

**DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES**

PAPERLESS OFFICE AUTOMATION

by

Kamil Serhan BİLMAN

December,2006

İZMİR

PAPERLESS OFFICE AUTOMATION

**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of
Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Computer Engineering, Computer Engineering Program**

**by
Kamil Serhan BİLMAN**

December, 2006

İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**PAPERLESS OFFICE AUTOMATION**” completed by **KAMİL SERHAN BİLMAN** under supervision of **ASSIST. PROF. DR. ADİL ALPKOÇAK** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

.....
Assist. Prof. Dr. Adil ALPKOÇAK

Supervisor

.....
Prof.Dr. Cahit HELVACI
(Jury Member)

.....
Assist. Prof. Dr. Malik Kemal ŞİŞ
(Jury Member)

.....
Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor Assist. Prof. Dr. Adil ALPKOÇAK (who completely changed my vision), for his valuable help, suggestions and endless support to complete my thesis.

Finally, I would like to thank my mother Ümit, my father Kamil and my brothers Erhan and Gökhan for their lovely support and encouragement.

Kamil Serhan BİLMAN

PAPERLESS OFFICE AUTOMATION

ABSTRACT

In this study, a system has been designed and implemented that allows to serve PDF forms over a web site, enable users to fill forms and send form data over internet. The system gives advantage of saving and serving forms in PDF, makes office system paperless, and also allows users to digitally sign PDF documents and signature control.

On the server side Windows 2000 server installed with Active Directory, IIS (Internet Information Server), Certificate Server (Root CA) used to create and serve digital signatures. PHP (Hypertext Preprocessor) used to develop a web site which consists of administrator side and user side. Data is stored in a MySQL database. PDF templates stored in the web directory. Adobe Acrobat 7 Professional used to create PDF forms and to convert existing forms in other formats to PDF. php-fdf functions used to manage PDF forms data.

Keywords: Paperless Office Automation, Portable Document Format (PDF), Fillable PDF, Forms Data Format (FDF), Digital Signature, internet based paperless office system

KAĞITSIZ OFİS OTOMASYONU

ÖZ

Bu çalışmada, PDF formlarını web sitesi üzerinden sunan, kullanıcılara doldurdukları PDF formlarını internet üzerinden gönderme imkanı veren bir sistem tasarlanmıştır. Sistem PDF dökümanlarının dijital olarak imzalanmasına ve imza mekanizması geliştirilmesine olanak sağlar.

Sunucu tarafında Windows 2000 Server, Active Directory, IIS, Certificate Server (Root CA) kurulu olup, dijital imza oluşturulmasına imkan verir. İnternet sitesi PHP kullanılarak geliştirilmiş olup, yönetici ve kullanıcı kısımlarından oluşmaktadır. Veriler MySQL veritabanında depolanmaktadır. PDF örnek dosyaları sunucu tarafında tutulmaktadır. PDF formu oluşturulması ve diğer formatlardaki dosyaların PDF formatına dönüştürülmesi için Adobe Acrobat 7 Professional kullanılmıştır. PDF formu verilerinin kontrol edilebilmesi için php-fdf fonksiyonları kullanılmıştır.

Anahtar sözcükler: Kağıtsız Ofis Otomasyonu, Portatif Döküman Formatı (PDF), Doldurulabilir PDF, Form Veri Formatı (FDF), Dijital İmza, internet üzerinden kağıtsız ofis otomasyonu

CONTENTS	Page
THESIS EXAMINATION RESULT FORM	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZ	v
1 CHAPTER ONE INTRODUCTION	1
1.1 Motivation	1
1.2 What is paper?	1
1.3 Advantages of Paperless	2
1.4 Disadvantages of Paperless	3
1.5 Aim of Thesis	4
1.6 Previous Works	4
1.6.1 POEM	4
1.6.2 The Paperless System In Japan Patent Office	4
1.6.3 DocuXplorer	4
1.6.4 I.R.I.S. DocCenter	5
1.6.5 The Paper Tiger	5
1.6.6 WACOM	5
1.7 Thesis Organization	5
2 CHAPTER TWO DIGITAL SIGNATURE & DOCUMENTS	6
2.1 Introduction	6
2.2 Public key infrastructure (PKI)	7
3 CHAPTER THREE PDF & FDF	9
3.1 What Is PDF?	9
3.2 Acrobat Forms	10
3.3 Adobe Acrobat 7.0 Professional	10
3.4 Forms Data Format (FDF)	11
3.5 FDF File Structure	12
3.6 Submitting Data to a Server	12
4 CHAPTER FOUR DESIGN & IMPLEMENTATION	14
4.1 Architecture	14
4.2 Components	14
4.3 Database	14
5 CHAPTER FIVE EXPERIMENTATION	16
5.1 Experimentation	16
6 CHAPTER SIX CONCLUSION	27

CHAPTER ONE

INTRODUCTION

1.1 Motivation

Paperless office system is a collection of software packages for writing, drawing, searching, securing, browsing and latest technology like digital cameras, scanners, audio recorders.

In traditional form, the documents are created with lots of writing tools or software. Documents are in digital format or in printed form. These documents become the data of the paperless office system.

In recent years, the requirement of Paperless Office System has increased. Internet is an advantage and can be widely used in Paperless Office System, makes it possible to share documents online all over the world. Internet based Paperless Office System becomes useful to create and securely share documents online.

It seems that the promised “paperless office” is as much a mythical ideal today as it was thirty years ago. (Sellen, & Harper, 2002)

1.2 What is paper?

Paper is the symbol of the mankind after writing. Paper exists before writing in many forms like rocks and leaves in ancient time.

The term “paperless office” goes as far back as the 1970s. The Xerox Corporation promoted the term.

The fact that technologies are now beginning to look and feel more paper like is a step in the right direction. No longer are we limited by battery power, or large heavy screens. Battery and display technologies are now thinner and lighter than ever and wireless protocols make it necessary for devices to be secured to each other in order to communicate.

There are many advantages of paper. Some of them are:

- Authoring work and making draft printouts.
- Reviewing, annotating and adding comments to the paper is easy.
- Pen and paper used in plans and while organizing the work.

- Hard copy of the reports can be used in discussions and collaborative activities.
- People can communicate each other in an organization. They will print out and deliver it by hand to their colleagues.

1.3 Advantages of Paperless

There are many advantages of a paperless office system with the power of internet. The main goals are:

- No more space, filing cabinets, warehouses needed to store tons of paper, and also no more maintenance needed for storage equipments. Digital documents needs a bit of space.
- It is an easy task to search between digital documents. Comparing digital documents takes less time. Retrieving information from digital documents in many formats as needed.
- Less man power needed, it means labor saving. Man power can be involved into the other tasks in the paperless system.

Scanning documents, take its text and image data with imaging tools and Optical character recognition (OCR), and using a documents sharing software makes it easy to view documents with Personal Digital Assistant (PDA). Our technology makes it easier to become paperless.

Three key elements of the paperless office are electronic imaging, automated document management, and automated workflow-management products. (Shetterly, & Chmura, 1996)

There are many portable devices that allow persons to access, send, read and even mark documents as if it were in paper format. (Carr, 2005)

Internet makes online publication of digital documents possible. People who need information and who provide information can take the advantage of the internet publishing system. There may be people who makes printouts from online documents also. More digital documents online make people to create more paper as a result.

Online publication is a positive step toward achieving the paperless office. (Foster, 1990)

1.4 Disadvantages of Paperless

Digital documents are screen based documents and there are many disadvantages while reading these type of documents.

- Screens are harder to read, they are not so flexible, because of its size limitations a simple one page document can not be easily seen, reading a complex document in the screen is a frustrating task and makes our eyes tired
- Bookmarking in an electronic document is not easy to see and adding and reading comments is not an easy task in digital paper
- Comparing pages, and also comparing different sized pages is not an easy task in digital environment, screen size limitations makes vieving lots of pages at the same time more difficult, more screens needed to view different results at the same time
- Creating, sharing and reviewing digital documents needs complex software systems
- Securing digital documents is a difficult tasks

The paperless office is not always ideal. Also, it is a lot closer to myth than reality. The problem is that electronic document tools often increase the use of paper. (Fox, 2003)

While technology has made these fictions a possibility, the use of paper continues to rise.

