* Measure the time it takes for the computer to boot and show desktop.
  + Timing starts when the light indicator switches on and ends when the computer shows the lock screen.
  + Input the recorded times in a database.
* Shut down the computer for at least a minute.
* Repeat the procedure for 3 times with and without PO patches through 3 computers.

FIRST PHASE

Whereas,

T = total time of 3 trials

t = number of trials

SECOND PHASE

After getting both with and without PO boot time,

Ti = Average boot time without PO

Tf = Average boot time with PO

With each average boot time of with and without PB calculated, the average boot time without PB was subtracted to the average boot time with PB in order to know the difference in boot time of the subject.

FINAL PHASE

Then,

B1 = Average boot time difference of the subject 1

B2 = Average boot time difference of the subject 2

B3 = Average boot time difference of the subject 3

The general average boot time difference of all the subjects was calculated after getting the average boot time difference of each subjects. This data was then used for concluding the results.

If BD ≥ -1 then it is effective. However, if BD < -1 then it is not very effective.

#### Internet responsiveness and speed

Through the use of speedtest.net provided by Ookla we can test the internet speed and its responsiveness. The following scripts are initiated or used for this testing:

* DNS Configurator

The following were the conditions for the testing:

* (WITH PO) DNS was configurated with the current internet connection.
* The same internet connection was used for all of the trials and laptops.
* Disabled any background internet usage that may occur during the testing.

The procedure of the test is as it follows:

* Start the computer.
* Wait for at least 5 minutes for the computer to initialize.
* Connect the computer to an internet connection.
* Apply the patch (None, Cloudflare DNS, Google DNS, Open DNS).
* Go to speedtest.net and click go.
* Input the results in the tables below.
* Repeat the test 3 times.
* Shut down the laptop for at least a minute if a new patch will be tested.

FIRST PHASE

Whereas a table was used to record each trial and subject with ping,

|  |  |  |
| --- | --- | --- |
| Subject 1 / Subject 2 / Subject 3 | | |
| Without DNS patch / Server 1 (Cloudflare DNS) / Server 2 (Google DNS) / Server 3 (Quad9 DNS) | | |
| Ping | Download | Upload |
|  |  |  |
|  |  |  |
|  |  |  |
| Average: | Average: | Average: |

SECOND PHASE

Each average was compared and ranked to know if servers had an effect with the three properties being tested:

|  |  |  |  |
| --- | --- | --- | --- |
| Servers | Ping | Download | Upload |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 |
| None | 4 | 4 | 4 |

In order to generalize the performance of each server on a subject, the process of getting the sum of each server’s ranking was repeated. They were ranked again with the lowest sum being ranked the highest:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Servers | Ping | Download | Upload | Total | Ranking |
| 1 | 1 | 1 | 1 | 3 | 1 |
| 2 | 2 | 2 | 2 | 6 | 2 |
| 3 | 3 | 3 | 3 | 9 | 3 |
| None | 4 | 4 | 4 | 12 | 4 |

FINAL PHASE

To get the server’s general performance throughout all test subjects, the said process of totaling and ranking each was repeated to get the final ranking that was used to determine if a several, all, or none of the DNS servers provided by the app was more effective:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Servers | Subject #1 | Subject #2 | Subject #3 | Total | Final Ranking |
| 1 | 1 | 1 | 1 | 3 | 1 |
| 2 | 2 | 2 | 2 | 6 | 2 |
| 3 | 3 | 3 | 3 | 9 | 3 |
| None | 4 | 4 | 4 | 12 | 4 |

With the final ranking determined, it was then used as a basis for conclusion for answering the hypothesis.

If at least one or more DNS server was ranked higher than the internet provider’s DNS server, then that server is or those servers are effective. Otherwise, if the internet provider’s or stock DNS server was ranked as the highest or is tied up with a server at the top, then none of the servers are effective.

#### Immediateness

This tested if the effects of ‘performance optimizer patches’ are immediate or not. Thus, the patches were applied once and the subjects were used normally for at least a week.

The following conditions were applied:

* Apply the same conditions formerly done with the testing of Performance Optimizer enabled.
* Do not reenable / disable any options that were enabled.

After that period of time, the researchers proceeded to do all the tests again in order to see if there was no change or a major change after a week of not using the performance optimizer.

# CHAPTER IV

## Data Results and Analysis

### After PO

#### First Phase

On the first phase of calculation, it showed the varying performance of each laptop. Through 3 trials for each testing, the researchers were able to come up with averages that they could reliably use with comparing results between before and after the application of the patches.

The app launching and boot time results showed that subject 3 was the fastest, followed by subject 1, and the slowest being subject 2. This could be contributed to the specifications and the manufacturing of the laptops as they also vary among the subjects as shown at page 23. However, it seemed unnatural that the newer laptop, subject 1, was slower than subject 3 which was as old as subject 2. It may have been caused by the owner’s usage of the laptop, whereas, the owner of subject 1 may have used it more carelessly than the owner of subject 3. Nevertheless, the speed of subject 3 was clearly far ahead of subject 1 which was unexpected.

