

Bolton Printing System

by

Michael Hill
James Scott
Jacob Difrank
Nur Ahmed
Mekdelawit Woldyes

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/Cybersecurity

© Copyright 2022 Hill, Scott, Difrank, Ahmed, Woldyes

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

Nur Ahmed	<i>Nur Ahmed</i>	<u>April 28, 2022</u>
Michael Hill	<i>Michael Hill</i>	<u>April 28, 2022</u>
James Scott	<i>James Scott</i>	<u>April 28, 2022</u>
Mekdelawit Woldyes	<i>Mekdelawit Woldyes</i>	<u>April 28, 2022</u>
Jacob Difrank	<i>Jacob Difrank</i>	<u>April 28, 2022</u>
Professor Ryan Moore	<i>Ryan Moore</i>	<u>April 28, 2022</u>

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

April 2022

Table of Contents

<i>Abstract</i>	3
<i>Introduction</i>	4
<i>Project Summary</i>	4
<i>Problem Statement</i>	4
<i>Solution</i>	4
<i>Project Source</i>	4
<i>Discussion</i>	5
<i>Project Objectives/Goals</i>	5
<i>Project Scope</i>	5
<i>Quick Project Timeline</i>	6
<i>Technologies Used:</i>	7
<i>Technical Architecture Diagram</i>	8
<i>User Personas</i>	9
<i>Use Cases</i>	10
<i>Use Case Diagram</i>	13
<i>Testing Plan</i>	14
Overview	14
Methodology	14
Scope.....	15
Objectives	16
Test Logs and Procedures	16
Testing Review	17
<i>Change Management Plan</i>	19
<i>Budget</i>	20
<i>Problems Encountered and Analysis of Problems Solved:</i>	21
<i>Conclusion</i>	21
<i>References</i>	22

Abstract:

Bolton Printing System is a collection of open-source software that works together to provide a seamless printing experience, allowing users to print directly from their device with just one click. The administrator can easily keep track of all print data. Three fundamental programs, referred to as engines, will be required to make this happen. Common Unix Print System (CUPS), which is the print engine, allowing any computer to operate as a print server, handling print workloads, fault detection, and logging. Our data engine will be constructed by utilizing Zabbix, a data dashboard program that will interact with CUPS using APIs to track and visualize data, such as error and print logs. Finally, we'll create a security engine with pfSense, an open-source firewall and router. It will enable for buffer overflow protection, port management, whitelisting and blacklisting of IP addresses and barring IPs from specified countries from accessing the system.

Project Summary:

System of technologies to ease printer use for the soho market by using primarily open source and preexisting technologies, to keep costs low for small to medium sized offices. It can be thought of as a collection of software and hardware, especially in the networking space such as print servers. Software will be created as well to facilitate ease of use. Our product aims to diagnose and fix printer problems such as out-of-date drivers, printer jams, network compatibility issues, security, and cost analysis. If the issues cannot be fixed remotely, our software will notify your company's IT Department before it becomes a bigger issue. Our product will also track your company's usage of ink and paper to both notify when ink and paper are running low, but also to create an up-to-date cost analysis of your company's spending on printer supplies.

Problem Statement:

According to a study done by Gartner, a leading research and advisory firm for IT strategies and practices, up to 50% of calls to tech support are related to printer malfunctions. Often for small office home offices printing is a daunting task due to printers many points of failure and complexity. Printers are unresponsive, can print slowly, refuse to print from model OS's, stop due to paper jams, refuse to scan, insecure, difficult to connect to the network, and printing is often expensive. In today's business printing is a necessity that many of us have experienced the frustrations of when it does not work. Printing should not be a constant issue stemming from multiple confusing causes.

Solution:

To resolve this Bolton Printing System will create a print server that manages the network operations of the printers. The print server will create a straightforward process for adding new printers, such as auto detecting IP and saving print drivers for adding devices. It will monitor print status to ensure print jobs are completed successfully. It will do this by detecting the cause of the error and resolving it on its own if possible. If not, it will notify the IT department of the issue and send the print job to another local printer if applicable. The system will also be able to send print failure information to the user via email. It will know the user by authentication prior to printing. By tracking users printing data will be collected on who is printing and what they are printing. This will allow IT to know who has more issues and what kind of issues so they can create preemptive systems. IT also can track printer ink and paper alerting IT of low supplies. Finally, the system will use encryption and various forms of authentication to keep it secure. The various forms of authentication will primarily be dual factor authentication.

Project Source:

We are students who have experience working at a help desk and noticed the number of calls coming in based around printing problems. We saw that this was a problem as it was wasting our time as well as the users. James Scott then conceived an idea to fix this issue. He invited his friends, Nur Ahmed and Mekdelawit Woldyes, to join and found Michael Hill and Jacob Difrank through Microsoft Teams. We then came together to analyze and produce solutions for the problem.

Project Objectives/Goals:

Project goals are as follows:

- Improve company productivity by introducing a system to handle common printing errors.

