



**ASHESI UNIVERSITY COLLEGE**

**AFFORDABLE SOLUTIONS FOR COMPUTERS IN SCHOOLS:  
OPERATING SYSTEM AND TOOLS ON A USB FLASH DRIVE**

By

**EBENEZER AGOE-ASHIAKWEI**

Applied Project Report submitted to the Department of Computer Science,

Ashesi University College

In partial fulfillment of Bachelor of Science degree in Management  
Information Systems

**APRIL, 2014**

## Declaration

I hereby declare that this Applied Project Report is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature: .....

Candidate's Name: .....

Date: .....

The second half of the page should contain the following statement:

*I hereby declare that the preparation and presentation of the Applied Project Report were supervised in accordance with the guidelines on supervision of applied projects laid down by Ashesi University College.*

Supervisor's Signature: .....

Supervisor's Name: .....

Date: .....

## **Acknowledgements**

I would first of all like to thank Ashesi University College for the opportunities and experience provided to me during my time as a student. I would like to offer my utmost gratitude to Mr. Kwadwo Osafo-Maafo. Without his supervision and assistance, this dissertation would not have been successful.

I am also grateful for the knowledge imparted unto me by the entire Computer Science Department most especially by Mr. Aelaf Dafla for his help in clarifying my doubts when I was getting started with the applied project, Dr. Nathan Amanquah for the quality training provided and Dr. Ayorkor Korsah for her immense contribution to my programming knowledge.

I would also like to thank my colleague, Nii Lous Nelson for his assistance during the entire span of this project especially during data collection. My heartfelt thanks to my family and close friends for their support, encouragement and confidence in me. I would finally like to thank God for all the opportunities provided for me, for being my strength when things got tough and for helping make this project a success.

## **Abstract**

Over the past few years, there have been numerous developments of more efficient and portable ways of using computers. Hardware and storage have been designed to provide progressively smaller systems for computing while speed and data performance has greatly increased.

The purpose of this project is to provide a complete Operating System and tools on a USB flash drive. This is done with the aim of providing users with a much more portable, durable and affordable outlook to computing. Users can also share or reuse the same hardware while keeping their operating systems and data separate.

This project is necessary as other options require people to have and carry their own hardware, or rely strictly on the operating systems preinstalled on existing machines. Customers have preference for portable and cost-effective devices, while administrators want to simplify asset management. Overall, this project offers a new outlook on using IT hardware while bringing choice and convenience to end users.

## Table of Contents

Declaration.....	ii
Acknowledgements.....	iii
Abstract.....	iv
Table of Figures .....	vi
1.0 Introduction & Background .....	1
1.1 Motivation.....	2
1.2 Objectives.....	3
2.0 Background.....	5
2.1 Research Questions .....	5
2.2 Literature Review .....	6
3.0 Requirements.....	9
3.1 Functional Requirements.....	9
3.2 Non-Functional Requirements .....	9
3.3 Comparisons.....	10
3.4 Available Modes of data collection.....	10
3.5 Software Program Preferences .....	11
4.0 Approach.....	12
5.0 Implementation.....	14
5.1 LiveCD Installation.....	14
5.2 Persistence and Ubiquity .....	15
5.3 Customizing Ubuntu .....	16
5.4 Deployment and User Support.....	19
6.0 Evaluation, Testing and Results .....	21
6.1 Test Cases.....	21
7.0 Conclusions, Further work and Recommendations .....	25
Bibliography.....	27
Appendix.....	29

## Table of Figures

Figure 1: Ubiquity Screen .....	16
Figure 2: Ubuntu UI .....	17
Figure 3: Lubuntu theme .....	17
Figure 4: Customized UI .....	18
Figure 5: Customized Startup Screen .....	19

## **1.0 Introduction & Background**

Ashesi University College requires as part of its learning goals that students are technologically competent and as such, making use of computers is part of the day to day activities of its students. Currently, a large population of the student community have their own laptops with the smaller percentage being forced to make use of Student computers. Ashesi currently provides 12 desktop computers in the Library and 43 more in the computer labs for student use. These computers are supposed to help students who do not carry their computers around or those without personal computers or laptops to be able to get their work done.

The current setup of these desktop computers require students to log on using their Ashesi provided usernames and passwords. Students are also provided with a mere 200 megabytes of storage so they can download and save documents. Installation of important software requires administrator permission and as such all software installations need to be authorized by the IT department. This storage drive is also on the network and is therefore subject to failure when the network is down. This project aims at providing an alternate to this system using Ubuntu 12.0.4 and its LiveCD functionalities.

This Affordable Solutions for Computers project seeks to provide students with an alternate to the current computing system being used in Ashesi University College and other similar institutions. USB flash disks currently take up a large portion of the storage device market and can be considered

as the primary way by which students move documents around in Ashesi. They are also smaller, cheaper and contain no moving parts making it less fragile compared to traditional hard drives [1]. They also consume less power and can more importantly run a different operating system without making changes to a computer's hard drive.

### 1.1 Motivation

By providing an Operating System and tools on a USB flash drive, a system whereby one does not need to install the OS in order to use, is created. This way, students and lecturers will be able to work directly off the flash drive. They would also be able to carry around their work, programs and other important files in their pockets and continue work where ever they find themselves with a computer. Due to its low-cost functionality, students can always have a platform to go with minimal performance issues since it does not require the latest hardware to run. As such, there is no need for students and teachers to carry laptops with them as they just need to go to any computer and boot up. Support will also be easier since everyone would have the same system and students will be able to install software without any administrative assistance. Pen-drives, being solid state devices, are also much sturdier than hard-drives for people on the go [1].

