Virtual Machines in the Classroom

by

Jordan Nehring and Andrew Tamanko

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

Jordan Nehring — Date 4/17/17

Andrew Tamanko — Date 04 / 17 / 2017

Brian Verkamp, Faculty Advisor — Date 4/17/17

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

April 2017

Table of Contents

Table of Contents (cont.)

Tables

Figures

Abstract

School budgets tend to be tight, yet every year schools waste massive amounts of money trying to keep their computers up to date. With the fast pace of technology, it can be expensive to maintain. Virtual Machines in the classroom allows students to have the best user experience possible without having to swap equipment as frequently. Instead of schools channeling most their budget into purchasing brand new desktops and laptops that will see only a few years use, we can direct that budget into maintaining and up keep on servers at schools themselves. Instead of purchasing expensive computers, which students have direct access to and risk being damaged or stolen, students will only have contact with inexpensive thin clients, which are easily replaceable and do not have massive price tags attached to them. This makes troubleshooting easier and more sustainable for schools with tight budgets.

1. Problem Statement

1.1 Introduction

Schools spend a substantial amount of money upgrading their computers, they take equipment that may not be more than a few years old and replace it. In 2009, schools K-12 spent $385 Million on IT. For comparison, in 2015 schools K-12 spent $4.7 Billion on IT.[2] This is a massive increase in IT spending for schools in six years. Refocusing where the money gets placed on upgrading systems today is more simple and more efficient. There are many options available that can be used to cut down on costs without losing functionality. Instead of spending money on new equipment that the students have direct access to and can be damaged easily. Money can be used on upgrading server equipment and can instead be made available to students as they need it. Students are not always using computers in schools, but they are used a lot. Instead of having a ton of expensive computers we can use thin clients that will connect remotely to VMs and will be used as needed.

1.2 Project Description

Develop a way to successfully set up a thin client in a classroom environment using Hyper-V and other tools. Create a detailed plan as to how schools, or specifically Cincinnati Public schools will be able to set up an environment where they can use thin clients to run off a server running Virtual Machines. Figure out how much money can be saved by using thin clients as opposed to running laptops and desktops, including long term with PC refreshes. Figure out exactly how much money goes to “waste”, because the current system is set up to buy new products for the

students directly instead of upgrading server equipment. We will attempt to automate as much of this process as possible using scripts in PowerShell and Windows Deployment Services.

1.3 Problem

Schools have extremely tight budgets, they spend a lot of money upgrading PCs as they become outdated. When a computer breaks or there is an issue with the device, it is swapped with a brand-new system. The school will reimage the damaged computer and send it back out, but if it is an issue with hardware it will just be thrown out.

1.4 Solution

The solution to fix schools spending too much money on PCs is investing in the right areas, where the cost is initially more, in the end, it will save a lot of money by removing the need to spend thousands of dollars just to replace a few laptops or desktops. By changing where the money is funneled, we can direct it towards a better server infrastructure and allow them to buy inexpensive devices for the students to use and reduces the risk of students destroying expensive equipment and it also provides a more efficient way to deploy and machines for each student.

1.5 User Profile

1.5.1 Application Title

Student VM Deployment and Administration Tool

1.5.2 Potential Users

School Faculty

Students

1.5.3 Software and Interface Experience

Consists of a login screen. After logging in successfully, students will be connected to their own Virtual Machine. Each student will have their own personal virtual machine they will be able to use.

1.5.4 Experience with Similar Applications

They will be able to use the UI like it was a normal PC used previously to the VMs.

1.5.5 Task Experience

A user is required to have minimal knowledge on using a computer and the IT staff will need experience with certain IT applications. Hyper-V, using a simple program to create VMs with excel spreadsheets, as well as basic networking.