- Introduce a system that will easily alert an administrator to printing errors when they occur and then bypass the error to see if there is another location that can perform the task.
- Lower printing downtime costs by reducing downtime and enabling IT to resolve the issue quickly and efficiently or prevent the issue.
- Reduce company spending by creating a system to track the costs of printing materials such as paper and ink.
- Create a system to track printing expenditure as they are incurred to increase corporate productivity.
- Creates a secure method that ensures the work is delivered to the person who is printing it.

Team Members and Responsibilities:

Software Developer: Jacob Difrank

- Responsible for developing the software architecture.
- Responsible for database construction, monitoring, and management.

Network Administrator: Michael Hill

- Responsible for setting up a network and keeping everything connected.
- Responsible for researching hardware to be used.
- Responsible for writing programs and researching frequent network failure issues.

Printer Specialist: Nur Ahmed

- Responsible for having a high understanding of printers.
- Responsible for communicating with the team about general printing solutions.
- Responsible for communicating with Software Developer about how to best implement printing solutions.

Analyst: Mekdelawit Woldyes

- Responsible for finding associated metrics such as how effective the project is.
- Responsible for creating data dashboards and other visuals.
- Responsible for surveying users for data.
- Responsible for monitoring feature scope.

Cybersecurity Specialist: James Scott

- Responsible for creating security policies for the project.
- Responsible for maintaining the security posture of the system.
- Responsible for ensuring security is considered throughout the whole project.

Project Scope:

The system will allow offices to reduce printer downtime by automating the reporting of down printers, such as:

- When a printer fails to print, it will notify the user and provide further options, such as printing to a different printer.
- A dashboard will be included in the system to show problems and difficulties as they arise, as well as printer analysis, including cost analysis.

- It will also provide a security system to ensure that the person who prints is the same person who receives the printed papers.
- A cost analysis system to accurately display the current costs of printing such as ink and paper.

Quick Project Timeline:

Task #	Task Name	Duration	Start Date	End Date
Task 1	Research + Planning	10 Weeks	August 23	November 1
Task 2	Develop Proof of Concept	1 Week	October 4	October 11
Task 3	Develop User Personas	1 Day	October 4	October 4
Task 4	Research and Pick Technologies	1 Week	October 11	October 18
Task 5	Develop Use Cases and Use Case Diagram	1 Week	October 18	October 25
Task 6	Set Up a Working Environment	1 Week	October 18	October 25
Task 7	Develop Project Methodologies (Quality Assurance, Change Management, Iteration)	1 Week	October 25	November 1
Task 8	Develop Budget	1 Week	October 25	November 1
Task 9	Develop Final Report	4 Weeks	November 1	November 29
Task 10	Prepare for First Presentation	2 Weeks	November 1	November 15 - 29
Task 11	Network Setup	4 Weeks	November 29, 2021	December 13, 2021
Task 12	Printer Setup	2 Weeks	December 13, 2021	December 24, 2021
Task 13	Holiday Break	1 Week	December 24, 2021	January 3, 2022
Task 14	Set Up Print Server	12 Weeks	January 3, 2022	January 10, 2022
Task 15	Programming Interface	2 Weeks	January 10, 2022	January 24, 2022
Task 16	Programming Errors Alert on Interface	3 Weeks	January 24, 2022	February 14, 2022
Task 17	Programming Printing to	3 Weeks	February 14, 2022	March 7, 2022

	Different Printer on Error			
Task 18	Programming and Setting Up Security System	4 Weeks	March 7, 2022	April 4, 2022
Task 19	Programming Cost Analysis	4 Days	April 4, 2022	April 7, 2022
Task 20	Research Paper	8 Weeks	February 10, 2022	April 7, 2022
Task 21	Project Completion	1 Day	April 7, 2022	April 7, 2022
Task 22	Final Presentation	1 Day	April 12, 2022	April 12, 2022

Technologies Used:

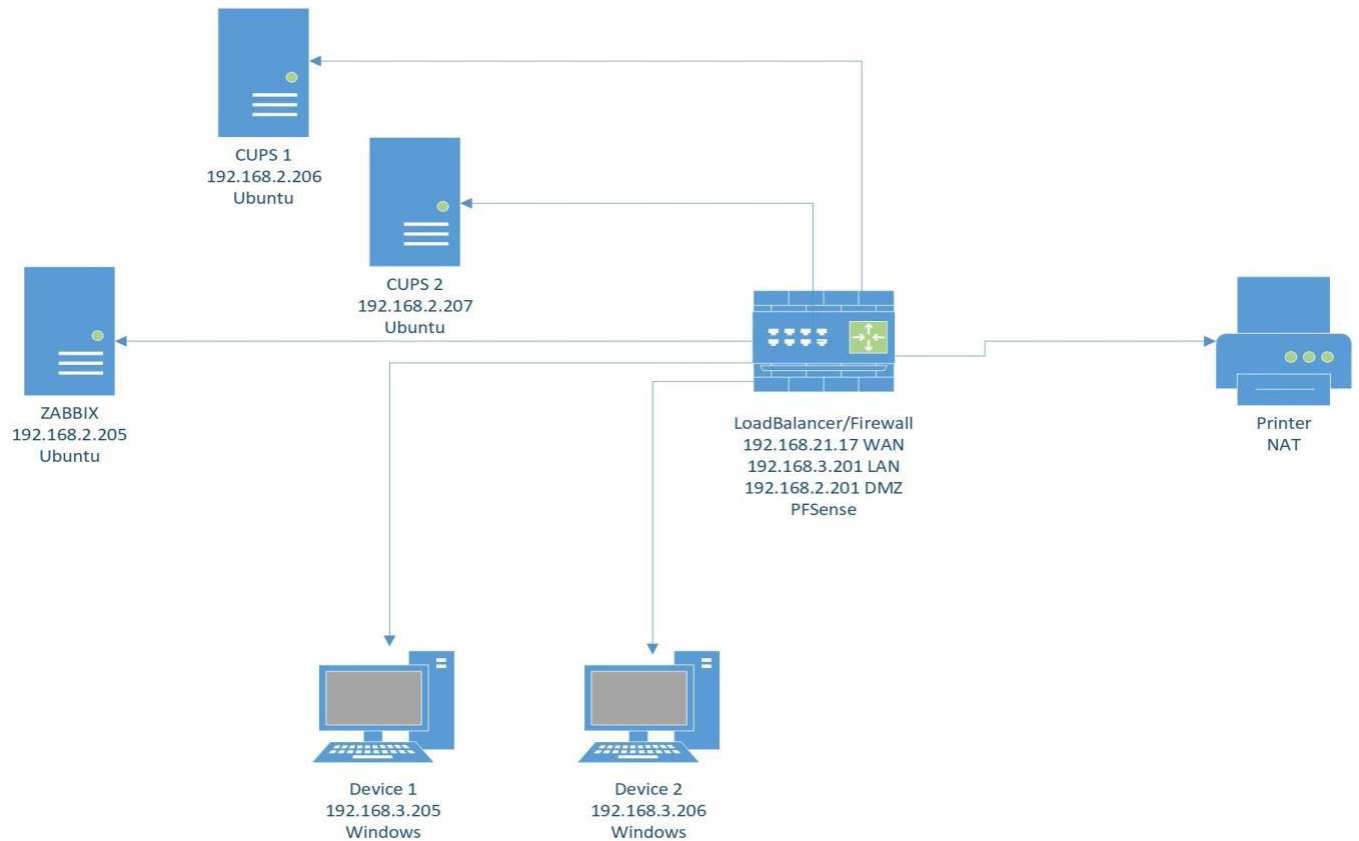
For our project we will need to set up a network with printers so that we can show how our system interacts with it and the printers. We will then need a programming language to have the printers do what we want, we will examine different languages to find the best fit in the research phase.

Short List:

- Router
- Computers
- Printers
- Print Server
- Programming Languages (Java, Python, PostScript)
- Common Unix Print System (CUPS)
- Zabbix
- Ubuntu
- PFSense
- Bootstrap
- VMWare

Technical Architecture Diagram:


The system architecture below tries to show how we interact with the systems or what solution our system provides. All the technologies we need to use for full system connectivity are labeled on the network diagram. Our solution works with three types of devices: Laptop, Phone, and Desktop. Then they will be connected to Wi-Fi. Then we have our Firewalls that can also prevent malicious software from accessing a computer or network via the internet. Finally, on our diagram you can see we will have Zabbix which will be used for error handling and CUPS will be used to create print server or automate printing, those tools play a vital role for our print management.





Network diagram showing the devices used for Bolton Print System.

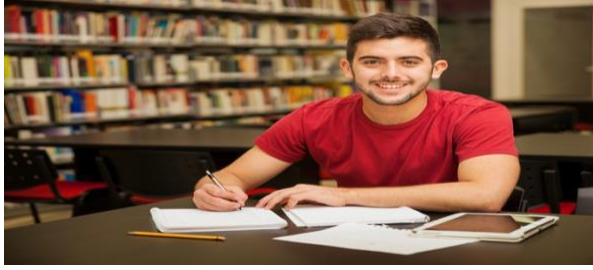
User Personas:

User personas are profile of the type of people who we envision will be drawn to our product and the types of people who will use and gain the most out of our system.

User Persona 1	
	Title: Marketing Director
	Name: Jennifer Wilson
	Age: 32
	Gender: Female
Behavior	Jennifer is a Marketing Director at a medium-sized company and gives reports to the board every week. For these reports she needs to print out a good amount of paper. She usually prints them out in the company.
Pain	Most of the time the printers do not respond, and Jennifer does not have the IT knowledge to fix it and the IT department takes too long to fix the problems.
Needs & Goals	Jennifer needs a reliable source to print from before her meeting.