While the competitive nature of the global marketplace demands new approaches to “business-as-usual,” most organizations resist changing current processes: thus, a continued reliance on paper. Any movement toward a paperless office will come only with an accompanying change in how an organization views work. (Smart, 1995)

When people are away from their offices, they need paper medium. Because computer hardware has not yet sufficiently evolved.

Paper is more convenient and comfortable. It has been used by all cultures for more than 2000 years. Will there ever be a paperless office?

Online publishing has different requirements in navigation and organization. It does not mean simply dumping text into a computer screen. Putting the documents

online can be done by complex software programs and needs a better work organization.

To be a suitable for paper, an electronic document must be as portable and convenient as paper. (Hayhoe, 1992)

1.5 Aim of Thesis

The aim of the thesis is to design and implement a prototype paperless office automation system based on PDF, running over internet.

We use PDF because of its flexibility. It is an easy task to convert existing documents to PDF. We serve PDF documents in Apache web server and serve them to users in our web interface. Users fill them and send the information only, not the entire PDF document. We get the information then store it in our MySQL database, and use it to populate PDF files when needed.

Data from the database used to create FDF file and then FDF file send to the browser to see the populated PDF file.

1.6 Previous Works

1.6.1 POEM

POEM is a versatile office system that may be characterized by the initials: “Paperless Office and Electronic Mail”. This powerful office system runs on the most popular, low cost computers – PCs – as well as on minis and micros. (Halm, & Schindler, & Zingler, 1988)

1.6.2 The Paperless System In Japan Patent Office

The objective of the Paperless Project of Japan Patent Office is to integrate a total system which provides a broad view and support to the clerical work of the Patent Office. The goal of this project is to achieve innovative efficiency in the field of clerical processing. (Muramatsu, & Takabayashi, & Kosuge, 1992)

1.6.3 DocuXplorer

It offers affordable document management systems that give any size business the ability to quickly find, share and access all document types, both scanned in paper and electronic computer-generated files, from one easy-to-use desktop screen.
<http://www.docuxplorer.com/>

1.6.4 I.R.I.S. DocCenter

I.R.I.S. DocCenter is an Electronic Document Management software suite offering huge archiving and storage capacities. The data is made available on shared drives, via the Internet, or stored on specific storage supports.
<http://www.irislink.com/c2-388/IRIS-DocCenter.aspx>

1.6.5 The Paper Tiger

The Paper Tiger is a revolutionary filing software product designed to help you overcome the problem of disorganization and paper overload. The software even allows you to find both paper files and electronic files (emails, spreadsheets, word processing and virtually any other type of file) on your computer using the new Google Desktop® software. The Paper Tiger combines an easy-to-use file indexing system with the proven paper management methods of the nation's leading authority on office management and organizing.

1.6.6 WACOM

Wacom's pen displays are a natural and efficient way to work with your digital documents. Using the pen directly on the screen gives you a intuitive way to use the electronic inking capabilities of such leading office applications as Word, Acrobat, and PowerPoint. Using a Wacom pen display you will be able to easily annotate and sign documents without the need to print them out, make your edits, and then convert the edited documents back into a digital format to be shared with others.
http://www.wacom.com/lcdtablets/e_paper.cfm

1.7 Thesis Organization

Post of the thesis is organized as follows. Chapter 2 focuses on Digital Signature. The ways to secure documents studied. Chapter 3 examines PDF (Portable Document Format) and FDF (Forms Data Format). The differences of the PDF forms were studied. The ways of sending form data explained. The new paperless office system introduced in Chapter 4.

CHAPTER TWO

DIGITAL SIGNATURE & DOCUMENTS

2.1 Introduction

Document security is a complex problem while working with electronic documents. Paper documents do not tend to have security problems. Computer files can by their nature be duplicated and altered without leaving any trace, and copies are indistinguishable from the original. Some of the important security problems are:

- **Authenticity.** By verifying a signer's digital identity provide authenticity with digital signatures. Recipients can verify the identity of the sender easily.
- **Integrity.** When a digital signature is applied to a document, recipients have more assurance that the information has not been altered since the document was signed. It is hard to make changes to a paper without being noticed.
- **Non-repudiation.** It prevents the signer of the document from denying that they signed the document. Authentication and time-stamping capabilities used.
- **Confidentiality.** You cannot really stop someone reading your mail in the paper world.
- **Availability.** An archived document will always be at hand in future?

We tend to rely on post offices and couriers having auditable processes. (Wilson, 1998)

Methods for maintaining integrity include:

- **Parity bits or cyclical redundancy checking (CRC) functions**—CRC functions work well for unintentional modifications, such as wire interference, but they can be circumvented by a clever attacker.
- **One-way hash**—A one-way hash creates a fixed-length value, called the hash value or message digest for a message of any length. A hash is like a unique fingerprint. With a hash attached to the original message, a recipient can determine if the message was altered by recomputing the hash and comparing his or her answer to the attached hash. Common hashing algorithms are MD5,

SHA-1, and SHA-256. Adobe has adopted the SHA-1 and SHA-256 algorithms because of their wide acceptance as a security standard.

- Message Authentication Codes (MAC)—A MAC prevents an attacker from obtaining the original message, modifying it, and attaching a new hash. In this case, a symmetric key is connected to the MAC and then hashed (HMAC). Without the key, an attacker cannot forge a new message. Adobe uses HMACs where appropriate.

Digital signatures are created using asymmetric key cryptography. For document encryption, a document's author encrypts a document using a public key. Because the recipient is the only person with the private key, he or she is the only one who can decrypt the message. Digital signatures reverse the use of public and private keys for document authenticity. The author encrypts the hash of the message with a private key. Only the public key can correctly decrypt the hash and use it to see if it matches a new hash of the document. Because recipients of the document have the author's public key, they gain greater assurances that the individual who signed the document was the person who encrypted the original hash.

The process that constitutes a digital signature is as follows:

- A hash is created of the original document.
- The digital signature is created, which encrypts the hash with a private key.
- The signature is included with the document.

Adobe Acrobat supports multiple digital signatures placed anywhere in the document. In fact, Adobe Acrobat tracks all previously "signed" versions within the document for easy verification of changes made during the document's lifecycle. Furthermore, Adobe offers a certified signature, which is the first signature on the document. With a certified signature, the author can specify what changes are allowed for integrity purposes. Adobe Acrobat will then detect and prevent those modifications.

2.2 Public key infrastructure (PKI)

Public key infrastructure (PKI) mainly provides a digital certificate that enables a document's recipient to know whether or not a specific public key really belongs to a specific individual. Digital certificates bind a person (or entity) to a public key.

Certificate authorities (CA) issue these certificates and recipients must trust the CA who issued the certificate. X.509 is the widely accepted certificate standard that Adobe uses.

If a certificate expires or a private key is compromised, the CA will revoke the certificate and record the revocation. As part of the process of authenticating a digital certificate, recipients can check the certificate's status. Certificate validity can be checked using the following standard methods:

- Certificate revocation list (CRL)
- Online Certificate Status Protocol (OCSP)

CHAPTER THREE

PDF & FDF

3.1 What Is PDF?

Portable Document Format known as PDF is a universal file format that preserves the fonts, images, and layout of source documents created on a wide range of applications and platforms. PDF is the standard for the secure, reliable distribution and exchange of electronic documents and forms around the world. Adobe PDF files are compact and complete, and can be shared, viewed, and printed by anyone with free Reader software from Adobe. You can convert any document to PDF document using Adobe Acrobat software products as shown in Figure 3.1.

