

IMPLEMENTATION OF A HIGH-QUALITY WEB-BASED DIGITAL PUBLISHING SERVICE FOR HIGHER EDUCATION

By

Michael A. Rodríguez-Meyer

A project submitted in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING

in

COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS

January, 2009

Approved by:

Nayda G. Santiago-Santiago, PhD
Member, Graduate Committee

Date

Manuel Rodríguez-Martínez, PhD
Member, Graduate Committee

Date

Jose Fernando Vega-Riveros, PhD
President, Graduate Committee

Date

Isabel Ríos, MBA
Representative of Graduate Studies

Date

Isidoro Couvertier, PhD
Chairperson of the Department

Date

ABSTRACT

Digital printing has become a very lucrative and technologically advanced industry in the past years. Many educational institutions currently use digital documents and printing methods. Although the majority of documents used in the University of Puerto Rico, Mayagüez Campus are in digital format, the methods for printing are manual. To meet these concerns we have developed a high-quality web-based digital publishing service that allows automatic printing of documents remotely without the need for printer drivers or manual printer interaction. The service can handle a diverse amount of document formats, uses a versatile account system, and uses mostly server-side programming, which not only makes it dynamic and useful, but also provides security and commercial application possibilities. This service also opens the doors for improvement and research in the digital publishing area as well as better resources for educational institutions.

RESUMEN

Impresión digital se ha convertido en una industria muy lucrativa y tecnológicamente avanzada en los últimos años. Muchas instituciones educativas actualmente utilizan documentos y métodos de impresión digitales. Aunque la mayoría de los documentos utilizados en la Universidad de Puerto Rico, Recinto de Mayagüez están en formato digital, los métodos de impresión son manuales. Para responder a estas preocupaciones desarrollamos un servicio de publicación digital de alta calidad utilizando la red cibernética que permite la impresión de documentos remotamente sin la necesidad de intérpretes o interacción manual con las impresoras. El servicio puede manejar una cantidad diversa de formatos de documento, utiliza un sistema versátil de cuentas y utiliza mayormente programación de lado del servidor, la cual no tan solo hace el servicio dinámico y útil, sino también provee seguridad y posibilidades de aplicación comercial. Este servicio además abre las puertas para mejoras e investigación en el área de publicación digital tanto como mejores recursos para instituciones educativas.

Copyright © 2009

By

Michael Rodríguez-Meyer

To my family and friends,
Who stood by me in best of times and worst of times

ACKNOWLEDGEMENTS

I would like to take this chance to thank my advisor and committee president Dr. José Fernando Vega for his guidance and help throughout the completion of this project and my master's degree. His knowledge, patience, and understanding were a beacon of light in the many dark times that I faced. I also give thanks to the rest of my committee members, Dr. Nayda Santiago and Dr. Manuel Rodriguez for their advice and assistance. Even though Dr. Wilson Rivera had to leave my committee before the completion of my project, I would also like to include him in these acknowledgements and thank him for his support as well.

I give thanks to the Network and System Administrators of the Department of Electrical and Computer Engineering. Without their help in computer, printer, and server related matters, this project would not have been possible. I thank my department and the Industrial Engineering department for their economic support throughout my degree. I give thanks to Dr. Nestor Rodriguez and the CISE Doctoral Program for both economic support as well as much needed practice in web development. I give thanks to my coworkers for their encouragement and help during this project and my degree.

Finally, but most importantly, I give thanks to my family and friends for their care and support before and throughout my entire degree and project. Without them I would not even be here today.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT.....	ii
RESUMEN	iii
ACKNOWLEDGEMENTS	vi
LIST OF TABLES.....	ix
LIST OF FIGURES	x
LIST OF ABBREVIATIONS.....	xiii
CHAPTER 1 - INTRODUCTION.....	1
1.1 Justification.....	1
1.2 Objectives	2
1.3 Outline.....	3
CHAPTER 2 – PREVIOUS WORK	4
2.1 Digital Publishing	4
2.2 Web-Based Printing Services	5
2.3 Project Background.....	6
2.4 Document Conversion & Implementation Changes	7
CHAPTER 3 – METHODOLOGY	10
3.1 Introduction.....	10
3.2 Materials	10
3.3 Project Phases	12
3.3.1 Server Set-up & Preparation	12
3.3.2 Web Service Implementation.....	13

3.3.3	<i>Database Communication</i>	15
3.3.4	<i>Data Testing</i>	15
3.3.5	<i>Printer Set-up & Communication</i>	16
3.3.6	<i>Document Testing</i>	17
3.4	Error Prevention & Security	18
CHAPTER 4 – SYSTEM DESCRIPTION		20
4.1	Introduction.....	20
4.2	Common Interfaces.....	20
4.3	Administrator Interfaces	25
4.4	Super Administrator Interfaces	42
4.5	User Interfaces	47
4.6	Database Model	58
4.7	System Workflow	60
CHAPTER 5 – CONCLUSIONS & FUTURE WORK		63
APPENDIX A – DPS DATABASE E-R MODEL DIAGRAM		67
APPENDIX B – JOB WORKFLOW ENGINE DIAGRAM		68
APPENDIX C – ACCOUNT BILLING RECEIPT EXAMPLE		69
APPENDIX D – DPS SECURITY MESSAGES EXAMPLES		70
APPENDIX E – EXAMPLE E-MAILS SENT BY DPS		72

LIST OF TABLES

<u>Tables</u>	<u>Page</u>
Table 1: Required software packages and versions on server.	12
Table 2: Job status descriptions and names.	39
Table 3: Database Tables and Functions.	59
Table 4: Truth table of possible job options in earlier DPS.....	60

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
Figure 1: Project phases and backtracking.....	10
Figure 2: Customer roles and associations.....	15
Figure 3: Log-In web page.....	21
Figure 4: Password Recovery web page.	22
Figure 5: Registration web page user information.....	23
Figure 6: Registration web page account information.....	23
Figure 7: Registration steps diagram.	24
Figure 8: Log-Out web page.	25
Figure 9: Administrator Home page and menu.....	26
Figure 10: Customer Search screen.	27
Figure 11: Sample search results (using All Customers link).....	27
Figure 12: Single user Notification screen.....	28
Figure 13: Create Accounts screen.	29
Figure 14: Account billing screen.....	30
Figure 15: System Configuration screen.....	31
Figure 16: Create Cost screen.	32
Figure 17: Edit Costs screen.	32
Figure 18: Create Color Option screen.	33
Figure 19: Edit Color Options screen.	34
Figure 20: Create Paper Type screen.....	35
Figure 21: Edit Paper Types screen.	35
Figure 22: Create Paper Size screen.	36
Figure 23: Edit Paper Sizes screen.....	36
Figure 24: Create New Paper screen.....	37
Figure 25: Edit Available Paper screen.	38
Figure 26: Job Activation screen.	39
Figure 27: Create Printer screen.	41
Figure 28: Edit Printer screen.	41

Figure 29: Printer Trays Management screen.	42
Figure 30: Super Administrator Home page and menu.	43
Figure 31: Activity Logs screen.	44
Figure 32: Log printing example.	45
Figure 33: Notification Removal screen.	46
Figure 34: Customer Removal screen.	46
Figure 35: Paper Sizes Removal screen.	47
Figure 36: Regular User Home page and menu.	48
Figure 37: Regular user My Workspace screen example.	49
Figure 38: Job Status screen.	50
Figure 39: Job submission process.	51
Figure 40: New submission job information screen.	51
Figure 41: Adding a file to a job example.	52
Figure 42: Changing file options example.	52
Figure 43: Job Preview information and file conversions.	53
Figure 44: Job submitted confirmation screen.	54
Figure 45: Customer Information screen with password changing tool.	55
Figure 46: Account Balance screen example.	55
Figure 47: Default File Preferences screen example.	56
Figure 48: Notification Preferences screen example.	57
Figure 49: Billing Options screen example.	57
Figure 50: Costs calculation diagram.	62
Figure 51: Database tables and attributes entity - relationship model.	67
Figure 52: Job workflow diagram.	68
Figure 53: Example billing receipt for flexible or unlimited accounts.	69
Figure 54: Session assurance error message example.	70
Figure 55: False page error prevention message example.	70
Figure 56: No printing account error prevention message example.	71
Figure 57: Administrator e-mail notification of new registered user.	72
Figure 58: Administrator e-mail from user via billing options tool.	72
Figure 59: User e-mail notification about preflight results.	73

Figure 60: User e-mail notification about proofing copy completion.	73
Figure 61: User e-mail notification about printing job completion.	74
Figure 62: User password recovery e-mail example.....	74

LIST OF ABBREVIATIONS

CUPS	Common UNIX Printing Solution
DP	Digital Publishing
DPS	Digital Publishing Service
GUI	Graphical User Interface
HTML	Hypertext Markup Language
JDK6	Java Development Kit 6
JSF	Java Server Faces
PC	Personal Computer
PDF	Portable Document Format
PHP	PHP Hypertext Preprocessor
POD	Print on Demand
MySQL	My Sequel
RoR	Ruby on Rails
UPR	University of Puerto Rico
UPRM	University of Puerto Rico, Mayagüez Campus

CHAPTER 1 - INTRODUCTION

1.1 Justification

Today documents are handled very differently than that in past years. Many documents are distributed, made, and read on computers and other technological devices. Although particular digital document formats seem to be almost standard, such as Microsoft Word Documents (.doc) and Portable Document Format (.pdf), there is still a diverse amount of formats currently being used. Just as documents have been converted to digital format, the printing processes have also evolved. Digital printing has recently become a very lucrative industry and many different companies have had significant advances in their services [1]. Even though most digital printing uses very advanced equipment, users must still manually interact with their printers, install custom applications and software, or physically connect to their printers or printer network. Portability is an issue that needs to be addressed given the increasing mobile technology and daily movement of users.

The University of Puerto Rico, Mayagüez Campus (UPRM) possesses a press facility for long run jobs. The press facility has added six digital printing systems, one black and white digital copier/printer more than two years ago and five full color digital printers in the year. The facility serves both the university community as well as external clients and can perform a variety of jobs, from brochures and posters to full books and presentation cards. Unfortunately, clients must bring their printing jobs to the facility personally. The facility has no web service or automated system, even though the digital printers are connected via network [2]. Although many web-based digital publishing services exist, most require the user to install applications and printer drivers or physically connect to the server's network. Other web-based services tend to be intermediary services that only receive the documents from the user and process the requests manually in some other way. From these concerns, a web-based digital publishing service would greatly benefit the university by providing a useful tool for printing digital documents without the need for drivers, software, or direct hardware interaction.

1.2 Objectives

There were three main objectives to be completed by the end of this project. These objectives were based, in part, on the idea that this work would be the continuation and completion of a digital publishing project that has been developed over the past three years. Although the previous versions of this project were outdated and required extensive re-implementation, the objectives remained the same. More information on the background of the project and changes made are stated later on. The objectives are as follows:

- *Install and implement the printing service server* - The computer system that would act as the server for the service was upgraded and therefore required preparation before the printing service could even be implemented. This new server uses a Linux-based operating system and many additional programs, such as Apache 2, PHP Hypertext Preprocessor (PHP), My Sequel (MySQL) Server, and Java Development Kit 6 (JDK6), needed to be installed and configured properly. The system also needed to communicate properly with the printers that would be used for testing. In order to accomplish this, the printer drivers and related printing software packages needed to be installed and configured as well.
- *Completion of the printing service components and interface* - Given that the service needed to be changed entirely, this objective included the complete implementation from the ground up of the service as well as completion of components that were not functional in earlier versions. Some of these components included the workflow engine, database tables, and graphical user interfaces (GUI). These changes also opened the doors to improvement of the service, such as adding document conversion and making processes more automatic.
- *Testing of the printing service* - The service actually needed to accept documents, change database information, and send the necessary commands to the testing printers. As with the second objective, this objective also opened the doors to

improvement upon earlier versions. The document conversion and other changes allowed testing of a wider range of document formats than what was originally allowed. Printing could also be done more dynamically at the moment of choosing the correct printer to use and sending commands.

A new goal that developed during the completion of this project was to make the service process request automatically with as little manual assistance as possible. Manual assistance would only be necessary for cases such as collection of service charges, hardware maintenance, special document binding or document pick up. Upon completion of these objectives, the service would be complete, fully functional, improved, and properly tested for use.

1.3 Outline

The remainder of this document is organized as follows: The second chapter will provide information about the area of interest this work falls in, Digital Publishing. This chapter will also present similar services and products currently on the market and provide a short historical background about the previous versions of this project. The third chapter will present the materials and methodology used throughout this work. In this chapter, changes to earlier versions of the project and decisions made will be explained and presented. The fourth chapter will present the results of this work. In this chapter, the interfaces, models, and workflow will be explained and demonstrated in more detail. The fifth and final chapter will present the conclusions reached in light of all the work completed and suggest possible improvements that could be made to this and similar projects in the future.

CHAPTER 2 – PREVIOUS WORK

2.1 Digital Publishing

As stated earlier, this work falls into an area of interest called Digital Publishing (DP). The main idea of DP is to link printing presses to computers by eliminating the use of films or plates and to raise the quality level for short-run printing jobs. Unfortunately, DP is hampered by a number of difficulties, such as getting the documents to print correctly without errors and managing the increasingly complex resulting workflow [3]. DP has several attractive advantages in relation to the printing of documents. It makes possible the idea of variable data printing, which is the essence of job personalization. Jobs can be modified at any point in the workflow once or multiple times with a minimum cost because job data remains digital until it is printed on the media. Traditional presses require a plate to be created for every job instance, making modifications during workflow not very cost-effective [4]. Another advantage is that customers using DP can obtain the desired services on demand. Print on Demand (POD) is a service created due to advances in digital printing. POD provides lower overhead, decrease in cost, and shorter run times [1]. Of course such benefits require a fast workflow in order to take full advantage of any hardware or software capabilities [5]. In attempts to achieve faster workflows in POD and web-based printing services, workflow automation is being used [1]. Productivity has clearly been accelerated thanks to online workflow, intelligent automation, and fast hardware [5].

In DP there are steps a job must go through in order to ensure that the quality of the job is the best one for the particular type of job that will be published. Although in this work we will not discuss all these steps in detail, it is useful to understand a few particular steps not traditionally used in past printing processes that will be present later on in this project. These steps are preflight and proofing. Preflight helps to ensure that all required files, such as fonts and graphics, are included in the package sent to the printer. Preflight also verifies document properties, tests validity, and tests the completeness of a prepared document using predetermined rules [3]. Proofing is a process where a single

example copy of the job is printed before the entire job. This copy is provided to the client for verification in order to physically view what the end result of the job will look like and to ensure all specifications are met without wasting resources on the entire job. This also allows the client to modify the actual job settings or any other changes if desired.

2.2 Web-Based Printing Services

It is very valuable to understand and study some of the products currently on the market with applications equal to or nearly equal to the printing service that will be presented in this project. One of these products is Lulu.com, which offers printing on demand and publishing services at a modest fee. Documents submitted to Lulu need to be in PDF format or Lulu can convert them for the client, also at a price. Lulu also offers many binding options and specifications, but such binding is not done automatically or directly from the web site. The client must submit their document with a certain format specified by Lulu on their web site and request the binding options he or she desires from the site's lists [6]. Another well known service is Parlor Press, which is an independent publisher and distributor of scholarly and trade books. Documents submitted must already be in PDF format and must have a specific formatting specified by the service [7].

Cornell University Library created an open-source software system designed to enable the organization, presentation, and delivery of scholarly journals, monographs, conference proceedings, and other common and evolving means of academic discourse called Digital Publishing System (DPubS). However, this service was meant to aid colleges and universities in managing and disseminating scholarly documents, not actual printing [8]. Another interesting system is Net2Printer. This system is a commercial software solution that allows clients to print documents to network printers from any location through the internet for an annual fee. The client requires the sender software in order to use the system and must configure it for use with the printers to be accessed [9]. The main disadvantage of these services is the fact that the document submitted must be either formatted a certain way or be of a certain format type (in most cases PDF). Also,

the services do not process the documents submitted automatically. Documents are only sent through the web application, but are handled independently using other methods before being sent back to the user. Other applications that do handle documents automatically tend to be software suites that require installation and configuration.

2.3 Project Background

Given that this project is a continuation of an earlier version of the DPS, it is important to understand the previous work done with this service and its current state. In the year 2005, Dr. José Fernando Vega began research in a web-based service in association with Purdue University which was funded and sponsored by Hewlett Packard (HP). This service was meant to be a revolutionary tool for printing any kind of publications with the desired quality and quantity. It would virtually eliminate the traditional time-consuming and highly expensive plate method for printing publications [10]. The DPS was developed by undergraduate students as part of an ongoing project under the Capstone course of the Department of Electrical and Computer Engineering (ECE). The first group to work on the DPS from August of 2005 to December of 2005 was called the Digital Publishing Research Group (DPRG), composed by the students Ricardo Veguilla and Diane Yopez. These students began the DPS implementation and defined part of the modules that would be used **Error! Reference source not found..** The second group worked on the DPS from January of 2006 to May of 2006 and was called the Nelson-Amaris-Fernando Team (NAFT Enterprise) composed by the students Nelson Quirindongo, Amaris Nieves, and Fernando Torres. This group continued the implementation, incorporated new designs, and attempted to improve the modules made in the earlier version **Error! Reference source not found..** The final Capstone group to work on the DPS was the Barbosa-Borges-Peña Team (B²P Enterprise) from August of 2006 to January of 2007 composed by Juan Barbosa, Xavier Peña, and Pedro Borges. This group made vast changes in the project, continued implementations, and reorganized all the lacking documentation for the project [13].

After the last Capstone group completed work on the DPS the service was functional, but had many incomplete features. The service would remain dormant for a

whole year before being handed to me for completion as part of my master's degree project. In the final report made by the B²P Enterprise, the most current state of the printing service was explained as well as the student's contribution to the service. According to this report, the previous teams lacked in documentation of their work on the DPS components and implementations. Important documents, such as E-R diagrams, class diagrams, and workflow engine details, were in fact missing. Some database tables contained redundant data and the workflow engine was implemented using Yet Another Workflow Language (YAWL), which was complex and only supported the database system Postgres [13]. The service was revised and updated in three basic phases by the B²P Enterprise team. First, the documentation of the database system, workflow engine, class documentation, and class diagrams were revised or created where there was no documentation. Second, the database design was upgraded, the workflow engine script was revised, and a test plan was created. Finally, the web interface and workflow engine were implemented and the test plan was used. The first two phases were completed successfully in the work period. However, the third phase was only partially completed. The web interface processed and returned the correct values internally, but the web page itself did not reflect the values. The workflow engine script was completed successfully, but the test plan was only done up to the point that was permitted by the interface. Clearly, there were still many tasks left to complete. The service currently needed the interface to be completely debugged, documents handled needed to be filtered and converted to a manageable format, and the workflow engine as well as the service as a whole needed to be tested thoroughly.

2.4 Document Conversion & Implementation Changes

One of the tasks of the printing service that required attention was investigated during an independent study at the University of Puerto Rico, Mayagüez Campus. During this study, the automatic conversion of documents was researched in order to find a useful method of conversion with the current printing service constraints (low or no cost and a UNIX-based operating system) [14]. The Portable Document Format (PDF) was found to be the easiest way to communicate with the printers given that it uses the

Postscript page description language. This fact allows the service to find document information such as number of pages and margins without complications. Three document conversion methods were studied and tested. The first method, using third party conversion software, was tested based on the software cost, result quality, need for an independent interface, and variety of document types that could be converted. The second method used special printer drivers, such as the Apple LaserWriter II NT v47.0. Results for this method were tested based on variety of document types supported and quality of the result. The final method, Application Programming Interfaces (API), was used in order to test the possibility of converting documents from within another application. The tests were based on the same criteria as the conversion software methods. The test results from all the methods demonstrated that the last two methods were the most cost effective and plausible methods for document conversion under the constraints given. The latter of the two methods was the most interesting given that the printing service would be implemented on a server with a UNIX-based operating system that already included a few open source APIs, such as Open Office, that handled documents and conversions.

An attempt to re-install the DPS using the documentation left by the last undergraduate group failed due to outdated programming and unclear documentation. During the summer of 2008, it was decided that the DPS would be re-done completely from the bottom up using Ruby on Rails (RoR) as the foundation while trying to conserve the overall ideas, functionality, and database implementation of the previous team. By the end of the summer of 2008 to August of 2008, it was noted that using RoR might have been useful for creating speedy web pages thanks to the use of the Model-View-Controller methods. However, the automatic scaffolding methods and files generated by RoR made debugging and code modification very difficult and time consuming without exact knowledge of where files were going to be placed and how they were all linked to each other. This led to the final decision of using a combination of PHP/Javascript languages and common Hypertext Markup Language (HTML) code as the most efficient method. The reason for this was the simplicity found while programming using scripting languages, the useful database and command line functions found in PHP, and the client-side execution of PHP code. Another advantage of using scripting was the elimination of

the need to compile part of the code before testing it as can be found while using Java applications. The basic database, interface, and workflow ideas were maintained. Now that we have a basic understanding of digital publishing, the previous work done on the DPS, and the changes that needed to be performed to the service, we can move on to the methodology used and the reasons for any decisions made during implementation of this work.

CHAPTER 3 – METHODOLOGY

3.1 Introduction

In this chapter we will learn about the system requirements needed to run the DPS and the methods used in order to implement the service from start to finish. All the problems and decision made throughout the project will also be presented and explained. The implementation of the DPS was divided into phases that needed to be completed in the order presented because each phase required programming or methods from the one that came beforehand. Figure 1 illustrates the phase order and some of the backtracking that was performed. As stated in the previous chapter, the service was using an earlier version as a reference, but the implementation was done almost from the bottom up.

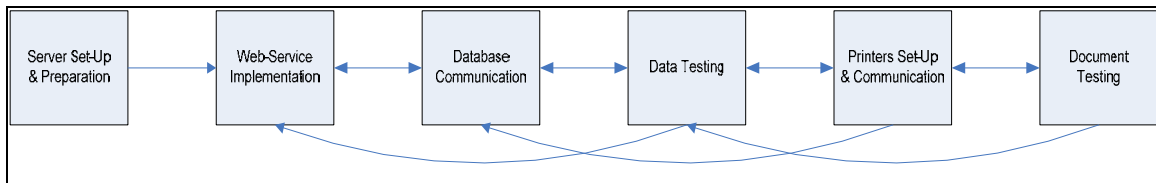


Figure 1: Project phases and backtracking.

3.2 Materials

The main materials used in this project were the printers, the server, and a test computer. Initially there were three HP Color Laserjet 9500 hdn printers set up in the reproduction center of the university. These printers have a 4,800 dpi maximum resolution in black and color, up to 24 ppm (page per minute) print speed in black and color, 3100-sheet standard input capacity, 500 MHz processor, and 288 MB standard memory. The printers are connected to the server via network, use Postscript drivers, and have the capability to automatically detect the paper size of media loaded in trays and color options of documents sent to them. For ease at recognizing which printer was being used during the tests, the two available printers were denoted *hp_laserjet9500_left* and

hp_laserjet9500_middle indicating their relative position to each other in the reproduction center of the College of Engineering.

The test computer, denoted *Asmodeus*, is a HP Pavilion dv6000 laptop that was used at different locations within and outside the university. This test computer has an AMD Turion64 X2 2.00 GHz processor, 3 GB PC2-5300 RAM, 250 GB hard drive, Integrated Wireless, Ethernet, and Infrared capability. The main programming used on this test computer was Windows Vista Home Edition, SSH Secure Shell Client, WampServer, and Adobe Master Collection 3. The Secure Shell Client was used to remotely upload content from the test computer to the server. Adobe Master Collection was used for web design and testing via the Dreamweaver and Fireworks programs. WampServer is a Windows-based utility that installs and incorporates Apache Server, MySQL, and PHP in one integrated package. This last program suite was especially useful for setting up the service for testing without the need for manually configuring each web-based utility individually.

The server, initially denoted *calima02* then changed to *hersrv*, is a Dell Precision 340 with a Pentium 4 processor 1.8 GHz, 512 MB RAM, 80 GB hard drive. The server was stored with the other computer laboratory and research servers of the Electrical and Computer Engineering Department. As with the servers of earlier versions of the DPS, the server used a UNIX-based operating system in order to better test portability, run with less system requirements, and reduce compatibility and configuration issues that are sometimes found under Windows-based systems. The specific operating system used was Ubuntu Server version 8.10, also known as Intrepid. The packages of software that were required on the server can be found on Table 1. Later on in this chapter we will learn about the use given to some of these software packages in relation to the DPS.

Table 1: Required software packages and versions on server.