User Persona 2	
	Title: IT Technician
	Name: Dan Johnson
	Age: 25
	Gender: Male
Behavior	Dan has worked in IT for a few years now and is familiar with the company's infrastructure.
Pain	Dan constantly receives calls and messages about printers malfunctioning, wasting his time and energy, costing the company money.
Needs & Goals	Dan wants to spend his time working on more pertinent IT initiatives. When there is a problem with a printer, Dan just needs a simple notification instead of multiple lengthy phone calls.

User Persona 3	
	Title: Small Business Owner
	Name: Patrick Miller
	Age: 50
	Gender: Male
Behavior	Patrick has just started his own business and has hired 5 employees. He bought a printer from BestBuy for the small office.
Pain	Patrick does not have any IT experience and employees complain of print issues that he does not know how to fix. Constantly on the phone with manufacturing support.
Needs & Goals	Patrick needs a printing solution that is both easy to understand and cost efficient for him and his employees to use to print off business documents.

User Persona 4	
	Title: Student
	Name: Brendan Anderson
	Age: 21
	Gender: Male
Behavior	A typical college student at the University of Cincinnati who is a little bit lazy and waits until the last moment to complete his work. He often likes to sleep in and with his hectic college life, he often forgets minute details.
Pain	Most of the time he needs to print off some homework a few minutes before class because he often wakes up too late, and in his rush, forgets to print off his homework at home.
Needs & Goals	He needs a hassle-free and quick way to print off his homework as the printing system at the University of Cincinnati is confusing and sometimes does not work at random times.

Use Cases:

Use cases are descriptions of how users will interact with our project. Below are five use cases describing how the system will interact with the printing user and the system administrators. These interactions will include what will happen when a print job is successful, when a print job is unsuccessful, how the system reports errors, how the system will handle cost analysis and how the system will handle security for print jobs that include sensitive data.

Use Case ID	1
Use Case Name	Print Paper with No Error
End Objective	To successfully print
User/Actor	Person Printing, Printing System
Trigger	User gives file to print
Frequency of Use	Multiple times per day
Preconditions	All printers must be working with no errors
Basic Flow	User wants to print file > User gives System file to print > System sees all printers working correctly > System prints to chosen printer > User receives printed paper
Alternate Flow 1 (Print from computer on printer network)	User wants to print file > User logs in to computer on network > User gives System file to print > System sees all printers working correctly > System prints to chosen printer > User receives printed paper

Alternate Flow 2 (Print from computer not on printer network)	User wants to print file > User logs in to computer on not on printing network > User logs in to printing network > User gives System file to print > System sees all printers working correctly > System prints to chosen printer > User receives printed paper
Alternate Flow 3 (Print from QR code on phone)	User wants to print file > User takes picture of QR code on desired printer > System logs user in via Bluetooth > User gives System file to print > System sees all printers working correctly > System prints to chosen printer > User receives printed paper
Alternate Flow 4 (Print from phone)	User wants to print file > User logs into printer network on phone > User gives System file to print > System sees all printers working correctly > System prints to chosen printer > User receives printed paper
Postconditions	Users receive printed paper and printers are ready to receive more print jobs

Use Case ID	2
Use Case Name	Print Paper with Printing Error
End Objective	To successfully print or to inform users and administrators that the system is not working
User/Actor	Person Printing, Printing System
Trigger	User gives file to print
Frequency of Use	Multiple times per day
Preconditions	Desired printer must be unavailable
Basic Flow	User wants to print file > User gives System file to print > System sees all printers working correctly > System prints to chosen printer > Error causes print job to fail > System then informs user and administrators of the error and sends a list of other printers to print to > User chooses different printer > Second printer prints successfully > User receives printed paper
Alternate Flow 1 (All printers are down)	User wants to print file > User gives System file to print > System sees all printers working correctly > System prints to chosen printer > Error causes print job to fail > System then informs user and administrators of the error and sends a list of other printers to print to > User chooses different printer > Second printer prints unsuccessfully > User receives error notice and notification that printing system is down > Network administrators are then informed that the printers are malfunctioning

Postconditions	Users receive printed paper or notice that all printers are down. Administrators are given notice that there are errors occurring.
----------------	--

Use Case ID	3
Use Case Name	Report Error to Administrator
End Objective	Administrator receives alert that one of the printers failed a job
User/Actor	IT Administrator, Person Printing, Printing System
Trigger	Printer Fails
Frequency of Use	How often do the printers fail
Preconditions	A printer must be unable to print due to error
Basic Flow	Printer fails > System reports the alert on the system interface “Errors” panel
Alternate Flow 1 (Alert through email)	Printer fails > System reports the alert on the system interface “Errors” panel > If “Alert Through Email” function is set up then sends the alert to chosen emails
Alternate Flow 2 (Alert through phone)	Printer fails > System reports the alert on the system interface “Errors” panel > If “Alert Through Phone” function is set up then sends the alert to chosen phone numbers
Postconditions	Administrators are informed that the printer is malfunctioning

