AngZuelaMus

Removable Drive File Recovery Software

Azuela, Piolo Andrew V.
Angquico, Justine
Musni, Roniell C.

AZM-ReDFiReS

TABLE OF CONTENTS

AZUELA, PIOLO ANDREW V., JUSTINE ANGQUICO AND RONIELL C. MUSNI. AZM-ReDFiReS: AngZuelaMus Removable Drive File Recovery Software. *Research paper for the Regional Science and Technology Fair 2015 (Team Category-Physical Science). Division of Cavite City. Cavite National High School, Cavite City, Region IV-A CALABARZON.*

ABSTRACT

Development of a software that could lessen the number of virus-infected flash drives is the primary objective of this study. The antivirus software is coded in Code::Blocks IDE and Notepad++ using C++ and Batch programming languages, and is programmed to remove virus and retrieve infected files in flash drives. This can be run on Windows XP and later operating systems, by copying the software files to the infected flash drive and opening the batch file.

Functionality and acceptability of the software was tested on 10 respondents tasked to use the software, given the proper usage instructions. Based on the prepared rubric, the respondents rated the acceptability based on their usage experience. Acceptability results showed that in average, the respondents rated the software a score of 3.9, which means the usage instructions given are easy to understand and in-program texts and guides are easy to read and follow. The software was run by the respondents with no glitches encountered throughout the testing proper.

` *As per the results stated, this antivirus software can now be used by computer users to disinfect their flash drives from viruses. Thus, this can now help students, teachers, printing shop owners, and other computer users in keeping their computers and flash drives away from virus infections.*

The simplicity of the programming and user interface of the software was considered for future improvements and studies by researchers and other computer programmers.

INTRODUCTION

Background of the Study

Only 55% of Filipinos have at least one computer in their homes (TekCarta–GeneratorResearch, 2015). Due to this, most of them tend to go to internet shops just to use one, not knowing that 3 out of 10 computers are vulnerable to viruses (Microsoft, 2013). In addition to that, people often use flash drives to connect to computers to save their documents into it. Unknown to them, transferring files from an infected computer to a flash drive will transmit also the infection alongside with the file being transferred. This may cause file corruptions and drive corruptions which will make the files or drive contents unreadable next time they will insert their flash drives to other computers.

In order to avoid these situations, some people are now more careful on saving files on computers in internet shops, while others install antivirus softwares in their computers. These measures effectively reduced flash drive infections. On the other hand, infections like Win32/Heur, Win32/Sality, and INF/Autorun, which automatically hides files and folders in the flash drive upon insertion into the infected computer, have now more complex coding (Kaspersky, 2014; Microsoft, 2013). That coding made these viruses unnoticeable and untreatable to commercial antiviruses right now.

With raised awareness and knowledge, programmers are now working to code antivirus softwares which will clean and disinfect flash drives from tougher viruses. One of those experimental programs is the AngZuelaMus Removable Drive File Recovery Software, which will address the problem of infections in flash drives and will help other programmers in resolving this issue.

Statement of the Problem

This study aims to develop a program that could lessen the number of virus-infected flash drives. Specifically, it aims to:

- create a software application that will disinfect and remove viruses in flash drives;
- test the functionality of the software; and
- test the acceptability of the software to the end users.

Significance of the Study

This project will help reduce the number of virus-infected flash drives. In addition to that, this will benefit the following individuals:

Ordinary computer users. These people, which include internet shop users, will greatly benefit from this software, for they can now safely transfer files from a rented computer to their flash drives without worrying to lose it afterwards due to virus infection.

Printing shop owners. After these computer shop users have saved their files into their flash drives, they will flock to printing shops to print their documents. However, printing shop owners are usually worried because these flash drives can make their computers infected. Upon using this software, these owners will be worry-free as they will be assured that their computers cannot be infected by flash drives. They can also help cleaning infected flash drives from their customers by using the software.

Programmers. This software will help programmers by modifying its codes to create a more powerful antivirus software that will be widely used worldwide.