Package Name	Package Version
MySQL	14.12 – Distribution 5.0.67 for Debian-Linux-GNU
PHP 5	5.2.6-2ubuntu4 with Suhosin-Patch 0.9.6.2 (cli)
phpMyAdmin	2.11.8.1-debian1
Apache Server 2	2.2.9-7ubuntu3
Build-Essential (Perl, C++, C, etc.)	11.4
CUPS	1.3.9-2ubuntu4
CUPS-PDF	2.4.8-1ubuntu1
GhostScript and GSFonts	8.63.dfsg.1-0ubuntu6
HPLip and PPDs	2.8.7-0ubuntu6
HPijs and PPDs	2.8.7-0ubuntu6
MySQL Server	5.0.67-0ubuntu6
Netbeans	6.1-0ubuntu1
OpenOffice.org Suite 3	3 – 1:2.4.1-11ubuntu2.1
OpenSSH Client & Server	1:5.1p1-3ubuntu1
PHP-Mail	1.1.14-1
Postfix	2.5.5-1
Sun Java Development Kit 6 (JDK6)	6-10-0ubuntu2
VNC4Server and X11 Fonts	4.1.1+xorg1.0.2-0ubuntu7
JODConverter	2.2.1

3.3 Project Phases

As stated earlier, the phases that the project was divided into needed to be completed in a certain order due to dependencies between each phase. Some phase components were linked to components of an earlier phase, therefore backtracking was necessary in order to ensure that functions communicated and worked properly throughout the project. Some errors that existed within a given phase would not be apparent until the next phase was being conducted due to the nature of the data or functions being executed.

3.3.1 Server Set-up & Preparation

During this initial phase, the server mentioned earlier was connected to the network and most of the packages in Table 1 were installed using the application package manager command (apt-get install) via the bash console screen. The Apache Server, PHP, and MySQL packages were configured by modifying their respective configuration files

to effectively integrate them with each other. During this phase the packages that would be needed for printer communication, such as CUPS and the PPD files, were also installed, but configured properly later on. Given that Ubuntu Server was installed on the server and no video peripherals (for example: a monitor) were necessary, communication with the server was done via the university network or remotely through the internet. Once all the programming and web-based packages were installed on both the server and test computer, the implementation of the service interfaces and basic functionality could begin.

3.3.2 *Web Service Implementation*

The web-based service components were initially written using Java Server Faces (JSF) and Java Web Services. After attempting to re-install these components using the guides left by previous DPS groups, it was clear that most of the methods used outdated software versions that would create conflicts with the updated packages installed on the server. It was decided that the components would need to be redesigned due to the fact that the new JSF version was drastically different from the one originally used in the DPS and would require extensive and time consuming re-programming. The first choice was Ruby on Rails (RoR) because it also used the Model-View-Controller (MVC) methodology that was also used in JSF and the relative ease it provided when creating interfaces thanks to the scaffolding ideas it implemented automatically. After testing RoR however, we realized that altering some of the interfaces it created automatically was complicated and time consuming due to the automatic files that were generated. This was especially true when using plug-in components that could be integrated in RoR, such as the authentication methods.

These inconveniences with RoR brought the decision to use a more simple programming language and re-design all the interfaces from the beginning with the older versions of the DPS as guidelines. PHP was used for the rest of the service implementation due to its integration with database components and modular processing speed. PHP is also a scripting language that is used by many web-based businesses and personal sites. This means that it has many functions built in that support common

interface methods, while scripting allows modification and testing without the need to compile components. Since PHP is a server-side language, clients would not need any programming libraries or packages installed on their computers in order to execute or view PHP functions. During the implementation of the interfaces, Javascript was also found to be a useful tool for providing more dynamic and friendly designs. Although Javascript is a client-side language, it was still useful for its scripting advantages and to provide more intuitive and smoother functionality for the users.

The users or customers that interact with the DPS are classified as Administrators and Regular Users. Administrators have the authority to modify system parameters, such as the available media, active printers, and costs, as well as user data parameters, such as account modification, job status modification, and user notifications. Given that the DPS would be handling possible confidential documents and monetary resources, administrators could not remove data from the system or use the printers directly. In order to provide a method for removing data with a highly controlled access, the DPS has a second type of administrator called the Super Administrator. There is only one super administrator and the data for this user can only be modified by directly manipulating the system's database. Regular users are the paying customers of the DPS and all have accounts associated to them in order to print documents. The different types and associated sub-types of customers mentioned can be viewed in Figure 2.

The first interfaces designed in this phase were the common interfaces shared by all users, such as log-in, log-off, and registration screens. The next interfaces to be implemented were the administrator screens because these interfaces mostly interacted with the database through simple queries and not with the printers. By this time, the database communication and modification would be implemented.

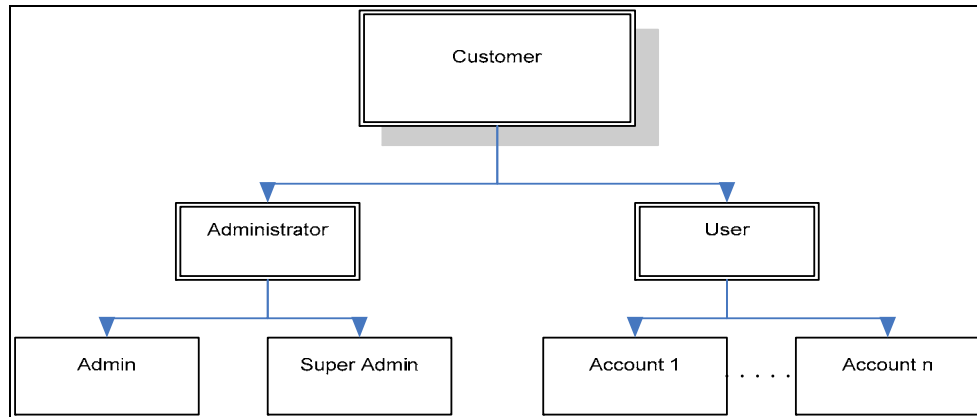


Figure 2: Customer roles and associations.

3.3.3 Database Communication

The main database tables and information was directly imported from the earlier version of the DPS because MySQL did not change much from the older version to the new version currently used by the server. Some relationships were modified later on due to incorrect relations, unused tables, or simplicity during data modification. During this phase the database was created, modified, and partially tested. The administrator interfaces and some of the user interfaces needed to properly call tables in the database and modify data already there. The tables that were viewed the most here were user data, account data, and media parameters. The package with phpMyAdmin provided a PHP-based interface for manipulation of the database data as well as query testing. Later on in this work we will learn more about the database model and tables. Once the database and interfaces could properly communicate with each other, tests of data manipulation and interface functionality were possible.

3.3.4 Data & Query Testing

During this phase the interfaces were verified to ensure that the various queries to the database were being executed correctly. This particular phase was very useful for correcting interface errors, adding and removing table columns, and modifying database relationships. Simple queries, such as account data and system settings modifications, were sent to the database through the interfaces in order to test the displaying and

manipulation of the data on the database tables. Although many queries were simple data selection, insertion, or updating queries executed on one table, there were a few queries that required joining three or more tables in order to produce a given result. These particular cases required the phpMyAdmin query interface in order to test the validity of the complex queries. Most of the work was done with a combination of the phpMyAdmin interfaces and Dreamweaver site testing methods. By the end of this phase the administrator interfaces were almost complete and the user interfaces, such as account options, were partially complete. In order to complete the interfaces and process jobs, the printers now needed to communicate properly with the service and integrate with the database and web interfaces.

3.3.5 Printer Set-up & Communication

Given that the server was running Ubuntu Server, the usual Windows-based printer drivers could not be used to install the printers. UNIX-based systems tend to use the Common UNIX Printing Solution (CUPS) software packages. The CUPS software provides various printer drivers and a graphical user interface (GUI) in order to install printers on a given system. However, additional drivers were needed for our printers to communicate effectively and understand documents sent to them. To this end, the HP Linux Imaging and Printing (HPLip) were also required because they contain many of the printer drivers currently supported for Linux systems by Hewlett-Packard. Once these packages were properly installed and the printers were added via the GUI provided by CUPS, a problem developed related to document handling. The printer drivers that existed for the test printers were Postscript drivers and could not interpret any document format sent to the printers. Only .pdf or .txt files could be interpreted correctly without the need for intermediary software. Another problem would be the need to obtain the number of pages in a document before being sent to the printers in order to properly calculate costs for jobs.

The solution found for these two problems was PDF conversion as researched in an Independent Study course [14]. In this study, we could see that the PDF format was a standard document format used by many industries that did not possess the large amounts

of meta-data found in other formats. This would make both page counting easier and be supported by the printer drivers. As we learned in the study, using an open source document package (in our case, Open Office 3) for UNIX-based systems as an intermediary would do the necessary conversions. The next problem was that the conversion needed to be command line-based and Open Office required a monitor and GUI to run properly. The solution found for these problems was the Java Open Document Converter or JODConverter combined with the Virtual Network Computing Server or VNCServer. The VNCServer would simulate the monitor needed by Open Office and allow Open Office to run as a service. The JODConverter is basically a script that accepts documents and uses the appropriate Open Office method in order to convert the document to another format. With this script running as a command line tool, the DPS could convert common document formats to PDF format. All command line functions were executed by a PHP method (exec) from within the DPS. This allowed printer communication and some of the document manipulation without the need for the Printer Job Language (PJP) that was used in earlier versions of the DPS. Once the printers were communicating properly, testing of actual documents and jobs could begin.

3.3.6 Document Testing

During this final phase, the handling of jobs by the DPS was tested and errors were corrected. A test user would send a new submission to the DPS and the service needed to verify all job information, convert the documents to PDF if necessary, extract the number of pages, calculate costs, and send the proper commands to the printers. Throughout document testing, the database communication was also critical because all account costs and changes in job status would require information from the database tables. During this final phase, the workflow engine was developed and tested as well. This engine is a separate PHP page that is called at different points throughout the DPS and decides what printer commands, database interactions, or notifications are executed via extensive programming conditionals. The main points that were checked during this phase included the workflow being executed in the proper logical order, the job information modified appropriately, costs verified, notifications sent at the proper moments, and the appropriate

commands sent to the printers. Many of the popular document formats, such as portable document formats (.pdf), Microsoft Word documents (.doc), and plain text (.txt), were used during testing. By the end of this phase the DPS was completely functional and required only simple modifications.

3.4 Error Prevention & Security

A very important aspect of the DPS is security. The DPS handles real accounts and costs, therefore, the verification of users and confidentiality of their documents is a priority. To this end, various security features were incorporated into the DPS. The service uses PHP sessions in order to ensure that only registered users are accessing the system and that a user cannot enter prohibited areas. An example message generated by the DPS to illustrate this can be found in Appendix D. This is especially important given that the DPS has an administrator interface that has access to system and account parameters. Another security feature is the notification tables of the service. These tables not only notify users about job or system matters, but also function as logs that record all actions users (administrators and regular users) execute while using the service. These logs can only be accessed or modified by the Super Administrator to ensure that even administrators do not damage the system.

Given that the DPS is a web-based service that will be used by different users, it is also important to minimize or completely eliminate any possible system errors that can occur. To this end, there are various error prevention methods in place throughout the DPS. Examples of these methods are data verifications for user entries to ensure that input fields are not blank or incorrect, web page conditionals to ensure that all possible interface cases are handled properly, and database verification functions to ensure that database queries that do not produce results still return feedback to users. An example of the Wrong Page screen that prevents users from manually trying to enter areas that do not exist can be found in Appendix D. The service also provides feedback in most cases with a distinctive sentence format (italics and red font) and explanatory short paragraphs above some web pages. These error prevention methods and security measures not only make the DPS cleaner and more presentable, but minimize future problems and the need

for service downtime for maintenance. With the end of this chapter, we now have a basic idea of how the DPS was implemented and some of the component functionality. We can now understand with more ease the actual interfaces and procedures that will be explained next.

CHAPTER 4 – SYSTEM DESCRIPTION

4.1 Introduction

In the previous chapters we have learned about the background, implementation, and features of the DPS. In this chapter the components, features, and interfaces will be presented in more detail to ensure that we have a complete understanding of the present state of the DPS. Most of the illustrations that we will view in this chapter are actual screenshots or examples of the working DPS system, not mock-ups. There are three types of users that can interact with the system: regular users, administrators, and a super administrator. It is also important to understand that there are three basic types of printing accounts available to regular users: prepaid, flexible, and unlimited.

A prepaid account is an account where a user has a static balance equaling the amount paid in advance by the user for printing. A flexible account is similar to a prepaid account, but a user designates a certain monetary amount that the printing jobs can go over the balance before considering the user to be out of funds. This amount a user can go over is called a permitted overdraft or maximum billing. Any cost amount that goes over the balance will begin to accumulate in the user's account information until the overdraft is reached. The final type, an unlimited account, is an account where all costs continue to add up on the user's account information until the user pays the amount owed. This amount is called an accumulated bill. The final note to keep in mind is the concept of preflight and proofing mentioned in chapter 2. A user has the option of obtaining a preflight test or a proofing copy of a submitted job before printing the actual job contents. The information that will follow begins with the interfaces of each type of user that interacts with the system, followed by the database model, and finally ending with the workflow engine.

4.2 Common Interfaces

All users of the DPS will interact with a few basic web pages that are the same no matter the user's role. These pages are the Log-In, Log-Out, Registration, and Password

Recovery pages. The Log-In page (see Figure 3) is where users confirm their identity to the system by entering their e-mail and password match. This page is the first page in the service and links to the interface appropriate to each user role by verifying a special column on the *customer* database table called *isadmin*. All database tables will be discussed later on in the Database Model section. The *isadmin* column is basically a Boolean value that identifies a user as an administrator or regular user and enables the correct menu interface for said user. If an administrator is also the first record on the *customer* table (*customerid* = 1), this user is identified as the super administrator. Beneath the log-in text boxes are the links *New Customer?* and *Forgot your password?*. These links take the user to the Registration page or Password Recovery page respectively. At any time throughout the DPS interfaces, a user can click on the Digital Publishing Service logo in order to be taken back to the Log-In page or initial menu interface if already logged on.

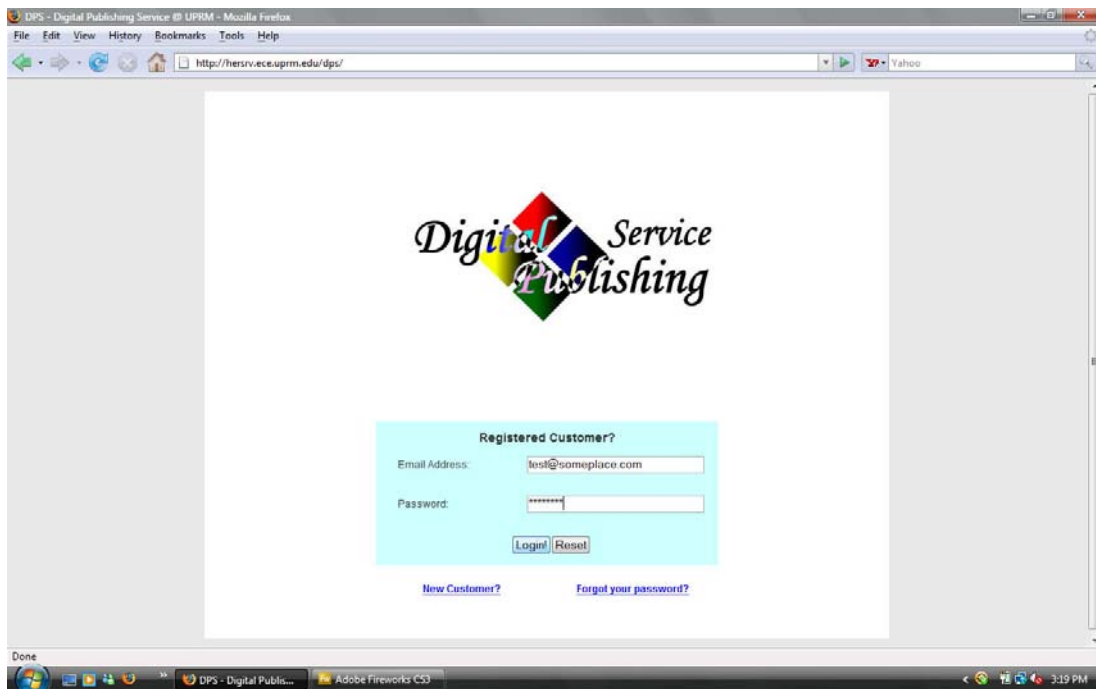


Figure 3: Log-In web page.

An existing user that does not remember the password can visit the Password Recovery page (see Figure 4) in order to receive an e-mail with their appropriate login

and password combination. The user must enter the e-mail used at the moment of registration for the e-mail to be sent.

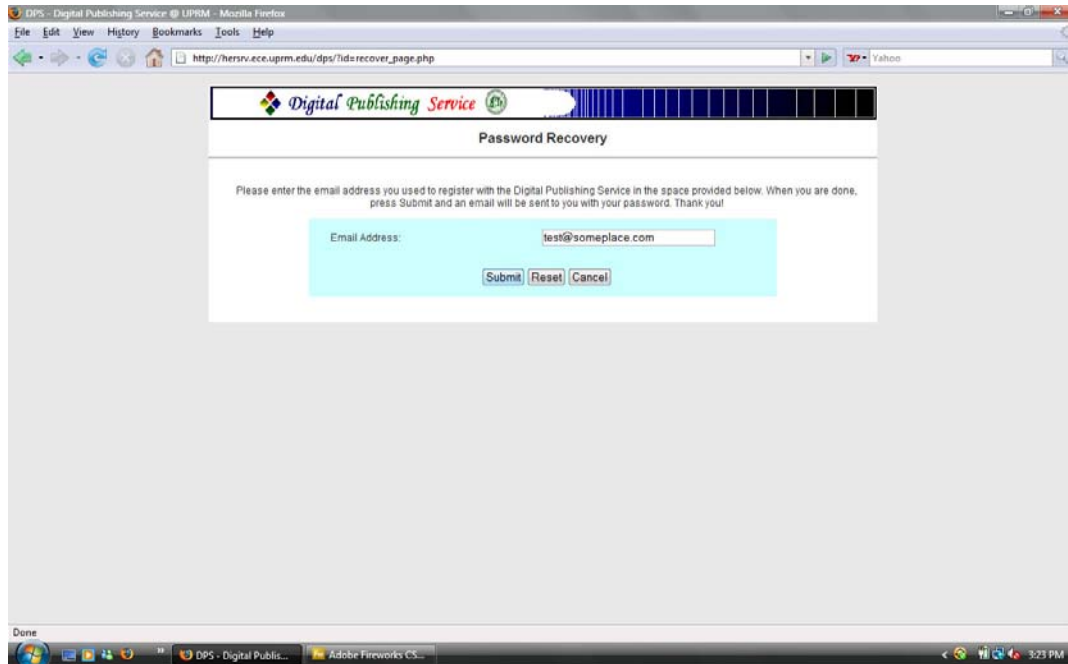


Figure 4: Password Recovery web page.

A new user must visit the Registration page (see Figure 5) and enter the appropriate information in the text fields provided. The information fields required are: the user's full name, e-mail address, password, full address, account type desired, and specific account information. The account type and information text fields (see Figure 6) change dynamically according to the account type select due to the fact that each of the three account types requires different information. Error prevention features, discussed in section 3.4, verify that the information given is correct.

Digital Publishing Service

Customer Registration

*Email Address:

*Confirm Email Address:

*Password:

*Confirm Password:

*First Name:

Middle Initial:

*Last Name:

Maiden Name:

*Postal Address:

*City:

Figure 5: Registration web page user information.

*City:

*State:

*Country:

*Postal Code:

*Type of Account:

*Department/Unit:

*Account Name:

*Account Number:

*Person Responsible for Account:

*Initial Balance:

*Permitted Overdraft:

Note: Fields marked with an asterisk (*) are required.

Figure 6: Registration web page account information.

User registration is a four step process that is illustrated in Figure 7. After a user submits the registration information, a message will be sent to the user's e-mail address

with a confirmation link. When the user enters the link, a Boolean column in the database table *customer* called *emailconfirm* will be set to true if the link's code is correct. After this, an administrator must edit the user's information and confirm that the user can use the account. Finally, the administrator must create a printing account for the user, otherwise the user cannot use the printing tools provided by the DPS.

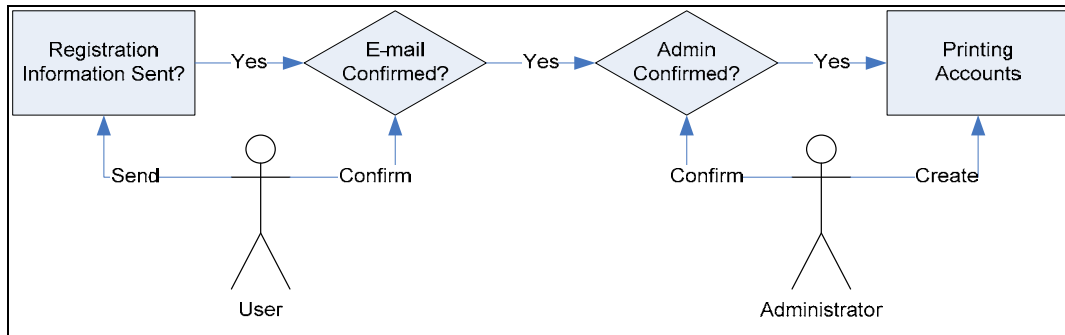


Figure 7: Registration steps diagram.

When a user enters the service, a session key is created to ensure system security. This key uses the default PHP automatic timeout limit of 1440 seconds or 24 minutes. When the user exits the service via the Log-Out page (see Figure 8), this key is destroyed. The key can also be destroyed if a user cleans out their browser's cache while a session is still running. This is a standard practice with PHP sessions in order to ensure that unauthorized users do not enter sensitive areas of web pages and to carry important information between related web pages. Clicking the designated link on the bottom of the page or the DPS logo will take the user once again to the Log-In page.

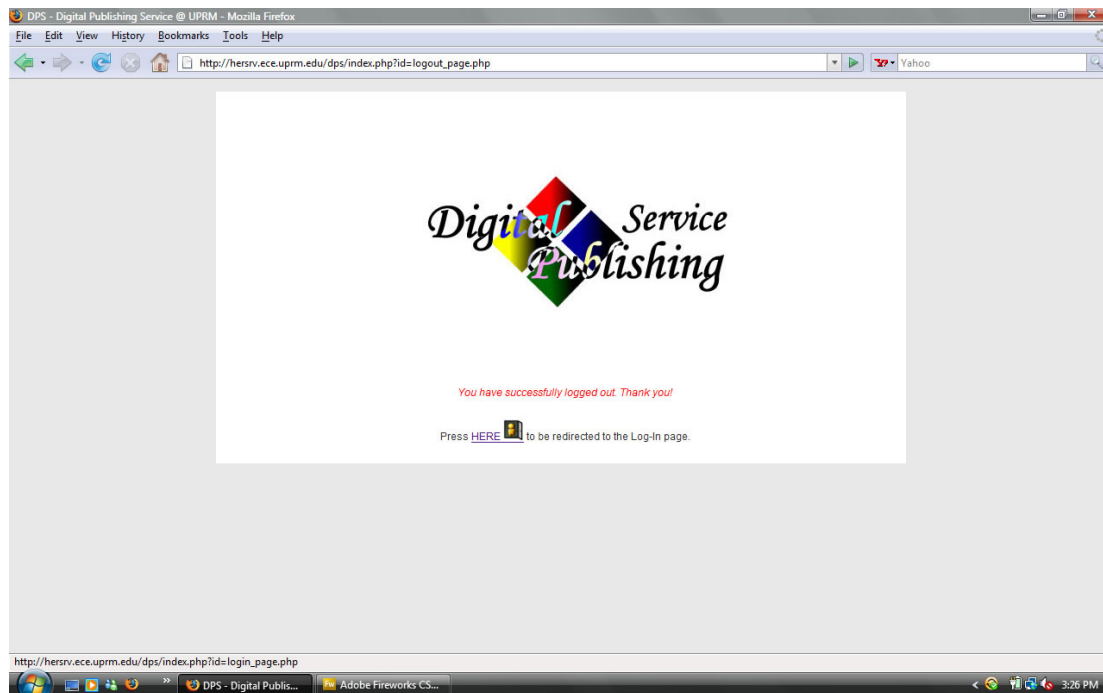


Figure 8: Log-Out web page.

4.3 Administrator Interfaces

Once a user has confirmed that he or she is an administrator, the system will determine whether or not the user is the super administrator or not. As stated earlier, this is done by verifying if the user is the first record on the *customer* database table. Note that the user must also have been found to be an administrator by verifying the *isadmin* value on the table. If both conditions are not met, the super administrator will not be found. This rule was established in this way because there must always be at least one user on the mentioned table and it allows modifications to the super administrator's information by directly modifying the table information in the future if needed. Administrators all have a beginning menu interface or Home page as seen in Figure 9. The left-hand menu allows administrators to modify the system's media, printers, customer information, settings, and submitted jobs. We will now learn about each option in more detail.



Figure 9: Administrator Home page and menu.

The first option and most important part of user registration is the Customers button. This option allows administrators to find users and modify their accounts. The Customer Search screen (see Figure 10) has a very simple search engine that can find a user by inputting any part of the desired user's e-mail or full name. The two top links provide a table of all users in the system or any users that require administrator confirmation in order to use the service respectively. Once a user or collection of users are found, an administrator can select a desired user and either edit the user's information, edit the user's accounts, or send the user a notification by using the button provided at the bottom of the screen (see Figure 11). Editing a user's information can be used to change the user's password, confirm that the user can now use the system, or change whether the user is an administrator or not. This last option is a clear reason why there must always be at least one administrator (the super administrator) in the system's database. Without an initial administrator, the interface cannot be accessed and other administrators cannot be created.