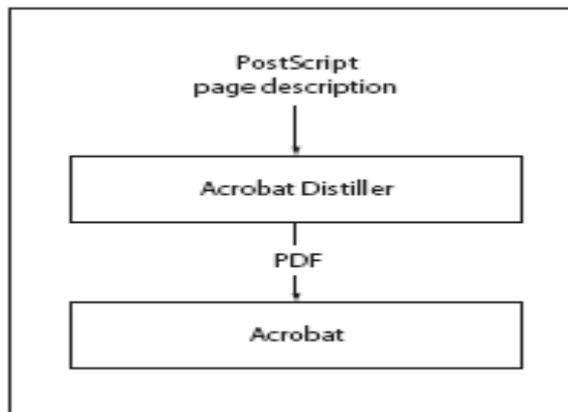


Figure3.1 Creating PDF files using Acrobat Distiller

PDF started off on the dream of a paperless office, as the pet project of one of Adobe's founders, John Warnock. Initially it was an internal project at Adobe to create a file format so documents could be spread throughout the company and displayed on any computer using any operating system.

Adobe already had PostScript as a device and platform independent technology to describe documents and Adobe Illustrator that ran on several platforms, could open and visualize fairly simple PostScript files, even if they were created using other applications. The engineers at Adobe enhanced these two technologies and created both a new file format PDF, and a set of applications to create and visualize these files.

There are several major differences between HTML forms and PDF Forms:

- HTML forms require a web browser, PDF forms require Acrobat or Adobe Reader to be filled out.
- PDF Forms require Adobe Acrobat or Adobe Designer for their creation, anyone with Notepad and a web browser can create/view HTML forms.
- Entered data in a HTML form is generally not saved for later use, the data is submitted to an online process. PDF Forms can save their data to various formats locally for later processing online or viewing in Acrobat/Reader.
- To prefill a HTML form requires the construction of the entire HTML document, to prefill a PDF Form an FDF file can be constructed and later used to open the PDF with its form field data prefilled.
- In a HTML form the data and structure are the same, PDF Forms have a separation between the form fields structure and its data.
- Construction of a HTML form is far simpler than a PDF form due to the underlying structures of HTML and PDF.

3.2 Acrobat Forms

Forms can contain annotations that represent text fields, action buttons, radio buttons, check boxes, list boxes and combo boxes. Acrobat Forms are a group of PDF extensions. Electronically filling out PDF based forms possible with Acrobat forms. These extensions are make a layer on top of a PDF file. You can create the underlying PDF file with any PDF producer. Then you can manually add the fields you want on the form.

3.3 Adobe Acrobat 7.0 Professional

PDF is an excellent program for document distribution. The reader program is freely available over internet, Acrobat Reader, allowing to view PDF documents over any platform. Adobe offers a range of tools to manage, create, secure, convert and distribute PDF documents.

With Acrobat security features, you can use passwords to restrict users from opening, printing, and editing PDF documents. You can use digital signatures to certify PDF documents, and you can encrypt PDF documents so that only an

approved list of users can open them. You can create a security policy that stores security settings and save them for later use,

For password protection, Acrobat supports 128-bit RC4 and 128-bit AES (Advanced Encryption Standard) security methods. You can choose which method to use when securing documents.

For digital signatures and document encryption, Acrobat supports public-key cryptography. Public-key cryptography uses two keys: a public key, which is stored inside a certificate that can be shared with other users, and a private key, called a digital ID, which you do not share with others. The public key certificate is used to encrypt (scramble) documents or to verify digital signatures, and the digital ID is used to decrypt (unscramble) encrypted documents or to create digital signatures.

The first signature in a document is called the author signature. When you add the first signature to a document, you have the option of certifying the document. Certifying a document lets you attest to its contents and specify the types of changes allowed for the document to remain certified. Changes to the document are detected in the Signatures tab. Subsequent signatures to the document are called ordinary signatures.

To sign a document, you must select a digital ID, which contains the signature information that you can share with other users in a certificate. You can create a self-signed digital ID, or you can obtain a digital ID from a third-party provider. Using certificates, other users can validate your signatures, and you should validate the signatures of others.

3.4 Forms Data Format (FDF)

FDF is used when submitting form data to a server, receiving the response, and incorporating it into the interactive form. It can also be used to export form data to stand-alone files that can be stored, transmitted electronically, and imported back into the corresponding PDF interactive form. In addition, beginning in PDF 1.3, FDF can be used to define a container for annotations that are separate from the PDF document to which they apply.

FDF is based on PDF; it uses the same syntax and basic object types, and has essentially the same file structure. However, it differs from PDF in the following ways:

- The cross-reference table is optional.
- FDF files cannot be updated. Objects can only be of generation 0, and no two objects can have the same object number.
- The document structure is much simpler than PDF.
- The length of a stream may not be specified by an indirect object.

FDF uses the MIME content type `application/vnd.fdf`. On the Windows and UNIX platforms, FDF files have the extension `.fdf`; on Mac OS, they have file type 'FDF'.

3.5 FDF File Structure

An FDF file is structured in essentially the same way as a PDF file but contains only those elements required for the export and import of interactive form and annotation data. It consists of three required elements and one optional element as shown in Figure 3.2.

- A one-line header identifying the version number of the PDF specification
- A body containing the objects that make up the content of the file
- An optional cross-reference table containing information about the objects
- A trailer giving the location of various objects within the body of the file

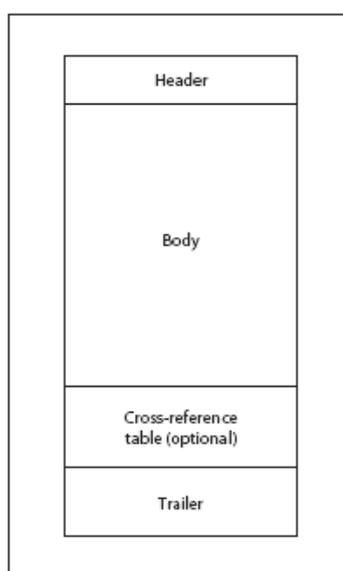


Figure 3.2 FDF file structure

3.6 Submitting Data to a Server

After filling in an Acrobat form, the user must click on a button to submit the data to a server. Submitted data may be in one of the following formats:

- HTML-compatible (MIME type application/x-www-form-urlencoded): The format submitted is identical to HTML form submissions. You can use existing CGI scripts for HTML forms to parse data in this format.
- FDF (an Acrobat-specific Forms Data Format: MIME type application/vnd.fdf).
- XFDF (for XML-based FDF files): MIME type application/vnd.adobe.xfdf.

The creator of the form decides which format to use.

The URL submission target is not restricted to the http scheme. For example, it can also be the mailto scheme like mailto:someuser@somecompany.com.

FDF exports as an FDF file. You can choose to export the form fields data, comments, incremental changes to the PDF file, or all three. The Incremental Changes To The PDF option is useful for exporting a digital signature in a way that is easily read and reconstructed by a server.

XFDF exports as an XML file. You can choose to export the form fields data, comments, or both. PDF exports the entire PDF file that is your form. Although this creates a larger file than the FDF option, it is useful for preserving digital signatures.

You can parse an FDF file to extract data from the individual fields. You can then use this data to populate a database or provide input to some other application. You can also generate FDF data and save it to a file or buffer to populate a PDF file. Exporting the data in FDF format allows you to do:

- Populate the same PDF file on the client with new data
- Send graphical information back to a PDF file
- Alter a PDF file on the client
- Construct a PDF file with templates

CHAPTER FOUR

DESIGN & IMPLEMENTATION

4.1 Architecture

Windows 2000 Server installed with Active Directory, Certificate Server (Enterprise Root CA), IIS to create certificates. Second server is an Apache web server, PHP, MySQL installed, used to store PDF document templates, to serve web site, and to store data on MySQL database. Adobe Designer 7.0 and Adobe Acrobat and Acrobat Distiller used to create PDF documents, to convert existing documents to PDF and to make forms on PDF documents.

On the server side Windows 2000 server installed with Active Directory, IIS (Internet Information Server), Certificate Server (Root CA) used to create and serve digital signatures. PHP (Hypertext Preprocessor) used to develop a web site which consists of administrator side and user side. Data is stored in a MySQL database. PDF templates stored in the web directory. Adobe Acrobat 7 Professional used to create PDF forms and to convert existing forms in other formats to PDF. php-fdf functions used to manage PDF forms data.