1.5.6 Frequency of Use

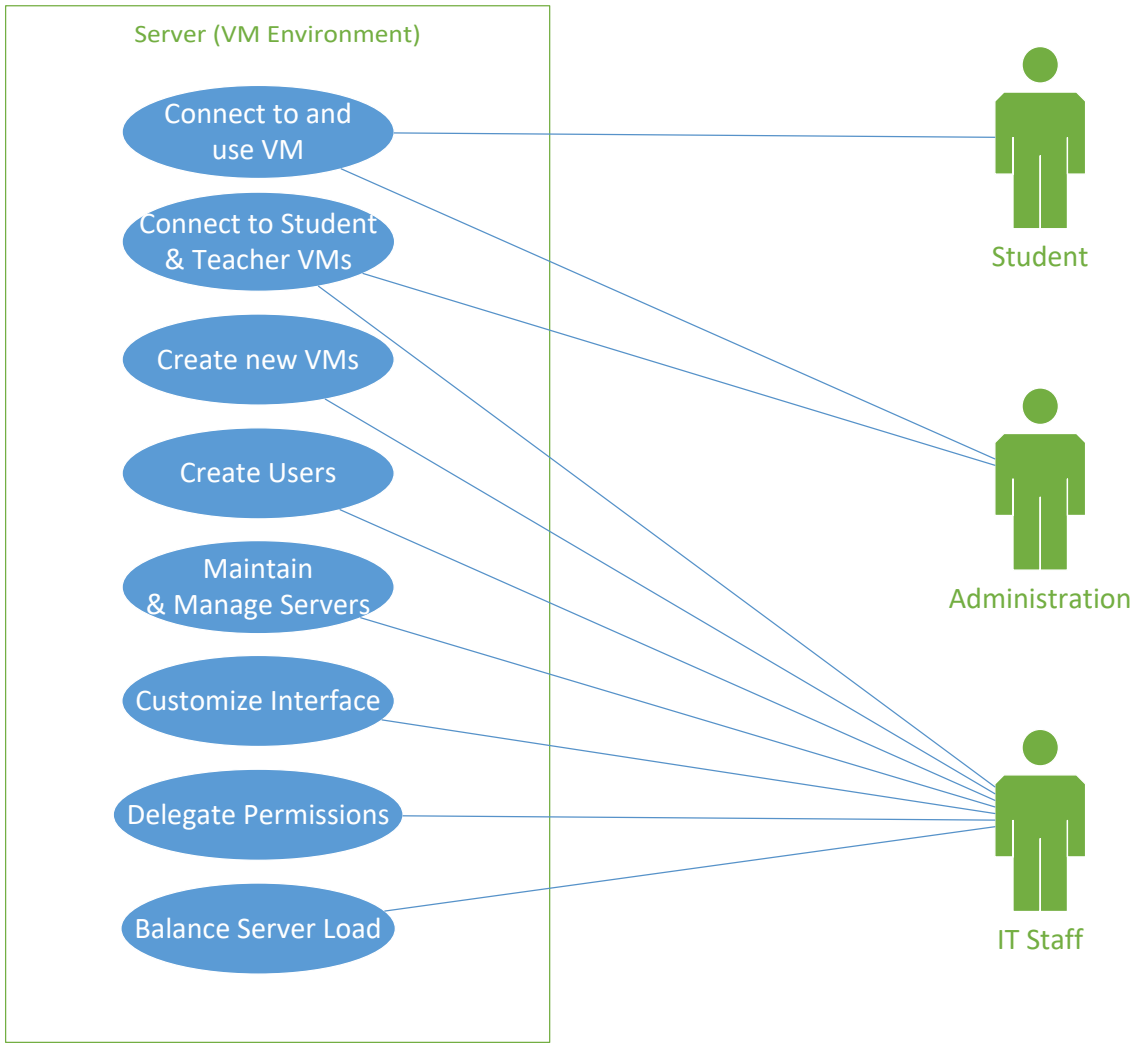
Students will access VMs daily as needed to complete assignments. IT staff will need to access student machines when a student requires assistance with an issue they may be having, or when the staff is creating VMs for the students and making accounts for the students as well.

1.5.7 Key Interface Design Requirements That the Profile Suggests

Separate interfaces are used for IT administrators and users. There will be a simple design so it will prevent distraction and reduce complication of the experience for the end user.