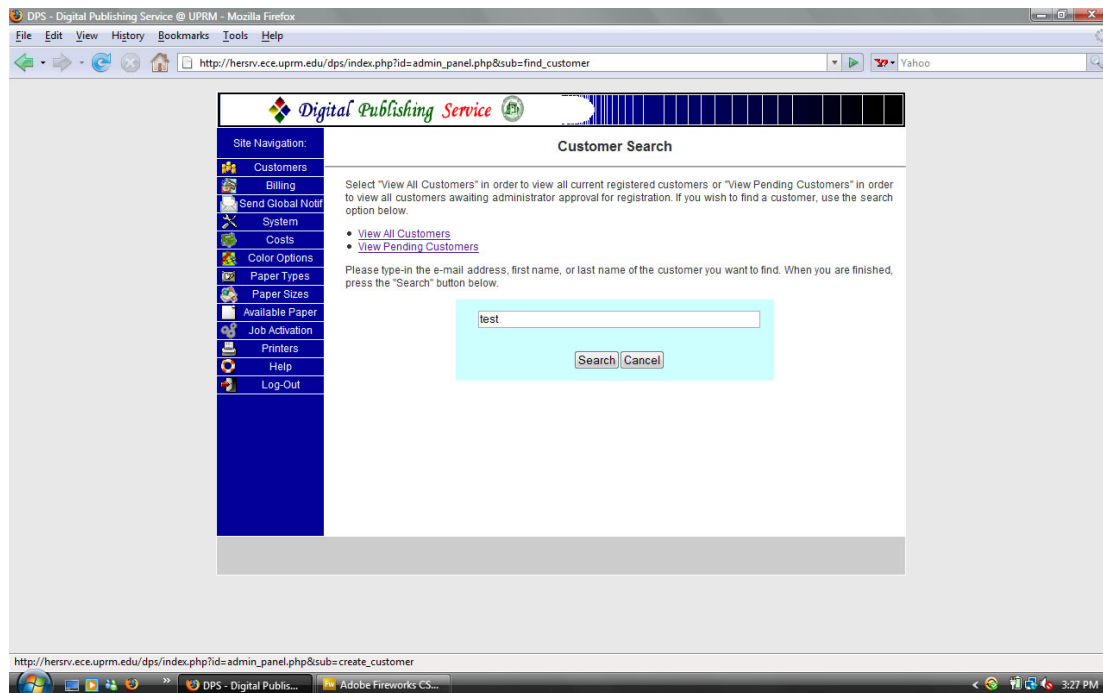


Figure 10: Customer Search screen.

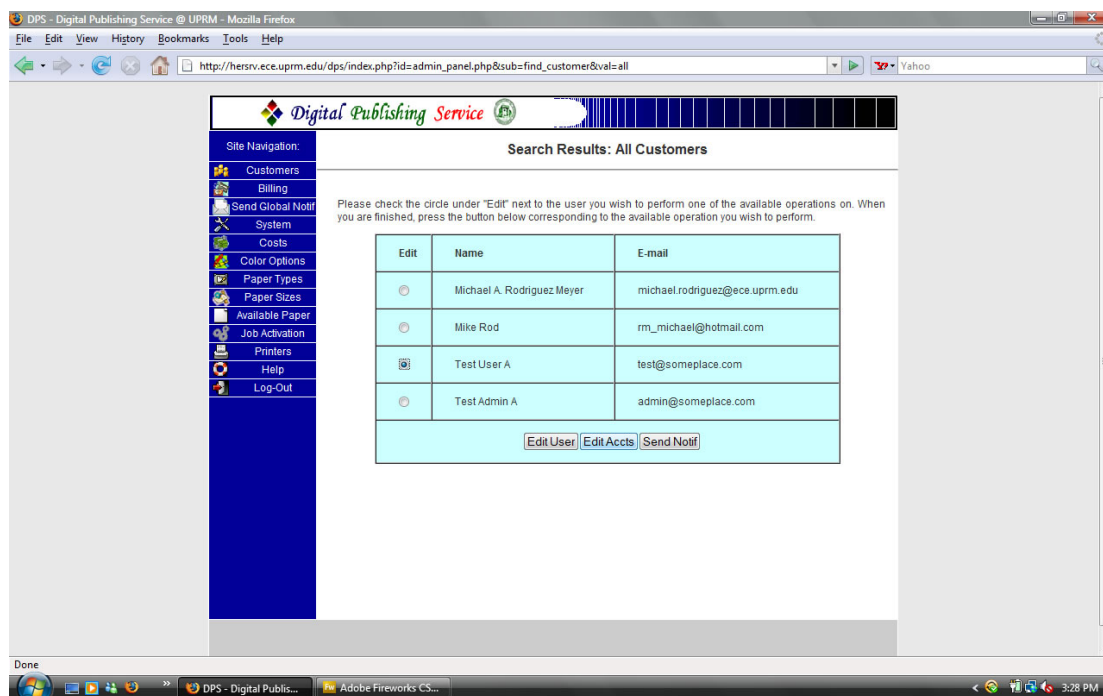


Figure 11: Sample search results (using All Customers link).

A notification can be sent to a selected user by pressing the Send Notif button. On the Notification screen (see Figure 12), the administrator must select the type of

notification, select the status if the message is in relation to a job, and enter the desired message in the space provided. The three types of notifications available are Error, Status, and System. Error notifications occur whenever a submitted job produces a system error that prevents it from proceeding forward. Status notifications inform users of changes in a submitted job's status, such as entering the preflight engine or printing the job. System notification occur when a submitted job or the system requires a user's interaction in order to proceed, such as waiting for approvals or requiring additional funds to be deposited. Notifications are often created automatically by the DPS during the job workflow and sometimes remain hidden from users, such as log entries when system parameters are changed.

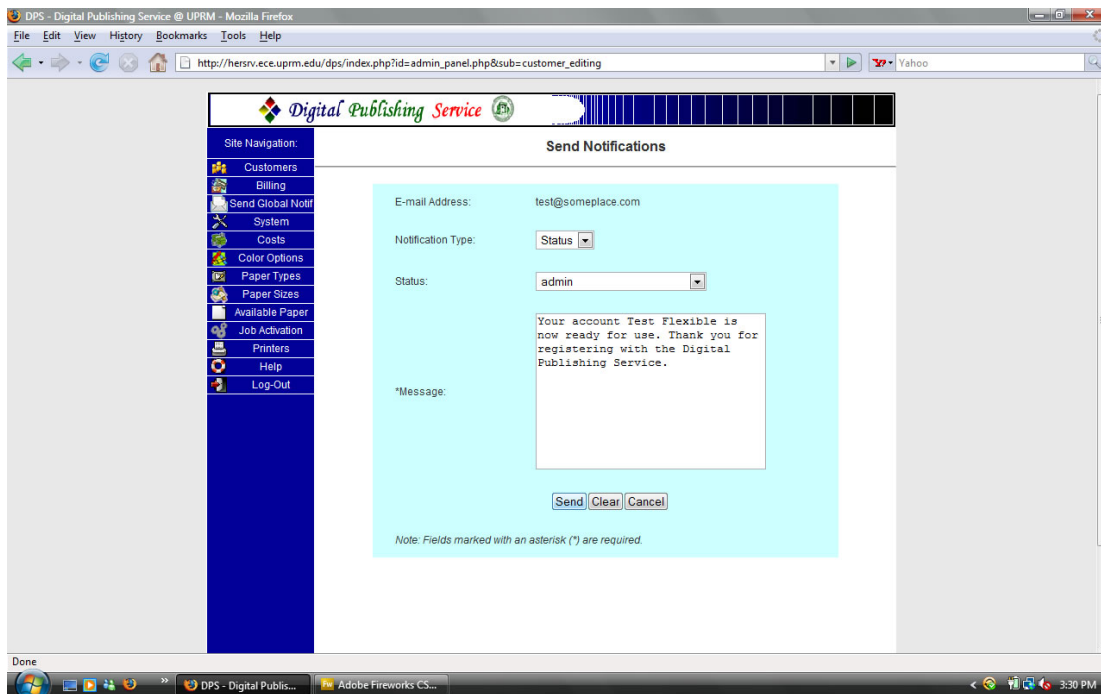


Figure 12: Single user Notification screen.

A necessary step after registration and confirmations is the creation of a user's accounts. Once an administrator clicks the Edit Accts button after selecting a user, a list of the user's accounts is displayed if any. Clicking the Create button will take the administrator to the Create Accounts screen (see Figure 13) where all the appropriate account information must be entered. As with registration, the information text fields

dynamically change depending on the account type selected. The max billing field is the permitted overdraft explained at the beginning of this chapter. The accumulated billing denotes the initial accumulated bill for a flexible or unlimited account. The cost type is a surcharge that is added to preflight, proofing, and printing costs. The cost calculation of a submitted job is explained later on in this chapter. Editing accounts is a similar process with similar fields, except that the account type will remain static and all fields will display the present account values.

Figure 13: Create Accounts screen.

The next administrator option is Billing. This option takes the administrator to the Account Billing screen (see Figure 14) where accounts with accumulated bills (flexible and unlimited) are shown. The administrator can then generate a billing receipt (see Appendix C for an example) in order to charge a user for funds owed or reset the selected account to an accumulated balance of 0.00 again. Performing any of these choices simply requires the administrator to select Receipt or Reset next to the account desired. A receipt can only be obtained for one account at a time as well as any resetting.

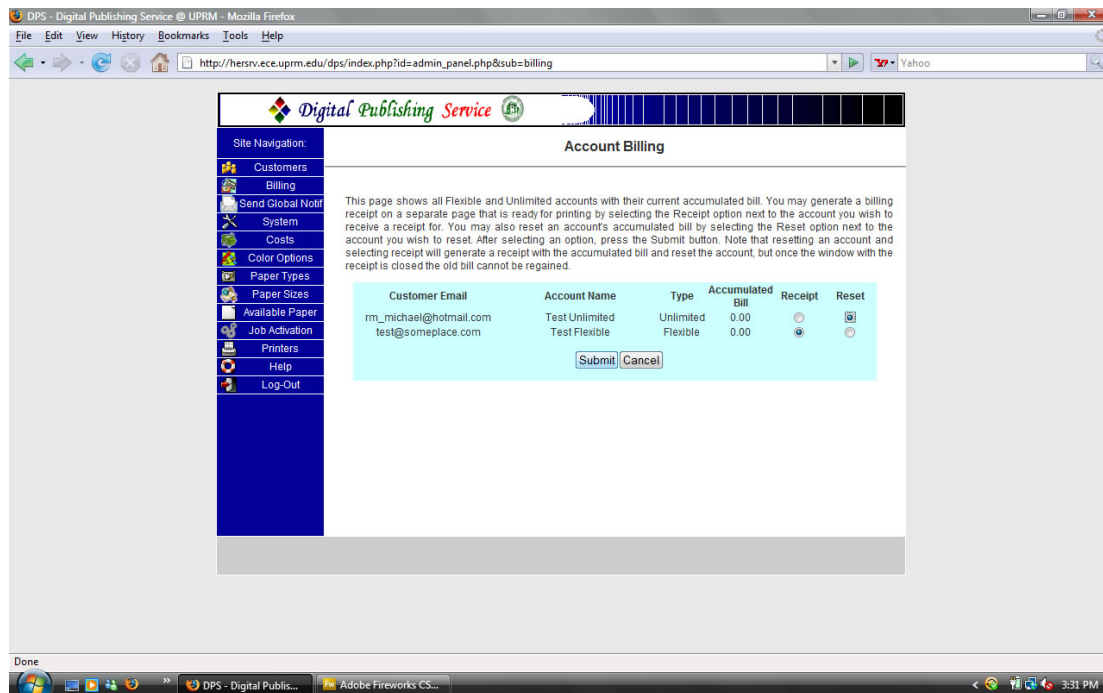


Figure 14: Account billing screen.

The next option, Send Global Notif, is similar to the tool for sending notifications to individual users. The main difference is that all valid users identified in the system that are not administrators will receive the notification created. Other details of this option, such as the notification type, function in the same way as individual user notifications. The System option takes the administrator to the Edit System Configuration screen (see Figure 15) where administrators can set the system cost for preflight, proofing, and printing. These costs are the initial costs added to any job submitted. As mentioned earlier, the calculation of a job's costs will be explained later on in this chapter. The max job threads value was meant to be a limitation on the amount of job system threads that could be executed at the same time. Given that the DPS was changed to PHP and system threads are not a limiting factor anymore, this particular value is no longer used. We decided to leave the value though in case future modifications to the DPS may require it.

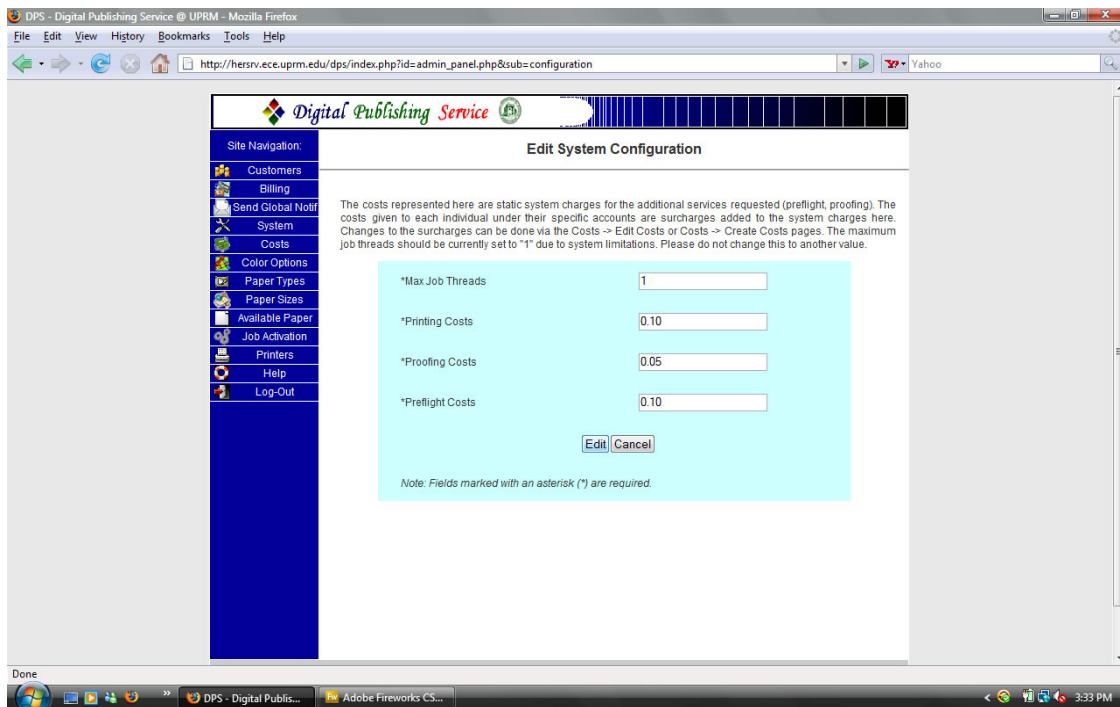


Figure 15: System Configuration screen.

The Costs option provides access to two screens related to the creation and editing of surcharges associated to each user account. The cost creation screen (see Figure 16) requires an administrator to enter the description or name of the new surcharge and the preflight, proofing, and printing costs associated to it. The cost editing screen (see Figure 17) displays all the present surcharge types in the system and their associated costs. An administrator can modify a specific preflight, proofing, or printing cost of a desired surcharge by entering the new amount and clicking submit. Most of the media (color and paper) options that follow have a similar layout and functionality. We will still view each particular screen in order to understand specific values associated to each option.

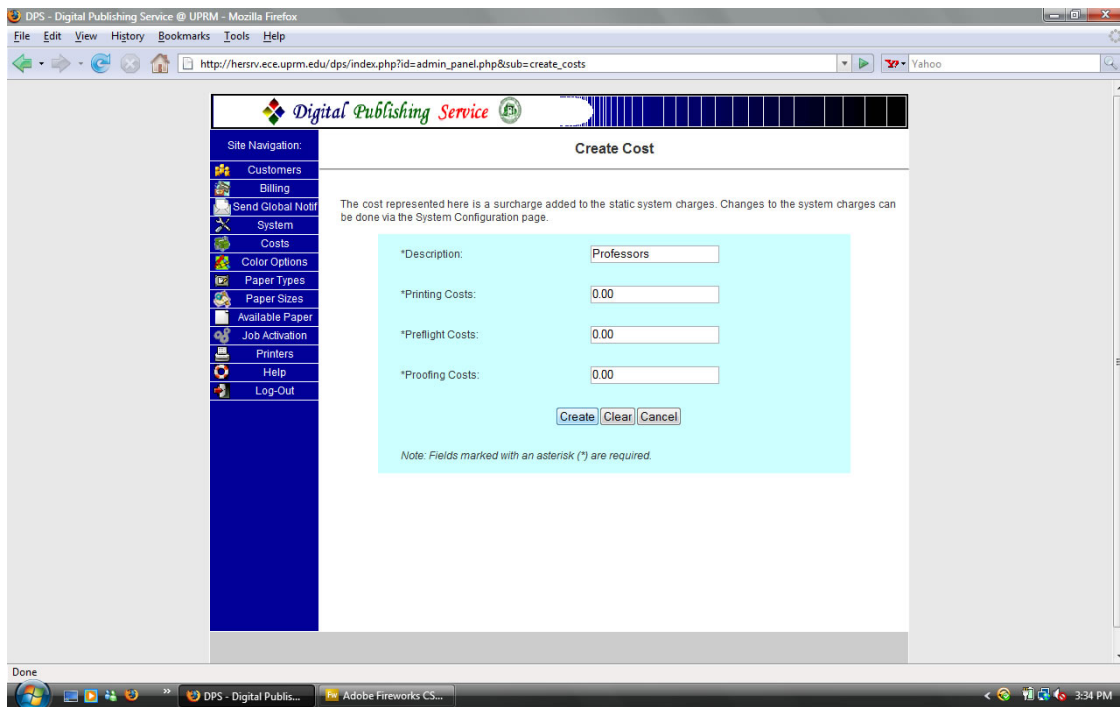


Figure 16: Create Cost screen.

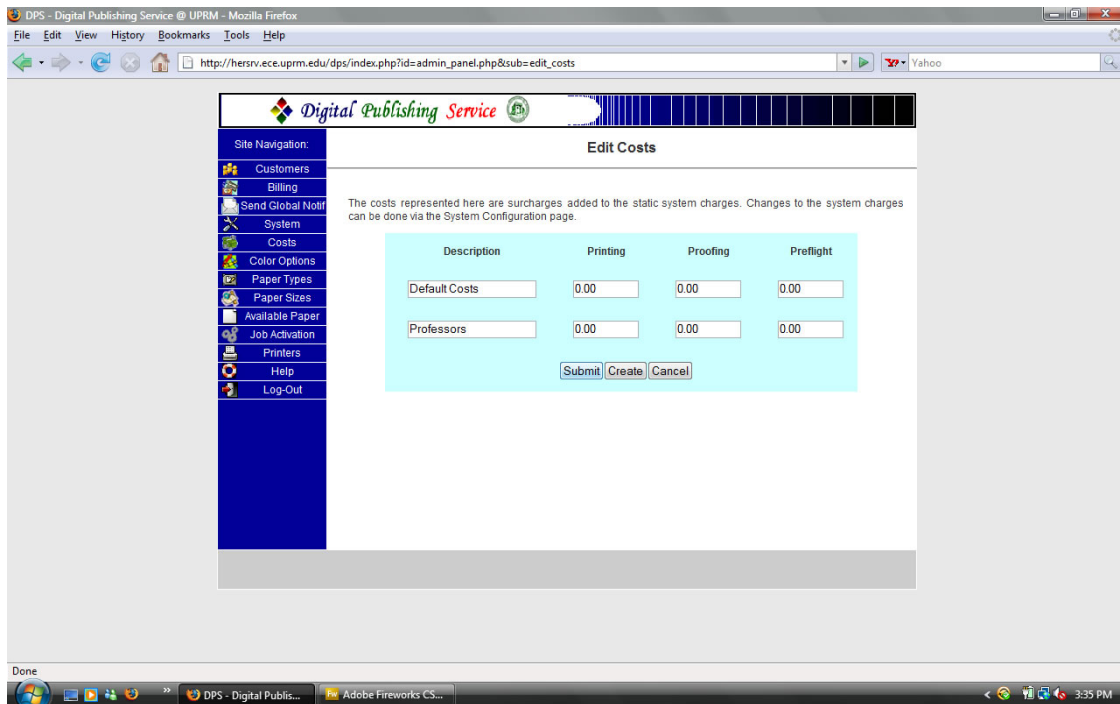


Figure 17: Edit Costs screen.

The Color Options screens allow an administrator to create or edit the color schemes available to users when submitting a job, their additional cost if any, and whether or not they are currently active. Users can only select active color schemes when submitting jobs, so the last value is important. As with Costs, the create color option screen (see Figure) provides an administrator with an interface for creating a new scheme, while the edit color options screen provides an interface for viewing the value of present color schemes in the system.

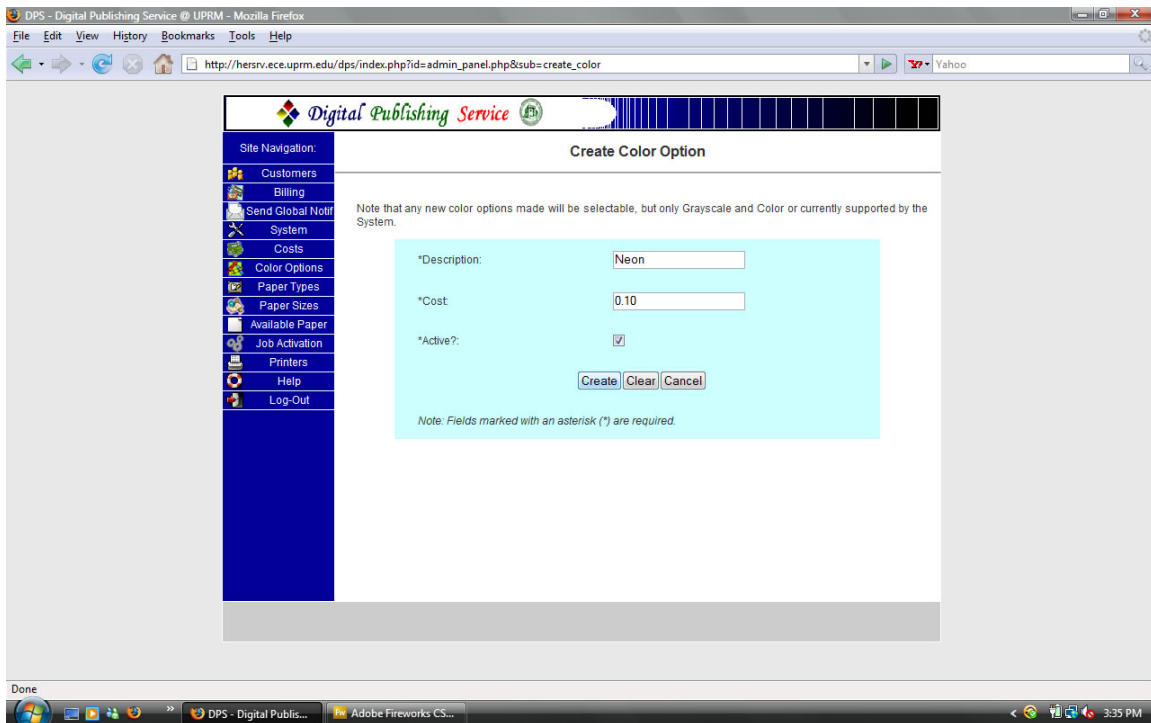


Figure 18: Create Color Option screen.

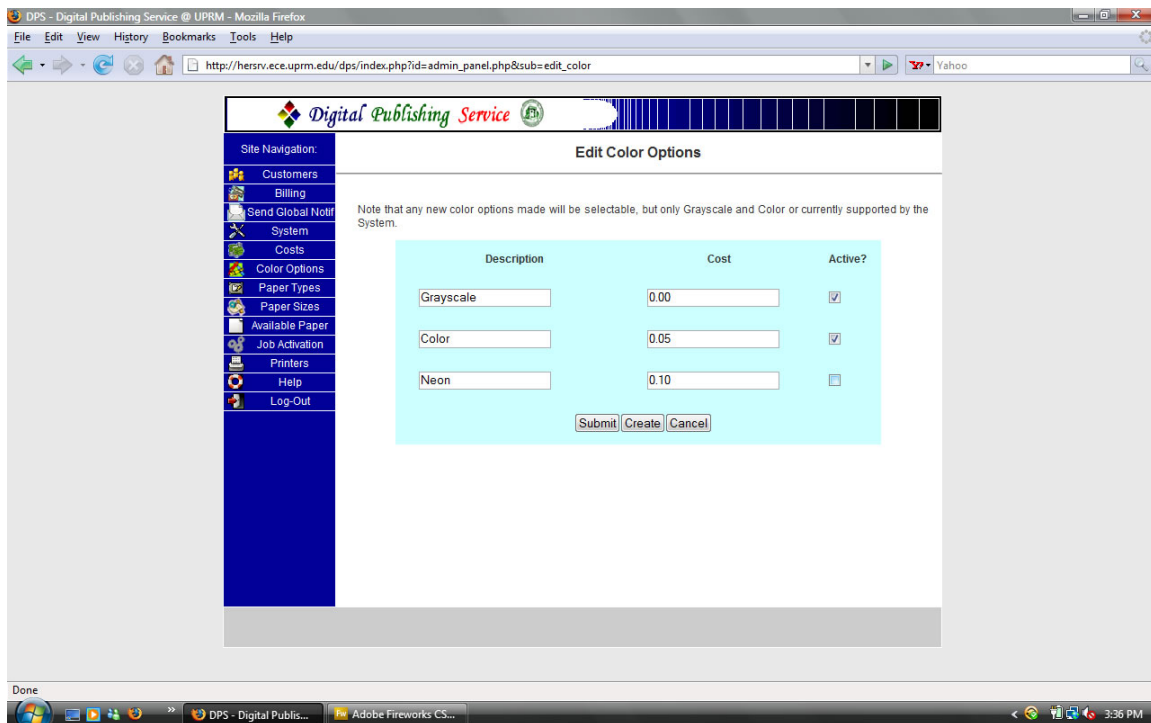


Figure 19: Edit Color Options screen.