Aside Ashesi University, the One Laptop per child (OLPC) initiative believes that every student should be able to own or use a computer in order to encourage learning [2]. The standard approach used is to build labs, purchase computers and hire/purchase programs and software in order to



provide students with computers to use. This project is meant to help provide alternate access to computing in order to encourage learning and practice, as well as access to necessary resources for learning.

## 1.2 Objectives

This section outlines the basic goals of the project in providing an alternate to affordable computing. These objectives were identified by taking into consideration the overall functionality of the USB Flash drive. These objectives include booting the OS from the Pen drive, enabling file storage and modification, providing users with full control of the OS, ease of use through UI modifications and providing user support and deployment tools.

### Bootting Operating System from Pen Drive on multiple platforms

Installing the operating system unto the flash drive registers only the drivers of the current computer and as such does not allow multiple computer bootting. As an option, the live CD feature was made to provide a preview of the operating system without installing. It does this by loading the necessary drivers for multiple hardware so it works on almost every machine. The operating system needs to be able to boot from the flash drive on any given computer as such it must be first installed to work on the USB flash drive.

### File Storage and Modification

In order to ensure full OS functionality, the chosen OS must provide the option to store and review previously saved files and documents. Storage is very important since students will need to be able to save their files across reboots and still be able to access them.

### Provide User with full control of the OS

By providing Users with full control of the OS, they will be able to install any software that they want and download files to any directory that they choose. This privilege can be managed by an IT manager if necessary and as such the IT manager will be able to provide the User with the password for the administrator of the OS thereby ensuring security.

### Making OS easy to use through UI Modifications

In order for users to get accustomed to the operating system, it has to be customized to give it a user friendly feel. Tools and Software that are needed by the user must also be provided so as to provide users with their expected functionality.

### Providing User Support and Deployment Tools

The importance of user support and tools objective is to explore the tools that would ensure that the implementation of this project can be built quickly and efficiently. There is also the need to provide support such as updating packages & issues and solutions pages within the operating system to help users update, get support and ask questions.

## **2.0 Background**

In order to ensure portability, there is the need to provide an operating system that can boot directly off a USB flash drive. This option must be able to load up the necessary drivers that ensure boot up with multiple hardware. The OS must also be customizable in order to ensure usability while providing support and quick build up tools that will help users to use the product efficiently.

### **2.1 Research Questions**

In order to implement the project, research has to be conducted on the feasibility and functionality of both the Linux operating system and the USB flash drive. Research must also be put into the customization and implementation of the user support tools. The following Research questions give us an understanding of what to look for in the research.

- I. How does one customize the Operating System to make it usable for most students? Majority of students in Ashesi are used to the Microsoft Windows OS functionality. Most of them have Windows on their laptops and the Ashesi student computers all have windows installed as its default operating system. A smaller number of students are more used to the Mac OS X operating system while very few students are familiar with the Ubuntu OS experience. That being the case, there is the need to be able to customize the operating system so users are comfortable with what they use.

- II. What tools and software are most suitable for students given the selected operating system? In order for students to find the project useful, there is the need to provide them with tools and software that they find useful in their studies. A step has to be made to provide these tools if not already available.
- III. How does one provide quick setup and configuration tools for students? Once the tools and software needed to make the OS efficient are discovered, there is the need to explore ways by which these tools can be easily setup on the operating system. Tools that help with configuration are important since they would help the user make most of the operating system.
- IV. How to build automation tools to help make creating and using the OS on the USB flash drives.

## 2.2 Literature Review

Although this project is unique to the Ghanaian education system, it is however, not the first of its kind. In researching on this project, two projects which were similar were discovered, i.e. Windows To Go and StormFly.

Windows To Go is a feature in Windows 8 Enterprise that allows it to boot and run from mass storage devices such as USB flash drives and external hard drives. This feature can be booted on a PC running on any operating system as long as it meets the Windows 7 or later certification requirements

[3]. One major difference between Windows To Go and this project is that this project focuses on providing a more affordable way to help students in schools while Windows To Go focuses on enterprises i.e. businesses looking to provide users with an imaged version of Windows 8 that reflects the corporate desktop [3]. When Windows To Go is initially booted on a host computer, it detects the host hardware and installs any needed drivers. As it is subsequently booted on that computer, it identifies the host and automatically loads the correct drivers. Although Windows To Go has made a name for itself, it has its setbacks, i.e.

- The drivers in Windows To Go are not mature so in many cases, Windows 7 drivers are used instead. This sometimes poses a problem.
- Windows To Go cannot be booted through a USB hub, it must be connected directly to the PC for it to run [4].

Introduced by Now Computing, Stormfly has been described as an operating system on your wrist. Working with a Linux operating system, it is contained on a USB 3.0 wristband with a size of 16GB. It has a couple of features i.e.

- A backup & replacement system which automatically backs up your files if the user subscribes to the service. If it is stolen, it is replaced with all the data it had as at the user's last backup [5].
- StormFly is built for speed i.e. apart from designing it with the latest USB standard (USB 3.0), it has also been tweaked to give it some properties of SSD (Solid State Drive) [5].

- A shared folder feature which provides users the opportunity of sharing files either on the operating system of StormFly or as a folder in any other operating System that the user plugs into [6].

Similar to this project, StormFly provides an affordable solution for schools by supplying portable PCs to students at a reasonable cost.