Scope and Limitations

The primary focus of this study is to develop a software that could disinfect virus-infected flash drives. The computers used in coding the program is a Windows 7 32-bit desktop computer and a Windows 8 64-bit laptop computer which are owned by the researchers. The effectivity of the program in other operating systems except Windows XP and later is beyond the scope of the study. The acceptability of the software to ordinary users was determined by test-running the software to beta users.

The creation and testing of the created program were done from June to August, 2015 at Cavite National High School, Cavite City.

METHODOLOGY

Material List

- Windows 7 desktop computer

- Windows 8 laptop computer

- Code::Blocks IDE Compiler

- Notepad++

- Infected flash drive

Preparation of Materials

The primary materials needed in creation of the antivirus program are a computer and an infected flash drive. Windows 7 32-bit desktop and Windows 8 64-bit laptop were the operating system specifications of the computers used. These computers will set as the facilitators for the coding and debugging of the program. Notepad++ software was installed in the computer for coding batch files. Code::Blocks IDE Compiler software was also installed for coding and debugging .exe files written in C++ programming language. Infected flash drive was used in testing the program's functionality. Figures 1 and 2 show the user interfaces of the coding programs used.

Figure 1. Code::Blocks software with retrieving codes written

Figure 2. Notepad++ with batch codes and a batch application running

Creation of the Antivirus Program

In starting the creation of the antivirus program, the computer was turned on and the coding softwares (Notepad++ and Code::Blocks Compiler) were run. The necessary command codes in detecting and deleting the computer worm or virus in a flash drive were written in the Notepad++ and saved as .bat or batch file. The command codes in retrieving the hidden files were written in Code::Blocks IDE Compiler and debugged as .exe. Figures 3 and 4 are the Batch and C++ codes used for the program.

```
@echo off
title AngZuelaMus Flash Drive Virus Remover
color 0a
::#######

:one
cls
echo Welcome to AngZuelaMus Antivirus == Virus Remover for Flash Drives
echo.
echo.
echo Please choose the drive location of the flash drive you want to clean.
echo.
echo D: = press 1
echo E: = press 2
echo F: = press 3
echo G: = press 4
echo H: = press 5
echo.
set /p ndrive=Enter number here:

if "%ndrive%" == "1" goto :D
if "%ndrive%" == "d" goto :D
if "%ndrive%" == "d:" goto :D
if "%ndrive%" == "D" goto :D
if "%ndrive%" == "D:" goto :D
if "%ndrive%" == "2" goto :E
if "%ndrive%" == "e" goto :E
if "%ndrive%" == "e:" goto :E
if "%ndrive%" == "E" goto :E
if "%ndrive%" == "E:" goto :E
if "%ndrive%" == "3" goto :F
if "%ndrive%" == "f" goto :F
if "%ndrive%" == "f:" goto :F
if "%ndrive%" == "F" goto :F
if "%ndrive%" == "F:" goto :F
if "%ndrive%" == "4" goto :G
if "%ndrive%" == "g" goto :G
if "%ndrive%" == "g:" goto :G
if "%ndrive%" == "G" goto :G
if "%ndrive%" == "G:" goto :G
if "%ndrive%" == "5" goto :H
if "%ndrive%" == "h" goto :H
if "%ndrive%" == "h:" goto :H
if "%ndrive%" == "H" goto :H
if "%ndrive%" == "H:" goto :H
```

```
:D
cls
attrib -r -a -s -h -i /s /d D:\*.*