With internet responsiveness and speed, each subject showed varying results. Subject 1 have shown that the custom DNS configurations had a substantial negative effect to the download speed. However, the upload speed fairly increased and the ping did not increase significantly. On the other hand, subject 2 had a contrasting result to subject 1 as the download speed improved after applying the patch. However, it also decreased the upload speed and the ping did not significantly change. Subject 3 only experienced minor effects in all aspects compared to the changes with subject 2 and 1. This could be attributed to the capacity of the laptop to process the amount of information it receives wherein it may not support higher bitrates because of the modem installed in the motherboard of the laptop. We have noticed as well that all of the laptops do not support 5ghz band connection which may had an effect with the testing of the laptops since they were all connected while testing the internet speed and it has been said that 2.4ghz band are mostly congested due to a lot of appliances using it thus resulting to slower-than-expected speeds (Klein and Glenn, 2018).

#### Second Phase

With the second phase of calculation, the researchers subtracted the generalized data of app launching and boot time of each subjects’ “with PO” or patched laptop data and “with PO (after a week)” or patched laptop data after a week to “without PO” or unpatched laptop data separately. The internet responsiveness and speed results, however, were calculated furtherly using the rank-point system (please research). Through these calculations, the difference between having the patches and without for each subject was shown.

First subject’s average app launching became worse after patch, it gained 6 seconds which is a very significant difference in performance. The average boot time of the laptop, however, did improve but insignificantly. It only decreased for just a millisecond which is very negligible and might be even contributed to the timer’s response time. The trend also continues when it comes to the internet responsiveness and speed ranking wherein the stock or unpatched data was the on the first ranking. Furthermore, the data from the first phase shows a very large gap between the download speeds before and after patching. Overall, the results were negative for subject 1.

On the other hand, the second subject got an opposite result. Both average app launching and boot time improved dramatically with the average app launching decreased by four and a half, and the average boot time by five and a half seconds. Same goes with internet speed, the Cloudflare data led the ranking followed by a tie between stock and Google. This data was also strongly supported by first phase data wherein there’s a big gap when it comes to download speed which is very noticeable when it comes to browsing the internet. In conclusion, subject 2 results were positive after patching.

With the third subject, the changes were not noticeable or were very minimal. App launching and boot time only increased for a second which is not that big of a change for a single subject average. The internet speed, however, did have a positive effect with Quad9 patch beating stock data. Although, going back to first phase data, the gap between both are close which may mean a negligible effect when it comes to casual browsing and video streaming. Therefore, subject 3’s results were neutral.

#### Third Phase

In this phase, the data from each subject from the second phase were then used in order to come up with a general difference that will summarize the results of all the data gathered from the testing conducted.

With app launching, the general average calculated was +0.98 seconds. Subject 2 may have saved a significant amount of time when it comes to app launching, still, the decrease in performance of subject 1 cannot be neglected which greatly affected the average. This given, the result did not satisfy the alternative hypothesis’ condition, thus, the null hypothesis was accepted.

As for boot time, the calculated general average data was -1.72 seconds. This result was mainly due to the large difference subject 2 with a difference of -5.63 seconds, whereas others had insignificant changes. Therefore, with the general average data, the alternative hypothesis’ condition was satisfied.

Using the second phase’s ranking data, rankings of each DNS patch from each subject were used in order to come up with a general ranking. The calculations resulted to the stock or unpatched having the lowest rank sum, thus, it was ranked first. It was followed by Quad9, then Google, and lastly Cloudflare. Stock may have been the lead with only one subject, it maintained its high ranking with only 2.5 as its lowest ranking compared to Quad9 which did not perform well with subject #2. In conclusion, the alternative hypothesis was not satisfied, thus, the null hypothesis was accepted.

### After a week with PO

#### First Phase

The same testing procedure was done with this part, only all of these testing was done after a week of PO being applied to the subjects wherein throughout the wait the laptops were normally used without installing any apps nor making significant changes to the laptop.

In comparison to the app launching and boot time data gathered after PO, the data gathered after a week followed the same trend wherein the subject 3 is the top performer followed by the subject 1, and lastly, subject 2. However, subject 1 significantly improved along with subject 2. Subject 3 did not have any significant change however it was slightly better than the data from ‘After PO.’ Overall, it was apparent that all of the subjects are faster than they were as shown by ‘After PO.’

The internet responsiveness and speed data gathered after a week with PO generally increased compared to the data from ‘After PO.’ However, the data gathered from subject 2 decreased, but it was still better than the data without PO. Nonetheless, all of the data was all similar if compared to the stock data as it was with ‘After PO.’

#### Second Phase

On this phase, the data showed the difference between stock and after a week with PO. The same calculations that were used after PO were done.

With the first subject, the average app launching showed an increase in performance which contradicts the data that was gathered after PO. It showed an average decrease of 3 seconds compared to the stock data which was 9 seconds faster compared to the data after PO. The same goes with the average boot time wherein it showed an increase of 2 seconds compared to stock and after PO. However, the internet responsiveness and speed data were the same as after PO, only Google improved slightly which brings it to a tie with Quad9. Overall, there were positive results.