Apart from the aforementioned projects, there are other technologies that have been implemented that also seek to provide affordability in the world of technology. Examples of these are:

- N-Computing: this is a desktop virtualization company that makes hardware and software to create virtual desktops which enable multiple users to share a single operating system instance concurrently [7]. It supports both the windows operating system and the Linux operating system through a version of vSpace Server (software that creates the virtual desktops in the shared PC by dividing the computer's resources into independent sessions that give each user their own PC experience) [8].
- Raspberry Pi: this is a single-board computer developed with the intention of promoting the teaching of basic computer science in schools. It supports quite a number of different operating systems e.g. Linux, Firefox OS, Haiku, WebOS, etc. Raspberry Pi is increasingly becoming a tool identified as an educational computer for students to learn to program, additionally, it is popularly known for its small size and low cost [9].

### **3.0 Requirements**

This chapter underlines the requirements of the project and their classifications. The requirements are broken down into Functional and Non-Functional Requirements.

#### **3.1 Functional Requirements**

This has to do with requirements that are important to the functionality of the project. In other words, these requirements affect the way the system works. The following are the identified functional requirements of this project.

- Need to be able to run an operating system from the pen-drive
- Should enable file modification and storage
- Able to install and run software
- Should be updatable

#### **3.2 Non-Functional Requirements**

The Non-Functional requirements of the project are features that do not directly affect the way the project functions but are necessary for the implementation to meet the specified objectives. The following are the identified Non-Functional requirements of the project.

- Low Cost
- Portable
- Easy to maintain and deploy
- Easy to use
- Customizable for different needs

### 3.3 Comparisons

This section involves analyzing the project in terms of:

Operating System: Comparing the chosen OS suites to that of the standard. For example, using *Netbeans* on Ubuntu vs. using *Netbeans* on Windows and so forth.

Using Pen-drives: Booting the operating system from a pen-drive as compared to booting an already installed version.

Using Smaller computers: Running project on a small or low end computer as compared to running on a faster high end computer.

Cost: In order to have a better idea of the affordability, an analysis of the costs involved compared to that of the standard operating system.

Installation: Comparing the ease of installing the Ubuntu operating system on the USB flash drive as compared to installing directly unto a computer.

Usability: How easy it is to use the operating system and its tools as compared to that of the standard.

Completion of tasks: The efficiency of the operating system in executing and completing tasks.

### 3.4 Available Modes of data collection

1) Case Study approach: This involves interviewing participants after they have made use of specific items or functionalities pertaining to the project.



2) Metrics Measurement approach: This involves recording time taken by participants in completing tasks and checking how they completed the work.

### 3.5 Software Program Preferences

In order to have a better understanding of the type of software used by students and teachers in Ashesi, a research was initiated to collect data from students. A sample of the questionnaire can be found in Appendix F.

In order to acquire data using these questionnaires, a sample of 5 students from each major are to be interviewed from the sophomore, junior and senior year groups, making a total of 45 students. The findings and results can be found in Appendix G.

## 4.0 Approach

This chapter provides an overview of how the project would be accomplished. In order to obtain an idea on how the project is to be implemented, a questionnaire has to be created. This questionnaire has to provide information of the software tools that has to be provided in the project as well as the type of operating systems users are comfortable with.

With the help of the information provided by the questionnaire, an operating system has to be chosen to be deployed on the pen drive. In order for this to be done, there is the need to find out if the OS can boot directly from the USB flash drive and which variation of the OS would be suitable for the project. It is also important to figure out if the OS needs swap space in order to effectively install on the pen drive.

Due to the importance of customization in the project, the operating system would have to support the option for customization and modification. One proposed customization implementation would be to customize the *Menu* bar in order to make it easy for users to find software installed. The following are two types of approaches to this:

**Task Approach:** This approach will involve categorizing the *Menu* bar of the chosen operating system according to tasks. As a result all programs used for the task of programming will be grouped in this category.

**Software Category Approach:** This would involve categorizing programs according to type or provider of software. For example, all Microsoft Office tools will be grouped in the same category.

In analyzing the USB flash drive, it is important to find out what size or range of storage space would be suitable for the deployment of the operating system. Portability is a key issue in implementing this project and as such, the operating system must be able to boot up on multiple machines. In making sure of this, an idea of what machines will work or boot up the OS is important.

An important approach to implementing the project would be to categorize the workloads for different users, for example, in Ashesi, categorization could be done by choice of major and may include Business Administration, Management Information Systems and Computer Science. Once this is done, focus can be put on the courses within these categories and software can be chosen as such. With the Computer Science category, the Programming class would definitely need a compiler, an editor and a debugger. The Database Systems class would need an editor, a diagram tool and a running database. The Design class would need a diagram drawing tool and a word processor to write reports. Using this information, software can be picked especially those with Linux version already available.

After the implementation of the identified objectives, a test must be conducted to ensure that the project works as expected.

Cost: In order to have a general idea of the cost of implementing such a project, a breakdown of the cost of the pen drive and the Ubuntu LiveCD must be provided.

## 5.0 Implementation

This chapter provides an overview of how the project was executed. In order to commence this affordable solution to computing project, I had to find a way to install an operating system unto the pen drive. I first tried to explore the option of using Windows as the project OS, however the option to install the feature known as *Windows To Go* could only be implemented in a registered enterprise edition of the OS. After much deliberation I decided to use the Ubuntu 12.0.4 Operating System. This is because unlike Windows and Mac OS X, Ubuntu has a portable OS feature known as LiveCD, which allows the user to try out the Ubuntu OS. In researching costs, I found out that the Ubuntu LiveCD can be downloaded for free at the Ubuntu website [10], however, the cost of an 8GB Pen drive was GHC 63.31 [11].