4.2 Components

Two servers used in our system. First server is a Windows 2000 server, installed with Active Directory, IIS (Internet Information Server), Certificate Server (Root CA) used to create and serve digital signatures. Second server is an Apache web server and MySQL database installed. PHP (Hypertext Preprocessor) used to develop paperless office web site which consists of administrator side and user side. PDF templates stored in the Apache server. Adobe Acrobat 7 Professional used to create PDF forms and to convert existing forms in other formats to PDF. php-fdf functions used to manage PDF forms data.

4.3 Database

Database created as shown in Figure 4.1.

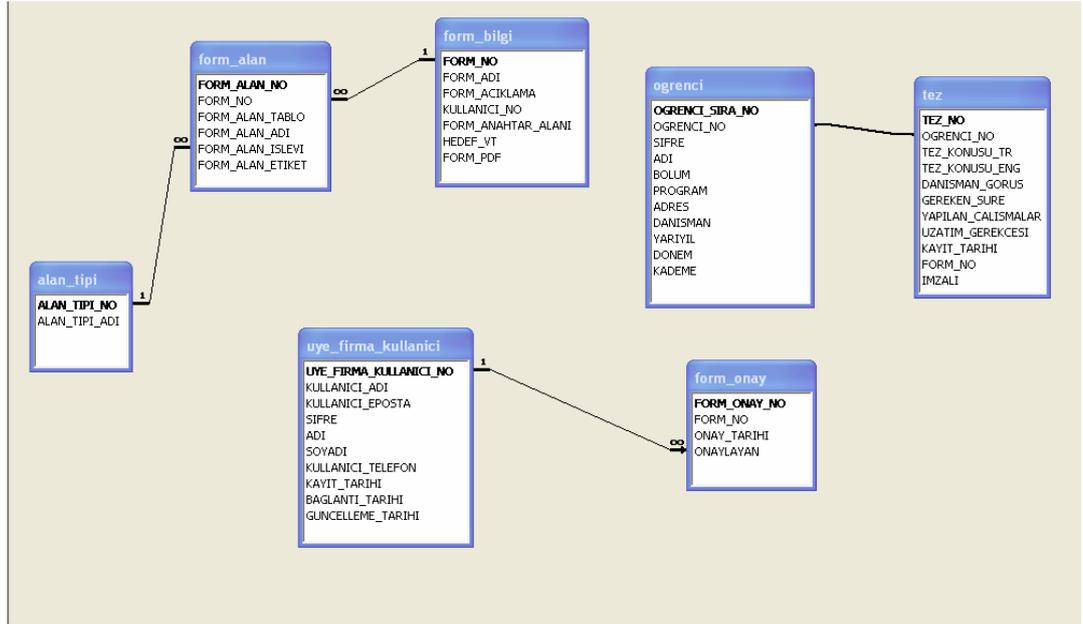
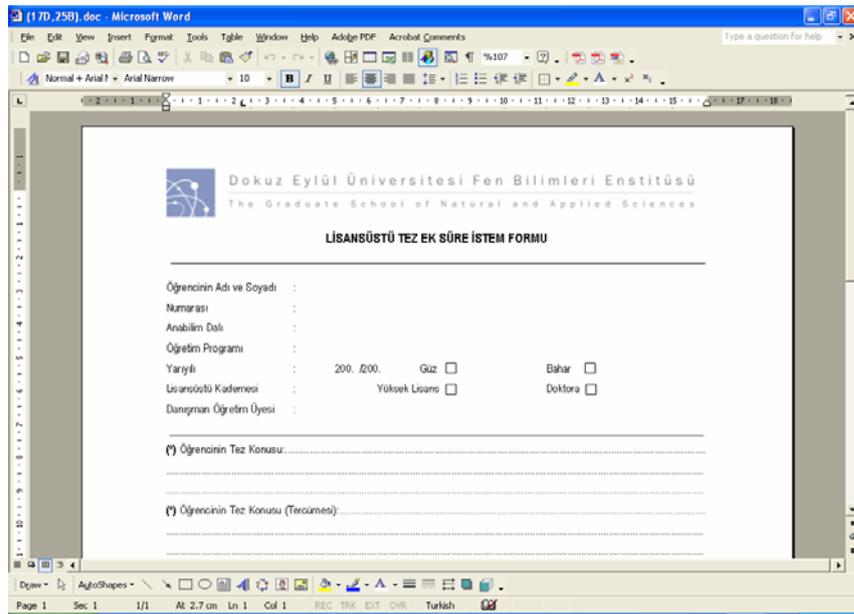


Figure 4.1 Database design

CHAPTER FIVE EXPERIMENTATION

5.1 Experimentation

At first, existing forms as shown in Figure 5.1 converted to PDF format using Acrobat Professional 7 program as shown in Figure 5.2.



The screenshot shows a Microsoft Word document titled "(170_258).doc". The document content is a form for a postgraduate thesis extension request. The header includes the logo of Dokuz Eylül University and the text "Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü" and "The Graduate School of Natural and Applied Sciences". The title of the form is "LİSANSÜSTÜ TEZ EK SÜRE İSTEM FORMU". The form contains several fields for personal and academic information, including name, number, department, program, year, and degree level. There are also checkboxes for "Güz" (Autumn) and "Bahar" (Spring) semesters, and "Yüksek Lisans" (Master's) and "Doktora" (PhD) degrees. The form ends with two lines for the student's thesis topic and its title.

Figure 5.1 Example form in MS Word doc format.

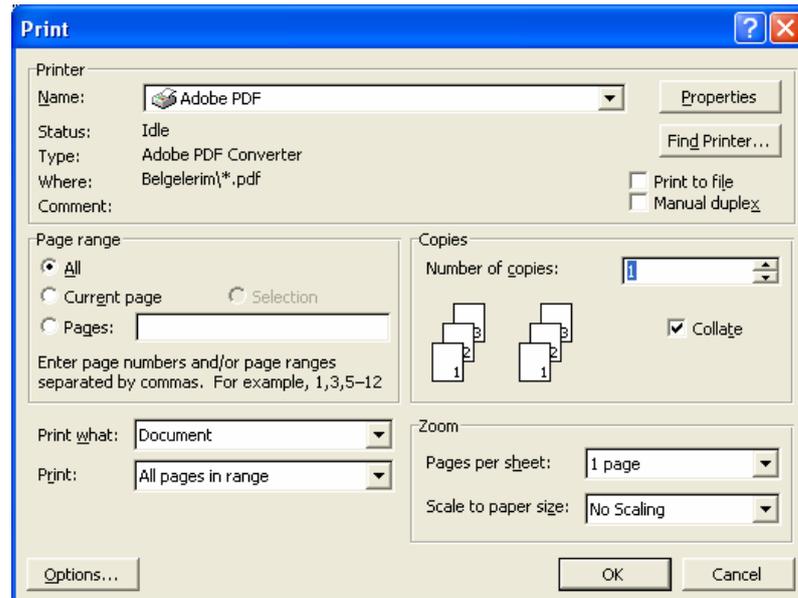


Figure 5.2 Document conversion done by PDF printing facility.

Adding form fields like button, check box, combo box, list box, radio button, text field, digital signature field to a PDF form with Adobe Acrobat forms toolbar as shown in Figure 5.3. A submit button added to the PDF form, and by the submit button, users can send FDF data by email or by HTML page which designed to store data to database. The method selection screen shown in Figure 5.4.