Figure 1: Use Case Diagram

This diagram showcases which each user would have access between the servers and VMs.



2. Project Management

2.1 Budget

Table 1 presents the project budget. With a real-world cost of $116,545, but totals to $0 since this is our senior project, which will not be sold.

Table 1 – Project Budget

No.	Item	Unit Each	Price ($)	Item Total
	Networking			
1	Labor	100	$80.00	$8,000.00
	Software			
2	Labor	150	$80.00	$15,000.00
3	Server 2012 R2	5	$1,209.00	$6,045.00
	Supplies			
4	Servers (Hyper-V)	5	$14,000.00	$70,000.00
5	Server(WDS)	1	$2,500.00	$2,500.00
6	Thin Clients	300	$50.00	$15,000.00
7	Misc. Supplies	-	-	-
			Subtotal	($116,545)
			Total EQM	$0

2.2 Objective/ Deliverables

Table 2 presents the deliverables deadlines for the Fall

Table 2 – Fall Project Objectives/Deliverables Due Dates

Fall Major Project Milestones (Deliverables)			
Pre Planning Milestone	9/20/16	Software Milestone	11/01/16
Pre Setup Milestone	9/27/16	User Interface Milestone	11/20/16
Network Setup Milestone	10/25/16	Presentation Milestone	11/28/16

Table 3 presents the deliverables deadlines for the Spring

Table 3 – Spring Project Objectives/Deliverables Due Dates

Spring Major Project Milestones (Deliverables)			
Sandbox Milestone	1/30/17	Quality Assurance Milestone	2/27/17
Improvements Milestone	2/13/17	VM Test Milestone	3/06/17
Raspberry Pi Milestone	2/20/17	IT Expo Milestone	4/11/17

2.3 Project Schedule

Figure 2 is our schedule with the major milestones listed for the Fall Semester.