The second subject also showed an increase in performance. The data showed that the average was less than 11 seconds compared to stock and 5 seconds compared to the data after PO. However, the average boot time increased 0.43 seconds compared to stock, 5 seconds if compared to after PO. The internet responsiveness and speed also showed that the DNS patches were better than stock which was also shown after PO. The only difference was that other patches were tied up with Cloudflare which had the best data with after PO. In conclusion, subject 2 also showed positive results.

Lastly, subject 3 showed minimal changes which were insignificant. App launching and boot time averages only showed less than 1 second difference compared to stock and after PO. However, there was a significant change in rankings in internet responsiveness and speed. Cloudflare was top 4 after PO, however, after a week with PO, it showed that Cloudflare had increased the download speed and upload speed the most which made it top 1, Google also climbed the ranks from top 3 to top 2, while Quad9 and Stock, which were formerly top 1 and 2, tied at the bottom. Thus, subject 3 had no significant difference with app launching and boot time, however, a great shift in rankings occurred with this subject.

#### Third Phase

With the final phase, the data from the second phase were used to come up with a summarized result that will then give a general idea about the effects of the performance optimizer.

The general average calculated for app launching was -4.81 seconds. Similarly, subject 2 greatly affected the average with app launching as it did in after PO results. However, the biggest difference was that subject 1 had a significant change with its performance as described in page 54. With this massive change, the result satisfies the alternative hypothesis’ condition, therefore, the performance optimizer has a significant impact to the subjects in general after a week of application.

On the other hand, boot time’s calculated general average data was -0.82 seconds. This was almost a second more than the general average data after PO which was -1.72 seconds. The increase in boot time was caused by subject 2’s increase in boot time after a week of PO which was 5 seconds more than after PO. However, the effect of this significant increase in boot time to the average was lessened by the 3 seconds decrease of subject 1 as well as the half second decrease of subject 3. Nevertheless, due to the increase of boot time average, the alternative hypothesis’ condition was not satisfied, thus, there was no significant change to the subjects’ boot time in general after a week of application.

Despite the new data, the ranking that was calculated was still similar to the results after PO. Stock still had the lowest rank sum of 6.5 which gives it the top rank. However, there was a significant change with the rankings as Cloudflare climbed up the rankings from top 4 to top 2 which lowered the rankings of other custom patches. Thus, the null hypothesis was still accepted due to the fact that having no custom DNS was faster than having a custom DNS in general.

# REFERENCES

Computer Hope. (2019, October 7). *Batch file*. Retrieved from Computer Hope: https://www.computerhope.com/jargon/b/batchfil.htm

Computer Hope. (2019, October 7). *Command prompt*. Retrieved from Computer Hope: https://www.computerhope.com/jargon/c/commprom.htm

Frew, J. (2018, February 26). *Why Changing DNS Settings Increases Your Internet Speed*. Retrieved from Makes Use Of: https://www.makeuseof.com/tag/change-dns-increase-internet-speed/

Garcés, M. F. (2019, October 19). *Impact of the Rapid Technological Change in the Achievement of the SDGs*. Retrieved from United Nations: https://www.un.org/pga/73/2018/10/19/impact-of-the-rapid-technological-change-in-the-achievement-of-the-sdgs/

Gershenfeld, N. (2003, July 21). *PERSONAL FABRICATION*. Retrieved from Edge: https://www.edge.org/conversation/personal-fabrication

K, J. (2018, June 1). *12 BEST SOFTWARE DEVELOPMENT METHODOLOGIES WITH PROS AND CONS*. Retrieved from Acodez: https://acodez.in/12-best-software-development-methodologies-pros-cons/

McCallion, J., & Hopping, C. (2019, July 18). *What is DNS?* Retrieved from IT Pro: https://www.itpro.co.uk/domain-name-system-dns/30228/what-is-dns

Meskauskas, T. (2017, December 18). *How To Get Rid Of Bloatware In Windows 10?* Retrieved from PCrisk: https://blog.pcrisk.com/windows/12022-how-to-get-rid-of-bloatware-in-windows-10

Moore, G. E. (1965, April 19). *Cramming more components onto integrated circuits.* Retrieved from Monolithic 3D: http://www.monolithic3d.com/uploads/6/0/5/5/6055488/gordon\_moore\_1965\_article.pdf

Museum of Applied Arts & Sciences. (2017, March 23). *‘Cupcake CNC’ 3D printer by MakerBot*. Retrieved from Museum of Applied Arts & Sciences: https://collection.maas.museum/object/539939

Phillips, W. (2012). *About*. Retrieved from Ask Windows Wally: http://www.windowswally.com/about/

Phillips, W. (2014, April 6). *Speed Up Windows using Registry*. Retrieved from Ask Windows Wally: http://www.windowswally.com/speed-up-windows-using-registry/