### 5.1 LiveCD Installation

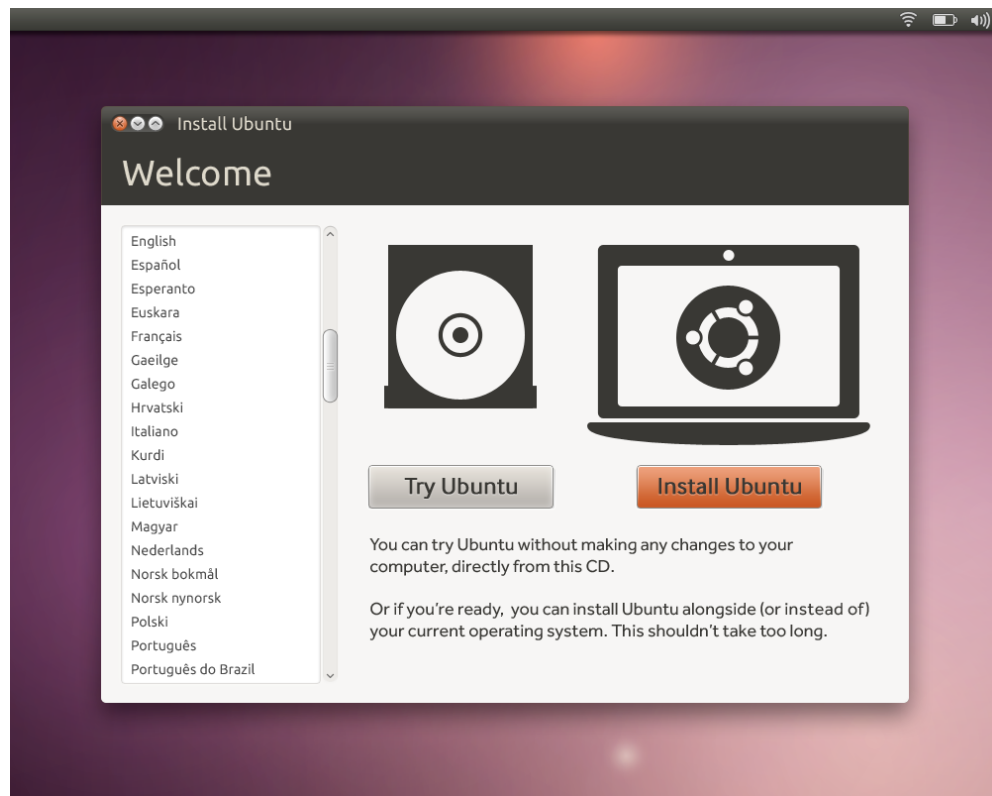
In setting up the LiveCD feature, I tried installing the Ubuntu OS on the USB flash drive and found out that this approach did not ensure that the Ubuntu would boot up on multiple machines. Apparently, installing directly unto the pen-drive registers the drivers of the current machine alone and as such does not work on others. Researching on LiveCDs, I found out the right way to set it up was to extract the contents of the LiveCD's ISO unto the USB flash drive. I decided to use a third party application called "Universal-USB-Installer" that does the extraction and makes the pen drive bootable. The application extracts the ISO and created a "liveUSB" on the Pen drive.

## 5.2 Persistence and Ubiquity

The Ubuntu LiveCD feature is designed for previewing the operating system. This being the case, any file stored on the LiveCD can be considered temporary as it will be discarded once the OS is shutdown. In order to provide the full operating system experience, a way has to be found around this ensuring that files stored on the OS are retrievable unless deleted by the user. While figuring a way to get around this issue, I tried to partition the pen drive. By doing this, I could specify how much space I wanted to use for the persistence storage and what was to be used for the installation. Making use of an 8GB USB 2.0 ScanDisk pen drive, I split the storage into two 4GB partitions. After which I tried to map the Documents, Desktop, Download, Music and Picture folders to the partitioned pen drive using shell scripts. I however abandoned this approach when I found out about Persistence. What persistence does is to make use of part of the main storage system, which in this case is the USB flash drive storage and uses it as a storage system while running LiveCD. In implementing persistence storage, I had to format the pen drive and label it "casper-rw", which is the name LiveCD looks for when it wants to make use of persistence storage. I then had to create a loopback file also named "casper-rw". The steps to create such a file can be found in Appendix A. The created file is supposed to store all data stored on the LiveCD while the OS is being used.

After ensuring that the LiveCD/USB boots up and provides the option to try Ubuntu, I went on to remove Ubiquity, which is the welcome and try/install

screen on startup by running the command "sudo apt-get install Ubiquity" in terminal.



*Figure 1: Ubiquity Screen*

By doing this I ensured that the LiveCD went straight ahead to load the Ubuntu OS on startup.

### 5.3 Customizing Ubuntu

The next step was to customize the Ubuntu operating system to look like the standard Windows User Interface. To help me do this I needed a Linux environment that looked like Windows or provided an easily customizable interface.