echo The Phase 1 of the Virus Removal has been finished.

if exist "D:\?" (
        goto :delD
        )else (
        goto :lobdelD
        )

:E
cls
attrib -r -a -s -h -i /s /d E:\*.*
echo The Phase 1 of the Virus Removal has been finished.

if exist "E:\?" (
        goto :delE
        )else (
        goto :lobdelE
        )

:F
cls
attrib -r -a -s -h -i /s /d F:\*.*
echo The Phase 1 of the Virus Removal has been finished.

if exist "F:\?" (
        goto :delF
        )else (
        goto :lobdelF
        )
:G
cls
attrib -r -a -s -h -i /s /d G:\*.*
echo The Phase 1 of the Virus Removal has been finished.
if exist "G:\?" (
        goto :delG
        )else (
        goto :lobdelG
        )

:H
cls
attrib -r -a -s -h -i /s /d H:\*.*
echo The Phase 1 of the Virus Removal has been finished.

if exist "H:\?" (
        goto :delH
        )else (
        goto :lobdelH
        )

:delD
del D:\*.lnk
del D:\*.tmp
del D:\*.exe
del D:\*.vbs
del D:\*.ini
del D:\*.db
start /d "D:\FDVRCleaner\FDVR-for-RenD\bin\debug\" FDVR-for-RenD.exe
echo.
echo Press enter to continue...
pause >nul
cls
xcopy "D:\fdvr" "D:\" /s /e /c /h /y /r /k
echo Press enter to continue...
pause >nul
cls
rd "D:\fdvr" /s
```

```
echo.
echo The cleaning has been finished. Thanks for using AngZuelaMus Flash Drive Virus
Remover!

pause >nul
exit

:delE
cls
del E:\*.lnk
del E:\*.tmp
del E:\*.exe
del E:\*.vbs
del E:\*.ini
del E:\*.db
start /d "E:\FDVRCleaner\FDVR-for-renE\bin\debug\" FDVR-for-renE.exe
echo.
echo Press enter to continue...
pause >nul
cls
xcopy "E:\fdvr" "E:\" /s /e /c /h /y /r /k
echo Press enter to continue...
pause >nul

cls
rd "E:\fdvr" /s
echo.
echo The cleaning has been finished. Thanks for using AngZuelaMus Flash Drive Virus
Remover!

pause >nul
exit

:delF
cls
del F:\*.lnk
del F:\*.tmp
del F:\*.exe
del F:\*.vbs
del F:\*.ini
del F:\*.db
start /d "F:\FDVRCleaner\FDVR-for-RenF\bin\debug\" FDVR-for-RenF.exe
echo.
echo Press enter to continue...
pause >nul
cls
xcopy "F:\fdvr" "F:\" /s /e /c /h /y /r /k
echo Press enter to continue...
pause >nul

:delH
cls
del H:\*.lnk
del H:\*.tmp
del H:\*.exe
del H:\*.vbs
del H:\*.ini
del H:\*.db
start /d "H:\FDVRCleaner\FDVR-for-RenH\bin\debug\" FDVR-for-RenH.exe
echo.
echo Press enter to continue...
pause >nul
cls
xcopy "H:\fdvr" "H:\" /s /e /c /h /y /r /k
echo Press enter to continue...
pause >nul
cls
rd "H:\fdvr" /s
echo.
```

```
echo The cleaning has been finished. Thanks for using AngZuelaMus Flash Drive Virus
Remover!

pause >nul
exit
:lobdelD
cls
del D:\*.lnk
del D:\*.tmp
del D:\*.ini
del D:\*.db
echo The cleaning has been finished. Thanks for using AngZuelaMus Flash Drive Virus
Remover!

pause >nul
exit

:lobdelE
cls
del E:\*.lnk
del E:\*.tmp
del E:\*.ini
del E:\*.db
echo The cleaning has been finished. Thanks for using AngZuelaMus Flash Drive Virus
Remover!

pause >nul
exit

:lobdelF
cls
del F:\*.lnk
del F:\*.tmp
del F:\*.ini
del F:\*.db
echo The cleaning has been finished. Thanks for using AngZuelaMus Flash Drive Virus
Remover!

pause >nul
exit

:lobdelG
cls
del G:\*.lnk
del G:\*.tmp
del G:\*.ini
del G:\*.db
echo The cleaning has been finished. Thanks for using AngZuelaMus Flash Drive Virus
Remover!

pause >nul
exit

:lobdelH
cls
del H:\*.lnk
del H:\*.tmp
del H:\*.ini
del H:\*.db
echo The cleaning has been finished. Thanks for using AngZuelaMus Flash Drive Virus
Remover!

pause >nul
exit
```