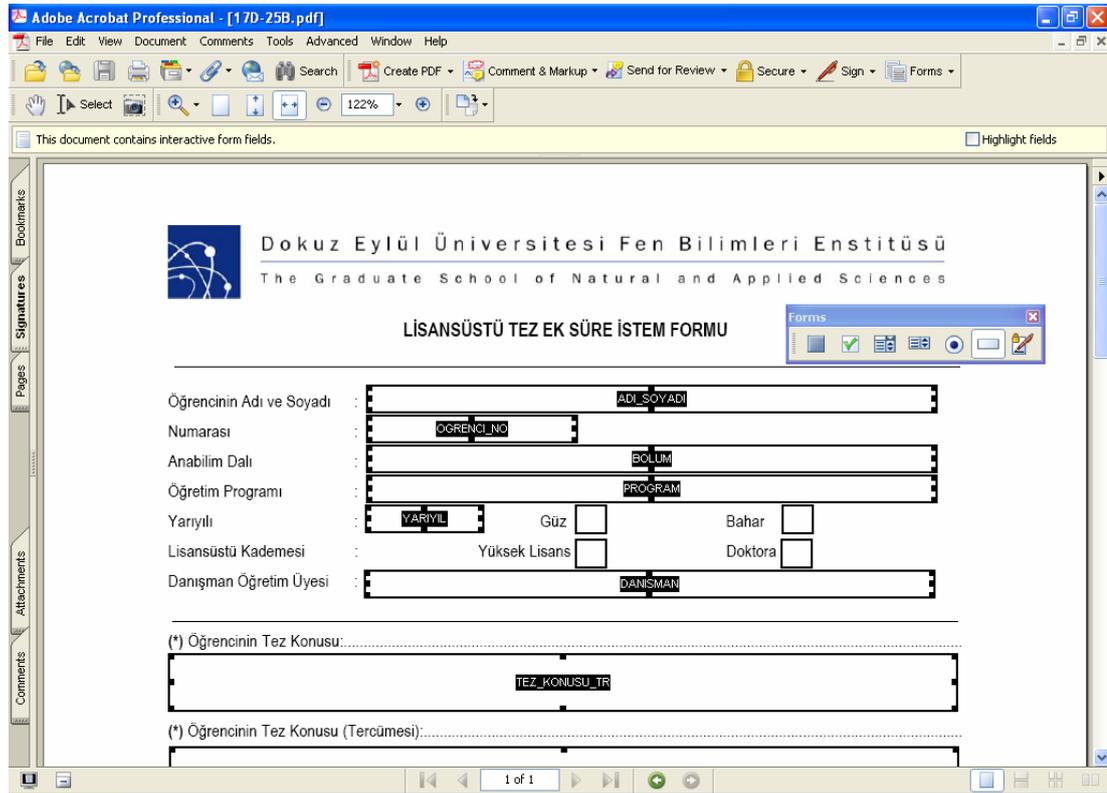


Figure 5.3 Adding form fields to a PDF form in Adobe Acrobat Professional

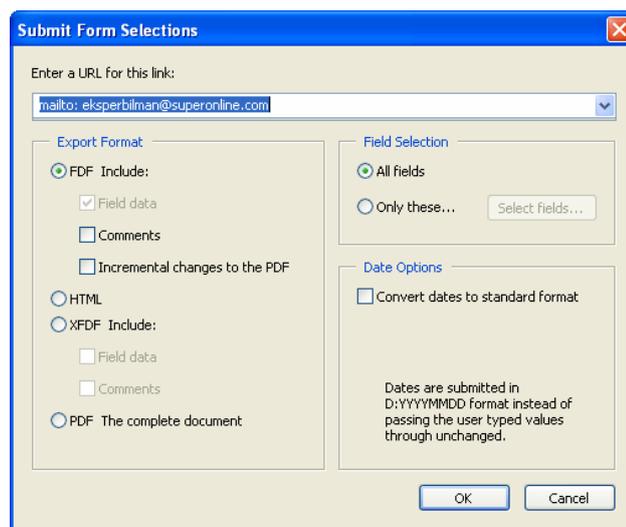


Figure 5.4 Form data export format selection

Completed PDF with form fillable as shown in Figure 5.5. User can submit the data by submit button. And a digital signature field exists in the form.

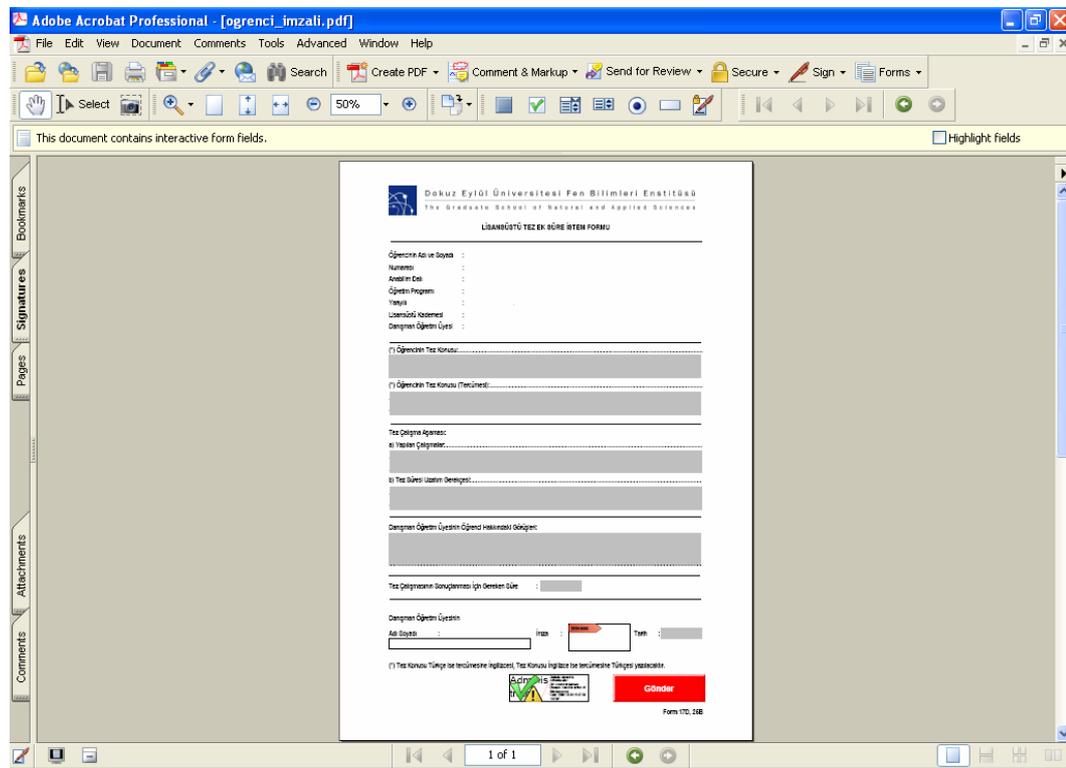


Figure 5.5 Completed fillable PDF form.

There are two types of users; admin and students in our system. Admin user has a username and a password. Admin user can log in to the system as shown in Figure 5.6.

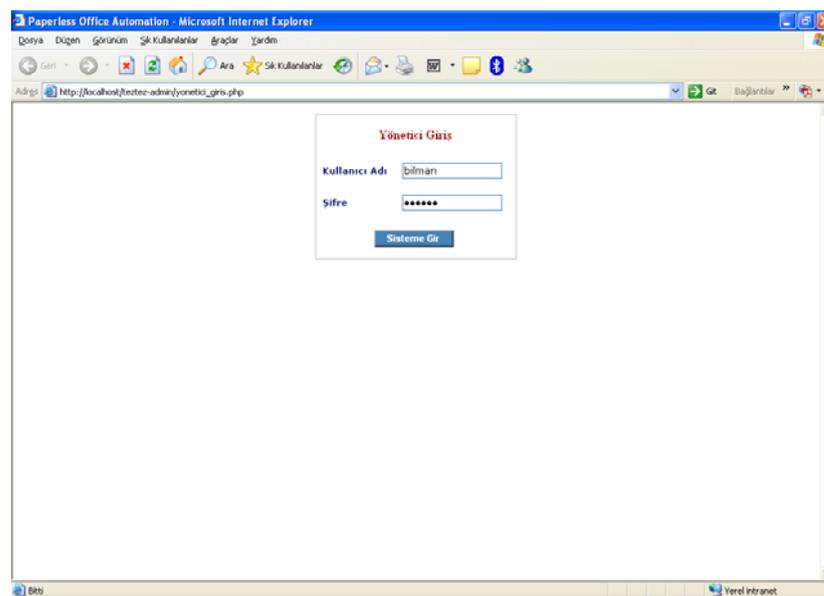


Figure 5.6 Admin user login screen

In the admin section, user can upload fillable PDF forms. A fillable PDF form icon appears in the list after upload as shown in Figure 5.7. Uploaded PDF forms stored in pdf_formlar directory at the web server. Users can give PDF form a name. Users can upload as many as PDF forms into the system. The list of the forms in the admin section as shown in Figure 5.7.