Figure 2 – Fall Schedule

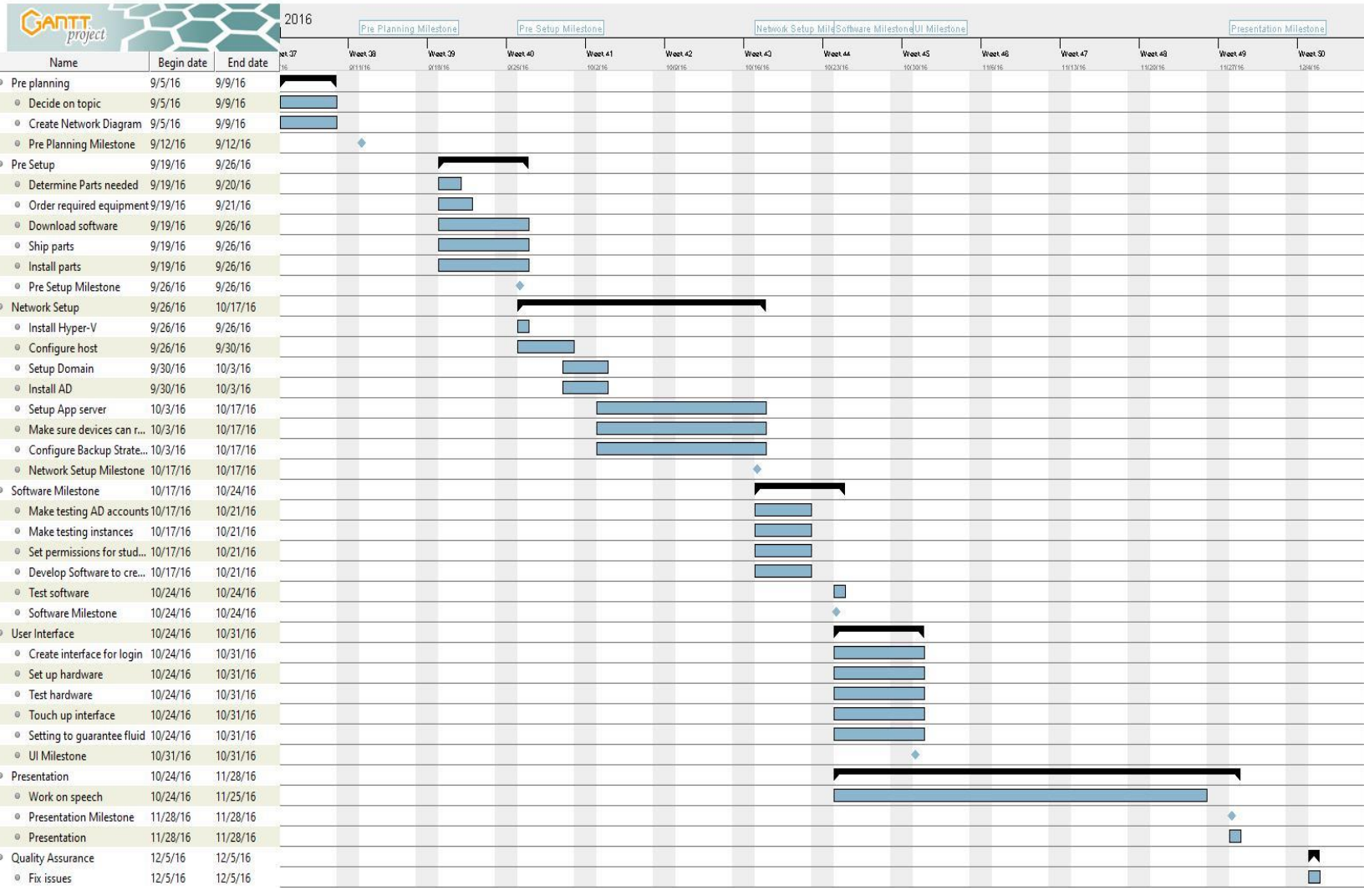
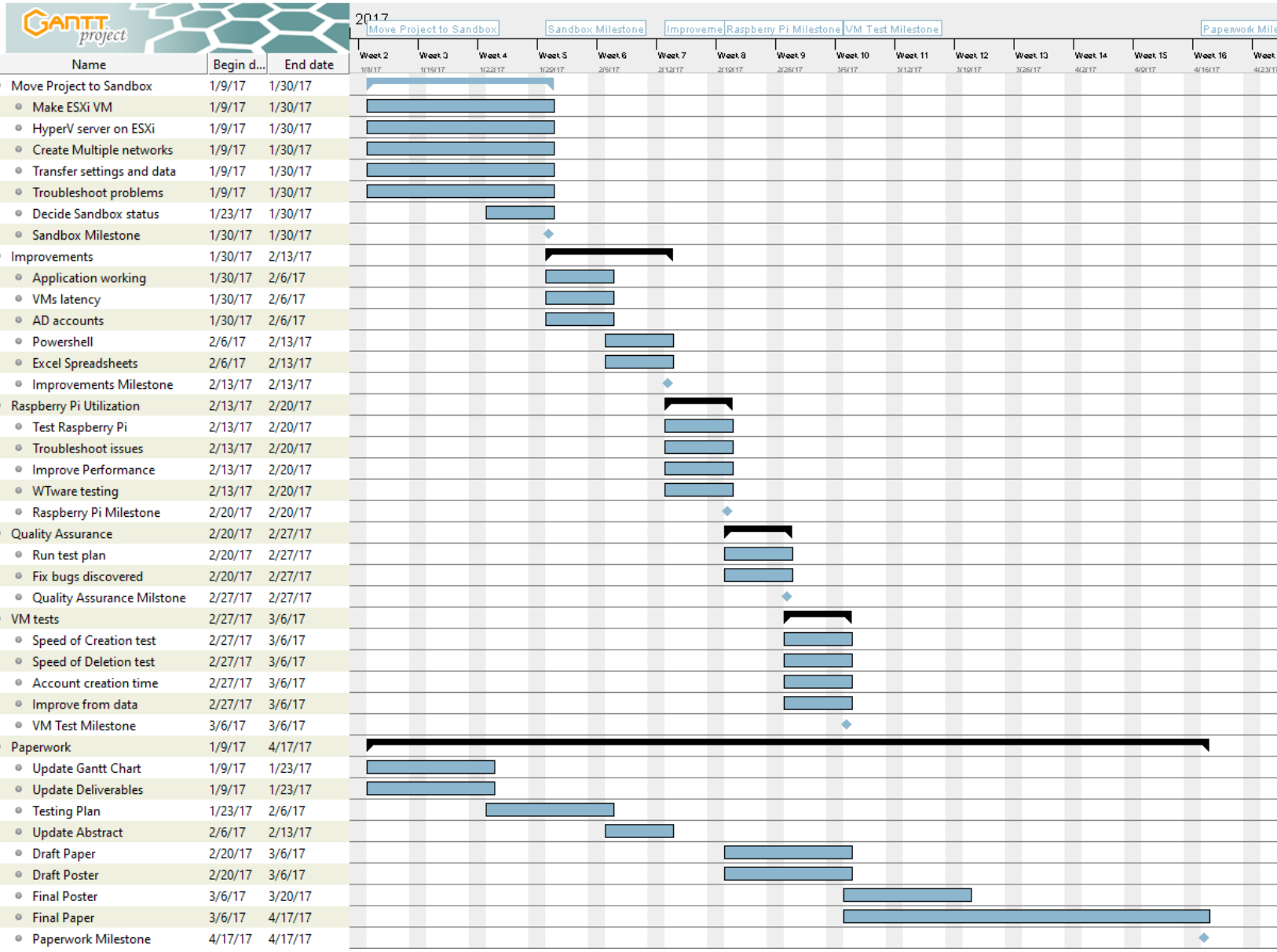