Rachiele, G. (2018, April 9). *Software Development Methodologies*. Retrieved from Medium: https://medium.com/@gianpaul.r/software-development-methodologies-a856883a7630

Registry Recycler. (2013, July 12). *Enhance Windows Memory Management Through Registry*. Retrieved from Registry Recycler: https://www.registryrecycler.com/blog/2013/07/enhance-windows-memory-management-through-registry/

Roser, M., & Ritchie, H. (2019). *Technological Progress*. Retrieved from Our World in Data: https://ourworldindata.org/technological-progress

Rubenking, N. J. (2019, May 17). *How (and Why) to Change Your DNS Server*. Retrieved from PC Mag: https://www.pcmag.com/review/364418/how-and-why-to-change-your-dns-server

Superuser. (2014, January 23). *Batch file with commands to run as administrator and standard user*. Retrieved from Superuser: https://superuser.com/questions/705813/batch-file-with-commands-to-run-as-administrator-and-standard-user?fbclid=IwAR1czUuyK6TORAwGkAKI9XFNT99jYS3EqDxILxiSqRFBieFPYppauFffoXM

# APPENDICES

## Tables of Data

### First Phase

#### Without PO

##### Subject 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SUBJECT 1 | | | | | | | | | |
| Without PO | App Launching | | | Boot Time | | Internet Responsiveness and Speed | | | |
| Stock DNS | | | |
| Word | Chrome | Explorer | Ping | | Download | Upload |
| Trial 1 | 13.96 | 32.88 | 7.66 | 20.63 | | 5 | | 42.26 | 16.04 |
| Trial 2 | 12.43 | 30.21 | 6.08 | 21.45 | | 4 | | 31.74 | 12.84 |
| Trial 3 | 10.24 | 33.41 | 5.94 | 19.63 | | 5 | | 40.94 | 18.95 |
| Average | 12.21 | 32.167 | 6.56 | 20.57 | | 4.667 | | 38.3133 | 15.9433 |
| Performance Mean | 16.9788 | | |  |  |  |  |  |  |

##### Subject 2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SUBJECT 2 | | | | | | | | | |
| Without PO | App Launching | | | Boot Time | | | Internet Responsiveness and Speed | | |
| Stock DNS | | |
| Word | Chrome | Explorer | Ping | Download | Upload |
| Trial 1 | 42.42 | 41.94 | 6.01 | 25.27 | | | 4 | 25.14 | 12.3 |
| Trial 2 | 39.03 | 72.65 | 4.71 | 27.54 | | | 8 | 51.25 | 19.45 |
| Trial 3 | 35.59 | 79.59 | 4.81 | 25.02 | | | 8 | 30.16 | 13.54 |
| Average | 39.013 | 64.727 | 5.177 | 25.943 | | | 6.667 | 35.517 | 15.097 |
| Performance Mean | 39.306 | | |  |  |  |  |  |  |

##### Subject 3

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SUBJECT 3 | | | | | | | | | |
| Without PO | App Launching | | | Boot Time | | | Internet Responsiveness and Speed | | |
| Stock DNS | | |
| Word | Chrome | Explorer | Ping | Download | Upload |
| Trial 1 | 3.49 | 8.21 | 3.38 | 16.94 | | | 4 | 49.69 | 18.89 |
| Trial 2 | 4.48 | 9.95 | 2.23 | 15.41 | | | 4 | 47.48 | 19.29 |
| Trial 3 | 4.29 | 8.22 | 1.51 | 13.14 | | | 4 | 48.62 | 18.09 |
| Average | 4.087 | 8.793 | 2.373 | 15.163 | | | 4 | 48.597 | 18.76 |
| Performance Mean | 5.084 | | |  |  |  |  |  |  |

#### With PO

##### Subject 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SUBJECT 1 | | | | | | | | | |
| With PO | App Launching | | | Boot Time | | | Internet Responsiveness and Speed | | |
| Cloudflare DNS | | |
| Word | Chrome | Explorer | Ping | Download | Upload |
| Trial 1 | 20.89 | 42.95 | 5.19 | 21.69 | | | 6 | 26.45 | 16.69 |
| Trial 2 | 15.66 | 55.16 | 4.99 | 20.04 | | | 6 | 23.05 | 18.98 |
| Trial 3 | 16.18 | 42.99 | 4.13 | 19.58 | | | 5 | 25.01 | 15.99 |
| Average | 17.58 | 47.03 | 4.77 | 20.44 | | | 5.67 | 24.84 | 17.22 |
| Performance Mean | 23.13 | | |  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SUBJECT 1 | | | | | |
| Internet Responsiveness and Speed | | | Internet Responsiveness and Speed | | |
| Google DNS | | | Quad9 DNS | | |
| Ping | Download | Upload | Ping | Download | Upload |
| 5 | 24 | 13.9 | 5 | 29.23 | 18.23 |
| 4 | 28.41 | 14.84 | 6 | 26.35 | 19.12 |
| 5 | 28.87 | 15.43 | 5 | 25.21 | 18.73 |
| 4.67 | 27.09 | 14.72 | 5.33 | 26.93 | 18.69 |
|  |  |  |  |  |  |