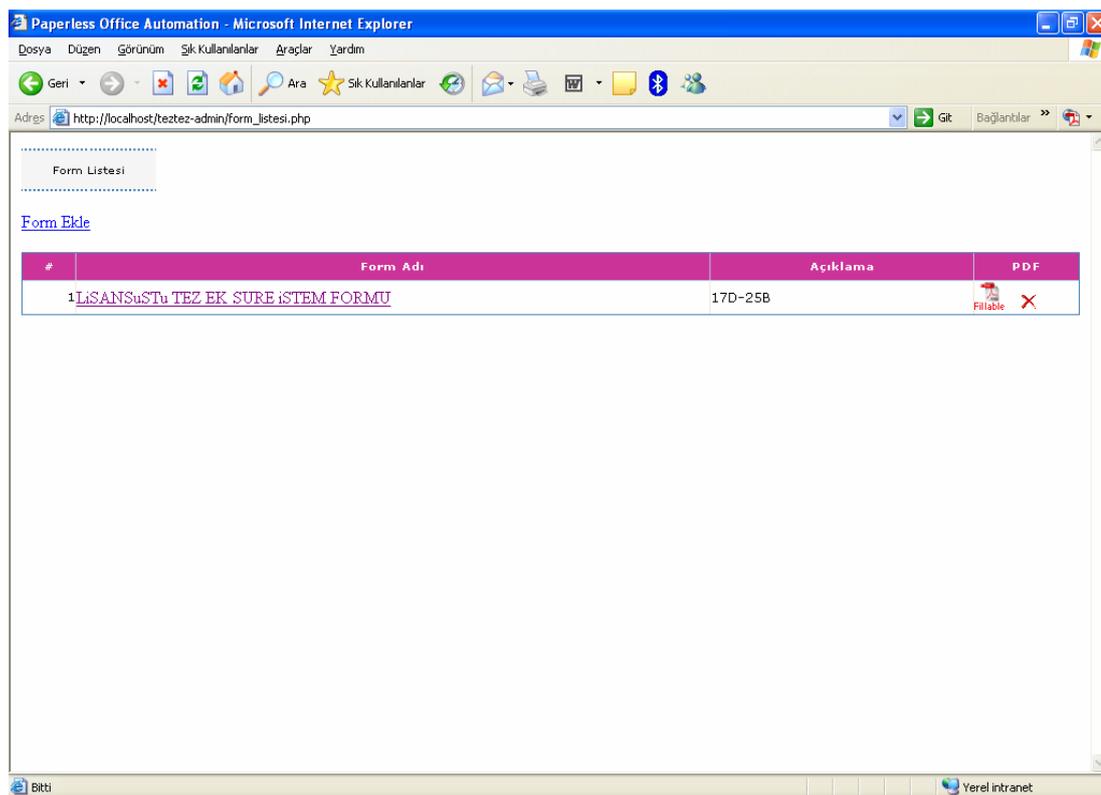


Figure 5.7 The list of the fillable forms in the admin section.

Every PDF forms have two descriptive fields, target database and key field. These fields are hidden and users can not see them on the PDF form. Target database field used when submitting the PDF form data into the database. The key field used to get and show the data on the PDF form. (Figure 5.8)

Form Bilgileri

Form Adı: LİSANSÜSTÜ TEZ EK SURE İSTEM FORMU

Form Anahtar Alanı: OGRENCI_NO

Hedef Tablo: tez

Form Açıklama: 17D-25B

Güncelle

Alan Ekle

#	Alan Etiketi	Alan Tablosu	Alan Adı	İşlevi
1	Adı	ogrenci	ADI	Etiket
2	Anabilim Dalı	ogrenci	BOLUM	Etiket
3	Öğretim Programı	ogrenci	PROGRAM	Etiket
4	Yarıyıl	ogrenci	YARIYIL	Etiket
5	Dönem	ogrenci	DONEM	Etiket
6	Lisansüstü Kademesi	ogrenci	KADEME	Etiket

Figure 5.8 PDF form fields list in the admin user's form section.

Users can describe two types of fields over PDF forms, input field and label fields. Input fields used to store information, label fields used to show information from the database. (Figure 5.9)

Alan Ekle

Alan Etiketi:

Alan Tablosu:

Alan Adı:

Alan İşlevi: Seçiniz

Kaydet

Figure 5.9 Inserting new fields to PDF form..

In the student section, student users can log in to the system with their user name and password as shown in Figure 5.10. After they log into, they see the list of the online fillable PDF forms (Figure 5.11).

Figure 5.10 Student section.

#	Form Adı	Açıklama	PDF
1	LİSANSUStu TEZ EK SURE İSTEM FORMU	17D-25B	Fillable

Figure 5.11 List of fillable PDF forms.

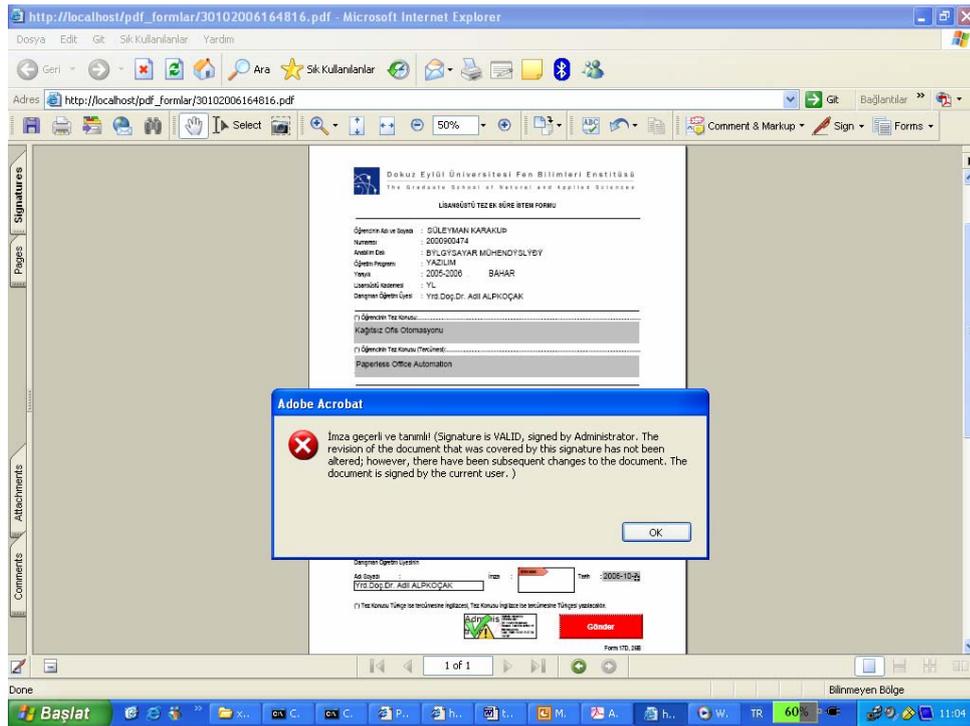


Figure 5.14 Javascript checking the signature, if it is valid or not.

Student user can save the PDF file into the local hard disk. The student fill in the input fields and sign the signature field in the PDF form, student submits the PDF form. A FDF file saved and send to admin user's email address as an attachment. (Figure 5.15) Admin user is a valid user in our database, student user also.