Figure 3. Batch codes used for the antivirus software

```
#include <iostream>
    #include <string>
    #include <cstdio>
    using namespace std;

    int main () {

      string srcName  = "H:/ /";
      string destName = "H:/fdvr/";

      int   ret       = rename(srcName.c_str(), destName.c_str());

      if   (ret == -1) { cout << "Unable to rename folder: "        << srcName ; }
      else             { cout << "Successfully renamed folder to: " << destName; }

      return 1;
    }
```

Figure 4. C++ codes used for the antivirus software

Testing of the Program and Data Gathering

To test the functionality of the device, the program was placed into a test computer with an infected flash drive inserted and was test run by 10 respondents given the proper usage instructions. Respondents were tasked to evaluate the ease of using the application and the result output of the program to determine its level of acceptability to the end users and its level of functionality. The acceptability and functionality of the program was evaluated by respondents using the following rubric:

Acceptability Rubric

Score		Verbal Description
4	=	Usage instructions easy to understand, in-program texts and guides easy to read and follow.
3	=	Usage instructions easy to understand, some in-program texts and guides too small to read yet still followable.
2	=	Usage instructions easy to understand, some in-program texts and guides too small to read or have complex terms.
1	=	Usage instructions have complex terms difficult to understand, some in-program texts and guides are too small to read or have complex terms.

Data Analysis

Qualitative data received during the test run process was transformed and converted into a quantitative distribution. Mean was used for all the obtained data in the ratings performed. Summary of the methodology used in this study was summarized in Figure 5.
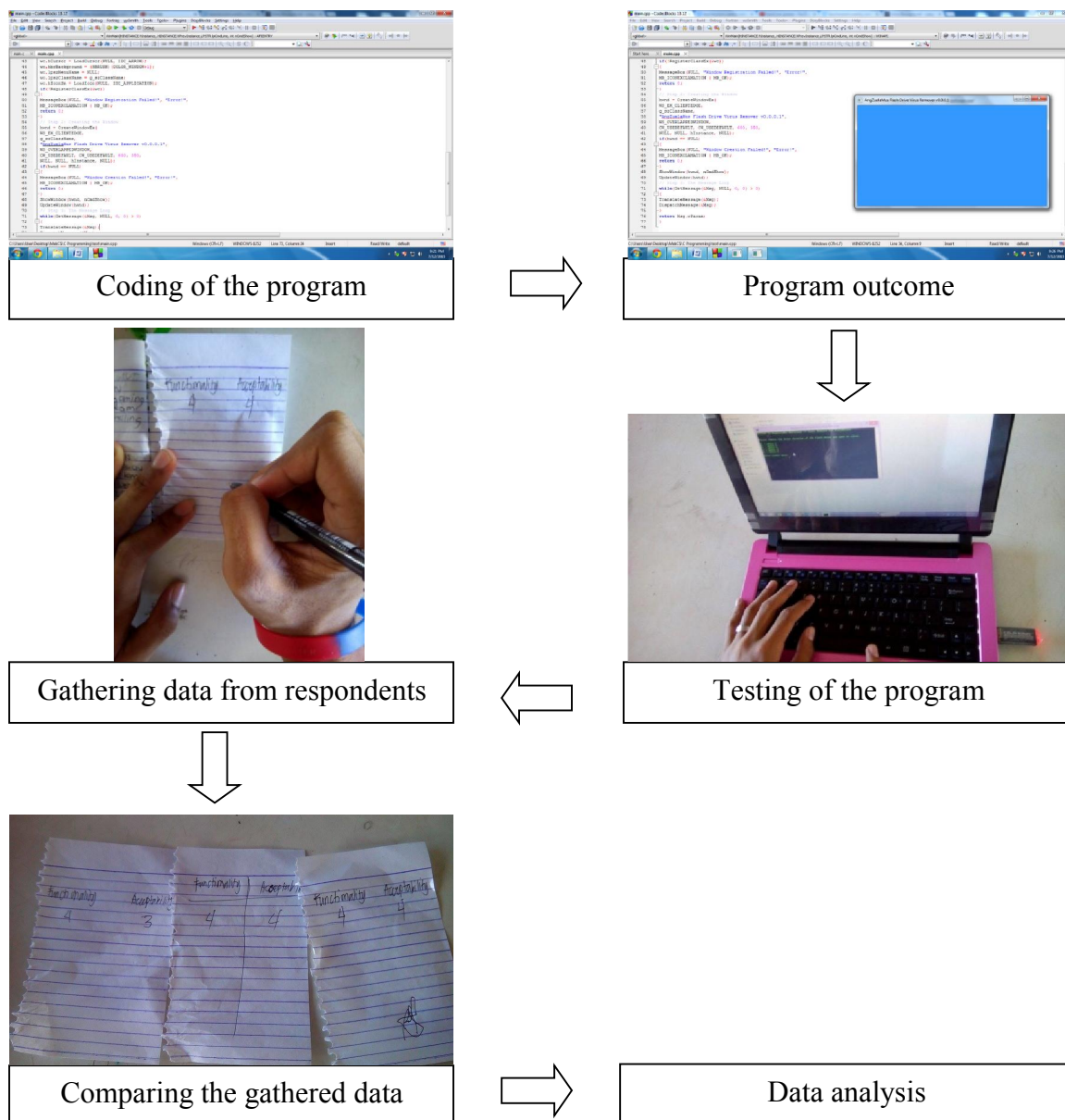
Coding of the program