##### Subject 2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SUBJECT 2 | | | | | | | | | | |
| With PO | App Launching | | | Boot Time | | | Internet Responsiveness and Speed | | | |
| Cloudflare DNS | | | |
| Word | Chrome | Explorer | Ping | | Download | Upload |
| Trial 1 | 31.35 | 40.13 | 5.25 | 19.95 | | | 5 | | 54.67 | 14.07 |
| Trial 2 | 31.22 | 76.03 | 4.28 | 21.06 | | | 8 | | 54.17 | 18.99 |
| Trial 3 | 32.72 | 59.48 | 5.37 | 19.93 | | | 8 | | 58.69 | 9.18 |
| Average | 31.76 | 58.54666667 | 4.966666667 | 20.31333333 | | | 7 | | 55.84333333 | 14.08 |
| Performance Mean | 31.75888889 | | |  |  |  | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SUBJECT 2 | | | | | |
| Internet Responsiveness and Speed | | | Internet Responsiveness and Speed | | |
| Google DNS | | | Quad9 DNS | | |
| Ping | Download | Upload | Ping | Download | Upload |
| 5 | 53.02 | 18.13 | 5 | 50.53 | 18.43 |
| 7 | 55.72 | 12.43 | 9 | 55.48 | 5.49 |
| 8 | 57.52 | 8.5 | 9 | 55.93 | 11.86 |
| 6.666666667 | 55.42 | 13.02 | 7.666666667 | 53.98 | 11.92666667 |
|  |  |  |  |  |  |

##### Subject 3

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SUBJECT 3 | | | | | | | | | | | |
| With PO | App Launching | | | Boot Time | Internet Responsiveness and Speed | | | | | | |
| Cloudflare DNS | | | | | | |
| Word | Chrome | Explorer | Ping | | | Download | | | Upload |
| Trial 1 | 4.66 | 15.67 | 1.8 | 15.97 | 4 | | | 49.1 | | | 17.55 |
| Trial 2 | 3.48 | 13.29 | 1.59 | 16.23 | 5 | | | 44.48 | | | 13.38 |
| Trial 3 | 3.94 | 11.6 | 1.65 | 15.13 | 5 | | | 42.94 | | | 19.9 |
| Average | 4.03 | 13.52 | 1.68 | 15.78 | 4.67 | | | 45.51 | | | 16.943333 |
| Performance Mean | 6.41 | | |  | |  |  | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SUBJECT 3 | | | | | |
| Internet Responsiveness and Speed | | | Internet Responsiveness and Speed | | |
| Google DNS | | | Quad9 DNS | | |
| Ping | Download | Upload | Ping | Download | Upload |
| 4 | 46.47 | 18.15 | 4 | 46.06 | 17.73 |
| 4 | 47.73 | 17.92 | 5 | 50.46 | 19.05 |
| 4 | 50.51 | 19.1 | 4 | 49.82 | 20.24 |
| 4 | 48.23666667 | 18.39 | 4.33 | 48.78 | 19.00666667 |
|  |  |  |  |  |  |

#### PO after a week

##### Subject 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SUBJECT 1 | | | | | | | | | |
| With PO | App Launching | | | Boot Time | | | Internet Responsiveness and Speed | | |
| Cloudflare DNS | | |
| Word | Chrome | Explorer | Ping | Download | Upload |
| Trial 1 | 21.03 | 18.05 | 4.48 | 18.1 | | | 4 | 27.72 | 13.85 |
| Trial 2 | 15.39 | 18.39 | 4.72 | 16.81 | | | 5 | 27.52 | 12.36 |
| Trial 3 | 15.9 | 22.88 | 4.06 | 18.24 | | | 5 | 25.11 | 16.73 |
| Average | 17.44 | 19.7733 | 4.42 | 17.717 | | | 4.67 | 26.783333 | 14.313333 |
| Performance Mean | 13.87777778 | | |  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SUBJECT 1 | | | | | |
| Internet Responsiveness and Speed | | | Internet Responsiveness and Speed | | |
| Google DNS | | | Quad9 DNS | | |
| Ping | Download | Upload | Ping | Download | Upload |
| 5 | 31.63 | 12.35 | 6 | 32.1 | 17.59 |
| 5 | 34.09 | 17.32 | 5 | 33.19 | 15.85 |
| 5 | 36.32 | 17.61 | 5 | 36.08 | 18.93 |
| 5 | 34.013333 | 15.76 | 5.333 | 33.79 | 17.456667 |
|  |  |  |  |  |  |