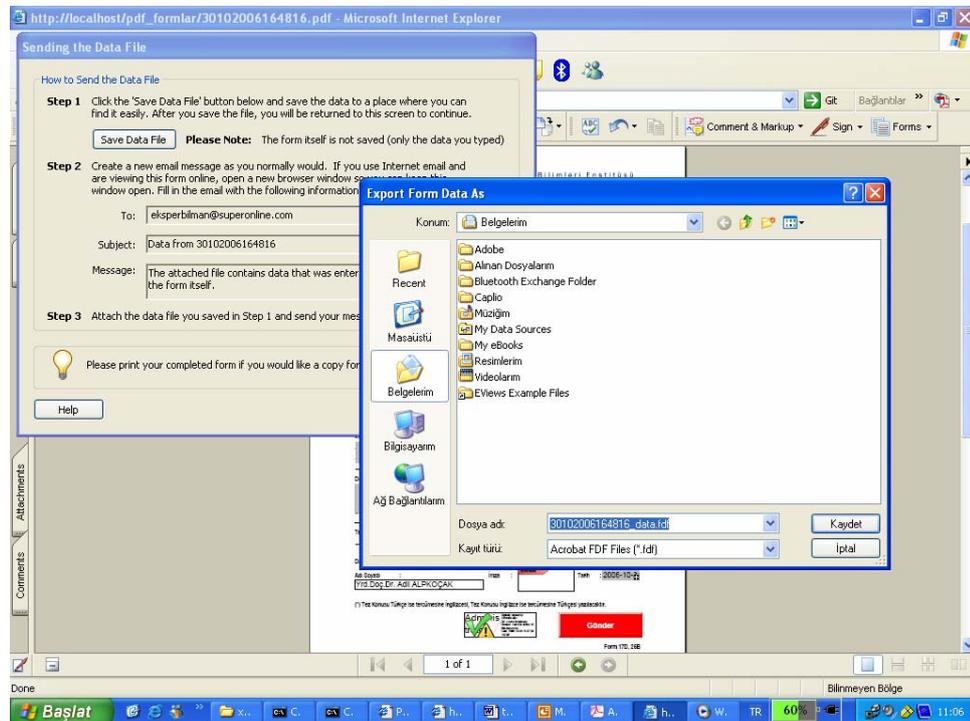


Figure 5.15 Student user saves the FDF file and send it as an email attachment.

Admin user take the FDF file from an email attachment, saves it to hard disk and upload it to the system. (Figure 5.16) A PHP file store the data in the FDF file into the database, by using the target database and key fields. After the form data uploaded into the system, data in the form stored in a temporary database as unchecked, until it will be checked by another admin user. (Figure 5.17) PDF document can be send to other admin users for aproving. Users for approving or reviewing can be described in the database.

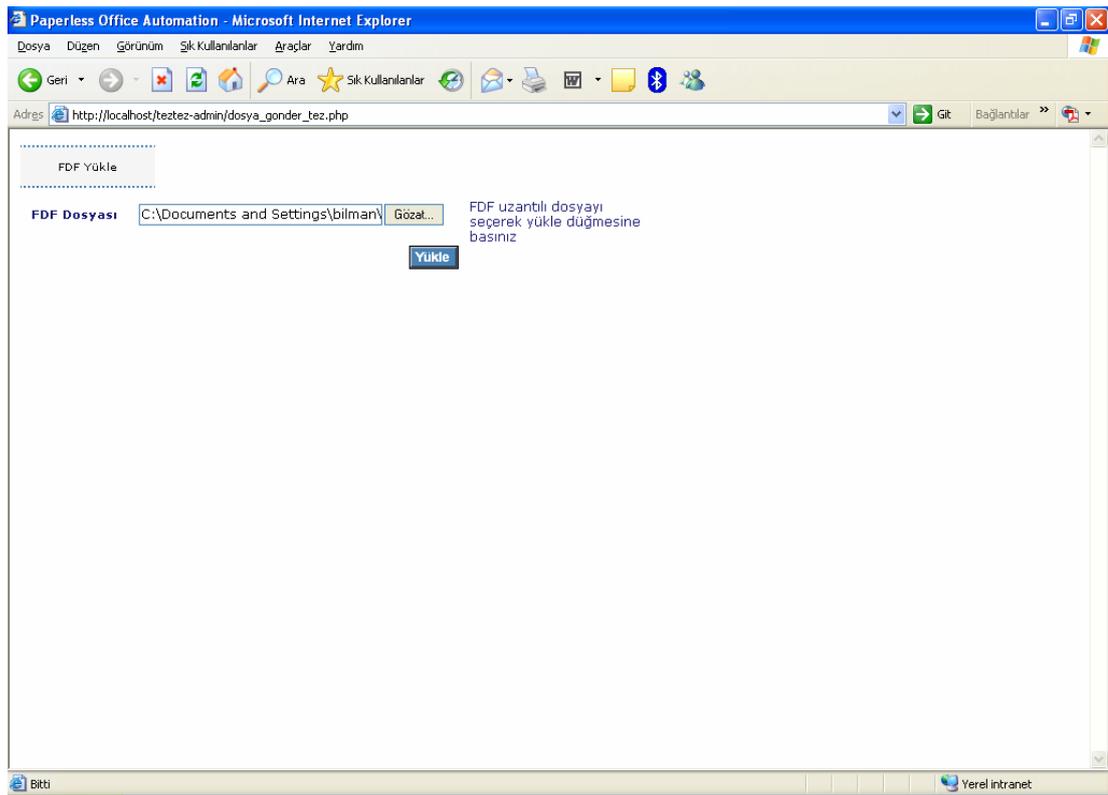


Figure 5.16 Admin user uploads FDF file to the system.

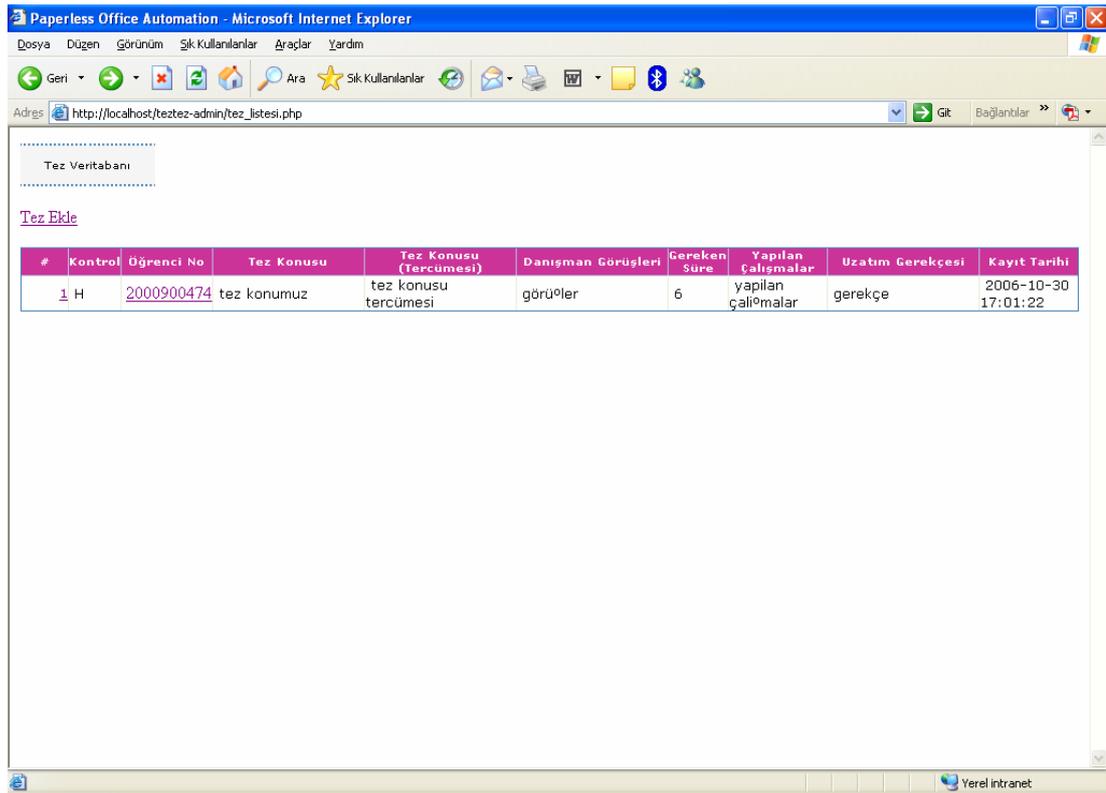


Figure 5.17 Admin user checks the incoming forms data after uploading FDF file.