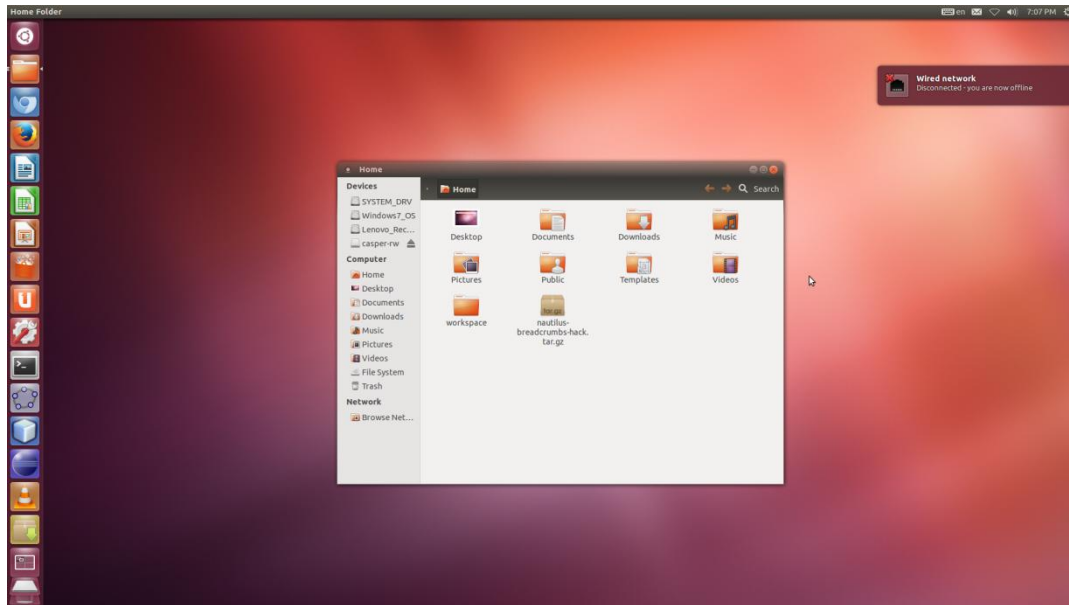


Figure 2: Ubuntu UI

I found two options, Xubuntu and Lubuntu. I decided to go with Lubuntu due to ease of modification and used terminal to install the theme.

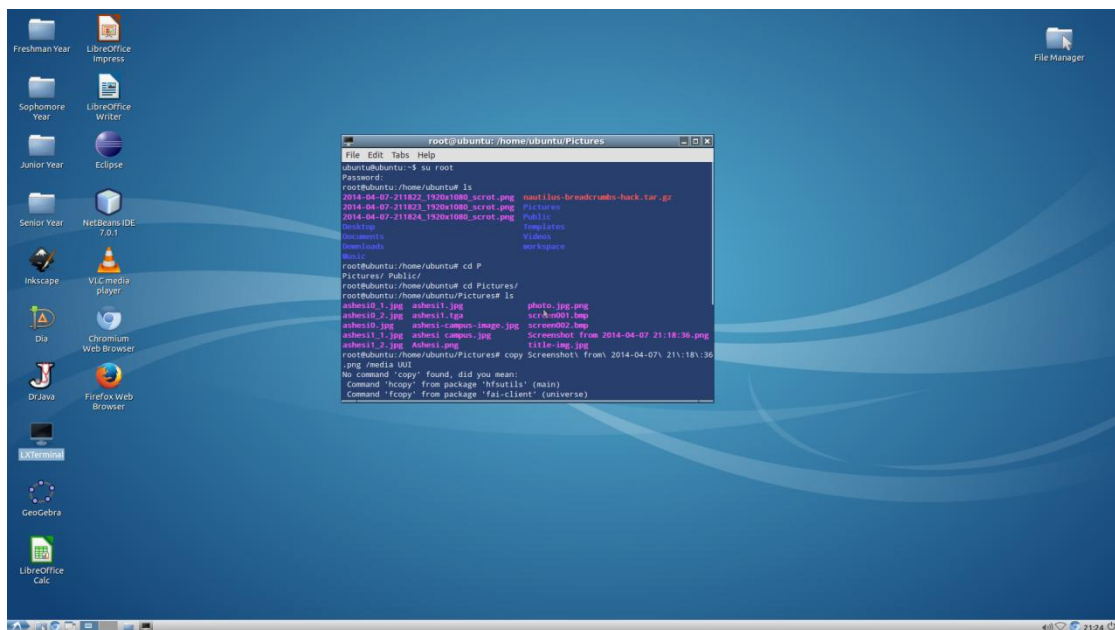
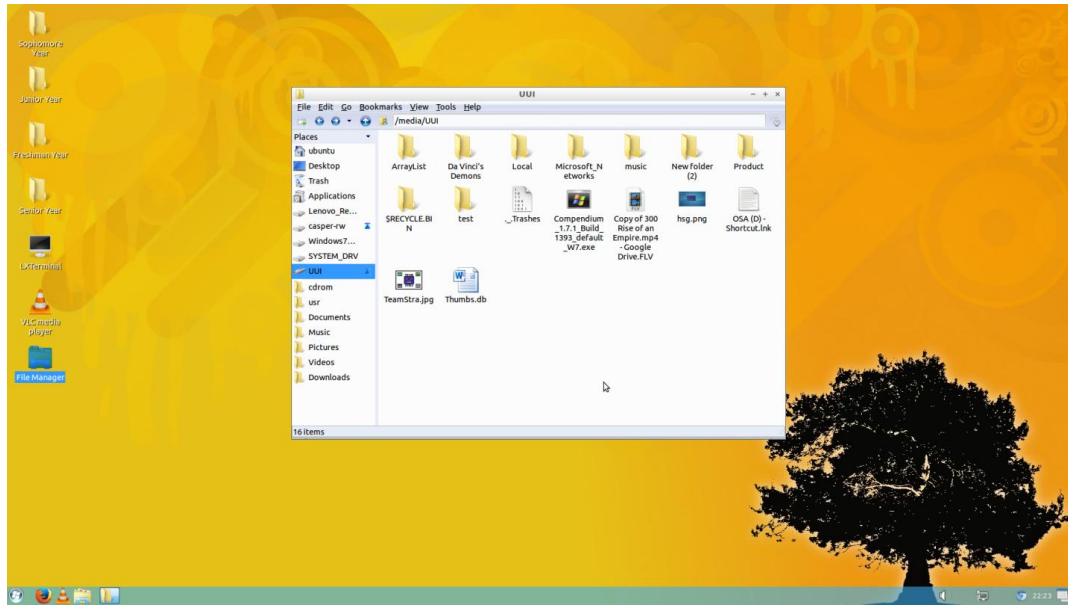


