Virtual Paper Based Filing System

By

Ali Haider	**Registration # 392**
Asadullah Ilyas	**Registration # 396**
Fazeel Ashraf	**Registration # 398**

A Project report submitted in partial fulfillment
of the requirement for the degree of
Bachelors in Information Technology

Department of Computing

School of Electrical Engineering & Computer Science
National University of Sciences & Technology
Islamabad, Pakistan
2014

CERTIFICATE

It is certified that the contents and form of thesis entitled **“Virtual Paper based Filing System”** submitted by *Ali Haider* **(***2010-NUST-SEECS-BIT-392***),** *Asadullah Ilyas* **(***2010-NUST-SEECS-BIT-396***)** *and Fazeel Ashraf (2010-NUST-SEECS-BIT-398)* have been found satisfactory for the requirement of the degree.

Advisor: Nasir Mahmood

Co-Advisor: Bilal Ali

DEDICATION

To Allah the Almighty

&

To my Parents and Faculty

ACKNOWLEDGEMENTS

We are deeply thankful to our advisor and co-advisor, Mr. Nasir Mahmood, and Mr. Bilal Ali for helping us throughout the course in accomplishing our final project. Their guidance, support and motivation enabled us in achieving the objectives of the project.

We are also thankful to ITS department, IGIS and Headquarters for their valuable suggestions and time. Finally, we are extremely thankful to Almighty Allah, for guiding us through every crucial and challenging time throughout the project.

Table of Contents

Table of Diagrams

ABSTRACT

If you go in any organization, whether it is government or private, you will see many cabinets filled with files. All those files are bursting with different documents on same subject. And main purpose of those files and documents to be there is to maintain history. That is, just in case if a previous document is needed in future, it should be there. But there are very few documents that are needed in future. Why save all of them and occupy physical space? Because, we do not know which few documents we will be needing in future. And hence, all the documents must be saved, and all the cabinets and space is filled. Now, if a document is needed in future, finding that document is also a tremendous and time consuming activity. So why not automate it.

We have created a web application that can do all that official paper work that was being done manually. As it is a web application, all the data is in digital form. Tons of data can be stored in a *physically small* hard disk now a days. So, all the physical space that was being wasted just to maintain history, can be shrank into very small device.

This web application has the three tier architecture (client, server and database). It is platform independent, user-friendly, and very compatible with every device including smartphones, tablets and computers, as it runs on a browser. We have created it using **Java's JSF technology**. The main reason to choose this technology was its rich libraries that would provide you with all the options that are needed in a simple web application. We have used **PrimeFaces**, as it was giving us all the things we wanted, including the Text Editor, where user can type that document. And now here comes the best part, we have done all this keeping the user experience as it is. It means, that user does not have to type always. If he/she is comfortable with writing with pen like they used to in manual system, he/she can do that using our **Pen Input Feature**.

INTRODUCTION

Our project is called online paperless filing system. It enables us to manage, store and forward official documents. We have also incorporated pen input device for writing on our documents. Over time documents, including file folders get lost, and as a result they are a cause of concern in issues of security and wastage of space. So our solution is making this system completely virtual by making a web portal that is accessible 24/7 from all devices using a web browser.

We have all seen, huge cabinets overflowing with old files full of documents that are overlooked or eventually go missing. Keeping track of such documents is a hassle, not to mention the effort it takes to search and maintain these cabinets is overwhelmingly difficult and frustrating.

As a solution to this problem, we have come up with a system that the user can access from anywhere using a browser and an internet connection that helps to create, edit, forward and store official documents such as minute sheets, E-IONS etc., while keeping a track of our document.

Our system is cost effective, easily accessible without compromising on security and most of all it is environment friendly, as we are shifting from physical paper to an online virtual document format, hence reducing cost of paper production.