Figure 3 is our schedule with the major milestones listed for the Spring Semester

Figure 3 – Spring Schedule



Name	Begin d...	End date
Move Project to Sandbox	1/9/17	1/30/17
Make ESXi VM	1/9/17	1/30/17
HyperV server on ESXi	1/9/17	1/30/17
Create Multiple networks	1/9/17	1/30/17
Transfer settings and data	1/9/17	1/30/17
Troubleshoot problems	1/9/17	1/30/17
Decide Sandbox status	1/23/17	1/30/17
Sandbox Milestone	1/30/17	1/30/17
Improvements	1/30/17	2/13/17
Application working	1/30/17	2/6/17
VMs latency	1/30/17	2/6/17
AD accounts	1/30/17	2/6/17
Powershell	2/6/17	2/13/17
Excel Spreadsheets	2/6/17	2/13/17
Improvements Milestone	2/13/17	2/13/17
Raspberry Pi Utilization	2/13/17	2/20/17
Test Raspberry Pi	2/13/17	2/20/17
Troubleshoot issues	2/13/17	2/20/17
Improve Performance	2/13/17	2/20/17
WTware testing	2/13/17	2/20/17
Raspberry Pi Milestone	2/20/17	2/20/17
Quality Assurance	2/20/17	2/27/17
Run test plan	2/20/17	2/27/17
Fix bugs discovered	2/20/17	2/27/17
Quality Assurance Milstone	2/27/17	2/27/17
VM tests	2/27/17	3/6/17
Speed of Creation test	2/27/17	3/6/17
Speed of Deletion test	2/27/17	3/6/17
Account creation time	2/27/17	3/6/17
Improve from data	2/27/17	3/6/17
VM Test Milestone	3/6/17	3/6/17
Paperwork	1/9/17	4/17/17
Update Gantt Chart	1/9/17	1/23/17
Update Deliverables	1/9/17	1/23/17
Testing Plan	1/23/17	2/6/17
Update Abstract	2/6/17	2/13/17
Draft Paper	2/20/17	3/6/17
Draft Poster	2/20/17	3/6/17
Final Poster	3/6/17	3/20/17
Final Paper	3/6/17	4/17/17
Paperwork Milestone	4/17/17	4/17/17

3. Technical Elements

3.1 Network

Will be using a standard network, basic modem router and will be using one or more servers running Hyper-V. Each VM will be running Windows, but the Raspberry Pi devices will be running an open source Linux OS called WTware to make the initial connection. We will run a WDS server, DHCP, DNS, AD, and Hyper-V server all hosted on 2016 server OS.