Figure 3: Lubuntu theme

The command to install the Lubuntu theme can be found in Appendix B.



*Figure 4: Customized UI*

The final step of customization was to create a custom startup screen and shutdown screen on Ubuntu. This is to be done to suit the preferences of the institution that the project is being made for. In the case of Ashesi University College, I used the Ashesi logo and a picture of the Campus as the startup screen as shown below.





*Figure 5: Customized Startup Screen*

I installed a program called Plymouth Manager which provided startup screen alternatives. Using Gimp, a design content editor, I modified one of the already existing templates to an Ashesi University standard. The commands for both Plymouth and GIMP can be found in the Appendix.

#### 5.4 Deployment and User Support

Considering the fact that the project has been setup to ensure an OS on a pen drive, an efficient storage system across reboots, and user friendly UI modifications, the next step would be to provide an efficient way of deploying software unto the system and providing support. Doing this a database had to be created to store information from the User Support and Deployment page which would be implemented in HTML and PHP. The Information in the database would include a list of Courses, Departments and Majors as well as the software necessary for deployment. This database was created using the MySQL engine. Users will be provided with a Web Page that will provide them

with support on their system. They also will have the option to choose what software they would like to deploy as well as other installation alternatives such as visiting the software website. Deployment can be done in two ways; the first will be to deploy all software under a given major. That being the case a Computer Science student will have all the software in the database linked to computer science courses. The second option will be to manually select the software that is needed to be deployed. Once the user submits the list of software, a shell script containing all the terminal commands for the installation of the software will be provided to the user. This can then be run on the OS in terminal using the command "bash filename.sh".

## 6.0 Evaluation, Testing and Results

This section documents the steps taken to test and evaluate the project as well as the outcome of these assessments. Majority of the tests were carried out in Ashesi University using 1 computer in the computer lab, 1 computer in the library, 1 Dell personal laptop and 1 MacBook pro laptop. A selected number of students were also made to test the project and give feedback. These tests were to be done by taking into consideration the objectives of the project, as defined in the Introduction section, the system functionality requirements defined in the Requirements section.

### 6.1 Test Cases

**File Storage and Modification:** As explained earlier on in this report, the Ubuntu LiveCD does not allow for storage and modification on a permanent basis. After making use of Persistence, as detailed in the Implementation section, a test was undertaken to ensure that it indeed stored and provided the user with previously saved files and software across computers. A word document was created on one of the PCs and saved. After booting up the OS, the file was still available in the Documents folder. The outcome of the test proved that the project did indeed ensure storage of files over reboots. This test was done with an 8GB pen drive with 4GB persistence, however due to the fact that there will be a number of software deployed on the pen drive, a larger pen drive size will have to be used.

**Usability and Suitability of Interface:** This test was carried out in order to find out if users were comfortable with the customized version of the project. This test was carried out on a total of 8 students, consisting of 2 students

from each year group in Ashesi. The test plan included ease of users to save files, starting up the system and logging out among others. During the test, majority of the users did not even realize that they were not using the standard Windows operating system and as such found it easy to perform majority of the tasks. All users were able to connect to the internet using the pen drive, however one user pointed out the fact that printing using the Ashesi network had not been set up. Most of the users were impressed at the boot up and shut down speed of the operating system as compared to Windows. Users were also asked to rate the usability of the project after the tests and the average rating recorded is 8.625/10.

<b>User</b>	<b>Bootng OS</b>	<b>Saving files</b>	<b>Connect to Internet</b>	<b>Average rating</b>
Test User1	Pass	Pass	Pass	10
Test User2	Pass	Pass	Pass	8.5
Test User3	Pass	Pass	Pass	8
Test User4	Pass	Pass	Pass	10
Test User5	Somewhat	Pass	Pass	7
Test User6	Pass	Pass	Somewhat	8
Test User7	Pass	Pass	Pass	9
Test User 8	Pass	Pass	Pass	8.5
Average Rating				8.625

**User Software Preferences:** After conducting the interviews with the students from each major using the questionnaire documented in the Requirements section, the software suggested was put into a list of software with a count on how many people in each year group suggested the software. A detailed documentation of this list can be found in Appendix F.

Although many of the software on the list have Ubuntu variations, a number of them cannot be represented on the project since their manufacturers have not provided Linux variations. That being the case, similar programs will be provided to the user.

## Test Results

The Table below shows the functionality tests and their results.