We have made two separate panels, one for Admin, and another for user. If the user wants to use the system, he must send a physical request form to the admin, requesting to register him as a user, by providing him a valid email address. The admin then activates the user and sends him his password. The user will then use his email id and password to login to the system.

Moreover we have implemented proper business rules, to grant access to the users according to their security clearance. We have made alerts that remind users to forward documents within a given timeframe. Another important feature of our system is the use of pen input to sign off or write comments on our document.

Pen input is another state-of-the-art feature that we have added in our system. We looked at previous existing systems and scrutinized them for not having such an important feature. Almost 70 % of all the mobile phones used today have touch input so it was important that we add such feature in our system.

It enables us to write on our document just like on any normal physical paper i.e. using a pen that is capable of writing digitally.

REVIEW OF LITERATURE

As we were going to automate a manual system, we must had to learn about the currently deployed manual system. After studying about this system and getting all the help from Clerical Staff, we found that:

- There are four basic types of official documents
    - Letter
    - Application
    - Minute Sheet
    - Inter Office Notes (IONs)
- Any faculty member can initiate any of the above mentioned document.
- Every document is associated to a file.
- Every paper in a file is called Enclosure.
- There can be 100 enclosures in one file. After that, a new volume of that file is created, and previous volume is safely kept in cabinet.

- While learning about this manual filing system, we find out that there is a web-based and intranet software already deployed in SEECS that is used to keep history in databases. But that was its only use. One cannot initiate a document using that software.
- From IGIS, we found out that there is another software, also web-based and on intranet, that is used for IONs. One could initiate, forward, reply and save as draft using that software. It had no Pen-Input Feature.
- We brought a file for learning more about it. This file was titled **Misc.** and was of 2009. In that file, there were not only Minute Sheets, IONs, Letters and Applications, but also date sheets, time tables, faculty and student allocation during exams.
- After learning about this system we tried to integrate our knowledge of the JSP backing beans architecture to try and improve the system.
- In our system we allowed users to create their own documents, forward them and receive them according to proper business rules and system hierarchy that was followed in SEECS.
- The user got notification when he received the document.
- A document details view table allowed for details of the sent and received documents as it showed the user the time at which they were sent or received and whether the document had been viewed by the receiver or not.
- Pen input was an added feature that allowed for writing on our paper virtually just like on any physical paper. It was another feature that was missing from the previously deployed system, but was an integral part of our system.

METHODOLOGY

This is our core part of the report where we will be discussing in details the technical aspects of our application. We will share with you the techniques and methodologies we came up with in making our application plausible and technically feasible.

We have added code snippets to give an idea to the reader how we programmed our system, we have added use case diagram to show the functionality our system is capable of performing.

We have added ERD (Entity Relationship Diagram) that shows the basis on which our database operates. Since database is an essential part of our application and relies heavily on it for info storage, we have explained in detail our database through tables and how they operate.

We have also added sequence diagrams that enable the reader to understand better the flow of our program. We have added some of the GUI (Graphical User Interface) images, mock ups, of the application so that the reader knows how our system looks like in practice.

Our very abstract architecture is very simple.

- **Client**
- **Server**
- **Database**

Client

On the client side, the user logs in to the system. Hence sends a request to the Login Bean java class which is a backing bean class on the server side, in the form of HTML form data.

Server

After getting the request, the server processes this request. Backing bean converts the HTML data that the user sent into a Java object. After the data has been converted into an object, the server can use it to process different functions and manipulate it to call different processes on it. After the data has been received on the server side, a new entry is made of it in the database for future usage.

Database

There is a function in the backing bean that obtains a Database connection object, which is also a Java class and is a Managed bean. Its function is to make a connection with the database and performs all the operations and queries on the database. After the connection has been made with the database, the database can be queried. After the data has been processed by the server, an entry of it is made in the database so that it can be used for future accessibility and it can be queried.