3.2 Software

We used Microsoft Visual Studio, and Visual Basic to develop a basic application to automate our entire process. We used Windows Deployment Services on Windows Server 2016, as well as Hyper-V to run the VMs. WTware is an Open Source Pi Operating System, which we used as the OS for our Raspberry Pi devices. PowerShell is the language we used on the backend to run all the automation tasks. This is used to complete all the tasks ran in the application. When the user runs a task, it kicks of PowerShell to complete it.

Technology Used

4. Problems Encountered

4.1 PowerShell

For our project, we decided to use PowerShell to automate the process of deploying the new Virtual Machines. At first we didn't know what we could use to efficiently deploy virtual machines that could give the administrator full management of deploying the VM's while also making it as simple as possible. PowerShell accomplished this for us and then some. With PowerShell, we could fully automate the process of creating, starting, and configuring a virtual machine with very little input from the Administrator. We also tied in the functionality of adding the users to Active Directory at the same time.

4.2 Deployment

Deployment of the Virtual Machines was tricky for us. We did not know what program we could use to deploy the Virtual Machines operating system and completely automate the process of doing this. After some research, we found that Windows has their own software to boot operating system install disks over the network. We set this up to run on its own server and then worked to make it install the operating unattended. By modifying the unattend.xml file, we could make a complete install of Windows 10 deploy completely without having to touch anything.

4.3 Management

One of the benefits to putting all the student machines as virtual machines is easier management. We wanted to take advantage of this and find a way to be able to make management of the Virtual Machines much easier. Managing the machines is much easier because the admins access to the machines is easier. He can access all the machines from one computer or even a remote computer. Also, every student will have their own Virtual Machine and will only have the rights to log into their own virtual machine. This means that when a student is having a problem, the administrator will be able to remote into their VM directly and fix the problem, or the administrator will be able to administrate the student a temporary VM to use until he gets their VM fixed.

5. Testing

5.1 Overview

The testing chapter explains the methodology in which we tested Virtual Machines in the Classroom. It includes the objective of testing our final product and each test case in full detail.

5.1.1 Scope

The scope of testing is to test the operation of VMs in the Classroom running on Raspberry Pi systems as well as testing on the backend as well. The test will be organized based on requirements to use the devices and to use the applications on the server side as well.

5.1.2 Objective

The objective of testing is to remove known bugs and fix any problems from normal use of our application as well as our final product. This would consist of three different types of users Teachers, Students, and IT Staff.

5.1.3 Logging Test and Reporting

If a bug is found during testing, then the individual will document the bug. The team will meet after the test and decide if the bug reported is a bug or a feature not yet designed. Once the bugs are decided on then the developers will fix the bugs based on severity.

5.1.4 Entry and Exit Criteria

Entry Criteria	Exit Criteria
• Build Complete • Self-testing complete • Test Environment setup	• All tests are run • All bugs found are documented and fixed

5.2 System Testing

Virtual Machines in the Classroom will be tested using the Raspberry Pi systems to make sure that they work and can be used properly by the person being tested. As well as the backend, will test that the IT staff will be able to use the application and that the GUI for the application is self-explainable or easily learned.

5.2.1 Testing Procedures

The following are steps that are needed for testing which consists of:

- Create all the test scenarios and test cases
- Create a document of the steps to use to conduct the test and the expected results
- Describe the bugs found in the correct report

Below are the tests that will be performed:

1. **Server Stability Test** – This will focus on the backend being stable.
2. **Pi Test** – This will focus on the Raspberry Pi devices to make sure that it can be properly logged into and logged out of as a user.
3. **Server Functionality Test** – This test will focus on the features on the server side so that they are easy to use for IT staff and they perform as intended.

5.2.2 Pass/Fail Conditions

It is expected that Virtual Machines in the Classroom will pass all tests in the categories present above to be successful. If it does not pass, the testers will document any issues found.