The next two options, Paper Types and Paper Sizes, are similar to Color Options. The difference is that these two options pertain to the paper loaded in the printers. The create paper type screen (see Figure 20) will create a new paper type, such as glossy or plain, and denote the cost for using the new type if any. The edit paper type screen (see Figure 21) allows an administrator to change the description or cost of paper types already present in the system. In similar a fashion, an administrator can create a new paper size, such as letter or legal, with the create paper size screen (see Figure 22) or change existing paper size parameters with the edit paper size screen (see Figure 23). We must note however, that the paper type and size options do not have an active value box. We will understand why after viewing the next option, Available Paper.

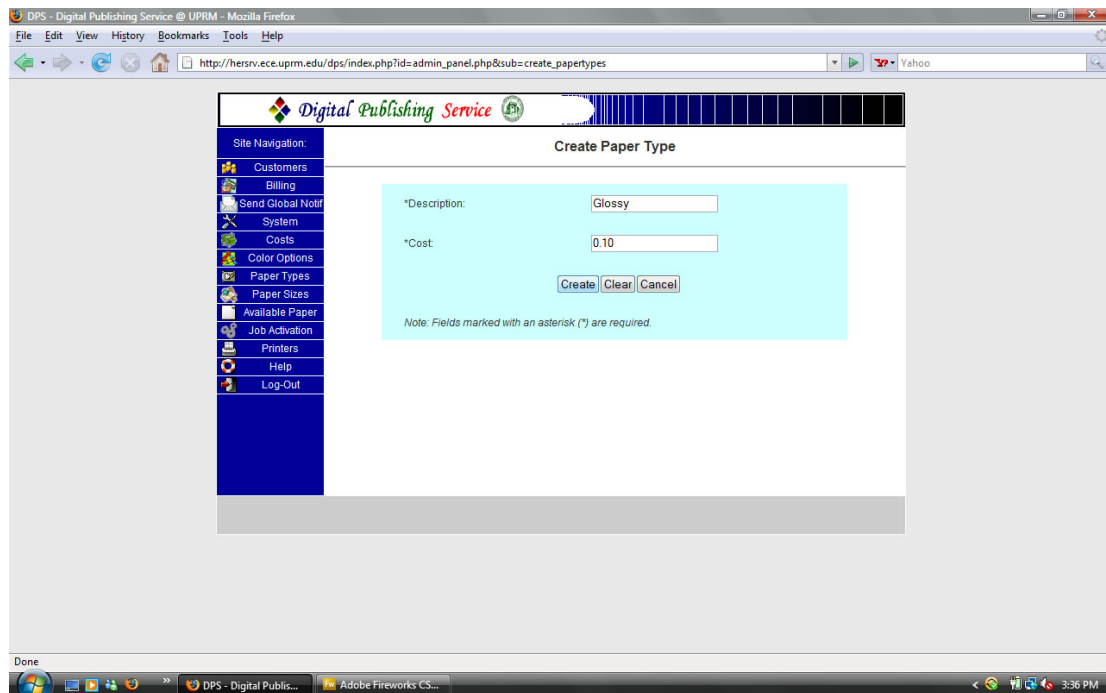


Figure 20: Create Paper Type screen.

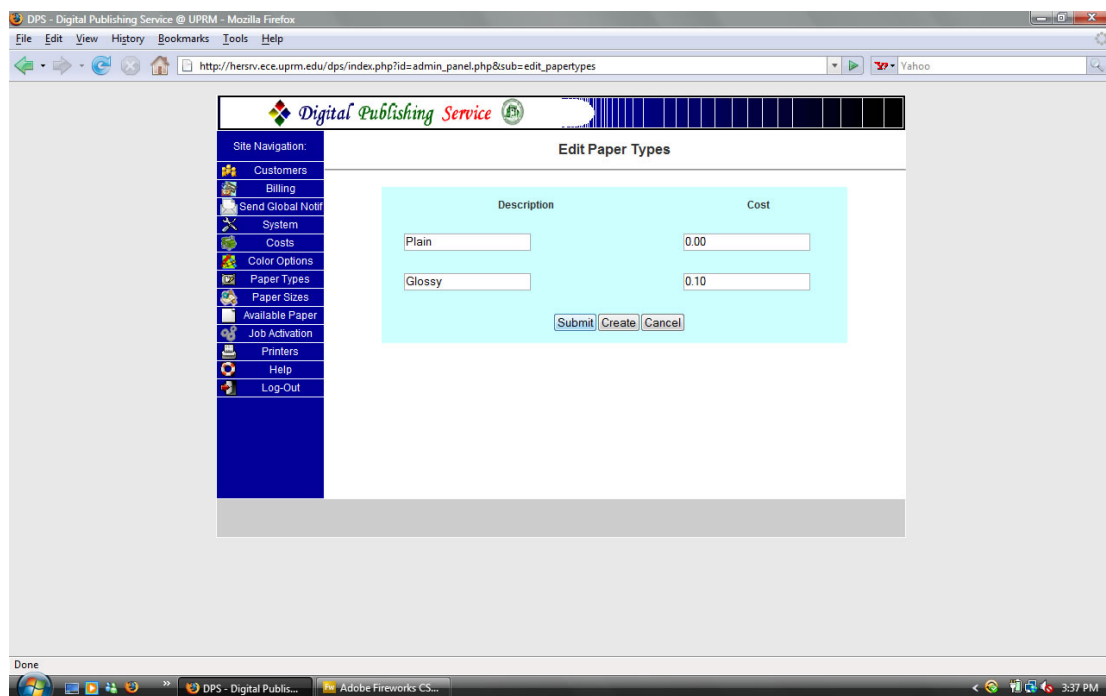


Figure 21: Edit Paper Types screen.

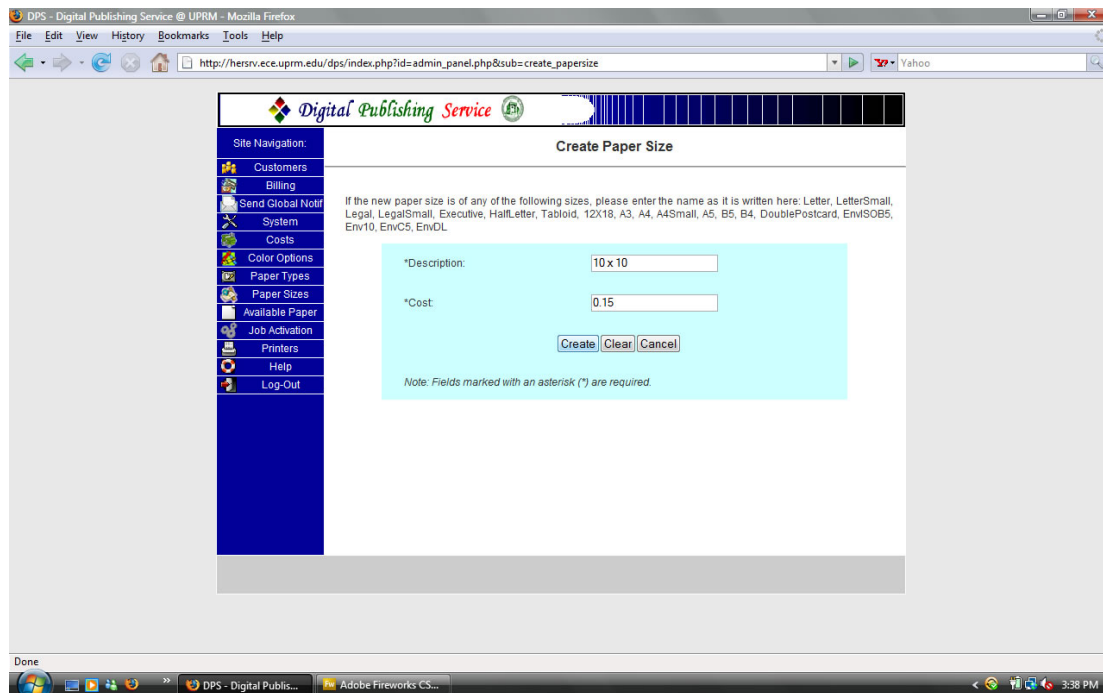


Figure 22: Create Paper Size screen.

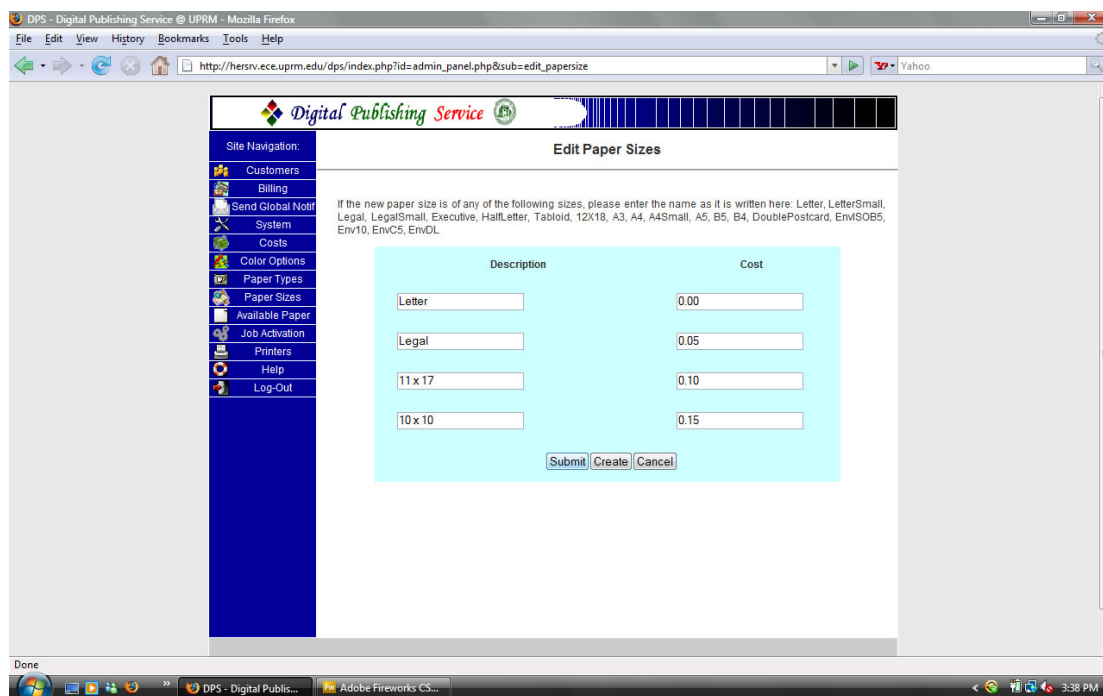


Figure 23: Edit Paper Sizes screen.

The next option, Available Paper, allows administrators to create and modify the paper type and size combinations that are currently available for use. These combinations

denote the paper media that will be loaded into the printer trays. Similar to earlier options, the create new paper screen (see Figure 24) is where an administrator can add a new paper combination by selecting the paper type and size from the available drop down menus. These menus are created with the information present in the system that was created with the Paper Types and Paper Sizes options. In this particular option, an administrator must also specify if the paper combination is available for use, otherwise it will not appear later on for users to select during document submissions. The edit available paper screen (see Figure 25) is where an administrator can view the present combinations in the system and change whether they are active or not.

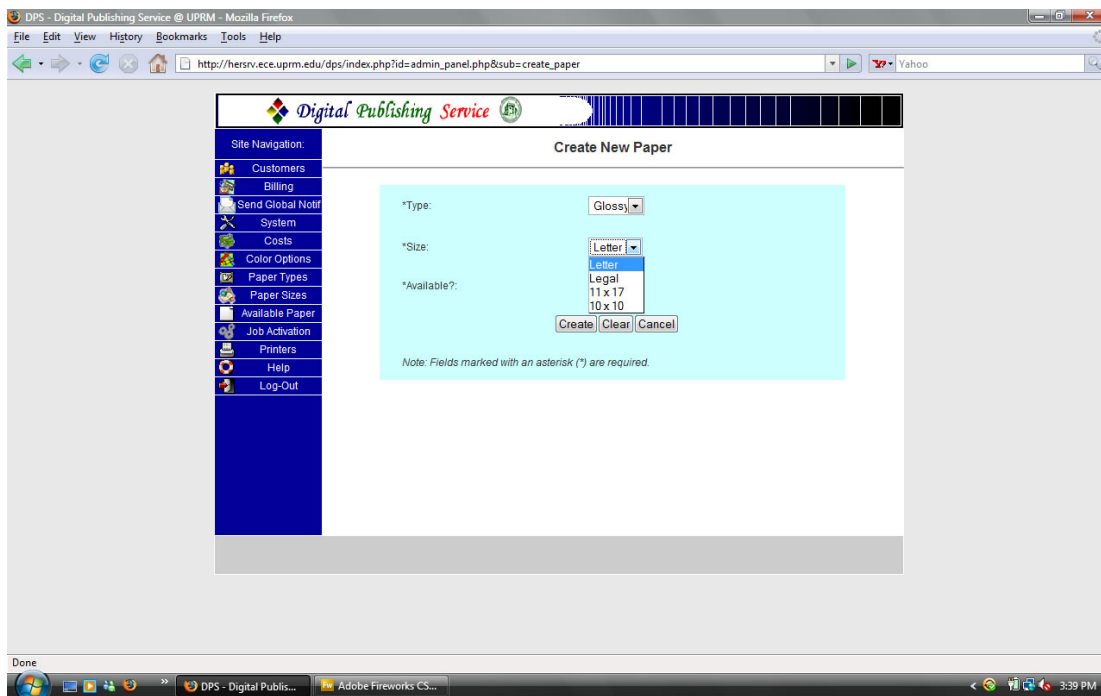


Figure 24: Create New Paper screen.

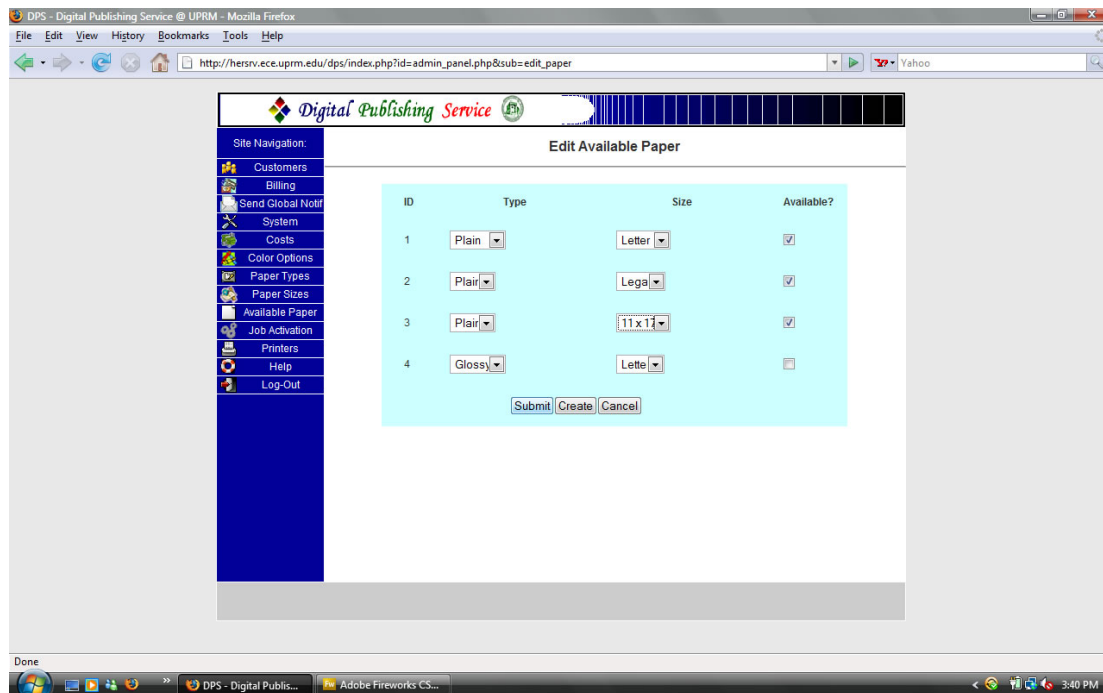


Figure 25: Edit Available Paper screen.

The next option on the menu is the Job Activation option. By entering the job activation screen (see Figure 26), an administrator can view all unfinished jobs that are present in the system, view the user and account used for the submission, and change the job's status to a later or earlier stage. Table 2 contains all the possible status names and their meaning. Whenever a job's status is changed, the workflow engine is often called again in order to process the new job stage.

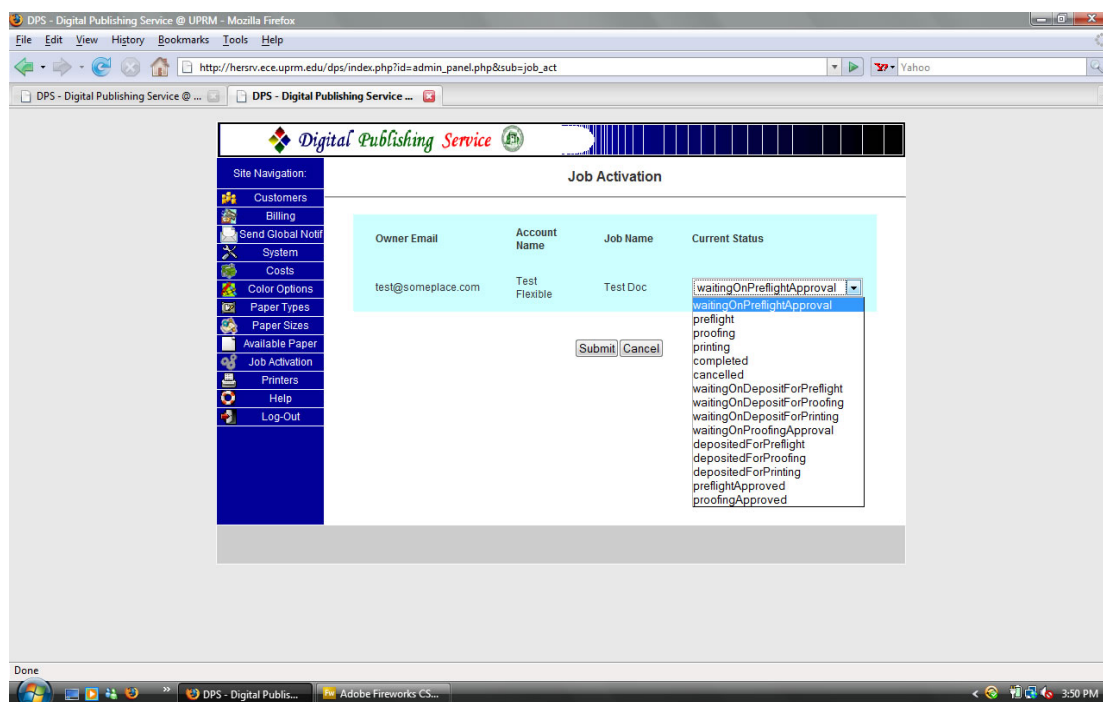


Figure 26: Job Activation screen.

Table 2: Job status descriptions and names.

Job Status Name	Status Description
Preflight	Indicates that the job has begun to enter the preflight engine.
Proofing	Indicates that the job has begun to make a proofing copy of the job.
Printing	Indicates that the job has begun to make the actual print copy of the job.
Completed	Indicates that the job was printed successfully.
Cancelled	Indicates that the job was cancelled.
WaitingOnPreflightApproval	The preflight results were sent and the results require approval in order to continue.
WaitingOnProofingApproval	The proofing copy was completed and the copy requires approval in order to continue.
WaitingOnDepositForPreflight	Preflight stage halted due to lack of funds.
WaitingOnDepositForProofing	Proofing stage halted due to lack of funds.
WaitingOnDepositForPrinting	Printing stage halted due to lack of funds.
PreflightApproved	Preflight results approved and job can continue.
ProofingApproved	Proofing copy approved and job can continue.
DepositedForPreflight	Funds received and preflight stage can continue.
DepositedForProofing	Funds received and proofing stage can continue.
DepositedForPrinting	Funds received and printing stage can continue.

The next option available is the Printers option. This option has three screens that pertain to the printers communicating with the system. The create printer screen (see Figure 27) is where an administrator adds the name, network IP number, amount of trays, and active status of new printers. The edit printer screen (see Figure 28) allows an administrator to change printer names, status, and network IP number for printers currently on the system. The active status of a printer denotes whether the printer will be considered for use by user's jobs and the number of trays is used to manage which available paper is loaded in the printers. Although the information given in these screens is not verified by the system to ensure if printer names are correct and actually communicating with it, the names of the printers are used directly during the workflow execution. Therefore, it is important that the printer names be the same as the names provided when the printers are added via the CUPS interface. The final screen, printer trays management (see Figure 29), generates tables automatically based on the active printers and their number of trays. This screen allows administrators to specify what paper media was placed in each printer tray. The paper media available in each drop down list will be taken only from active paper found in the Available Paper option described earlier.

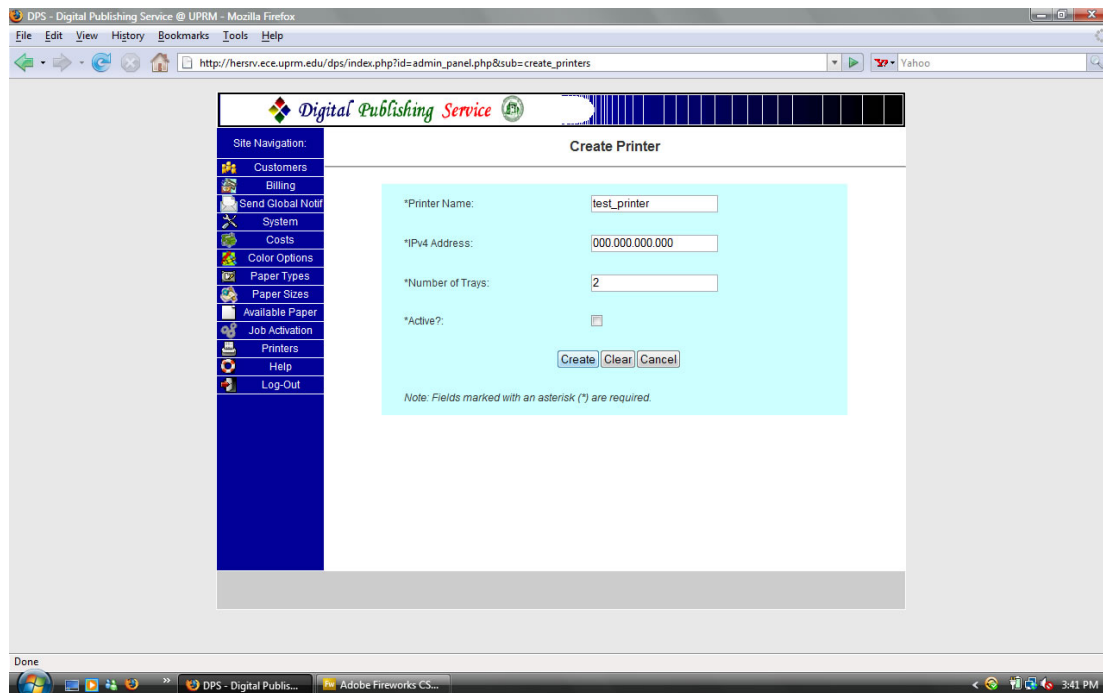


Figure 27: Create Printer screen.