Program outcome

Gathering data from respondents

Testing of the program

Comparing the gathered data

Data analysis

Figure 5. Methodology flowchart of the antivirus program

RESULTS AND DISCUSSION

Specification, Installation, and Operation of the Flash Drive Virus Remover

The AngZuelaMus Flash Drive Virus Remover uses two components for proper functioning – the computer and the software itself. After several bug fixes and code revisions on the program, it was observed that the software can only function with the drives placed on its start screen, and a folder with a blank name always exist after the program has removed the virus. From four drive name selections, it was increased to five, adding D: drive to the selection. Blank named folder was also renamed for a faster file retrieval process. Table 1 shows the complete basic information of the software. Figure 6 shows the complete interface of the software.

Table 1. Specification, installation and operation of the program.

Specifications	Computer	Windows 8 64-bit touch screen laptop with no virus infection
	Program	Runs on Windows XP, Windows 7, and Windows 8
		Runs in a CMD interface
		Coded using Batch and C++ programming languages
Installation		The program and its files are copied to the "Desktop" folder of the computer.
		(File path – X:\Users\%User%\Desktop\)

|	
Operation	Infected flash drive is inserted to the computer. The program and its files were copied from the computer to the main directory of the flash drive. The program is then run. The drive name of the infected flash drive is identified by the user and is chosen the option for the said drive name. The program will then remove the virus and retrieve the files infected. After the prompt turns out, the program has finished cleaning the drive.
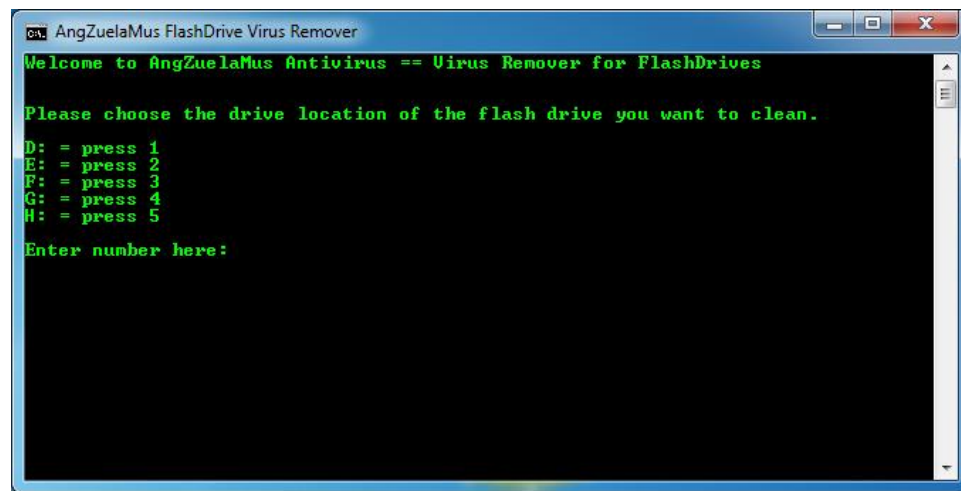
Figure 6. Flash Drive Virus Remover program running on Windows 7