Test	Expected Result	Actual Result	Pass/Fail
Bootling across Windows computers	Make use of computer drivers and load OS	Made use of computer drivers and load OS	Pass
Bootling across MacBook Pro Computers	Make use of computer drivers and load OS	Did not allow OS from Pen drive to load	Fail
File Storage	Store files across Reboots	Stored files across Reboots	Pass
Usability of UI design	Average Usability rating of 7/10 or greater by Users	Usability rating of 8.625/10	Pass
Connect to the internet(Networking functionality)	View wireless and LAN connections and connect to the network	Able to view wireless and LAN connections and connect to the network	Pass
Printing	View printers connected to network be able to print	Not able to view printers on network	Fail
Software installed on Pen-drive OS vs. Personal Computer installed OS	Software installed on the OS should work as well as that installed on the OS of any hard drive computer	Using Dr. Java, Eclipse, Inkscape and Firefox browser as test cases, all software user functionalities on the pen drive were met.	Pass

## **7.0 Conclusions, Further work and Recommendations**

This Affordable Solutions for Computers project explores an alternative solution to portable computing by making use of a USB flash drive. Having completed the implementation, the project is able to boot up on multiple machines. It also enables data storage, can connect to the network and make use of all features of an operating system installed directly onto the hardware. This project was completed using the Ubuntu gnome Linux environment.

I however think it would be interesting to explore this concept on other operating systems such as Windows and OS X. In such a case it would be interesting to compare its functionality and efficiency against that of this project. Since the Windows OS is the standard OS on most machines, it would be interesting to explore Windows' portable USB option known as the Windows To Go and how it can be made to work in more educational situations. One limitation of this project is that it does not boot up on Apple MacBooks. This could be an area of research as to how to get the MacBook to boot from the USB device.

In order for a project along the lines of this one to be successful, an in-depth understanding of operating systems and its concepts is important. As a result of the Ubuntu OS used for the project, a lot of work had to be done in terminal. It is therefore important to be familiar with the basic UNIX commands.

Instructions on how to setup such a system have been documented along with the commands that can be used in the Appendices below.



## Bibliography

- [1] RedOrbit, "USB Flash Drive," RedOrbit, [Online]. Available: [http://www.redorbit.com/education/reference\\_library/technology\\_1/computer-tech/2583534/usb\\_flash\\_drive/](http://www.redorbit.com/education/reference_library/technology_1/computer-tech/2583534/usb_flash_drive/). [Accessed 3 March 2014].
- [2] One Laptop per Child, "About the project | mission," One Laptop per Child, [Online]. Available: <http://one.laptop.org/about/mission>. [Accessed 15 March 2014].
- [3] Microsoft, "Windows To Go: Feature Overview," Windows, [Online]. Available: <http://technet.microsoft.com/en-us/library/hh831833.aspx>. [Accessed 25 February 2014].
- [4] S. O'Neill, "Windows To Go: The Do's and Don'ts," CIO, 13 June 2012. [Online]. Available: [http://www.cio.com/article/708391/Windows\\_To\\_Go\\_The\\_Do\\_s\\_and\\_Don\\_ts?page=2&taxonomyId=3081](http://www.cio.com/article/708391/Windows_To_Go_The_Do_s_and_Don_ts?page=2&taxonomyId=3081). [Accessed 2 April 2014].
- [5] Now Computing, "Product," Stormfly, [Online]. Available: <http://stormfly.nowcomputing.com/product.html>. [Accessed 20 March 2014].
- [6] P. Ridden, "StormFly puts an operating system on your wrist," Gizmag, 4 February 2013. [Online]. Available: <http://www.gizmag.com/now-computing-stormfly-wristband-operating-system/26093/>. [Accessed 20 March 2014].
- [7] NComputing, "NComputing Products," NComputing, [Online]. Available: <http://www.ncomputing.com/products/overview>. [Accessed 5 April 2014].
- [8] NComputing, "vSpace Server," NComputing, [Online]. Available: <http://www.ncomputing.com/products/vspaceserver>. [Accessed 5 April 2014].
- [9] S. Watkiss, "Raspberry Pi - Linux computer for learning programming," Penguin Tutor, [Online]. Available: <http://www.penguintutor.com/linux/raspberrypi>. [Accessed 5 April 2014].

- [10] Ubuntu, "Download Ubuntu Desktop," [Online]. Available: <http://www.ubuntu.com/download/desktop>. [Accessed 27 January 2014].
- [11] Ghana Electronics Company, "Sandisk 8GB Cruzer USB Flash Drive, Black," [Online]. Available: <http://www.ghanaelectronics.com/sandisk-8gb-cruzer-usb-flash-drive-black/>. [Accessed 2 April 2014].
- [12] S. Technologies, "Durable Mobile Computing: Driving Real-Time Value," *Symbol*, pp. 1-6, 2003.
- [13] Honeywell International Inc., "The Importance of Mobile Computer Durability and Ergonomics—Why Rugged Devices Cost Less," *Honeywell*, pp. 1-4, 2008.
- [14] Worple, Doug, "StormFly: Like a PC On Your Wrist," Kickstarter, [Online]. Available: <https://www.kickstarter.com/projects/750308586/stormfly-like-a-pc-on-your-wrist>. [Accessed 20 March 2014].