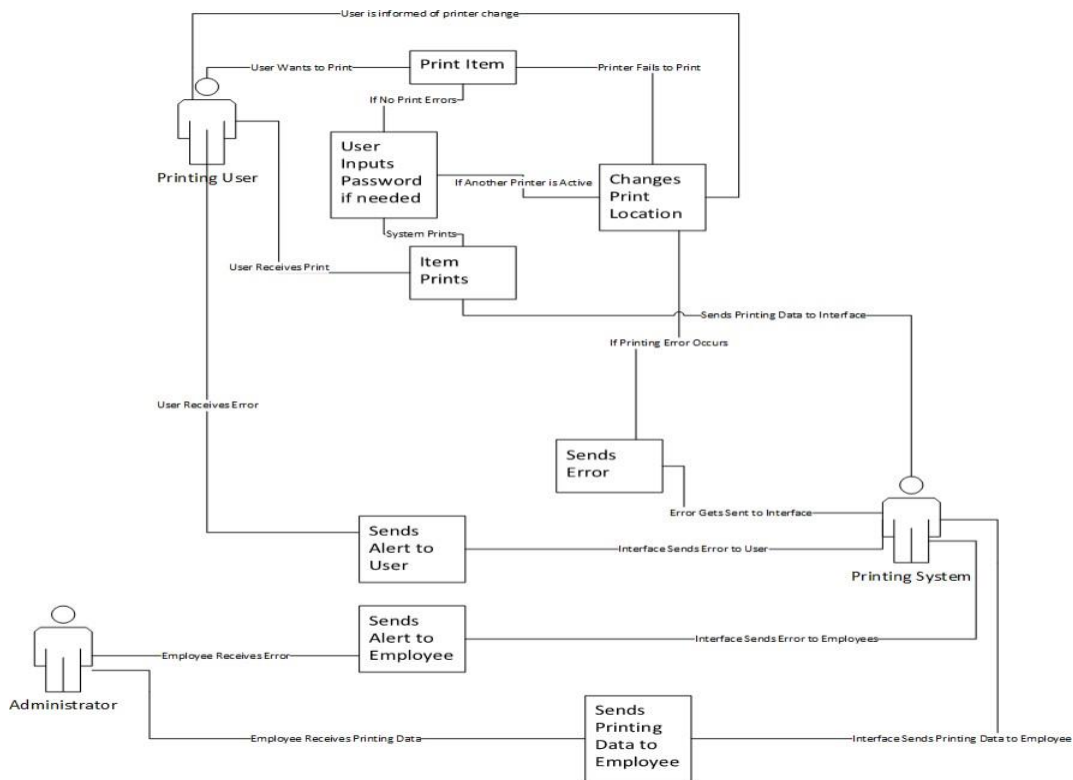
Use Case ID	4
Use Case Name	Report Cost Analysis
End Objective	Accurately track cost analysis
User/Actor	IT Administrator, Interface
Trigger	Successful print job
Frequency of Use	Multiple times throughout day
Preconditions	System must be able to print
Basic Flow	Print job is successful > System takes in how much ink and paper are used and sends information to Interface > Interface then displays information > Administrator observes cost analysis through Interface
Alternate Flow 1 (Through Email)	Print job is successful > System takes in how much ink and paper are used and sends information to Interface > Interface then sends an email to select addresses if Email function is set up > Administrator observes cost analysis through the email
Postconditions	The Administrators are informed of the cost analysis

Use Case ID	5
Use Case Name	Security

End Objective	To securely print paper with sensitive information
User/Actor	Person Printing, Printing System
Trigger	When User prints and security is enabled
Frequency of Use	Multiple times per day
Preconditions	Users must enable system security
Basic Flow	User attempts to print paper > User is asked at the printer the password given to them by the system > User inputs password > System prints paper > User receives printed paper
Postconditions	User receives printed paper and needs to input password by printer. The system is ready to print again.

Use Case Diagram:

The Use Case Diagram is a drawn map of how all the Users and Actors will interact with each other. Shown below is the diagram of the combination of all the Use Cases in the section above.



Testing Plan:

Overview

In the Testing Plan section readers may expect to find out how our team approached this project and our methodology for project management. You will explore how we chose our methodology, how we executed our iteration testing.

Methodology

For this project we decided to go with the Agile methodology. We went in this direction because our project can be easily split down into iterations that can have a substantial product at the end of each iteration. Our iterations will have three main phases, 1. Planning, 2. Construction, and 3. Testing.

1. **Planning:** In the planning stage the team will have a meeting to discuss the goals of the iteration. In this meeting the team will discuss the iteration objective and do initial research on that planned objective. Initial research and planning will include iteration budget, iteration timeline, and personal objectives.
2. **Construction:** In the Construction stage the team will begin production towards the iteration goals. Each team member will work towards their personal objectives assigned in the iteration planning stage. All group members will track and log what was completed in the group log and the team progress reports.
3. **Testing:** In the Testing stage the team will test the progress of the construction phase and log results and problems. The team will then have an end-of-iteration meeting where the team members will explain what they worked on, what worked, and what problems they encountered.