##### Subject 2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SUBJECT 2 | | | | | | | | | | |
| With PO | App Launching | | | Boot Time | | | Internet Responsiveness and Speed | | | |
| Cloudflare DNS | | | |
| Word | Chrome | Explorer | Ping | | Download | Upload |
| Trial 1 | 34.48 | 33.11 | 5.55 | 26.02 | | | 5 | | 47.32 | 13.22 |
| Trial 2 | 36.71 | 34.56 | 5.93 | 26.87 | | | 7 | | 42.76 | 19.31 |
| Trial 3 | 37.09 | 35.66 | 5.48 | 26.22 | | | 9 | | 45.11 | 11 |
| Average | 36.0933 | 34.4433 | 5.653 | 26.37 | | | 7 | | 45.063 | 14.51 |
| Performance Mean | 25.39666667 | | |  |  |  | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SUBJECT 2 | | | | | |
| Internet Responsiveness and Speed | | | Internet Responsiveness and Speed | | |
| Google DNS | | | Quad9 DNS | | |
| Ping | Download | Upload | Ping | Download | Upload |
| 6 | 42.63 | 18.19 | 5 | 43.11 | 13.25 |
| 10 | 41.4 | 12.85 | 7 | 44.96 | 18.38 |
| 7 | 57.28 | 9.12 | 7 | 43.49 | 11.78 |
| 7.6666667 | 47.103333 | 13.386667 | 6.3333333 | 43.853333 | 14.47 |
|  |  |  |  |  |  |

##### Subject 3

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SUBJECT 3 | | | | | | | | | | | |
| With PO | App Launching | | | Boot Time | Internet Responsiveness and Speed | | | | | | |
| Cloudflare DNS | | | | | | |
| Word | Chrome | Explorer | Ping | | | | Download | | Upload |
| Trial 1 | 5.13 | 8.67 | 2.78 | 14.94 | 4 | | | | 47.76 | | 20.15 |
| Trial 2 | 2.16 | 7.18 | 1.54 | 15.21 | 5 | | | | 53.59 | | 20.5 |
| Trial 3 | 4 | 9.05 | 1.4 | 15.25 | 4 | | | | 54.3 | | 20.57 |
| Average | 3.76333 | 8.3 | 1.90667 | 15.1333 | 4.33 | | | | 51.883333 | | 20.406667 |
| Performance Mean | 4.656666667 | | |  | |  |  |  | |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SUBJECT 3 | | | | | |
| Internet Responsiveness and Speed | | | Internet Responsiveness and Speed | | |
| Google DNS | | | Quad9 DNS | | |
| Ping | Download | Upload | Ping | Download | Upload |
| 4 | 53.75 | 19.9 | 4 | 49.86 | 18.95 |
| 4 | 48.32 | 20.25 | 5 | 51.69 | 18.63 |
| 5 | 52.62 | 20.46 | 4 | 49.97 | 20.05 |
| 4.3333333 | 51.563333 | 20.203333 | 4.3333333 | 50.506667 | 19.21 |
|  |  |  |  |  |  |

### Second Phase

#### App launching and initialization

##### Subject 1

|  |  |
| --- | --- |
| SUBJECT 1 | |
|
| Performance Mean | App Launching |
|
| w/ PO | 23.12666667 |
| w/o PO | 16.97888889 |
| Performance Difference | 6.14777777777778 |

##### Subject 2

|  |  |
| --- | --- |
| SUBJECT 2 | |
|
| Performance Mean | App Launching |
|
| w/ PO | 31.75888889 |
| w/o PO | 36.30555556 |
| Performance Difference | -4.54666666666667 |

##### Subject 3

|  |  |
| --- | --- |
| SUBJECT 3 | |
|
| Performance Mean | App Launching |
|
| w/ PO | 6.408888889 |
| w/o PO | 5.084444444 |
| Performance Difference | 1.324444444 |

#### Boot time

##### Subject 1

|  |  |
| --- | --- |
| SUBJECT 1 | |
|
| Average | Boot Time |
|
| w/ PO | 20.43666667 |
| w/o PO | 20.57 |
| Average Difference | -0.133333333333329 |

##### Subject 2

|  |  |
| --- | --- |
| SUBJECT 2 | |
|
| Performance Mean | Boot Time |
|
| w/ PO | 20.31333333 |
| w/o PO | 25.94333333 |
| Average Difference | 0 |

##### Subject 3

|  |  |
| --- | --- |
| SUBJECT 3 | |
|
| Performance Mean | Boot Time |
|
| w/ PO | 15.77666667 |
| w/o PO | 15.16333333 |
| Average Difference | 0.613333333333335 |

#### Internet Speed and Responsiveness

##### Subject 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Internet Responsiveness and Speed Ranking | | | | | |
| Subject 1 | | | | | |
| DNS | Ping | Download | Upload | Total | Ranking |
|
| Cloudflare | 4 | 4 | 2 | 10 | 4 |
| Google | 1.5 | 2 | 4 | 7.5 | 3 |
| Quad9 | 3 | 3 | 1 | 7 | 2 |
| Stock (Without PO) | 1.5 | 1 | 3 | 5.5 | 1 |