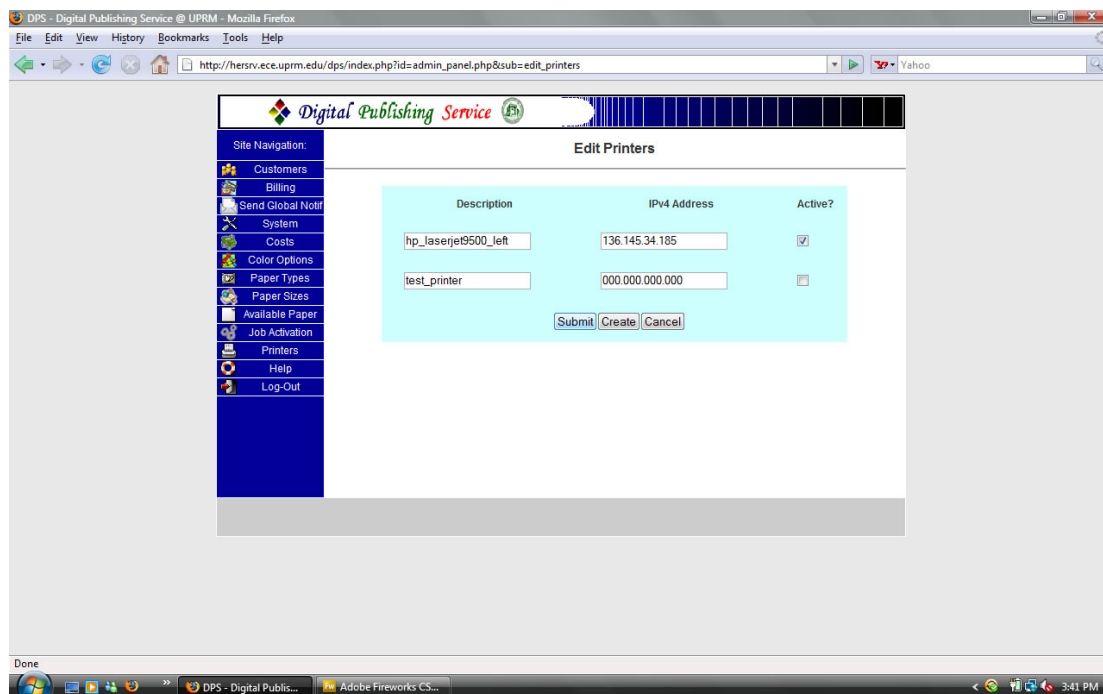


Figure 28: Edit Printer screen.

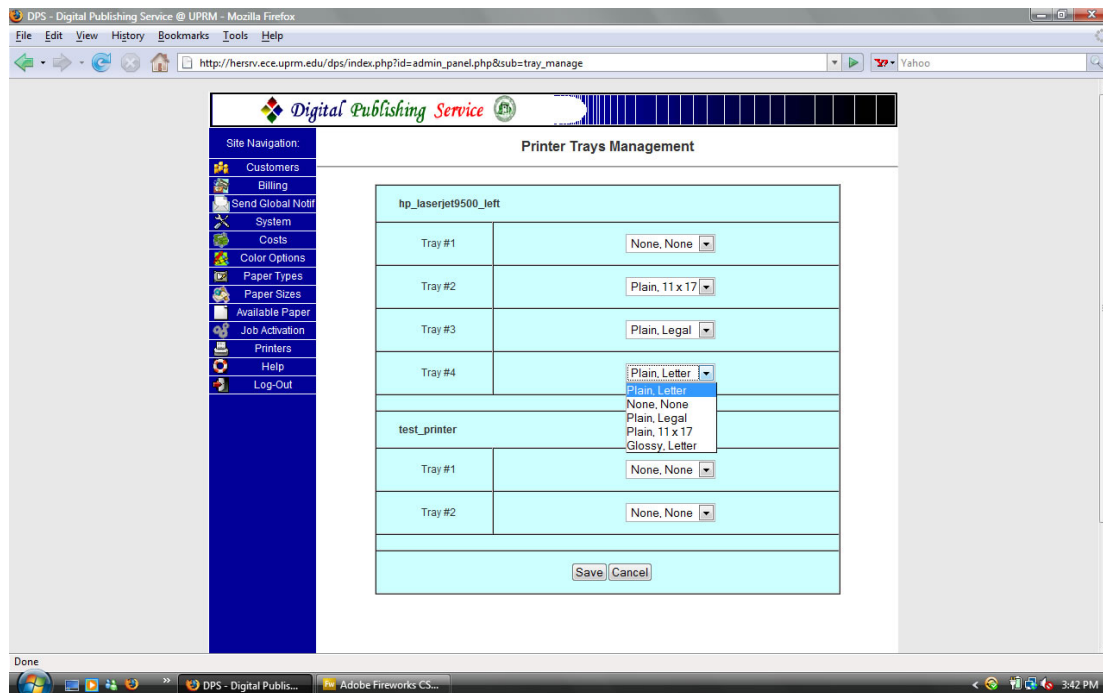


Figure 29: Printer Trays Management screen.

The last option, Help, contains three tools with information about the DPS in order to guide users or provide help. The FAQ screen contains Frequently Asked Questions or basic questions that can arise during use of the DPS and the User Manual and Administrator Manual buttons provide an information manual that can be downloaded related to the regular user's or administrator's interfaces and processes respectively. With these interfaces explained, we can now move on to the additional interfaces the Super Administrator has access to.

4.4 Super Administrator Interfaces

The unique Super Administrator has a slightly different left-hand menu compared to regular administrators (see Figure 30). Although all the options described during the administrator interfaces section are available, there are two new options: Activity Logs and Data Removal. The Activity Logs option provides a screen (see Figure 31) where the super administrator can review all the stored system notifications separated by account, customer, error, status, and system notification categories. Account notifications are all

notifications that are associated to jobs that have been submitted and therefore, associated as well to user accounts. Customer notifications are notifications that are usually sent by administrators to users, but may not relate to any particular job. Error, status, and system notifications are notifications of each respective type that have not been sent by administrators or have no job associated with them.

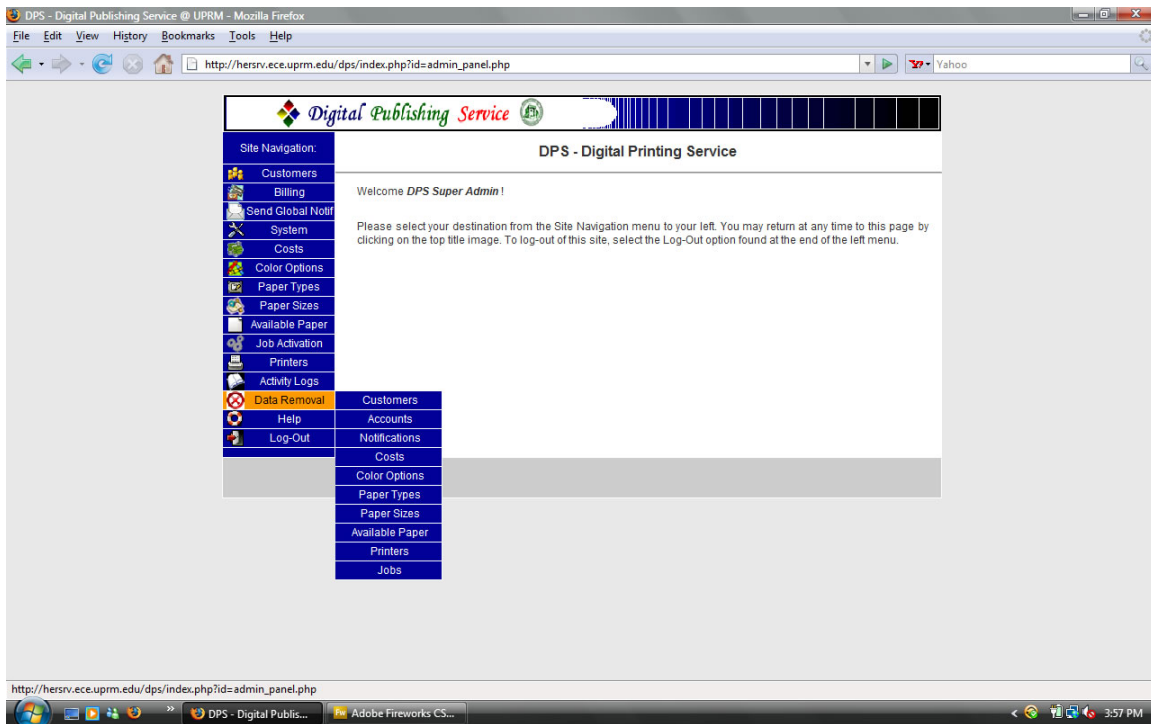


Figure 30: Super Administrator Home page and menu.

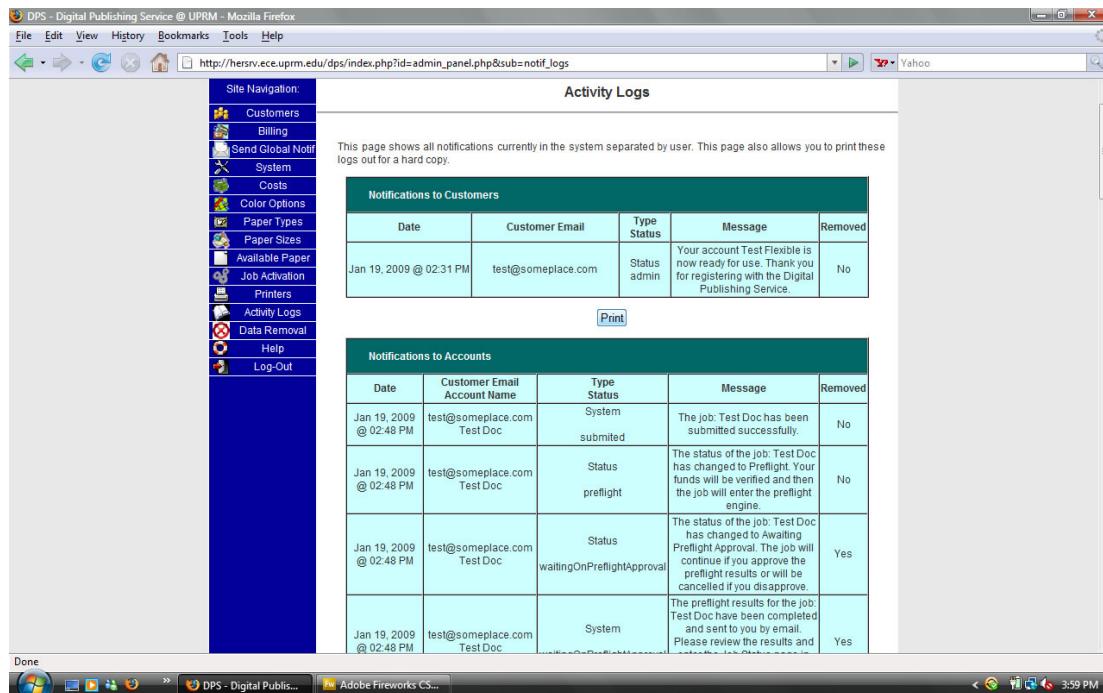


Figure 31: Activity Logs screen.

All the diverse log tables generated have a Print button below them that opens a new window (see Figure 32) with only the selected table. This new window permits printing a specific table without the clutter of other tables or the rest of the super administrator interface. When the new window is opened, the user's default printer window is also opened for quicker printing. At the very bottom of the activity logs screen is a Print button for all the logs at once. This will open a new window with only the log tables in order to print the information without the rest of the super administrator interface components.

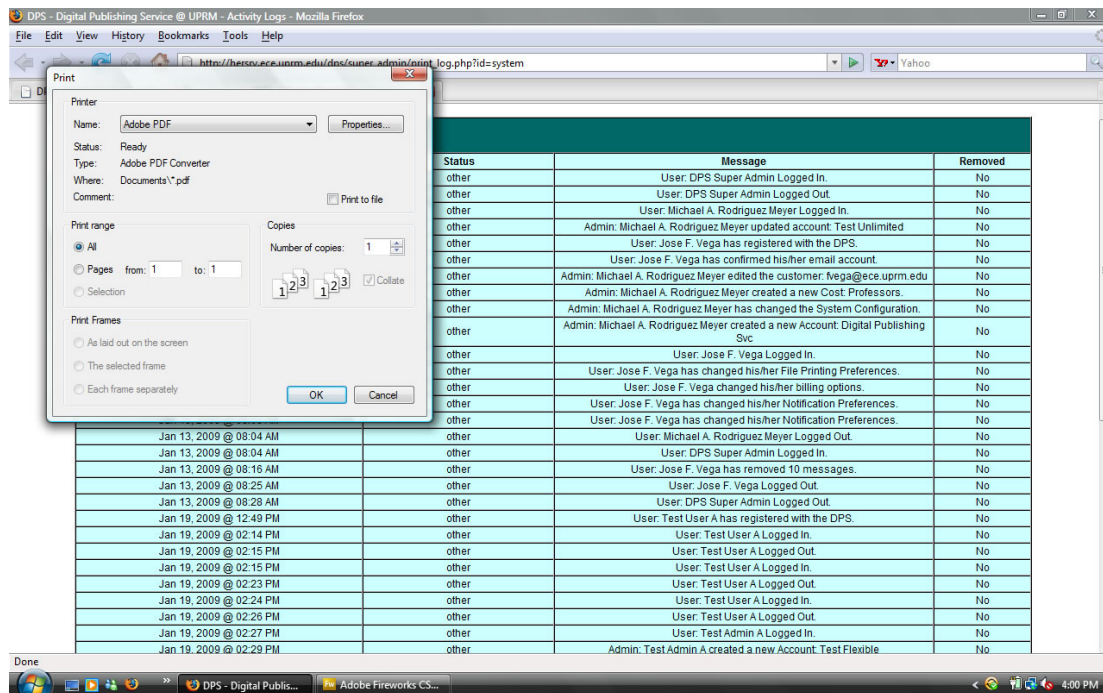


Figure 32: Log printing example.

The Data Removal option contains 10 screens for removing different types of information found in the system database, such as customers, jobs, color options, or notifications. All of these screens work in a similar fashion by showing the super administrator a table with the data that can be removed followed by a box to select the desired items. Some screens are simple, such as the notification removal (see Figure 33). However, others warn the super administrator that removing data may also remove associated data found elsewhere on the system, such as in the case of removing customers (see Figure 34). The media parameter removal screens (see Figure 35) warn that any users with preferences containing the parameter to be removed will result in the preference settings changed automatically to the default values set by system for that particular parameter. This error prevention method ensures that errors do not occur due to information removal and verifies that a default value is always in the system. These default values do not appear on the removal lists for obvious reasons.

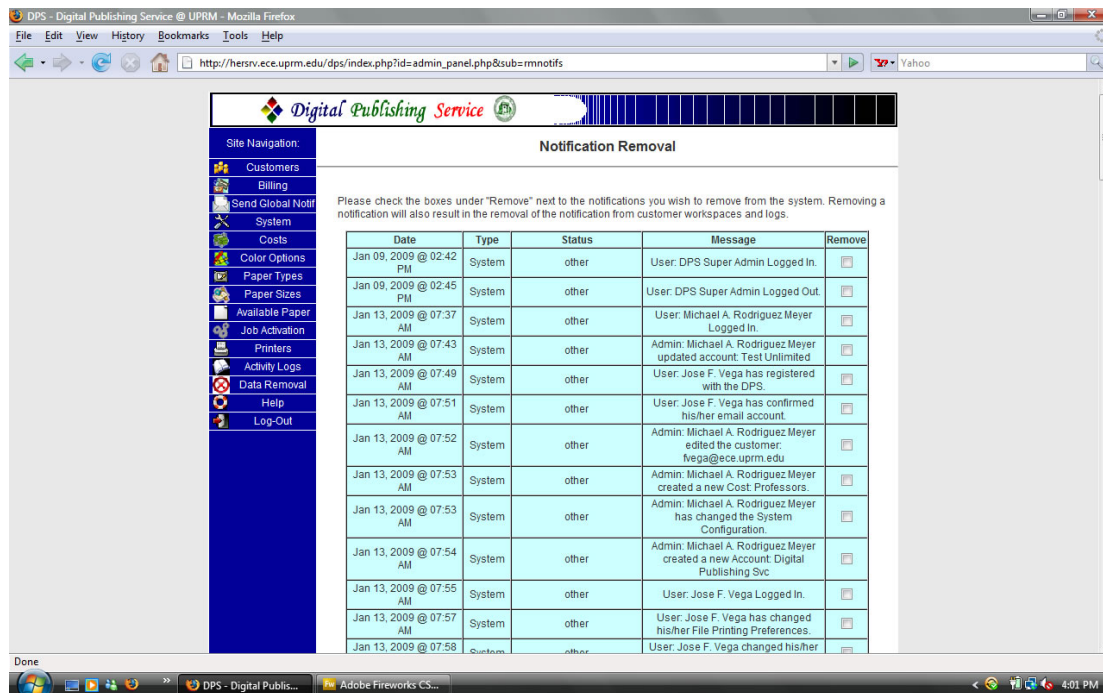


Figure 33: Notification Removal screen.

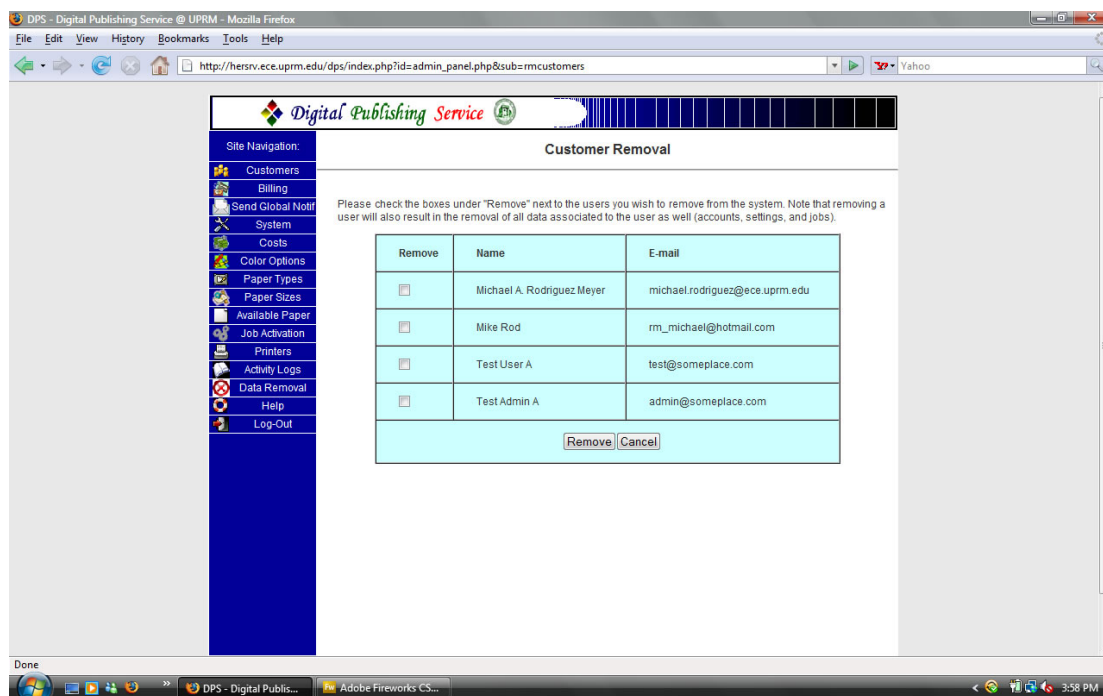


Figure 34: Customer Removal screen.

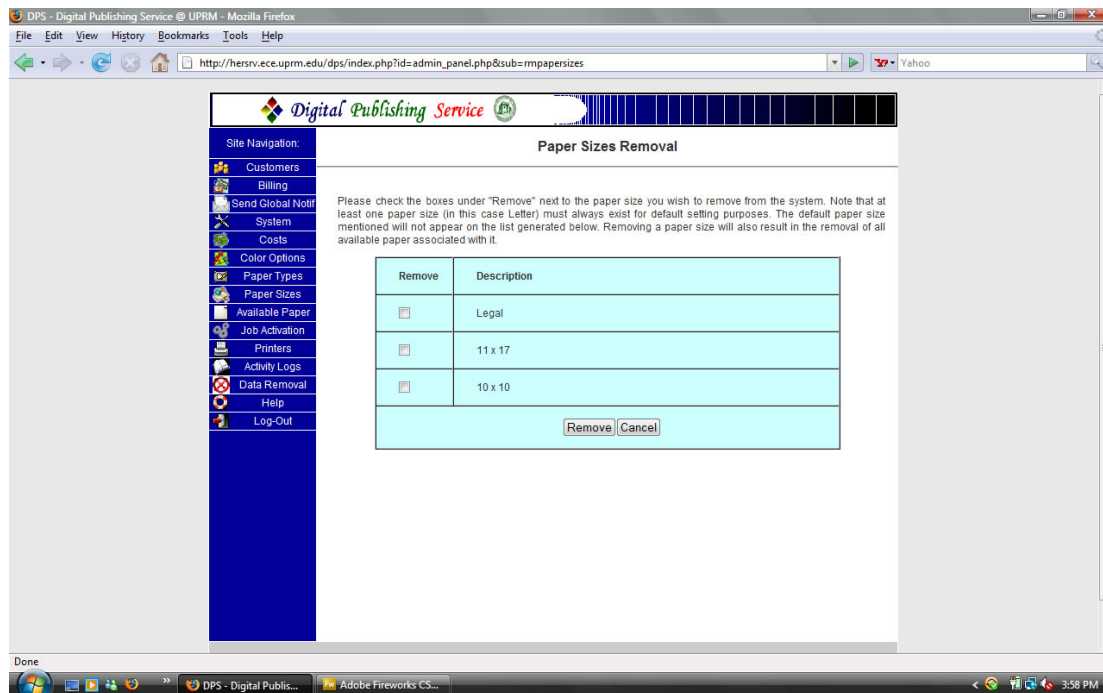


Figure 35: Paper Sizes Removal screen.

This concludes the Super Administrator interfaces. We may now continue with the regular user interfaces in order to fully understand the DPS.

4.5 User Interfaces

If a user has confirmed that he or she is a regular user, the system will take the user to the beginning menu interface or Home page for regular users as seen in Figure 36. The left-hand menu allows regular users to view and remove notification sent to them by either the administrators or the system, view the status of the jobs, submit new jobs, or change information and preferences associated with the accounts. We will now describe each option in more detail.

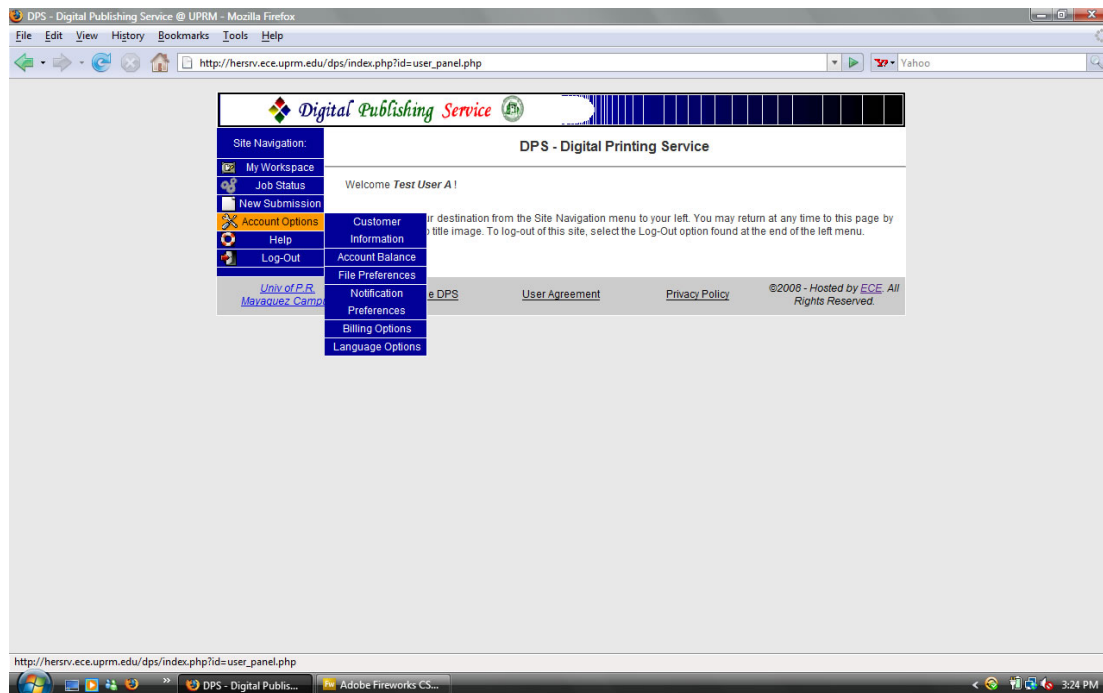


Figure 36: Regular User Home page and menu.

The first option available to regular users is the My Workspace. This option provides a screen (see Figure 37) with all notifications separated by accounts the user has received from administrators or the system. These notifications are displayed depending on the notification preferences set by the user for each account. A user can also remove any notification by selecting the box next to the desired notification. Notifications removed in this fashion will not be shown again to the user, but they are still kept in the system for log purposes. This is possible because each notification in the *notifications* table of the DPS database has a Boolean column called *user_rm* that denotes whether it was previously removed by the user or not.