Our deliverables at the end of each iteration will include the iteration objective as described by the team in the planning stage with little or no bugs. Progress will also be tracked and applied to future iterations.

Scope

The scope will include all the use cases and features that we will be testing along with a description of each feature and its importance.

1. **The System Can Print:** The first thing we need to test is if the system can correctly print with no errors, if it cannot, then we move to the next step.
2. **The System Handles Errors Correctly:** The biggest part of our system is how it performs error handling. We need to make sure that when a printing error happens multiple things happen: 1. Administrators are notified of the problem, 2. The system can and will still print even with a printer down, and 3. In the case that all printers are down, that it is easy to install a different printer in the system. The system will also have some self-corrective behaviors. It will be able to notify administrators of necessary updates and power cycle.
3. **The System Handles Security Correctly:** Security is a secondary objective for our system. The main goal is to protect someone's print job if it is printing valuable information from other people if the system is in a public setting.
4. **The systems 4 core pillars are as follows:**
 - a. Availability
 - b. Cost
 - c. Ease of Use
 - d. Security

Each of these pillars is to be considered for all decisions throughout the project.

Availability is the idea that the printers will keep up with minimal downtime. Cost is the system should be affordable for small to medium businesses. It should be easy to use

for any user whether it is the IT office or Sales department. Finally, it needs to be secure. Printers should not be a vulnerability on the network.

Objectives

The team testing strategy and end goals will be what the team is expecting at the end of each iteration and what objectives are expected at the end of this Senior Design Project.

After each Iteration:

- A. All major features must be accounted for
- B. All user cases need to be accounted for
- C. All major bugs must be resolved
- D. Documentation must be completed

By the IT Expo:

- A. All major features must be accounted for
- B. All noticeable bugs must be resolved
- C. All documentation must be accounted for and arranged in an organized manner
- D. All group members must have a clear understanding of all aspects of the system
- E. All use cases must be accounted for and arranged in an organized manner

Test Logs and Procedures

Iteration Number	Iteration Description
1	Printer Connection Error
2	Error Log Process
3	Error Menu
4	Error Log
5	Backup Printer

Iteration Number	Test Number	User	Expected Outcome	Actual Outcome	Pass /Fail	Reason For Failure/Success	Date
1	1	Michael Hill	The system can read the error and report it.	System correctly read file and displayed some sort of message	Pass	The script correctly displayed a message to CUPS system.	11/20/21
2	1	Michael Hill	Create a script that has a menu that can run and stop a script as a process.	Correctly Accomplished starting and stopping the process	Pass	Script ran successfully	1/6/22
3	1	Michael Hill	Create a menu that can access and run the different	Created a menu using two different methods.	Pass	Successfully created a menu to locate the different	1/23/22

			features of our product.			functions of the product.	
4	1	Michael Hill	Create and be able to display an error log.	Created an error log that will display in the menu created in iteration 3.	Pass	Successfully created an error log that will display in the menu.	2/18/22
5	1	Michael Hill	Set up a printer to automatically print to a different printer if it is unable to print.	Created script that will print to a different printer on fail. This is set up in the menu of iteration 3.	Pass	Successfully created a system to print to a different printer on a printer fail.	3/11/22

Testing Review

This section will review in detail the iterations and what we expected, what we learned, information on any reworks, and what we would change in the future.

Iteration 1: Printer Connection Error

In this initial iteration, we are attempting to fix a basic error that could happen when printing, the printer not being connected to the network. This will happen when the system has a printer saved to the network and the user can print to it, but the printer is not turned or connected to the network. We wanted to start out with it because we viewed it as an easy first step that can help gain knowledge about the system and that we can have more knowledge for when we try to fix harder mistakes. Through some tinkering we produced a bash script that can detect and alert the CUPS machine that something is not connected right. The current alert is there as a place holder and the main part of the code is detecting the error.

Bash Code:

```
#!/usr/bin/bash
While inotifywait -q -e modify /var/log/cups/error_log > /dev/null; do
    If [[ "$(tail -1 /var/log/cups/error_log)" == *"The printer is not responding"* ]]; then
If [[ "$(xmessage -buttons Yes,No -title 'Print Error' 'Printer Connection Error! Do you want to
cancel print job?' -print)" == "Yes" ]]; then
Cancel "$(lpstat -o | awk '{print $1}')"
Fi
Fi
Done
```

This code will detect if the `/var/log/cups/error_log` file has been changed by using `iNotify-tools` which can be downloaded by the command `sudo apt install inotify-tools`. If it has it will try and detect the error that says, "The printer is not responding". If it does detect it, it will display some code, like in the code above it will display a `xmessage`.

Note: The message feature has been removed as of iteration 2.

Iteration 2: Error Log Process

In this iteration there have been some scripts that have been created to use as a blueprint to future features. These include a script to alert the Linux system if there is an error and a script to log errors. These scripts will be enabled and disabled through a menu that the admin can access. This menu will be built in a future iteration.