Functionality of the Program

The software was used by 10 respondents and was tasked to run the software in infected flash drives, given the proper program usage instructions. After the test proper, the functionality rubric was given to the respondents and they rated the functionality of the software. Figure 7 below shows the proper usage instructions of the program.
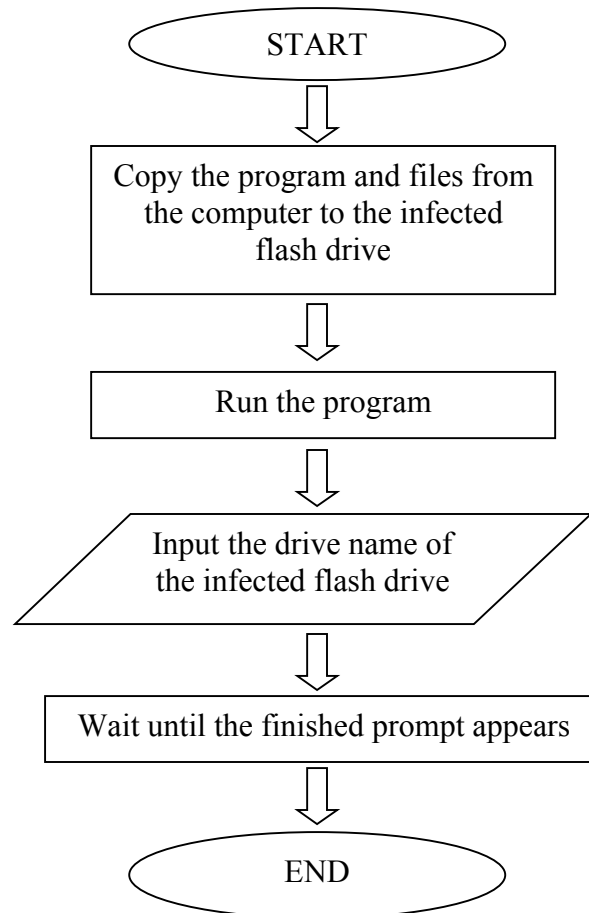

Figure 7. The flowchart of usage instructions of the program

Acceptability of the Program

The software was used by 10 respondents and was tasked to run the software in infected flash drives, given the proper program usage instructions. After the test proper, the acceptability rubric was given to the respondents to examine the ease of usage of the software. Table 2 below shows the acceptability rubric used.

Table 2. Acceptability rubric given to the respondents.

Score	Description
4	Usage instructions easy to understand, in-program texts and guides easy to read and follow.
3	Usage instructions easy to understand, some in-program texts and guides too small to read yet still followable.
2	Usage instructions easy to understand, some in-program texts and guides too small to read or have complex terms.
1	Usage instructions have complex terms difficult to understand, some in-program texts and guides are too small to read or have complex terms.

Table 3. Acceptability score given by the respondents.

Respondent No.	Score
1	4
2	3
3	4
4	4
5	4
6	4
7	4
8	4
9	4
10	4
Average	3.9

Table 3 shows the acceptability score given by the respondents. Nine of the respondents gave a rating of 4 while one gave a rating of 3. In average, the respondents gave a rating of 3.9.

Table 4. Average ranged acceptability rubric

Score	Description
3.26 − 4.00	Usage instructions easy to understand, in-program texts and guides easy to read and follow.
2.51 − 3.25	Usage instructions easy to understand, some in-program texts and guides too small to read yet still followable.
1.76 − 2.50	Usage instructions easy to understand, some in-program texts and guides too small to read or have complex terms.
1.00 − 1.75	Usage instructions have complex terms difficult to understand, some in-program texts and guides are too small to read or have complex terms.