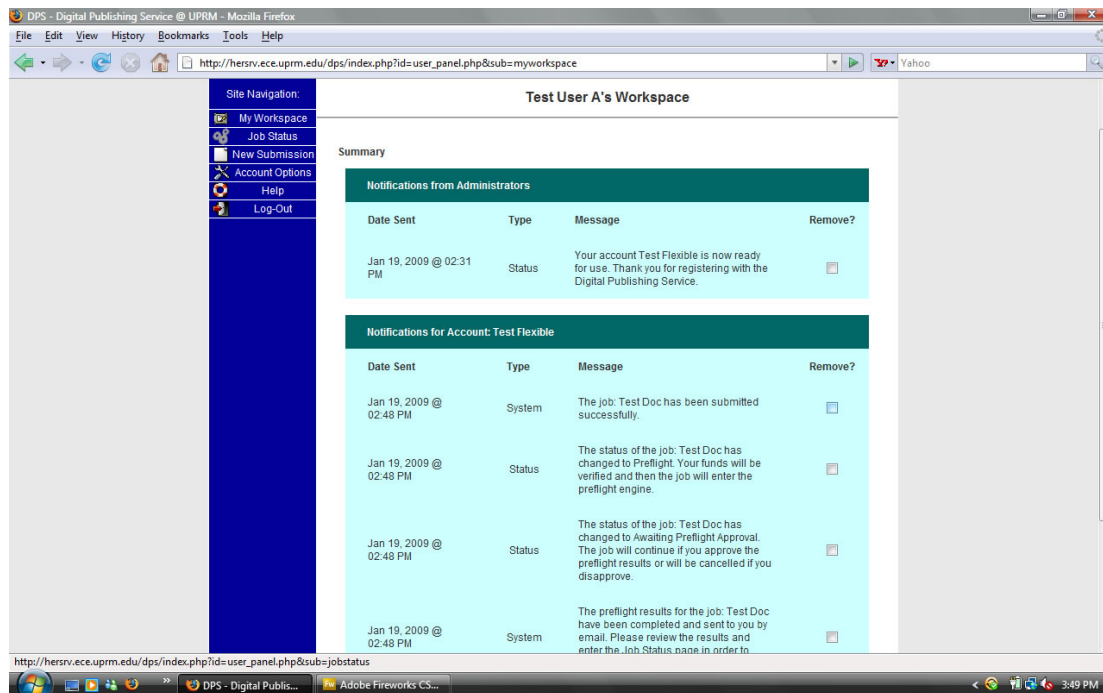


Figure 37: Regular user My Workspace screen example.

The next option available is the Job Status option. This option provides a screen (see Figure 38) for a user to view the current status of all submitted jobs. This screen also provides users with a tool for canceling jobs and approving preflight or proofing tasks. If a given job is currently awaiting preflight or proofing approval, a Yes or No option will appear in the respective column next to the job information. This function occurs dynamically and automates the approval phase and gives users more control over the jobs. If a user does not approve preflight or proofing results, the job will automatically be canceled. Any decision taken by a user will result in the workflow engine being called and properly executing the appropriate phase of the job based on the answers.

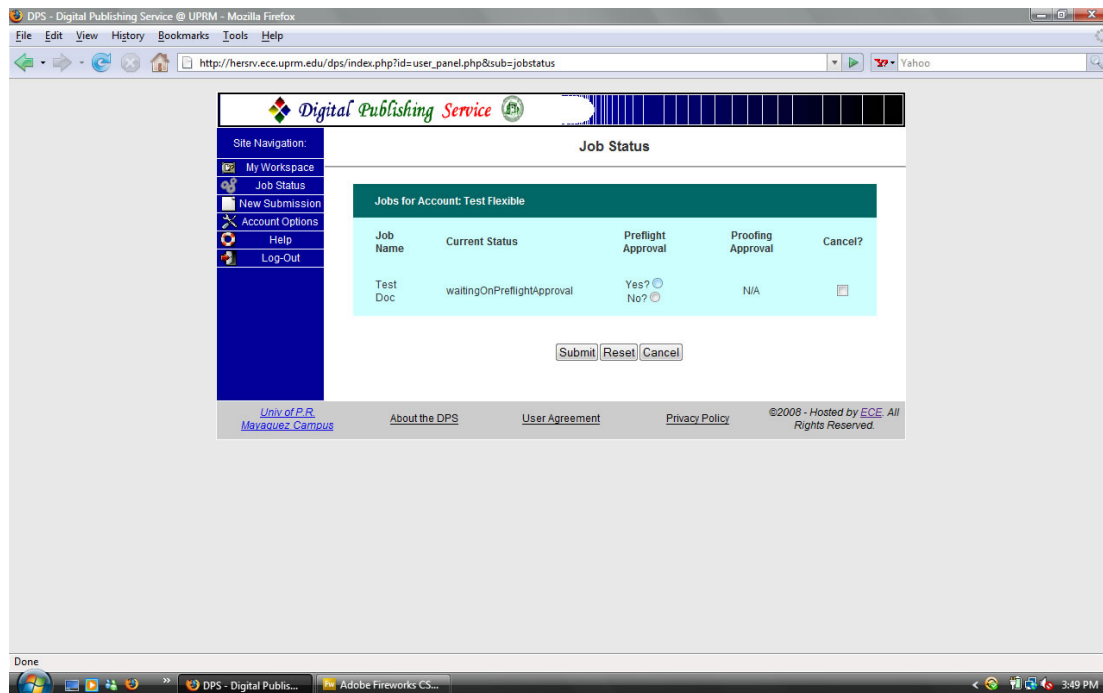


Figure 38: Job Status screen.

The next option is one of the most important options available to regular users because it allows them to submit jobs for printing. The New Submission option provides the screens for creating a job, adding files, and changing job parameters. Figure 39 illustrates the job submission process in the correct order. The first screen (see Figure 40) is where a user enters the identifying name for the job, which account will be used for payments and preferences, and whether a preflight or proofing copy will be requested. The next screen (see Figures 41 and 42) is where the user can add files to the job and change the options for each file, such as number of copies, paper to be used, and color option. The Both Sides option tells the DPS if sheets will be printed on both sides or not. When job or proofing costs are calculated, each sheet of paper used might have an additional charge associated with it due to size and paper type. Therefore, the Both Sides option will also reduce the amount of paper a user is charged for if there are any additional paper option charges.

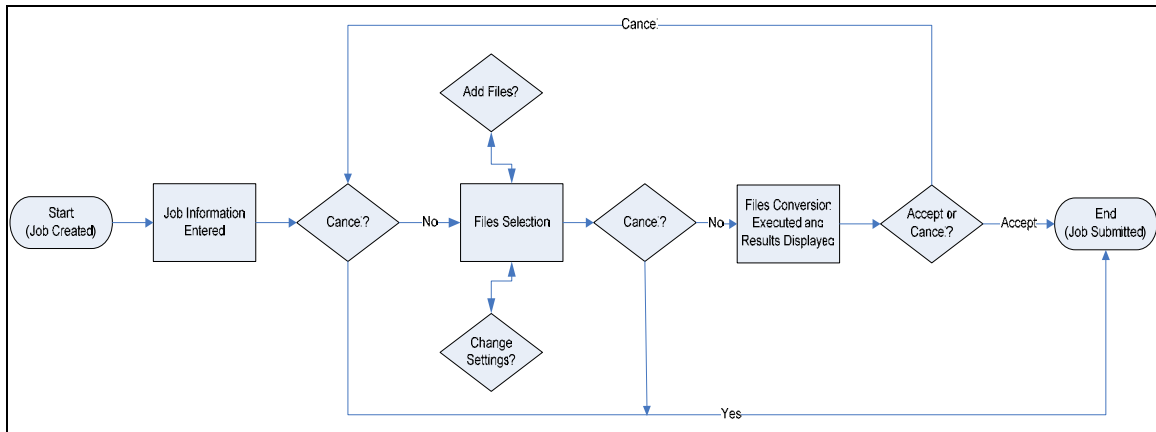


Figure 39: Job submission process.

The screenshot shows a web browser window titled "DPS - Digital Publishing Service @ UPRM - Mozilla Firefox". The address bar shows the URL: http://hersrv.ece.uprm.edu/dps/index.php?id=user_panel.php&sub=submission. The page has a blue sidebar with navigation links: Site Navigation, My Workspace, Job Status, New Submission (highlighted), Account Options, Help, and Log-Out. The main content area is titled "New Submission" and contains a welcome message: "Welcome to the Digital Publishing Service Job Submission pages. Please enter the required information below and press the 'Next' button when ready to continue. Please note that the job you are about to begin must have a name. At any moment during the job submission you may press the 'Cancel' button in order to cancel the job process and return to the initial customer page. Doing this will remove the job from the system and any files uploaded as part of the job." Below the message is a form with the following fields: Name (text input with "TestDoc"), Choose an Account (dropdown menu with "TestFlexible"), Preflight? (radio button with "Yes" selected), and Proofing? (radio button with "Yes" selected). At the bottom of the form are "Next" and "Cancel" buttons. The footer of the page includes links for "Univ of P.R. Mayaguez Campus", "About the DPS", "User Agreement", "Privacy Policy", and a copyright notice: "©2008 - Hosted by ECE. All Rights Reserved." The browser's status bar at the bottom shows "Done" and the system clock indicates "3:46 PM".

Figure 40: New submission job information screen.

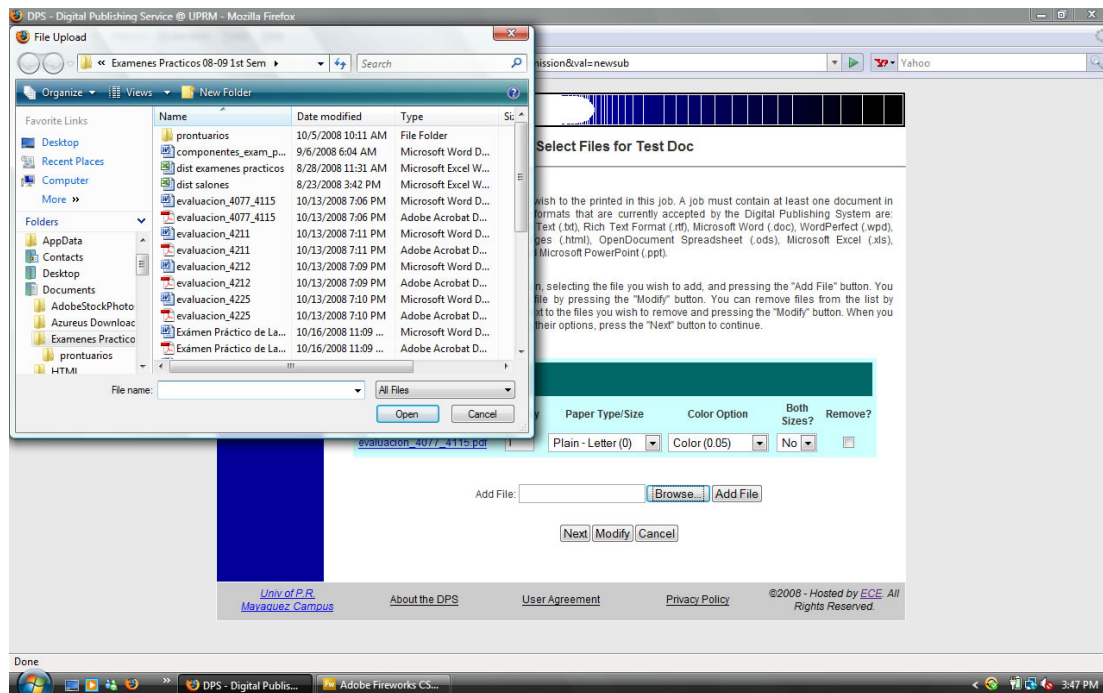


Figure 41: Adding a file to a job example.

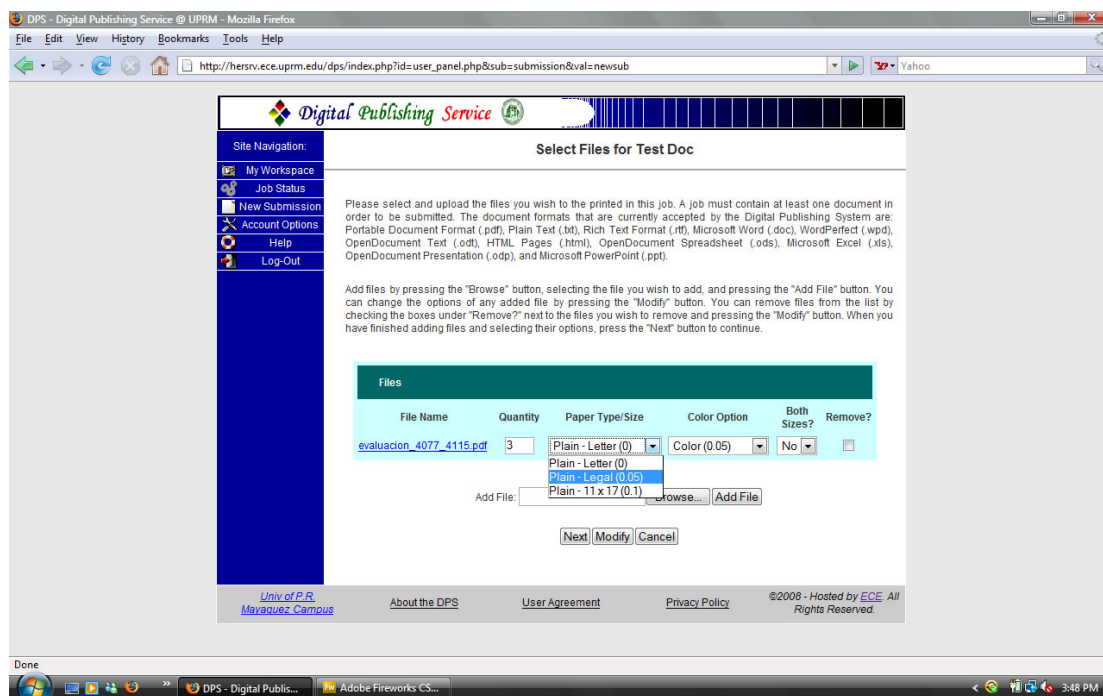


Figure 42: Changing file options example.

The following screen (see Figure 43) shows the user all information related to the job, such as costs and selected account information. Given the nature of the printers

currently being used by the DPS, all files must be converted to PDF format. The DPS does this automatically and shows the user the resulting converted file or if any errors occurred in case the user does not approve. Up to this screen, a user can press the Back button to return to an earlier screen or cancel the job without any problems. Canceling a job in this manner eliminates the job information from the system and deletes files that were uploaded by the user. The final screen (see Figure 44) informs the user that the job was submitted and begins the workflow engine process based on the options selected.

The screenshot shows a web browser window titled "DPS - Digital Publishing Service @ UPRM - Mozilla Firefox". The address bar shows the URL: http://hersrv.ece.uprm.edu/dps/index.php?id=user_panel.php&sub=submission&val=newsub. The page has a blue sidebar with "Site Navigation" links: My Workspace, Job Status, New Submission, Account Options, Help, and Log-Out. The main content area is titled "Job Preview: Test Doc".

Below the title, there is a paragraph of text: "The Digital Publishing System's current system uses Postscript printers and a UNIX server. At the present time, all documents to be printed must be in PDF format. This system automatically converts documents to PDF format, although there may be small inconsistencies. Due to these inconsistencies, you will be shown the converted documents below as well as any documents that we could not convert. Unconverted documents will automatically be removed from the job. If you are not in agreement with the converted version of your documents, please press 'Back' and add your document again after using your own PDF conversion method."

Below this text is a table titled "Files" with three columns: Original File, Converted File, and Date Uploaded. The table contains one row of data:

Original File	Converted File	Date Uploaded
evaluacion_4077_4115.pdf	evaluacion_4077_4115.pdf	2009-01-19

Below the table is another paragraph of text: "Please note that a job submitted without the necessary balance will be halted until funds are obtained. You will receive a notification if this occurs as well. If you selected preflight and/or proofing, the charge of these options will be deducted in addition to the amount needed for the complete job. If you are not in agreement with the preflight or proofing of the job, you may state this in the 'Current Jobs' page and no further funds will be deducted from your account. The Printing Cost below is the cost for the entire printing job not including any preflight or proofing made."

Below this text is a table titled "Account and Job Information" with two columns: Name and Account Name. The table contains two rows of data:

Name	Account Name
Test Doc	Test Flexible
Submission Date: 2009-01-19	Account Type: Flexible

Below the table is another table with two columns: Current Status and Account Balance. The table contains two rows of data:

Current Status	Account Balance
Pending	\$ 10.00
Proofing: Yes	Proofing Cost: \$ 0.15

Below the table is another table with two columns: Preflight and Preflight Cost. The table contains two rows of data:

Preflight	Preflight Cost
Yes	\$ 0.1
Documents: 1	Printing Cost: \$ 0.6

At the bottom of the page, there are three buttons: Submit, Back, and Cancel.

Figure 43: Job Preview information and file conversions.

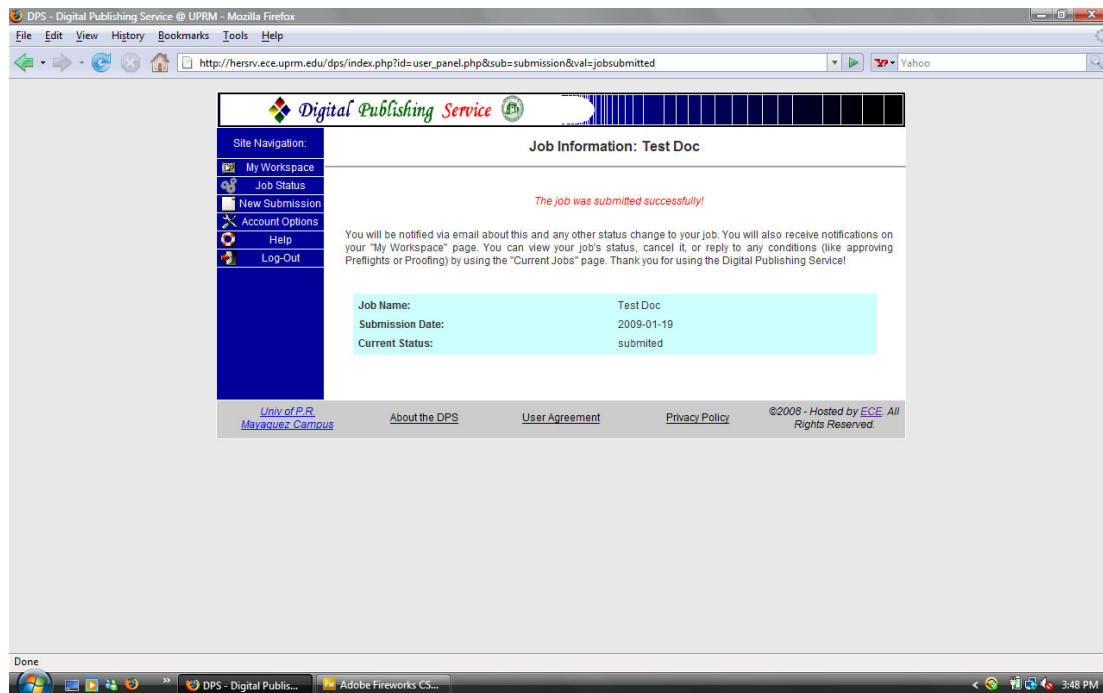


Figure 44: Job submitted confirmation screen.

The next option available to users is the Account Options. This option provides six tools that users can use in order to view their information, accounts, and change their preferences. The customer information screen (see Figure 45) displays a regular user's registration information and provides a tool to change said user's password. The other information provided can be viewed, but not changed by a user once it has been submitted during registration. The account balance screen (see Figure 46) displays the balance or accumulated bill for the user account that is currently selected. If the user has multiple accounts, the information of a different account can be viewed by selecting the account from the topmost drop down menu bar. The information displayed is only meant to inform a user; therefore it cannot be changed by the user. The file preferences screen (see Figure 47) allows a user to set the initial options that will appear when adding files to new job submissions. As with the account balance screen, the account chosen for the file preferences can be selected from the topmost drop down menu bar. All screens with the menu bar functionality change dynamically as an account is selected. There is no need to re-enter the option screens again.

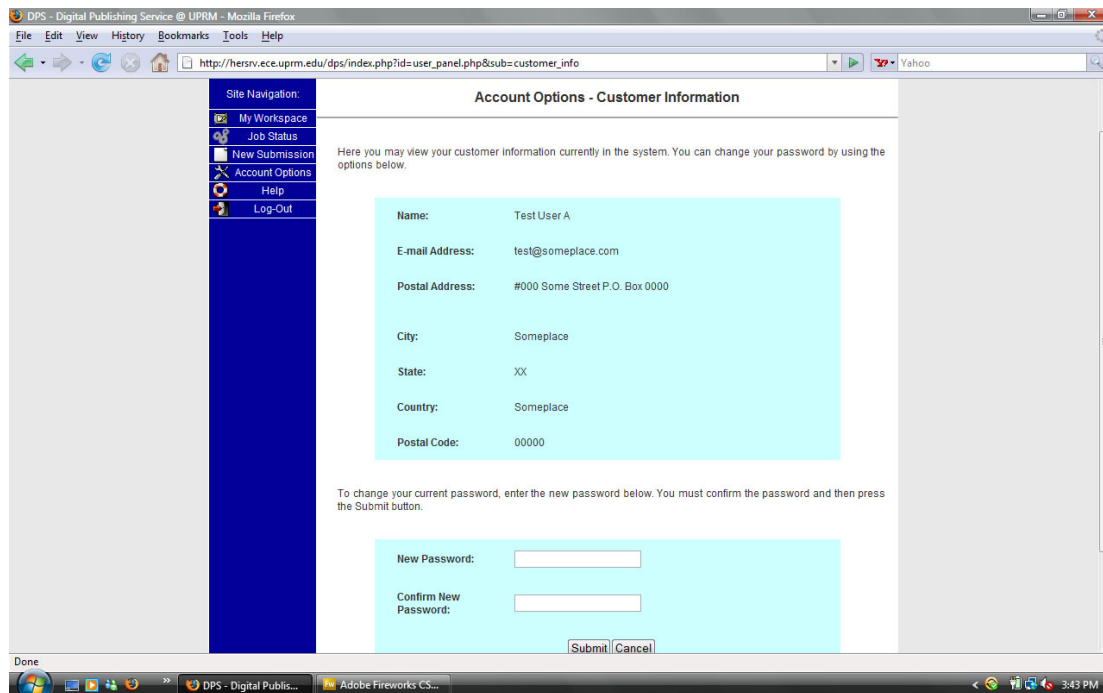


Figure 45: Customer Information screen with password changing tool.

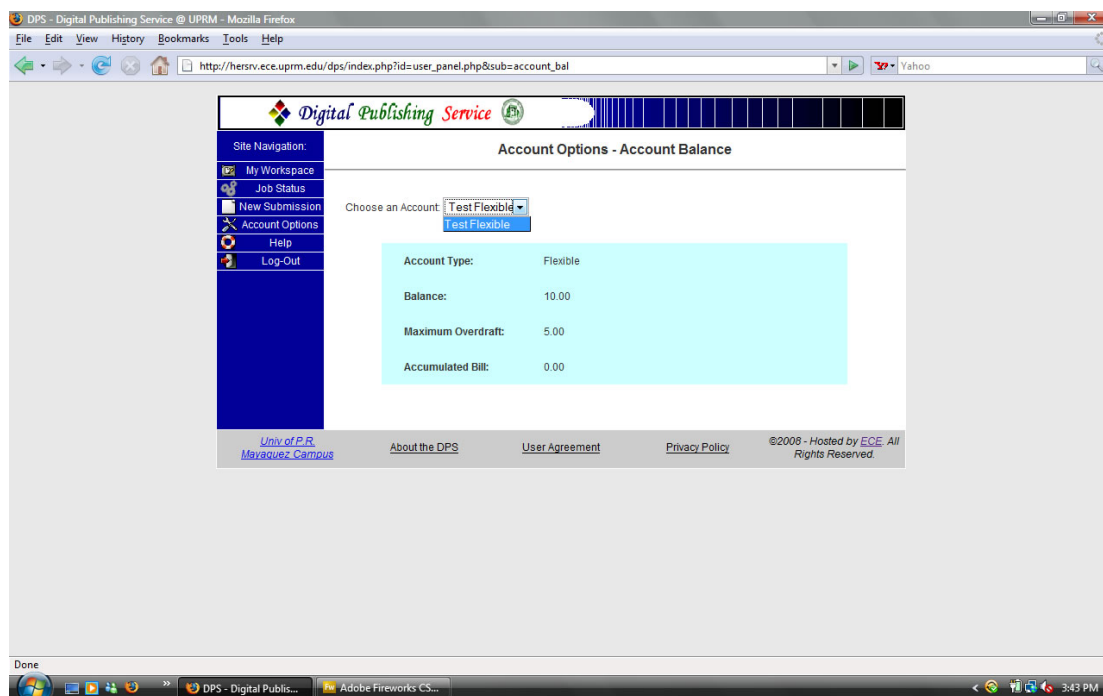


Figure 46: Account Balance screen example.