Figure 5.18 Signing a signature field and writing a reason for signing.

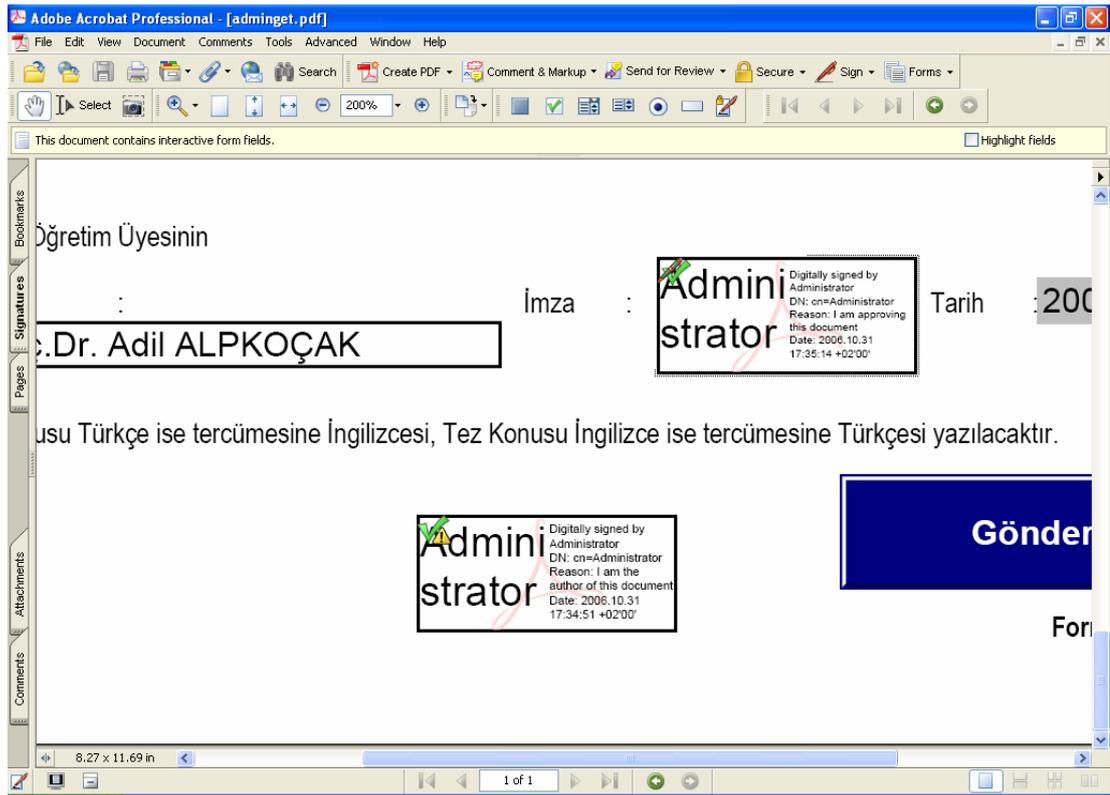


Figure 5.19 There are two signature fields exists in the fillable PDF form, and they are successfully signed.

Javascript can be used in Adobe PDF forms. The script can be embedded on the fly. The following script used to check the signature field, before sending data to the php which stores the data to the database and set the record as checked, and moves the record from the temporary table.

CHAPTER SIX

CONCLUSION

In this study, a prototype system for paperless office automation based on PDFs, has been designed and implemented with two servers. First server is a Windows 2000 server, installed with Active Directory, IIS (Internet Information Server), Certificate Server (Root CA) used to create and serve digital signatures. Second server is an Apache web server and MySQL database installed. PHP (Hypertext Preprocessor) used to develop paperless office web site which consists of administrator side and user side. Data is stored in a MySQL database. Fillable PDF documents stored in the Apache server. Administrators and users have username and password to sign in to the system.

Documents in other formats like word, converted into the PDF format with Adobe Acrobat and Adobe Distiller. Forms created on the PDF documents with Acrobat Forms Tool. After forms created, PDF documents become fillable PDF forms. Administrator can put digital signature before uploading the form to the system. After the form uploaded to the system, users able to download fillable PDF forms.

Two types of fields used while creating PDF forms. One of them is a label field, which shows personal information of the user who want to download the form. Another field is input field that the user must complete before sending the form over internet. FDF send as an attachment to the described reviewer's email address.

Documents already used in other formats converted to PDF easily. No documents transfered in our paperless office system, only the forms data send. Form feature added to PDF documents. Fillable PDF documents served over internet. PDF forms filled with user's data and if needed. Users filled the PDF documents and than send only form data in FDF format by a submit button or email it. Digital signatures can be used to secure documents.

Javascript used to enable the PDF forms to check digital signatures if they are valid or not, before the forms are submitted to the database.

The system gives you the ability to serve the latest version of a document. One empty fillable PDF form served on the Apache server. The change in one document effects all of the documents created for users.

Distribution list can be implemented as a future work. Users can be described as form creator, reviewer or approver. Document modification history can be saved and incremental changes can be viewed. Field locking may be possible against some users.

REFERENCES

- Active Directory Operations Overview. Retrieved September 19, 2006, from <http://www.microsoft.com/technet/prodtechnol/windows2000serv/technologies/activedirectory/maintain/opsguide/part1/adogd01.msp#ECB>
- Adobe Acrobat. Retrieved September 19, 2006, from <http://www.adobe.com/products/acrobat/>
- Adobe Acrobat Pro. Retrieved September 19, 2006, from <http://www.adobe.com/products/acrobatpro/>
- Adobe Acrobat Security. Retrieved September 19, 2006, from <http://www.adobe.com/security/>
- Apache. Retrieved September 19, 2006, from www.apache.org
- Carr, M. R. (2005). *An analysis of the feasibility of a paperless environment – the case of the Mona School of Business*. *Management of Environmental Quality: An International Journal*, Vol. 16 No. 4, pp. 286-290.
- FDF. Retrieved September 19, 2006, from <http://partners.adobe.com/public/developer/acrobat/fdf/topic.html>
- FDF functions. Retrieved September 19, 2006, from www.php.net/fdf
- Foster, M. S. (1990). *Hardcopy to online publication – It can be done*. Los Alamos National Laboratory, IEEE, pp. 51-57.
- Fox, S. (2003). *Is the latest technology the only good technology?* *IEE Manufacturing Engineer*, Volume October/November, pp. 24-27.
- Halm, J., & Schindler, S., & Zingler, K. (1988). *POEM – An office System for International Use*, IEEE, pp.87-93.

Hayhoe, G. F.(1992). *There's got to be a desk here somewhere: Dispelling the Myth(?) of the paperless office*. IEEE, pp.654-658.

Microsoft Certificate Server (MCS). Retrieved September 19, 2006, from <http://www.windowsitlibrary.com/Content/405/17/1.html>

Muramatsu M., & Takabayashi O., & Kosuge K. (1992). *The construction of the paperless system in Japan Patent Office*. NTT Data Communications Systems Corporation, IEEE, pp. 278-287.

MySQL. Retrieved September 19, 2006, from www.mysql.com

PDF reference site. Retrieved September 19, 2006, from http://partners.adobe.com/public/developer/pdf/index_reference.html

Sellen, A., & Harper, R. (2002). *The Myth of the Paperless Office*. MIT Press, Cambridge, MA.

Shetterly, R., & Chmura, A. (1996). *Creating the paperless office*. IEEE Software, pp.124-127.

Smart, K. L. (1995). *The Paperless Office: Facts and Fictions*. Brigham Young University Department of Management Communication. IEEE, pp.141.

Wilson, S. (1998). *Digital signatures and the future of documentation*. MCB University Press, Information Management & Computer Security, Volume 7/2, pp.83-87.

Windows 2000 Certificate Services. Retrieved September 19, 2006, from <http://www.microsoft.com/technet/prodtechnol/windows2000serv/depoy/depopt/2000cert.msp>