When the system detects an error it will grab the CUPS print job and save some variables that it can save to a log file. We do this by running these commands:

```
Job="$(lpstat -R | cut -d ' ' -f4 | cut -d ' ' -f1 | head -1)"
jobNumber="$(lpstat -R | cut -d ' ' -f4 | cut -d ' ' -f2 | head -1)"
printer="$(lpstat -R | cut -d ' ' -f4 | cut -d ' ' -f1 | head -1)"
```

We can do a lot with these variables in the future, but the main use is to create a basic error log in a future iteration.

Iteration 3: Error Menu

In this third iteration we have completed a menu that displays multiple features in the back end that will allow more experienced users a more in-depth experience with our product. For our menu we used two different methods to build it. The first and most widely used method was to use case statements to automatically build the lists that we knew what the options were. An example of one of our examples is as follows:

```
select number in Process Log Backup Exit
do
case $number in
Process)
#Process Code
;;
Log)
#Log Code
;;
Backup)
#Backup Code
;;
Exit)
#Exit Code
```

While using the case statements we encountered a problem where we couldn't figure out how to use case statements when we didn't know how many menus we wanted. In the future we plan on having some menus based on how many printers are set up. To work around this, we have hardcoded some while loops that simulate the case statements based on how many printers are set up in CUPS. To detect how many printers are assigned in CUPS we created a new file that has the names of all the printers. This while loop will then grab those names and create a list in the same format as the other menus.

At the end of this iteration, we have created a menu that will contain the different parts of our product.

Iteration 4: Error Log

CUPS has its own error log, but it is very confusing for an inexperienced computer user to understand and contains information that the user doesn't need to see. To work around this, we decided to make our own error log that is easier to understand and that will be displayed in a clean fashion.

Previously, in iteration 2 we gathered a bunch of variables that we are going to use to create a basic error log. We will combine the variables to create a folder that contains a file for every printer. These logs will be in the format of:

[Date Time] Error.

This will be displayed when you enter the menu and go to `Log>Printer> View Errors`

A second function was created to clear a log file. This will be done by going into the menu and going to `Log>Printer>Clear Errors`

This is done by overwriting the printer error file.

In this iteration we accomplished creating an error log for each printer in CUPS and writing it into the menu, along with some other functions like clearing the error logs.

Iteration 5: Backup Printer

One of our features that we discussed when planning this product is to be able to work around a printer failure and still be able to receive the print job.

We have added code to the list menu that when you assign a printer as another printer's backup, there is a folder that will save this. Then, when the error detection encounters an error, it will check the folder to see if there is a backup. If there is a backup assigned to the printer, code is run that will move the print job to another printer in CUPS.

Example code in the error detection process:

```
if [[ $(cat ./pBackup/$ePrinter) != "" ]]
then
bkPrinter=$(cat ./pBackup/$ePrinter)
lpmove "$ePrinter"-"$jobNumber" $bkPrinter
sleep 2
fi
```

After this iteration we have a print job sent to a printer that will send an error to the system. The system will then check if there is a backup printer associated with it. Assuming there is, then the system will move the job to that printer instead.

User Testing Plan:

Once we achieve a working prototype, we will run some user testing in which we will have some people with no experience with our project try and achieve a list of goals that we have given

them. This will be to test how user friendly our project is. We will also ask them to rate different sections of the project based on usability and practicality. Had a user with office suite experience run through a set of objectives that was provided. Users were instructed to create a word document. Then use their method to print. After they were told to follow instructions, we provided to add the printer and print. The users were given instructions how to view admin console and basic admin functions. Once the test was over a survey was given to rate the experience

Change Management Plan:

In case of a requested change, all team members will need to approve and understand the proposed changes. Changes cannot be requested of items that have a significant impact on past iterations to prevent substantial portions of the project getting changed close to the end of the project. Depending on the impact of the change, it will impact the urgency of the change. For instance, a change that will impact the rest of the project will receive a higher importance than a simple change that will not impact the project besides the slight change. Changes will be reported to advisors after a team meeting and the team has produced a solution to the change.

Budget:

Our budget is the expected cost of the completed project if our team were hired to complete the project. There are no labor costs for this project as this is our senior design project. We calculated our theoretical labor costs by imagining every group member will work for 10 hours every week of the remaining project. With all five members working 10 hours a week for 25 weeks (about 5 and a half months) at \$20 an hour, we imagine it will take \$20,000 in labor costs. We have blacked out the external labor row because of the smaller scale of the project. Due to it being a senior design project, we do not view our project as needing to hire external assistance. The software cost as of now is zero dollars. The team is making a conscious effort to keep costs as low as possible. One of the areas we found effective in doing this is by using openly free versions of software or open source. For the external hardware, there will be costs for a router for the printing network and printers. For cheaper printers and routers our team decided that \$1000 would be the mean price for all hardware. The miscellaneous costs currently include a \$200 bill for a logo from a graphic designer. This budget is subject to change as new software/hardware and features are decided. The data engine has not been decided thus it is not included in this budget.