Table 3 shows that nine of the respondents rated 4, which means software's usage instructions, and in-program texts and guides were easy to read, understand, and follow, according to their experience. However, one respondent rated 3, which means that the usage instructions were easy to understand yet some in-program texts and guides are too small to read. In average, the respondents rated it 3.9, which means, based on Table 4, the program's instructions, texts, and guides are easy to be read, understood, and followed. In the testing process, the respondents did not encounter any glitches on the software, meaning that the software ran smoothly and functioned properly.

Overall, the created antivirus software runs its functions smoothly and properly as expected, with no glitches and bugs encountered. With its basic user interface, the

respondents easily followed the usage instructions and executed commands properly to make the software function properly.

SUMMARY AND CONCLUSION

An antivirus software that could lessen the number of virus-infected flash drives was developed. The antivirus software was programmed to remove computer viruses, such as Win32/Heur, Win32/Sality, and INF/Autorun, and recover infected files in flash drives. This software was coded in Code::Blocks IDE and Notepad++ using C++ and Batch programming languages in Windows 7 32-bit desktop and Windows 8 64-bit laptop computers. This can be run on Windows XP, 7, and 8 operating systems, by copying the software files to the infected flash drive and opening the batch file.

Functionality and acceptability of the software was tested on 10 respondents tasked to use the software, given the proper usage instructions. Based on the prepared rubric, the respondents rated the acceptability based on their usage experience. Acceptability results showed that nine out of 10 respondents rated the software a score of 4, which means the usage instructions given are easy to understand and in-program texts and guides are easy to read and follow; while one rated 3, which means the usage instructions given are easy to understand and in-program texts and guides are too small to read yet still followable. In average, the respondents rated the software a score of 3.9, which means the usage instructions given are easy to understand and in-program texts

and guides are easy to read and follow. The program was run by the respondents with no glitches encountered throughout the testing proper.

As per the results stated, this antivirus software can now be used by computer users to disinfect their flash drives from viruses. Thus, this can now help students, teachers, printing shop owners, and other computer users in keeping their computers and flash drives away from virus infections.

RECOMMENDATIONS

This antivirus software was coded using Batch and basic C++ programming languages only. This runs on Windows XP, 7, and 8 operating systems in a Command Prompt window. As of latest beta testing, the program can run smoothly with no glitches and performs deletion of virus and recovery of infected files. However, researchers can improve and enhance this study in their future studies by modifying and improving:

1.   software graphical user interface;

2.   software program codes for virus removal;

3.   software program codes for file recovery;

4.   software virus removal and file recovery processes; and

5.   software compatibility on other operating systems (e.g. Mac, Linux, Firefox OS)

REFERENCES

Kaspersky Lab. (2014). *Kaspersky Security Bulletin 2014*. Accessed 24 June 2015.

Retrieved from https://www.securelist.com/files/2014/12/Kaspersky-Security-

Bulletin-2014-EN.pdf.

Microsoft Corporation. (2013). *Microsoft Security Intelligence Repor*t (Vol.15). Accessed

24 June 2015. Retrieved from

http://download.microsoft.com/download/7/1/A/71ABB4EC-E255-4DAF-9496-

A46D67D875CD/Microsoft_Security_Intelligence_Report_Volume_18_Worldwi

de_Threat_ Assessment_ English.pdf.

TekCarta – GeneratorResearch Ltd. (2015). *Personal Computers per Household (68*

countries) [Data Table]. Accessed 24 June 2015. Retrieved from

http://www.generatorresearch.com/tekcarta/databank/personal-computers-per-

household.

ACKNOWLEDGEMENT

We, the researchers would like to express our deepest gratitude to all people who helped and aided us in making this study, anonymous or known.

First, to God who guided us and gave us the will, determination, and knowledge to finish this project.

Second, to our Research teacher, Mr. Joald G. Calpo, for all the guidance and teaching he gave us in order to complete this project.

Third, to our parents, who gave us moral support to finish our project.

Fourth, to our beloved classmates, especially, Marcus Alvarez, Jeannine Natividad, Andrew Nodado, Kate Signo, Lester Tebangin, and Andrea Valenzuela, who helped improve some parts of our project, and provided us some of the materials we need in accomplishing this.

Lastly, to the computer shop owners and test respondents, who let us use their computers and time for beta-testing for this project.