## Appendix

### A. Creating Persistence Storage

1. Plug Pen drive into Computer
2. Run the following command in Terminal to view all drives connected to the PC

```
df -h
```

3. Take note of the File System of the pen drive eg. sda1
4. Unmount the drive so it can be formatted using the command:

```
sudo umount /dev/sda1
```

5. Run the following command to format the drive:

```
sudo mkfs.ext3 -b 4096 -L casper-rw /dev/sda1
```

6. Once the output is done running, run the following command to create the persistence loopback file. To change the size of the file, modify the count parameter.

```
dd if=/dev/zero of=/media/hda1/casper-rw bs=1M count=1024
```

7. Run the following command once file has been created:

```
mkfs.ext3 /media/hda1/casper-rw
```

8. Install LiveCD on the Pen drive using Universal-USB-Installer
9. Boot up LiveCD with persistence.

## **B. Installing Lubuntu theme**

1. Open Terminal (Ctrl +Alt + T).
2. Run the following command to install Lubuntu LXDE theme

```
sudo apt-get install lubuntu-desktop
```

3. Log out after install is done and select Lubuntu LXDE session from log on-screen.

## **C. Installing Plymouth Manager**

1. Open Terminal (Ctrl +Alt + T).
2. Run the following command to install Plymouth manager's repository

```
sudo add-apt-repository ppa:mefrio-g/plymouthmanager
```

3. Run an update on the system:

```
sudo apt-get update
```

4. Install Plymouth Manager:

```
sudo apt-get install plymouth-manager
```

5. Install themes:

```
sudo apt-get install plymouth-theme
```

6. Run the following to pick or change theme:

```
sudo update-alternatives --config default.plymouth
```

7. Run to apply changes:

```
sudo update-initramfs -u
```

## D. Installing GIMP

1. Open Terminal (Ctrl +Alt + T).

2. Run the following command to install software repository

```
sudo add-apt-repository ppa:otto-kesselgulasch/gimp
```

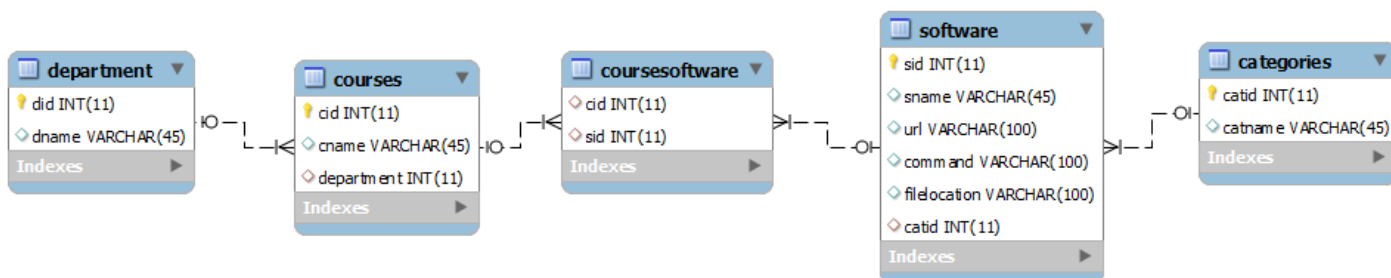
3. Run an update on the system:

```
sudo apt-get update
```

4. Install GIMP:

```
$ sudo apt-get install gimp
```

## E. EER Diagram for Software deployment database



## **F. Student Questionnaire**

1. Student ID:
2. Student Major
3. Year of Graduation
4. What Courses did you take in Ashesi last year?
5. What tasks associated with each course did you require the use of a computer software?
6. What software did you use in accomplishing this task?

## **G. List of Software suggested by students**

After conducting interviews on 45 students (15 from each major), the following software was suggested to be deployed on the project. The list of software has been categorized by year groups and has a count of student suggestions. The software with the highest number of suggestions has a higher priority during deployment.

Software Preferences- Freshman year.

Software	Total Count
MS Word	15
SPSS	15
Visual Studios	15
MS PowerPoint	12
MS Excel	7
Adobe Reader	4
Prezzi	3
Calculator	3
Offline Dictionary	1
Microsoft Math	1
Math Input Reader	1
R	1
DIA	1
MS Access	1

Software Preferences- Sophomore year.

Software	Count
MS Word	15
MS Excel	12
MS PowerPoint	12
Adobe Reader	12
Chrome	12
VLC	12
Dr. Java	8
Firefox	6
Eclipse	3
Snip tool	2
SPSS	1
Photo Viewer	1
Foxit Reader	1
Sorting Simulator	1
Corel Draw	1
R	1
Blender	1

Software Preferences- Junior year.

Software	Total Count
MS Word	13
MySQL Work Bench	10
MS Excel	9
xampp	8
Netbeans	8
Notepad+ +	8
MS PowerPoint	7
Inkscape	6
Balsamiq	6
Indigo Studio	6
Dropbox	6
Dev Cpp	5
Logism	5
phpMyAdmin	5
Eclipse	5
Processing	4
Mars	4
Chrome	4
Photoshop	3
Illustrator	3
Compendium	2
VLC	2
roomeon 3D	1
MS Project	1
Geogebra	1
Codeblocks	1
MS picture manager	1
MS Outlook	1
Prezzi	1



Software Preferences- Senior year.

Software	Total Count
MS Word	12
Netbeans	9
xampp	9
MS PowerPoint	9
MS Excel	9
Dr. Java	8
mysql workbench	7
notepad++	7
Indigo Studios	6
DevCPP	5
myPhpAdmin	5
Chrome	5
Argo UMLs	4
Adobe Reader	4
Virtual Box	4
DIA	3
dreamweaver	3
pencil	3
Compendium	3
MS Outlook	3
JavaME Mobile Emulator	3
VMWare	3
illustrator	2
komposer	2
JustInMind	2
android ADT	2
Git	2
SPSS	2
LucidChart	1
caret	1
Ripple (chrome store)	1
Live Mail	1
Aptana	1
phpStorms	1
sqlite database browser	1
FileZilla	1
VLC	1
R	1