##### Subject 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Internet Responsiveness and Speed Ranking | | | | | |
| Subject 2 | | | | | |
| DNS | Ping | Download | Upload | Total | Ranking |
|
| Cloudflare | 3 | 1 | 2 | 6 | 1 |
| Google | 1.5 | 2 | 3 | 6.5 | 2.5 |
| Quad9 | 4 | 3 | 4 | 11 | 4 |
| Stock (Without PO) | 1.5 | 4 | 1 | 6.5 | 2.5 |

##### Subject 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Internet Responsiveness and Speed Ranking | | | | | |
| Subject 3 | | | | | |
| DNS | Ping | Download | Upload | Total | Ranking |
|
| Cloudflare | 4 | 4 | 4 | 12 | 4 |
| Google | 1.5 | 3 | 3 | 7.5 | 3 |
| Quad9 | 3 | 1 | 1 | 5 | 2 |
| Stock (Without PO) | 1.5 | 2 | 2 | 5.5 | 1 |

#### PO after a week

#### App launching and initialization

##### Subject 1

|  |  |
| --- | --- |
| SUBJECT 1 | |
|
| Performance Mean | App Launching |
|
| w/ PO | 13.87777778 |
| w/o PO | 16.97888889 |
| Performance Difference | -3.10111111111111 |

##### Subject 2

|  |  |
| --- | --- |
| SUBJECT 2 | |
|
| Performance Mean | App Launching |
|
| w/ PO | 25.39666667 |
| w/o PO | 36.30555556 |
| Performance Difference | -10.9088888888889 |

##### Subject 3

|  |  |
| --- | --- |
| SUBJECT 3 | |
|
| Performance Mean | App Launching |
|
| w/ PO | 4.656666667 |
| w/o PO | 5.084444444 |
| Performance Difference | -0.427777777777779 |

#### Boot time

##### Subject 1

|  |  |
| --- | --- |
| SUBJECT 1 | |
|
| Average | Boot Time |
|
| w/ PO | 17.71666667 |
| w/o PO | 20.57 |
| Average Difference | -2.85333333333333 |

##### Subject 2

|  |  |
| --- | --- |
| SUBJECT 2 | |
|
| Performance Mean | Boot Time |
|
| w/ PO | 26.37 |
| w/o PO | 25.94333333 |
| Average Difference | 0.426666666666669 |

##### Subject 3

|  |  |
| --- | --- |
| SUBJECT 3 | |
|
| Performance Mean | Boot Time |
|
| w/ PO | 15.13333333 |
| w/o PO | 15.16333333 |
| Average Difference | -0.0300000000000011 |

#### Internet Speed and Responsiveness

##### Subject 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Internet Responsiveness and Speed Ranking | | | | | |
| Subject 1 | | | | | |
| DNS | Ping | Download | Upload | Total | Ranking |
|
| Cloudflare | 1.5 | 4 | 4 | 9.5 | 4 |
| Google | 3 | 2 | 3 | 8 | 2.5 |
| Quad9 | 4 | 3 | 1 | 8 | 2.5 |
| Stock (Without PO) | 1.5 | 1 | 2 | 4.5 | 1 |

##### Subject 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Internet Responsiveness and Speed Ranking | | | | | |
| Subject 2 | | | | | |
| DNS | Ping | Download | Upload | Total | Ranking |
|
| Cloudflare | 3 | 2 | 2 | 7 | 2 |
| Google | 4 | 1 | 4 | 9 | 4 |
| Quad9 | 1 | 3 | 3 | 7 | 2 |
| Stock (Without PO) | 2 | 4 | 1 | 7 | 2 |

##### Subject 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Internet Responsiveness and Speed Ranking | | | | | |
| Subject 3 | | | | | |
| DNS | Ping | Download | Upload | Total | Ranking |
|
| Cloudflare | 3 | 1 | 1 | 5 | 1 |
| Google | 3 | 2 | 2 | 7 | 2 |
| Quad9 | 3 | 3 | 3 | 9 | 3.5 |
| Stock (Without PO) | 1 | 4 | 4 | 9 | 3.5 |

### Final Phase

#### App launching and initialization

|  |  |
| --- | --- |
| App Launching | |
|
| Subject | Performance Difference |
|
| #1 | 6.147777778 |
| #2 | -4.546666667 |
| #3 | 1.32444444 |
| General Average of Performance Difference | 0.98 |

Boot time

|  |  |
| --- | --- |
| Boot Time | |
|
| Subject | Average BT Difference |
|
| #1 | -0.133333333 |
| #2 | -5.63 |
| #3 | 0.613333333 |
| General Average of BT Difference | -1.72 |

#### Internet Speed and Responsiveness

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Internet Responsiveness and Speed | | | | | |
|
| DNS | Subjects | | | Total | Final Ranking |
| #1 | #2 | #3 |
| Cloudflare | 4 | 1 | 4 | 9 | 4 |
| Google | 3 | 2.5 | 3 | 8.5 | 3 |
| Quad9 | 2 | 4 | 1 | 7 | 2 |
| Stock | 1 | 2.5 | 2 | 5.5 | 1 |