Table 4 presents our testing strategy. This is used to test and make sure everything we needed to be tested was tested. This is more detailed in section **5.2.1**.

Table 4 – Testing

Stability Test						
Test	**Explanation**	**Expectation**	**Pass/ Fail**	**Date**	**Reason**	**Logged by**
Server stability	Servers up and running	Server will turn on and remain on with little to no down time	PASS	3/2/17	Servers have been constantly on since earlier in the semester	Jordan
Pi Test						
Test	**Explanation**	**Expectation**	**Pass/ Fail**	**Date**	**Reason**	**Logged by**
User Login	Pi is on and has log in screen that can be used	Login to Pi and connect	PASS	3/4/17	Login works as intended	Andrew
Correct VM	VM for the correct user	Did it connect to the user's VM	FAIL & PASS	3/4/17 & 3/11/17	Some IP issues when VM name is reused, working as intended now	Andrew
Running smoothly	Any delay in inputs or keystrokes	No delay or very little delay	PASS	3/11/17	Once VMs were installed on SSD, removed delays	Andrew
Applications Available	Student applications installed	All student applications are installed	PASS	3/4/17	All applications installed after VM created	Andrew
User Log off	Log out of VM	VM logs off completely and sets to login screen	PASS	3/4/17	When user signs off it disconnects, and logs off VM	Andrew

Functionality Test						
Test	**Explanation**	**Expectation**	**Pass /Fail**	**Date**	**Reason**	**Logged by**
VM creation	Create VMs	The VMs will be created and begin running upon completion	PASS	3/4/17	Working as intended	Andrew
Create new user account	Using credentials from old username	Account will be created	PASS	3/4/17	Account added to AD and given correct permissions	Andrew
Change User password	Change password for a user	Password will change	PASS	3/4/17	In AD and working as intended	Andrew
Delete User	Remove user from AD	When user is deleted VM associated will be deleted as well.	PASS	3/4/17	Working as intended	Andrew
Create VM in bulk	With excel doc and PowerShell create multiple VMs at once	Application will use excel doc and PowerShell to create mass amount of VMs	PASS	3/4/17	Excel spreadsheet creates all VMs without errors	Jordan
Delete VM in bulk	With excel doc and PowerShell delete multiple VMs at once	Application will use excel doc and PowerShell to delete mass amount of VMs	PASS	3/4/17	Excel spreadsheet deletes all VMs without errors	Jordan
Create User in bulk	With excel doc and PowerShell create multiple users at once	Application will use excel doc and PowerShell to create mass number of users	PASS	3/4/17	Excel spreadsheet creates all users without errors	Jordan
Delete User in bulk	With excel doc and PowerShell delete multiple users at once	Application will use excel doc and PowerShell to delete mass amount users	PASS	3/4/17	Excel spreadsheet deletes all users without errors	Jordan

6. Conclusion

VMs in the Classroom will allow schools to save a lot of money using a relatively old technology that would have been more expensive several years ago. The process is streamlined to make it easier for all users involved in the experience from the teachers and students to the IT staff. Having a simplified experience allows less stress on IT departments in schools as well as their budgets. Troubleshooting will be a lot simpler for IT staff and will allow them to more accurately fix problems. The automation is one of the biggest factors of our project and will aid IT staff in quickly creating environments for students and teachers with minimal effort. IT staff will be able to, with a just few clicks, create accounts and VMs for students to use.

7. Bibliography

[1] McCarthy, Shawn P. *Pivot Table: U.S. Education IT Spending Guide*. Report no. Doc #US41089016. IDC. V. 2 ed. 2014–2019. IDC, 2016.

[2] McCandless, Julia. "U.S. Education Institutions Spend $6.6 Billion on IT in 2015." Center for Digital Education. May 22, 2015. http://www.centerdigitaled.com/higher-ed/US-Education-Institutions-Spend-66-Billion-on-IT-in-2015.html.

[3] Winske, Chrissy. "IT Spending on the Rise in K-12 Schools." TechDecisions.co. October 12, 2016. https://techdecisions.co/it-infrastructure/it-spending-on-the-rise-in-k-12-schools/.