	Rate Per/Hr.	Work Effort Hr.	Costs	Ongoing Annual		
				Rate P/Hr.	Work Effort Hour	X Support Cost
Labor IT	20	1000	\$ 20,000.00			\$
Labor External						\$
Software External			\$ 0.00			
Hardware - External			\$ 1000.00			
Misc.			\$ 200.00			
TOTAL			\$ 21,200.00			\$

Table 3- Project Budget

Problems Encountered and Analysis of Problems Solved

The problems we encountered occurred mostly when we attempted to install software on our Ubuntu machine. We had a lot of problems creating the html and bash script for CUPS. We had to utilize other software to see how the templates looked as we worked on them because the html pages are saved as .tmpl files. There were also a few sites, such as the add, change, and remove printers' pages and the add classes page, that were mostly functional but occasionally had strange issues. Furthermore, because the notify bash script above examines the error log file for revisions before searching for the error message if there is a connection fault, we could not figure out how to read the printer when it is out of paper. The printer, not CUPS, generates the out of paper notification. There is a cupsd.conf file that changes to show that the printer is out of paper, but it does not cause an error or the notify wait command to be triggered. Apart from that, installing Zabbix was a bit of a challenge because we were getting a lot of log errors and the commands were almost non-functional till we updated the system.

Conclusion:

Throughout the first semester of the senior project, we have learned many important lessons that will impact our future careers. One of the important lessons we have learned was to work as a group to study and implement the project management skills that we learned in our classes at the University of Cincinnati. Learning how to plan and complete a professional project is a crucial lesson that we have learned throughout this semester.

There were many skills that each of us have gained throughout the project. We have gained a better understanding of bash scripting as we use it in the CUPS system to control error management. We have also learned Unix interaction through our management of an Ubuntu server. The biggest most used skill we learned is how to research, look up and explore problems that we encountered throughout the project. This skill is so important because in the future, we will encounter problems that our current skills and knowledge will not be able to solve. Being able to correctly research the problem will enable us to be able to solve problems faster and increase productivity of future projects.

In the Spring we plan to connect more parts of the project together to create a nicer looking and complete project. This includes continuing to configure and customize our CUPS interface to create a more refined and user-friendly look and feel, as well as really digging into the other pieces of software we intend on using (Zabbix and Shadow Daemon), to find out their limitations and capabilities. The learning curve we encountered at the start of this semester is now mostly behind us. We now intend to use our knowledge of our chosen software, and of print management, to continue to develop our solution to the problem we encountered when we began this project. Throughout the spring semester, we added some major features to the product. We hooked up Zabbix with CUPS, we used Bootstrap to change up the CSS of the CUPS interface. We added a backup feature, expanded usability through menus, and created a log system.

Bolton print system is not a one size fits all software, but instead a set of tools to ease the burden of print management for IT administrators in the small to medium business. These tools are open source to keep costs down while also allowing further customization based on company needs. Each tool was chosen for its ease of use and their ability to make some of printing's most challenging tasks easier, like security and availability. Bolton Print system is software suite for the modern-day print management needs. It uses three primary software or engines. These are Common Unix Print System (CUPS), Zabbix, PFSense. Each engine has a core functionality that creates a fully functional system. CUPS is the print engine. This is the print server that manages the print jobs. Then there is Zabbix which is the data engine. This tracks all data throughout the print job. PFSense is the last engine. It is the network and security engine. It is the router as well as the firewall and is used to ensure secure print jobs with high availability.

Throughout these two semesters, we applied our knowledge to the best of our abilities to create this product. It was a great challenge and learning process for every member. We learned new software and areas of IT. Our team was predominantly. We applied the knowledge from our courses to create a fully functional solution to a problem that we have all experienced. We also saw firsthand the difficulties of bringing an IT project to fruition. We were able to work through that and create a product that all of us are immensely proud of. From learning to code, script, use new software's, and develop each of us grew our skills and has a deeper understanding of completing an IT project.

References

- *Brazil's third-largest bank transforms lengthy processes into just minutes with Xerox® accounts payable services.* Xerox. (n.d.). Retrieved April 24, 2022, from <https://www.xerox.com/en-us>
- Error. (n.d.). Retrieved April 24, 2022, from <https://my.ibisworld.com/us/en/industry/32311b/abot>
- *5 printing solutions for common business printer problems.* Marco. (n.d.). Retrieved April 24, 2022, from <https://www.marconet.com/blog/5-printing-solutions-for-common-business-printer-problems>
- *Five fast tips to improve your printing.* Xerox. (n.d.). Retrieved April 24, 2022, from <https://www.xerox.com/en-us/small-business/insights/printing-problems-and-solutions>
- Flesch, P. (n.d.). *How to fix printer problems yourself: Gordon Flesch.* How to Fix Printer Problems Yourself | Gordon Flesch. Retrieved April 24, 2022, from <https://www.gflesch.com/blog/common-printer-problems>