#### PO after a week

#### App launching and initialization

|  |  |
| --- | --- |
| App Launching | |
|
| Subject | Performance Difference |
|
| #1 | -3.101111111 |
| #2 | -10.90888889 |
| #3 | -0.427777778 |
| General Average of Performance Difference | -4.81 |

#### Boot time

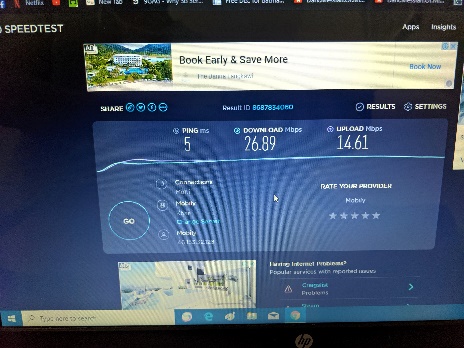
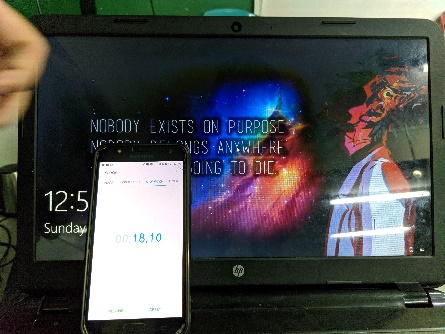
|  |  |
| --- | --- |
| Boot Time | |
|
| Subject | Average BT Difference |
|
| #1 | -2.853333333 |
| #2 | 0.426666667 |
| #3 | -0.03 |
| General Average of BT Difference | -0.82 |

#### Internet Speed and Responsiveness

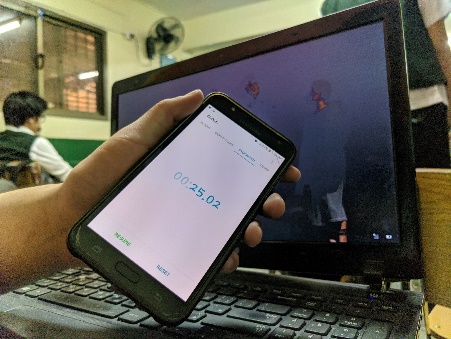
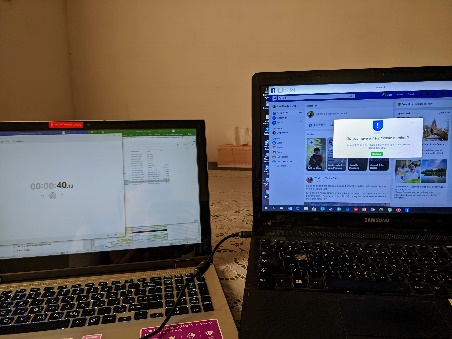
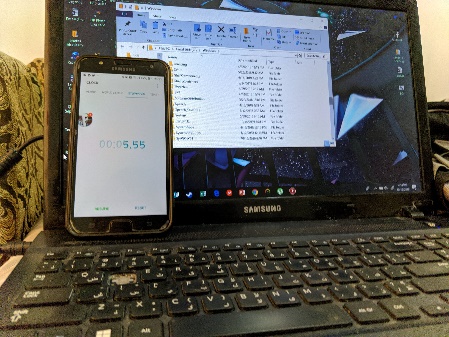
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Internet Responsiveness and Speed | | | | | |
|
| DNS | Subjects | | | Total | Final Ranking |
| #1 | #2 | #3 |
| Cloudflare | 4 | 2 | 1 | 7 | 2 |
| Google | 2.5 | 4 | 2 | 8.5 | 4 |
| Quad9 | 2.5 | 2 | 3.5 | 8 | 3 |
| Stock | 1 | 2 | 3.5 | 6.5 | 1 |

## Short Experiment Gallery

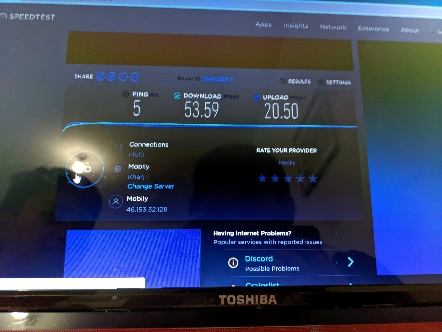
### Subject 1

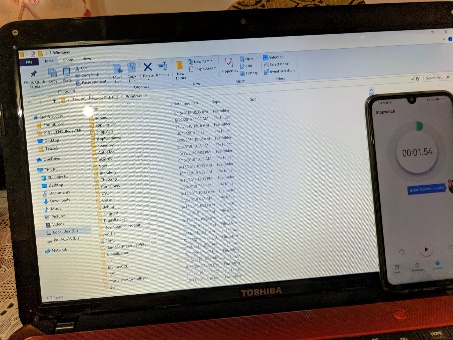


### Subject 2



### Subject 3





### Researchers

Full Album Link:

https://photos.app.goo.gl/mngQq4KWvWJN2Lnw9