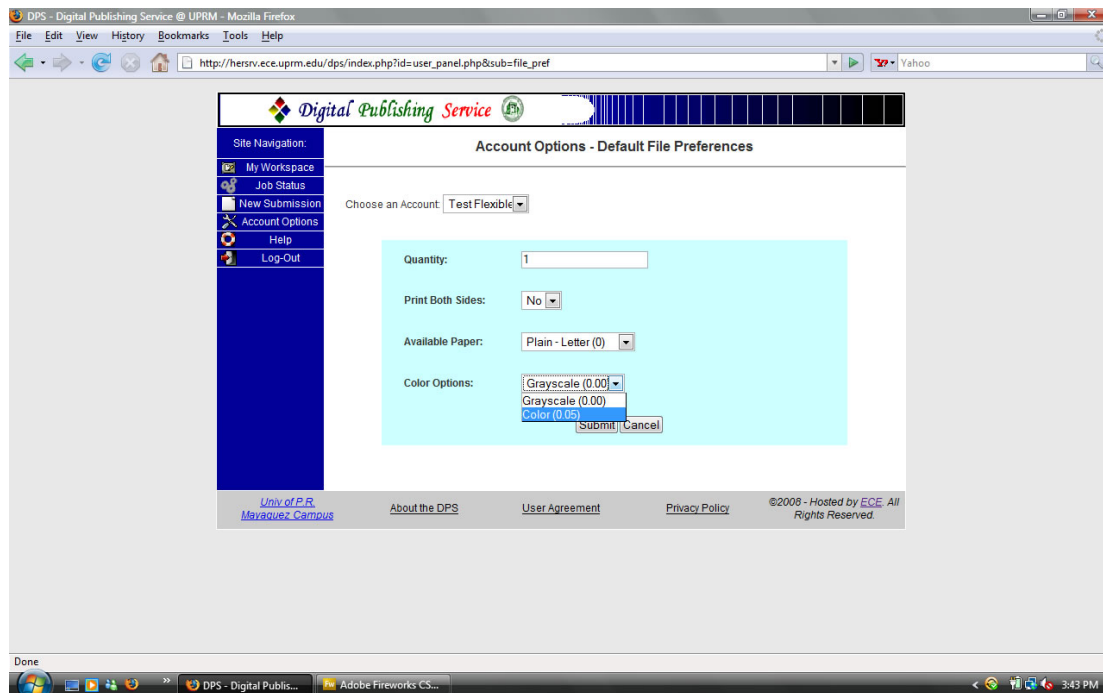


Figure 47: Default File Preferences screen example.

The notification preferences screen (see Figure 48) allows a user to determine which types of messages will be displayed on the My Workspace screen for each account the user has. As with other screens in the Account Options, a topmost drop down menu allows a user to dynamically select a different account to customize. It should be noted that choosing to not receive certain messages may result in a user not knowing the status of jobs submitted. Also note that even though a user selects not to display certain types of messages, the notifications will still remain in the system and can be viewed by changing the notification preferences again. The next screen, called Billing Options (see Figure 49), is meant to be a tool for users to add funds to their account. Since the DPS does not have payment-type support (example ATM or Credit Card interfaces), this screen provides users with a messaging system in order to notify administrators of their intentions to add funds to their account. Sending a message through this tool will send an e-mail to all administrators identified in the system with the message content. The final screen under Account Options, language options, is meant to be a tool that allows users to change the default language that is used to display the DPS. Currently the DPS only

supports the United States English language; therefore this tool has no actual function. The screen was still added for future improvements.

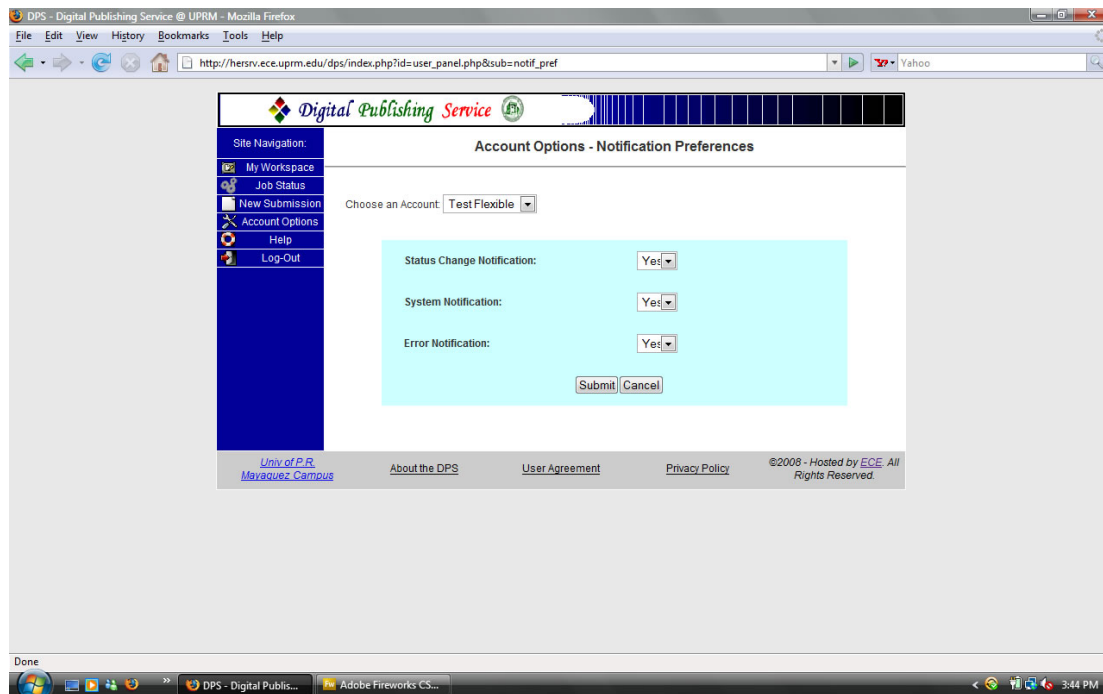


Figure 48: Notification Preferences screen example.

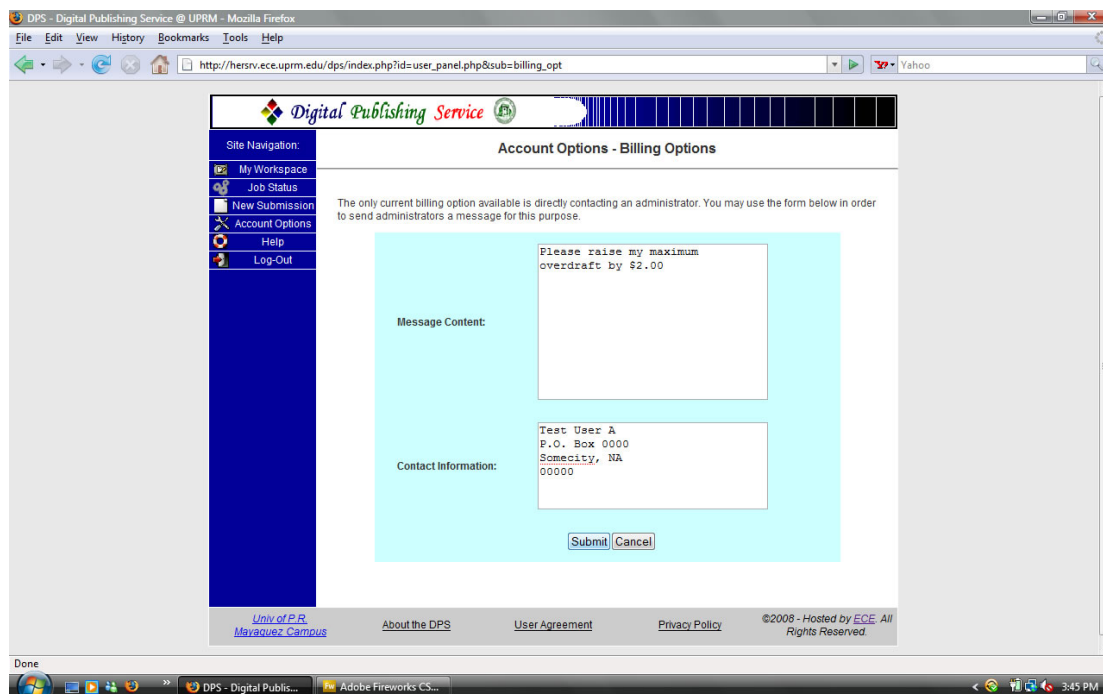


Figure 49: Billing Options screen example.

The final options found in the user interfaces are Help options. These options are similar to the administrator options because they provide a Frequently Asked Questions screen and User Manual download tool. With this we should now have a clear understanding of all the interfaces that are present in the DPS and their functionality. The next sections will explain in more detail the DPS database tables and workflow engine. We will also learn how job costs are calculated for submitted jobs.

4.6 Database Model

As can be appreciated throughout this work, the DPS interacts greatly with a database in order to dynamically print documents, have account functionality, and determine system parameters. An E-R diagram of the entire database, tables, and attributes can be found in Appendix A. In this diagram we can see how many tables are dependent on information from other tables. This is known as Foreign Key (FK) relationships and they ensure that important data is not modified in one table without child data being modified in another related table first. The dependencies can be viewed by the arrows pointing from one table to another. The attributes of each table can be viewed inside each table box with their respective data type. Attributes that appear in bold are required by each record, while non-bold attributes indicate that the value can be left blank when performing queries.

Another important note is that all tables have a Primary Key (PK) that is usually of the type Counter. This means that each record has a unique id associated with it that is automatically generated with a number higher than the id before it. This eliminates duplicate data rows, provides a faster method for searching through tables, and helps ensure table row organization. Most queries made to the database are simple one or two table Select, Update, or Insert instructions. At times, three or more tables are queried, but all queries tend to be basic Inner Join type instructions. The only exceptions to these basic queries are the Data Removal queries, which tend to be simple one table Delete instructions. Given the large number of tables, the Table 3 will name each database table and summarize the use given to that particular table.

Table 3: Database Tables and Functions.

Table Name	Table Function
<i>Customer</i>	All general registration information of system users is contained in this table. E-mail and Admin confirmation flags are also contained here.
<i>Accounts</i>	Each account with its associated information is contained in this table. This table links to other tables with preferences and options.
<i>PrepaidAccount</i>	All prepaid accounts and their balances are contained in this table.
<i>FlexibleAccount</i>	All flexible accounts and their balances, maximum overdraft, and accumulated bills are contained in this table.
<i>UnlimitedAccount</i>	All unlimited accounts and their respective accumulated bills are contained in this table.
<i>Costs</i>	Surcharge types and their associated cost values are contained here.
<i>DefaultPreferences</i>	This table contains default file preferences associated with each account found on the <i>accounts</i> table.
<i>NotificationPreferences</i>	This table contains the notification preferences associated with each account found on the <i>accounts</i> table and their associated notifications.
<i>ColorOption</i>	All color schemes and their respective additional costs are contained in this table.
<i>PaperTypes</i>	All paper media types and their respective additional costs are contained in this table.
<i>PaperSize</i>	All paper media sizes and their respective additional costs are contained in this table.
<i>Paper</i>	All available combinations of paper types and paper sizes are held in this table and whether or not they can be actively used.
<i>FilePrintingSettings</i>	The media and printing options associated with each file from the <i>file</i> table is contained in this table.
<i>Job</i>	Each job information, costs, and general options are contained in this table. This table also links to the status each given job.
<i>File</i>	This table contains the information for each file added to different jobs. This information is also used to print the respective files and manipulate them in the system.
<i>Notifications</i>	This table contains all notifications sent by administrators or generated by the system. Notifications can be associated with jobs, users, or neither. Each notification is also linked to a <i>workingstatus</i> .
<i>WorkingStatus</i>	This table contains all the possible descriptions for status changes that jobs or notifications can be in at any given time.
<i>Printers</i>	The general printer information for each printer can be found in this table, along with whether the printer is active or not.
<i>TrayManagement</i>	This table links each tray of every printer contained in the <i>printers</i> table with the paper media contained in the <i>paper</i> table.
<i>SystemConfiguration</i>	This table holds the main system costs for each print, preflight, or proofing use.

The database model and associations should now be clear and understandable. We can now learn in the next section the details about the final DPS system component, the workflow engine.

4.7 System Workflow

The DPS workflow engine is perhaps one of the most important components in the system because it performs the logical execution decisions during the different phases of job submissions and printing. A diagram of the workflow engine decisions and possible phases can be viewed in Appendix B. As described earlier, there are three possible actions with a job: preflight, proofing, and printing. Each action must occur before the next since these actions provide a user with feedback and verification tools throughout printing job execution. This is the basis of digital publishing; to allow a user to dynamically and automatically print digital jobs with control over every phase of the job. Of course this means that a user has the option to either perform preflight, proofing, both, or neither before the actual job printing. In the earlier versions of the DPS it was assumed that if a preflight was done, then proofing wasn't necessary since the job was already verified. A simple truth table relationship can be obtained for representing these cases (see Table 4). In the current DPS a user can decide to obtain a proofing copy without a preflight if desired. This grants a user total control over the verification done on a job.

Table 4: Truth table of possible job options in earlier DPS version.

Job Options	
Preflight	Proofing
0	0
1	0
1	1

The workflow engine in the DPS is a PHP page that has no interface capability. It functions purely as a decision making function that uses inputted data and various conditionals in order to “decide” what course of action to take. Because the workflow engine is a simple PHP page, it can be called at any moment during the DPS executions by means of the PHP *include* function. This also makes an engine running for one user, completely independent of the engine running for another user since any number of instances of the workflow engine page can be called at once. Three basic data values are needed as input in order for the workflow engine to run at any instance: the workflow mode or option, the job id, and the account id. The job and account id are used to create

queries to the database and obtain information about accounts, balances, files, costs, and settings. The workflow mode tells the workflow engine from what point in the workflow diagram it will be continuing or beginning. The possible modes are: initial, preflight, proofing, and printing. These modes are somewhat self-explanatory.

An important factor to understand here is the cost calculation of a preflight test, proofing copy, and printing job. The easiest to understand is the preflight test. The costs for a preflight are static and independent of each file's settings. A preflight engine is often a software suite that checks a document and returns a report, therefore the cost is based on the cost a third-party company would charge for doing this work or for the license needed to have the software suite. The proofing and printing copies depend on the settings of each file in a job. The Figure 50 illustrates how proofing and printing costs are determined for a single file in a job. Note that the Both Sides Option means that if a user desires both sides of each sheet of paper to be used, then half the required sheets of paper will be needed for the job. Any additional costs associated with paper size and type will be halved, hence the division decision in the diagram. Also note that color options, surcharges, and system charges are needed every time the printers print a page. Therefore, this is independent of whether both sides are used or not. The sum of these costs will count always. Finally, note that a proofing copy is a single copy of a job; therefore all settings costs and paper costs will only be needed once since the quantity is automatically 1. Printing the entire job is the same as the proofing cost multiplied by the quantity specified by the user.

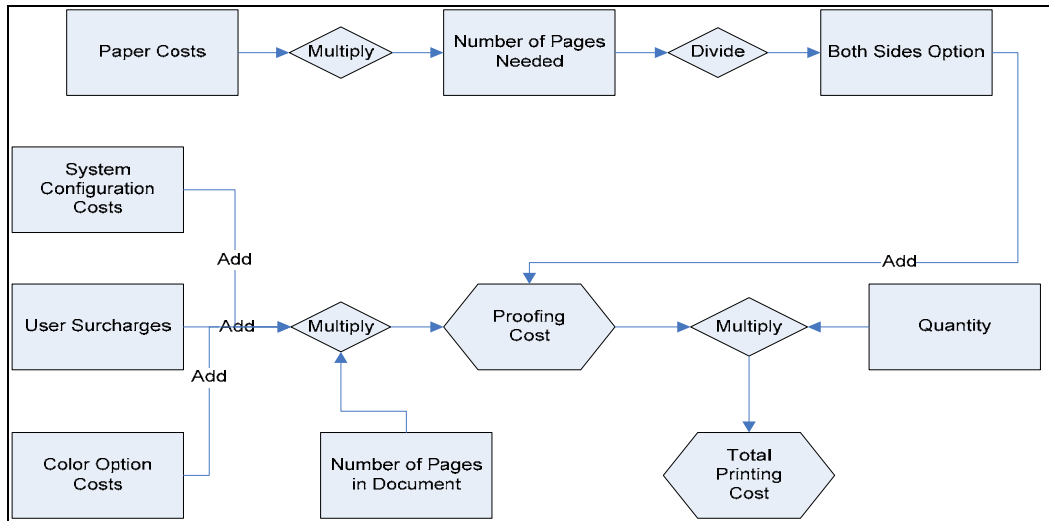


Figure 50: Costs calculation diagram.

This concludes the workflow engine description and functionality. This also concludes this chapter on the system description of the DPS. We can now move on to the conclusions for this work and future recommendations.

CHAPTER 5 – CONCLUSIONS & FUTURE WORK

As stated earlier in this work, the idea of digital publishing is to eliminate the manual aspect of printing documents and allow users to have more control over all aspects of the printing process. Users not only receive high quality job results, but also can dynamically personalize documents. The UPRM has a very capable printing center and many departments have very advanced computer technology at their disposal. However, a truly automatic, yet dynamic service is still a tool that cannot be found on the campus. Everyday we use the World Wide Web more and more for classes and investigations, yet no web-based printing services have been made in the past for campus use. The DPS has proved to not only allow network printing from outside the campus, but also provide automatic and remote interaction with the printing facilities it controls.

The DPS demonstrates that web-based services can function properly and be created with all the security features and individual components expected from today's technology. Also by accepting a large number of document formats and media types, it proves to be very versatile to user's needs or preferences. With the conclusion of this work, the possibility of a digital publishing service for educational institutions is now a reality. The current DPS has also improved from its earlier versions by gaining more automatic features, completed interfaces, and high quality results. The addition of document conversion packages and use of PHP functions has also made the internal processes easier to execute and open to more possibilities than static functions that are only useful for specific hardware. Appendix E illustrates examples of e-mail messages sent by the system, proving that the DPS also provides adequate feedback in addition to the error prevention features mentioned in earlier chapters.

Even though the objectives of implementing, completing the interfaces, and testing the DPS were successful and the service itself has improved, there is still plenty of room for future developments, improvements, and updates. More printers could be tested and merged with the present system, more document formats and settings could be added, and dynamic communication with the printers could be investigated. As web-based technologies and programming advance, the DPS and other digital publishing services could also advance. Everyday printing hardware becomes more digital and more

intelligent. The DPS opens the doors for more research in this vastly growing area of digital publishing. As stated before, it provides a possibility and ideas that can be built upon to create more accessible and greater quality tools for the area of higher education and beyond.

REFERENCES

- [1] M. W. Fleming, “Digital Print”, Digital Publishing Solutions Magazine, [online document] Jan. 2007, Available at: <http://www.dpsmagazine.com/content/ContentCT.asp?P=314>
- [2] University of Puerto Rico at Mayagüez, “Sección de Impresos”, Press Facility Web Site, [online document], 2003 - 2006, Available at: <http://www.uprm.edu/decadmi/secimp.html>.
- [3] A. E. Pereira, “Knowledge Representation for Digital Publishing Workflow”, Thesis M.S., University of Puerto Rico at Mayagüez, P.R., May 2006.
- [4] H. J. Santos, “Style-Dependent Artifact Recognition for Digital Variable Data Printing”, Thesis M.S., University of Puerto Rico at Mayagüez, P.R., May 2005.
- [5] C. F. Weston, “Print on Demand”, Digital Publishing Solutions Magazine, [online document] Dec. 2006, Available at: <http://www.dpsmagazine.com/content/ContentCT.asp?P=310>
- [6] Lulu, Inc., “Lulu”, Lulu Help & Services, [online document], 2002 - 2007, Available at: <http://www.lulu.com>
- [7] Parlor Press LLC, “Parlor Press”, Parlor Press Submission Guidelines, [online document], 2002 - 2007, Available at: <http://www.parlorpress.com>
- [8] Cornell University Library, “DPubS Digital Publishing System”, About DPubS, [online document], 2008, Available at: <http://dpubs.org/>

- [9] Net2Printer, “Net2Printer Immediate Document Delivery”, Net2Printer Product Information, [online document], 2003 - 2008, Available at: <http://www.net2printer.com/>
- [10] J. F. Vega, “High-Quality Web-based Digital Publishing Service for Higher Education”, Department of Electrical and Computer Engineering, University of Puerto Rico at Mayagüez, P.R., August 2005.
- [11] R. Veguilla and D. Yopez, “Digital Publishing System: Final Report”, Department of Electrical and Computer Engineering, University of Puerto Rico at Mayagüez, P.R., December 2005.
- [12] N. Quirindongo, A. Nieves, F. Torres, “High-Quality Web-Based Digital Publishing Service for Higher Education: Final Report”, Department of Electrical and Computer Engineering, University of Puerto Rico at Mayagüez, P.R., May 2006.
- [13] J. Barbosa, P. Borges, X. Peña, “High-Quality Web-Based Digital Publishing Service for Higher Education: Final Report”, Department of Electrical and Computer Engineering, University of Puerto Rico at Mayagüez, P.R., January 2007.
- [14] M. A. Rodriguez, J. F. Vega, “Automatic Document Conversion for Printing Service via World Wide Web”, Department of Electrical and Computer Engineering, University of Puerto Rico at Mayagüez, P.R., May 2007.

APPENDIX A – DPS DATABASE E-R MODEL DIAGRAM

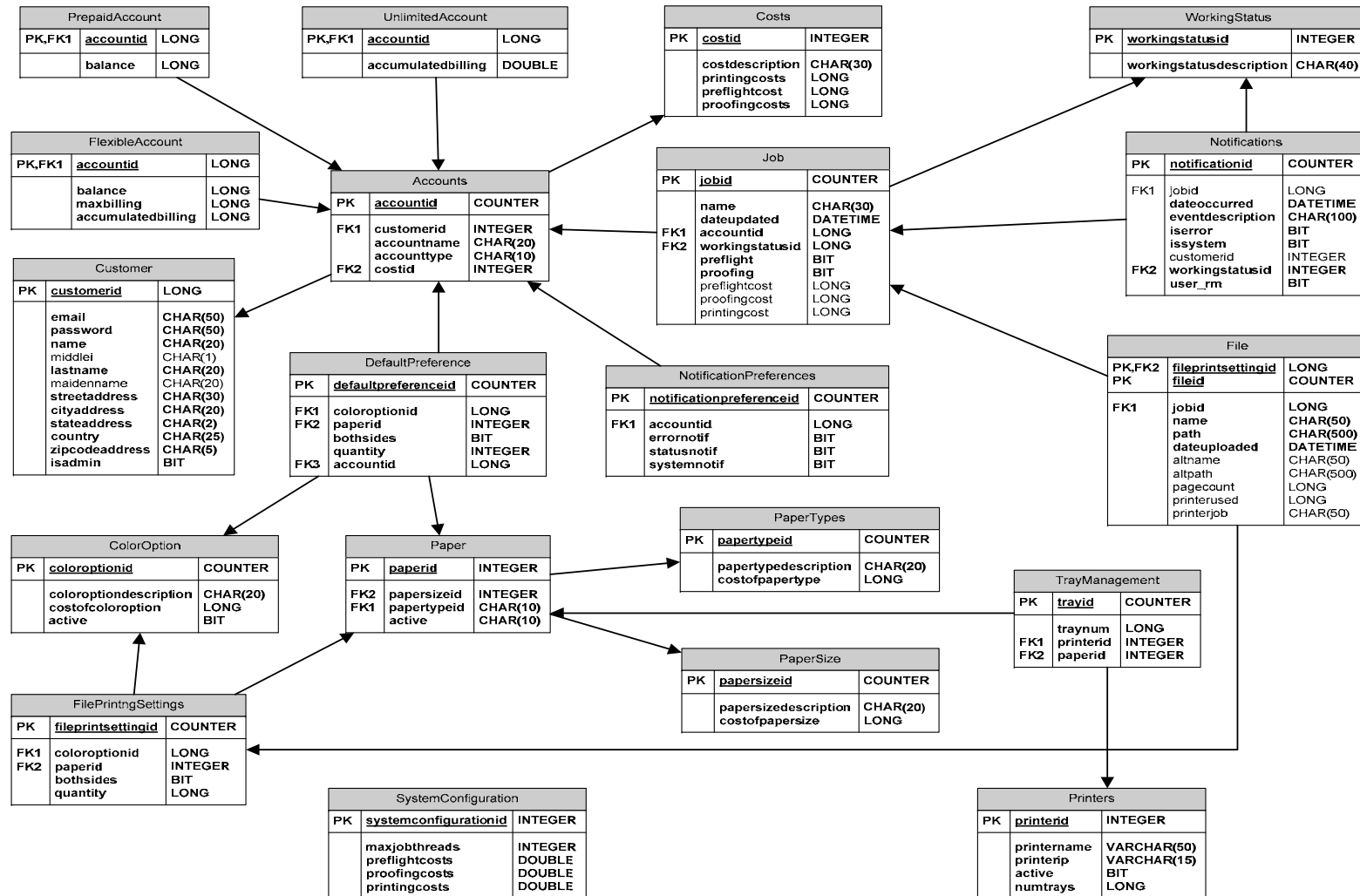


Figure 51: Database tables and attributes entity - relationship model.

APPENDIX B – JOB WORKFLOW ENGINE DIAGRAM

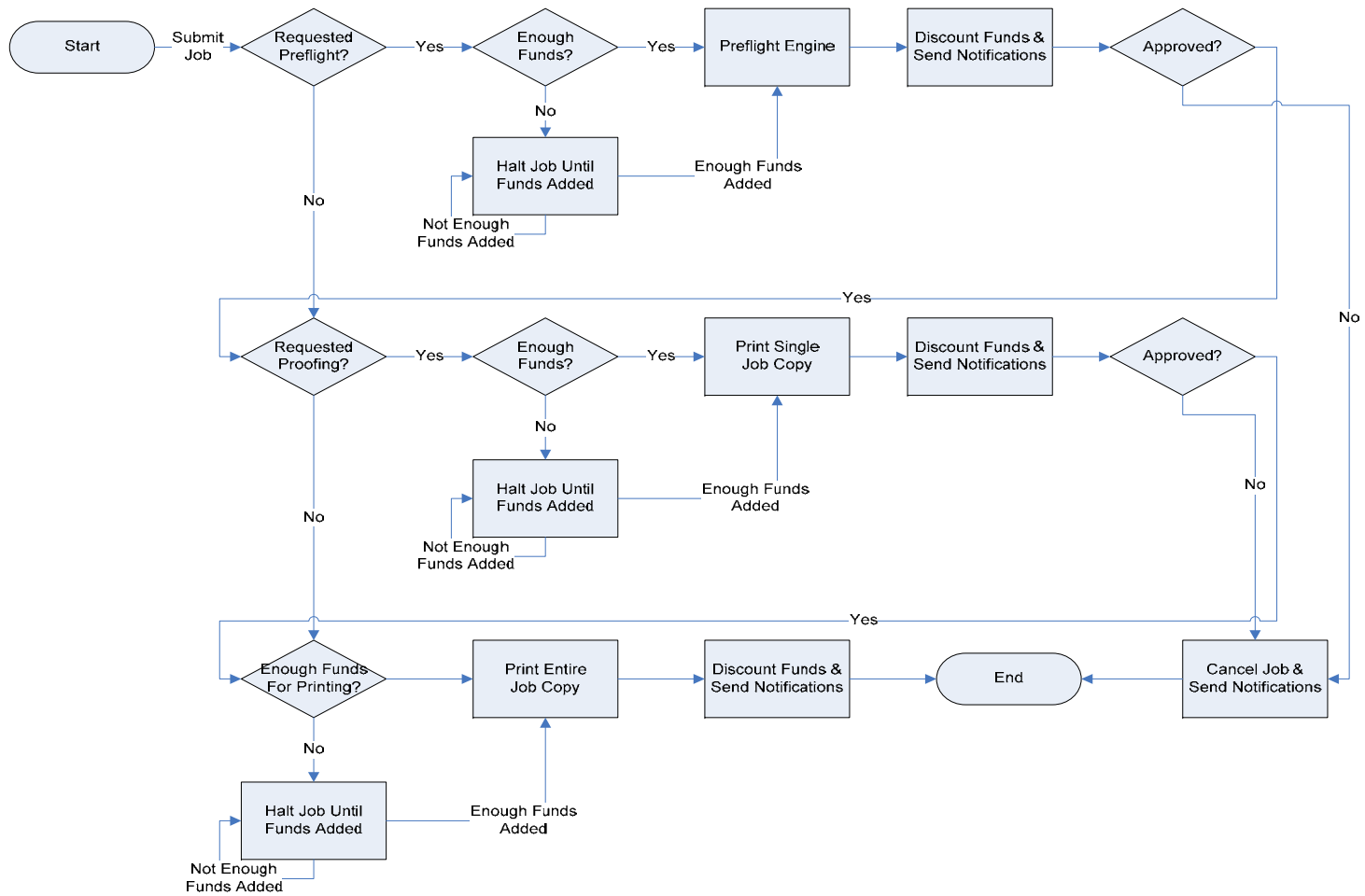


Figure 52: Job workflow diagram.

APPENDIX C – ACCOUNT BILLING RECEIPT EXAMPLE

DPS - Digital Publishing Service @ UPRM - Accumulated Bill Receipt - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://hersrv.ece.uprm.edu/dps/admin_pages/receipts.php?id=5&reset=3

DPS - Digital Publishing Service @ ... DPS - Digital Publishing Service ...

Digital Publishing Service @ UPRM

Accumulated Bill Receipt

Customer: Test User A

Address: #000 Some Street P.O. Box 0000

Someplace, XX

Someplace

00000

Account Name: Test Flexible

Account Type: Flexible

Customer Signature: Amount Owed: \$0.00

Done

DPS - Digital Publis... Adobe Fireworks CS...

3:32 PM

Figure 53: Example billing receipt for flexible or unlimited accounts.

APPENDIX D – DPS SECURITY MESSAGES EXAMPLES

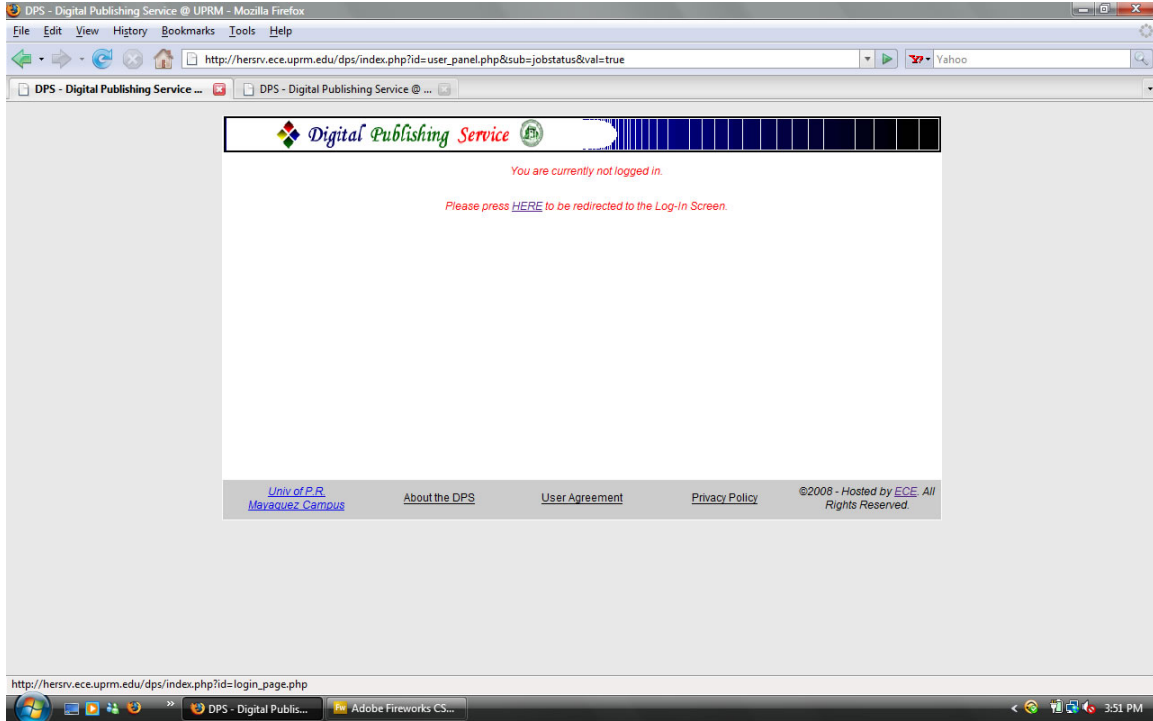


Figure 54: Session assurance error message example.

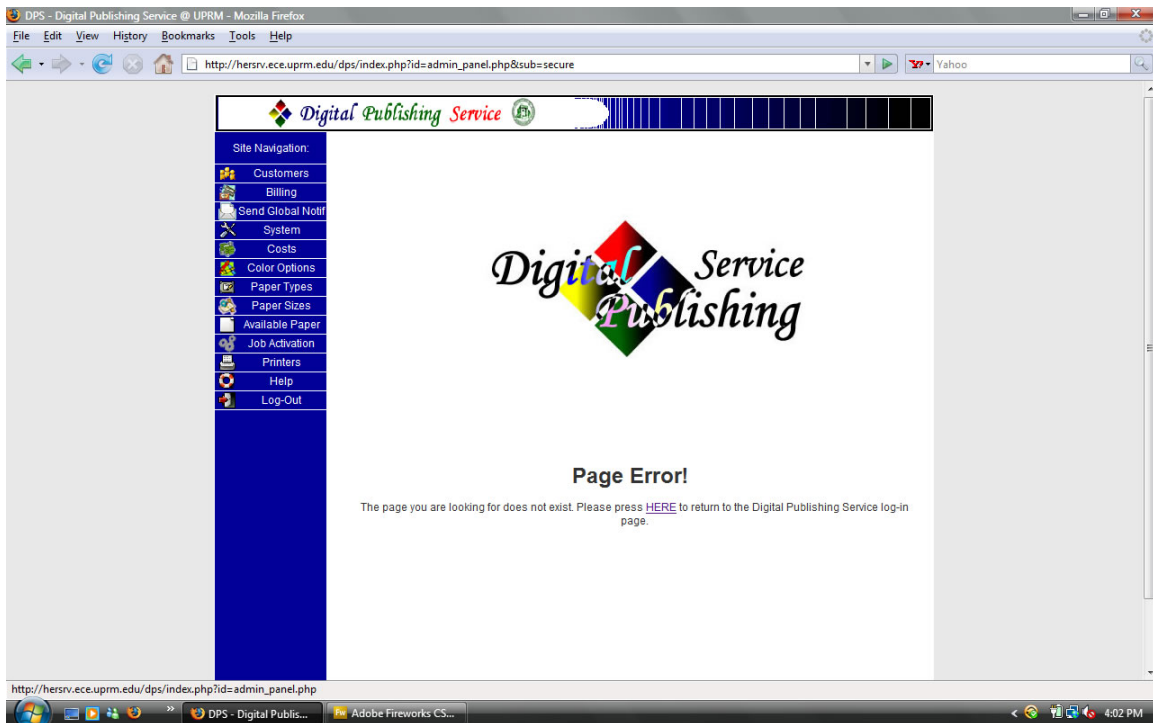


Figure 55: False page error prevention message example.

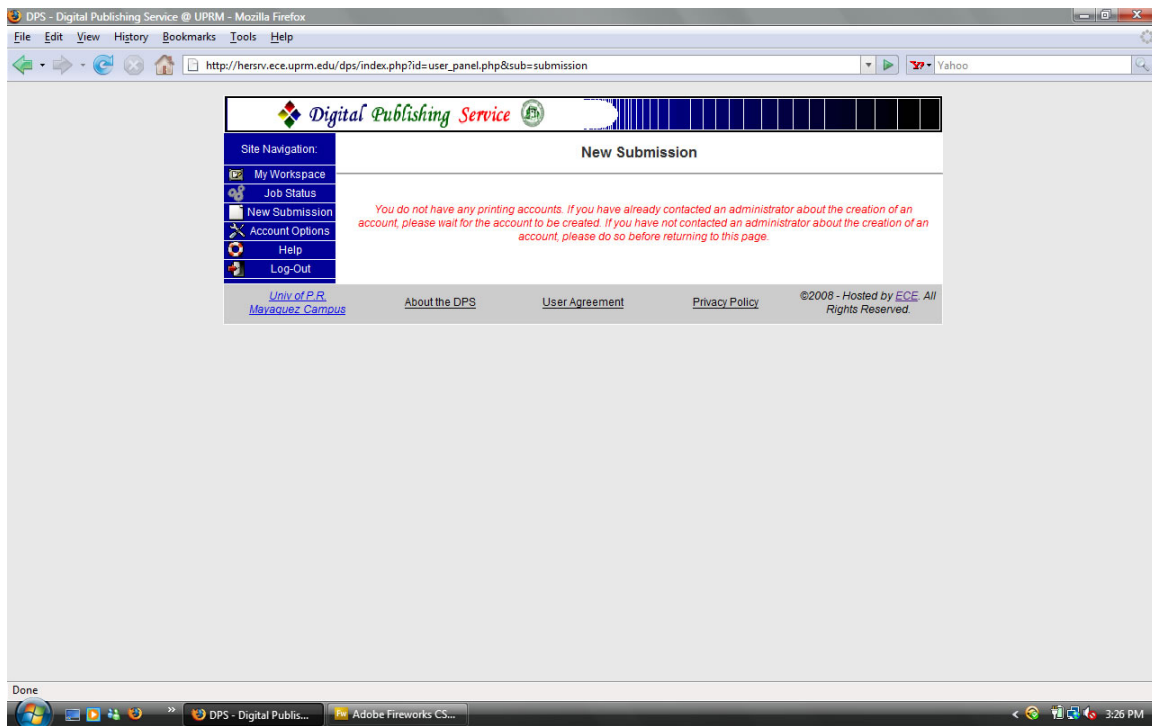


Figure 56: No printing account error prevention message example.

APPENDIX E – EXAMPLE E-MAILS SENT BY DPS



Figure 57: Administrator e-mail notification of new registered user.

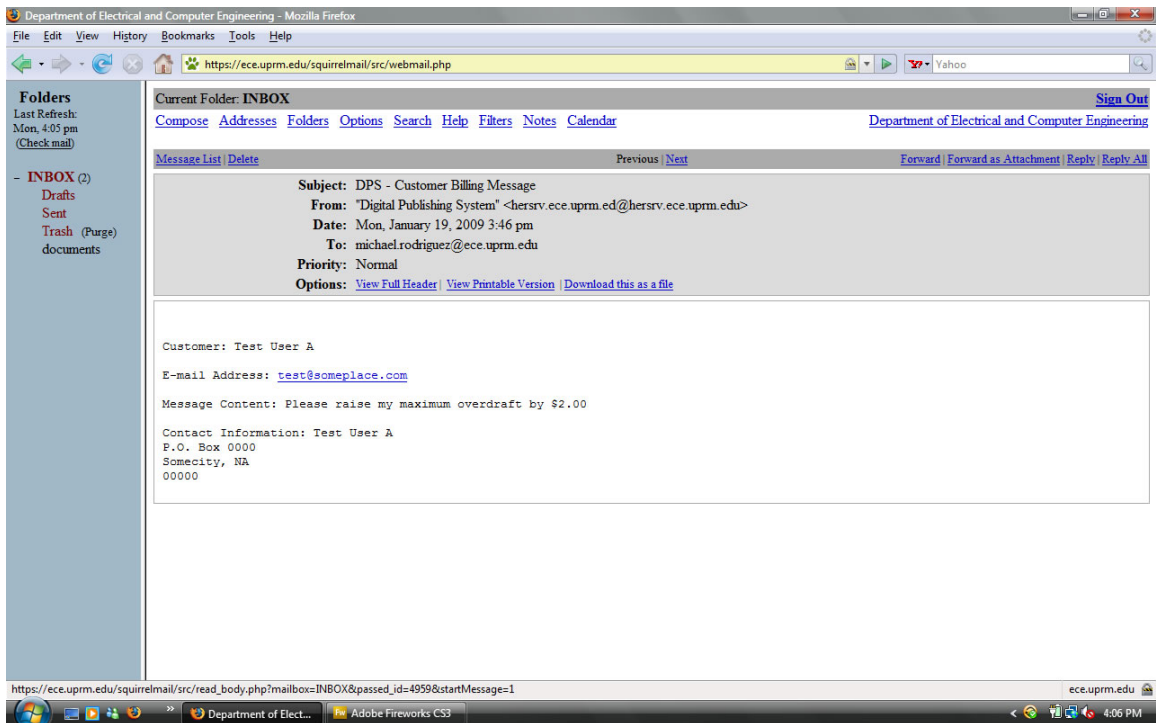


Figure 58: Administrator e-mail from user via billing options tool.

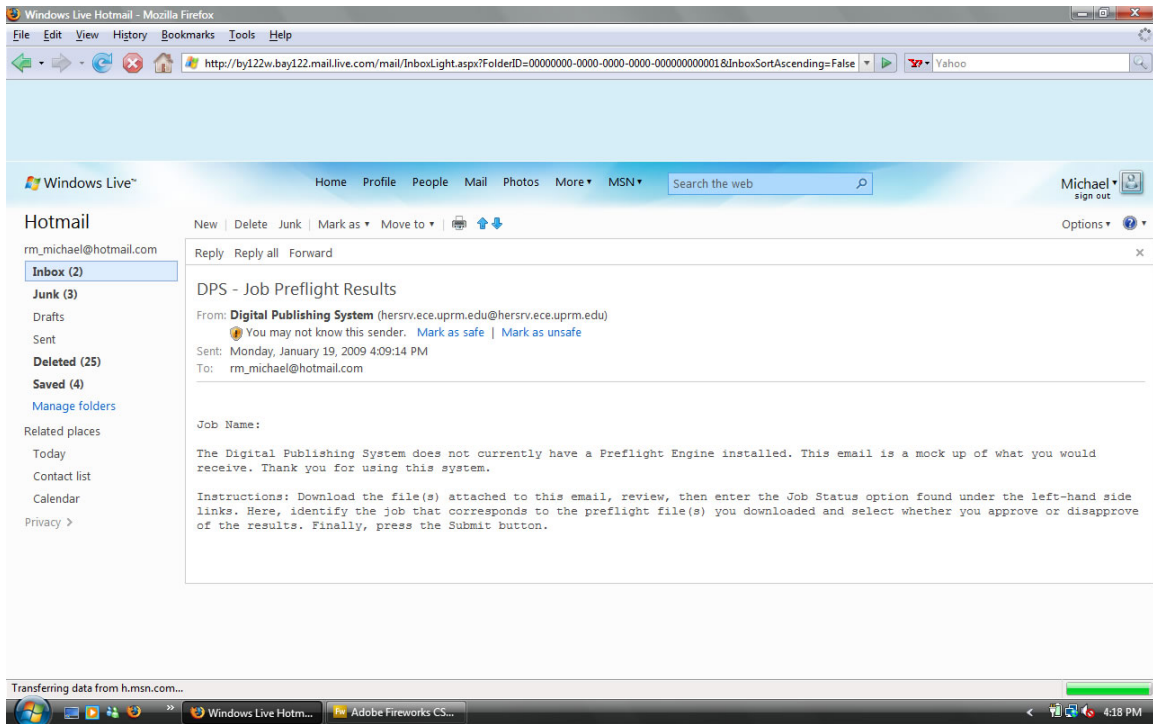


Figure 59: User e-mail notification about preflight results.

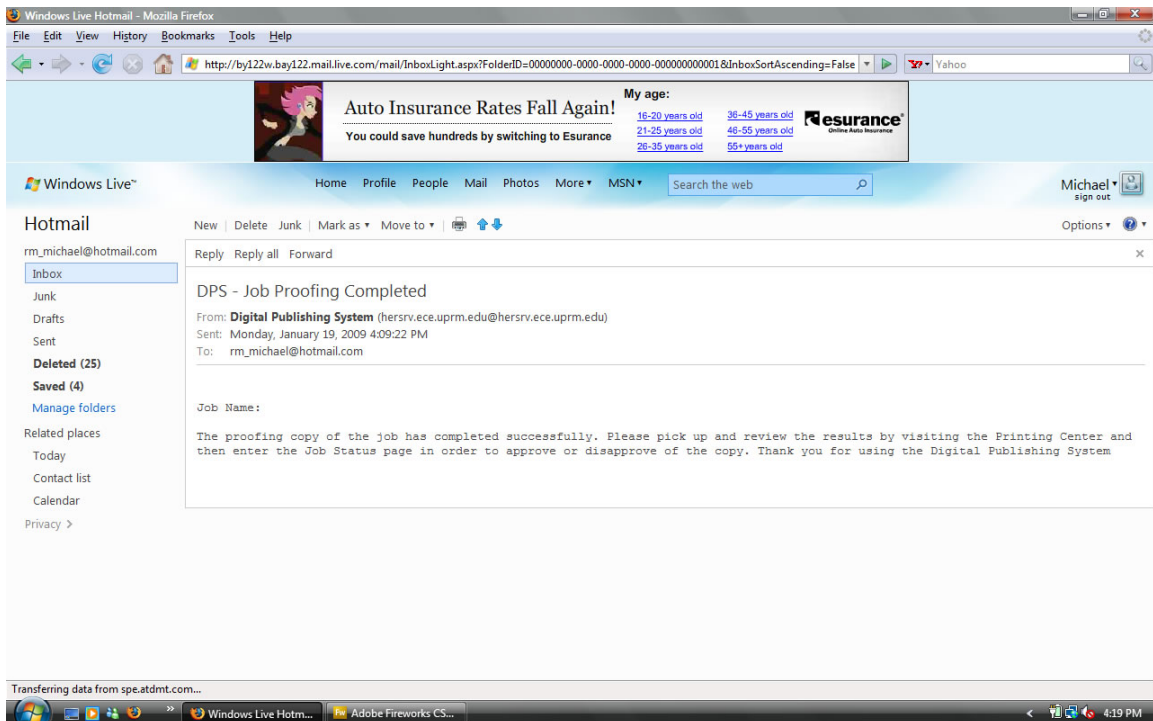


Figure 60: User e-mail notification about proofing copy completion.

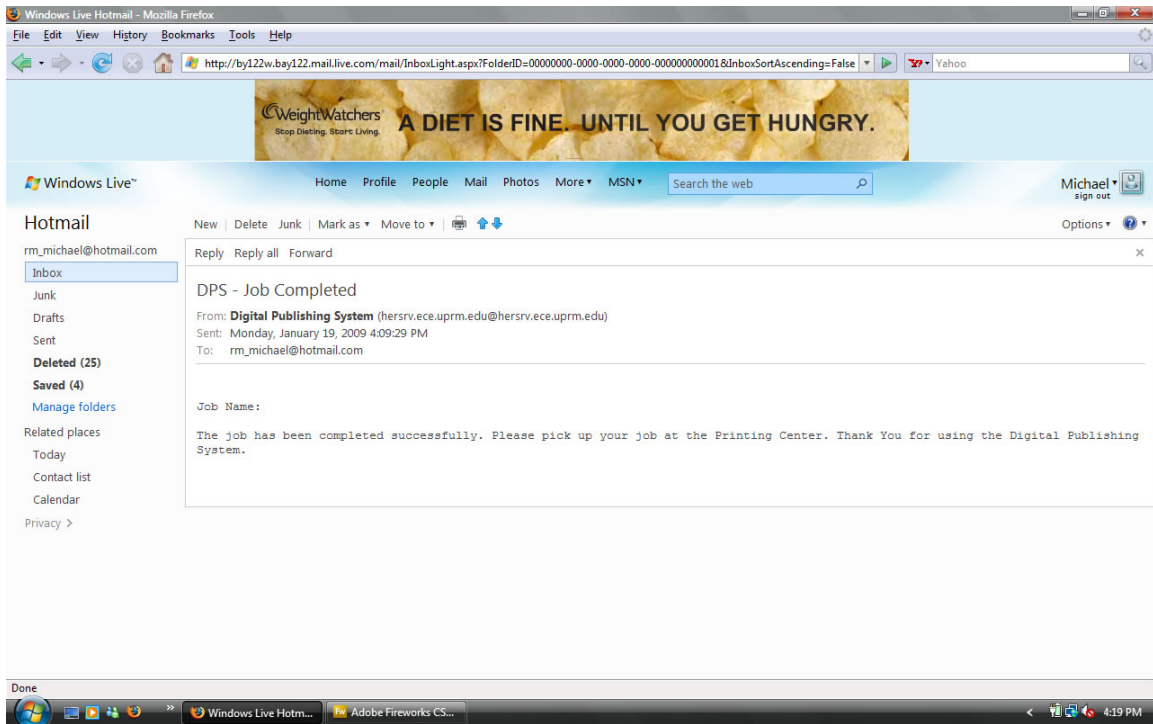


Figure 61: User e-mail notification about printing job completion.

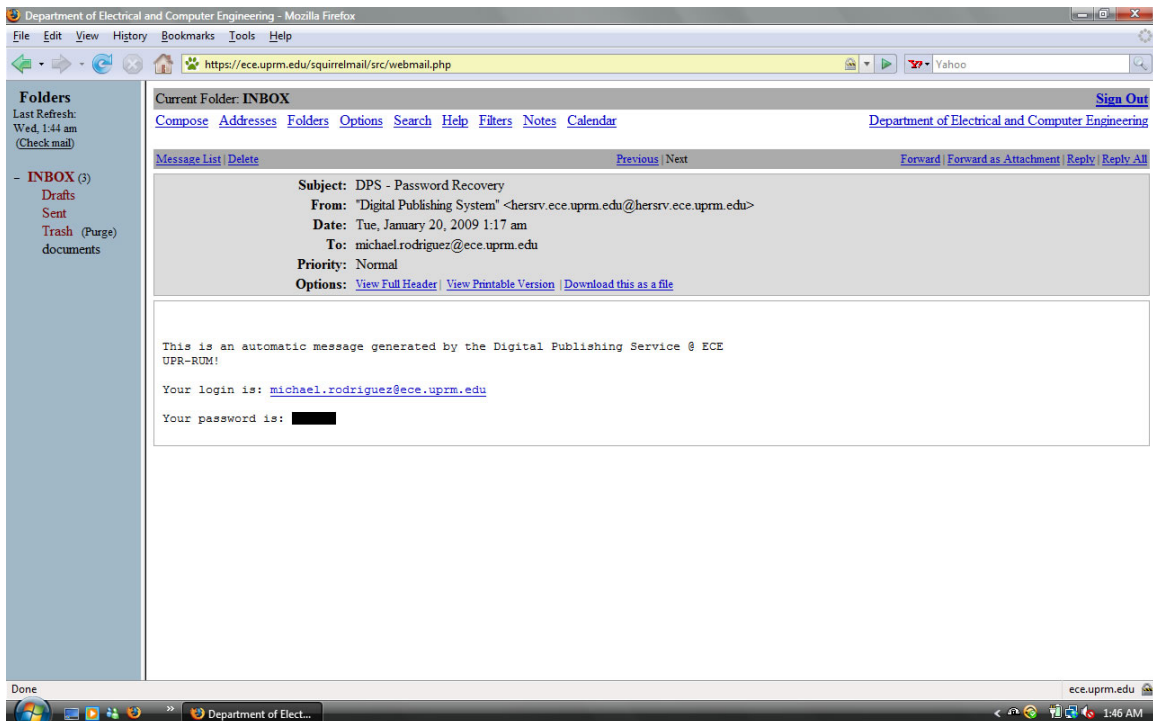


Figure 62: User password recovery e-mail example.