ERD (Entity Relationship Diagram)

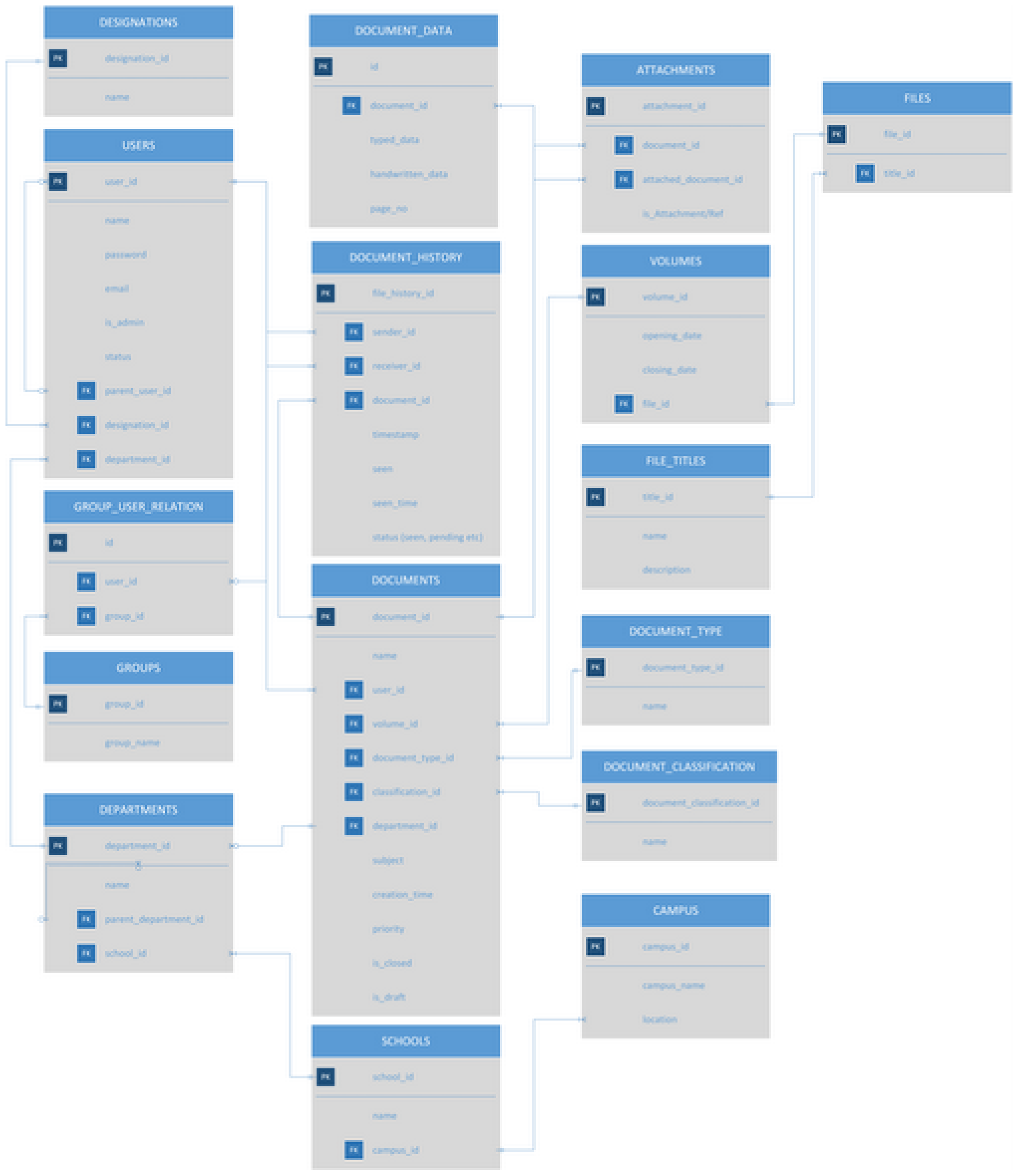


Diagram 1

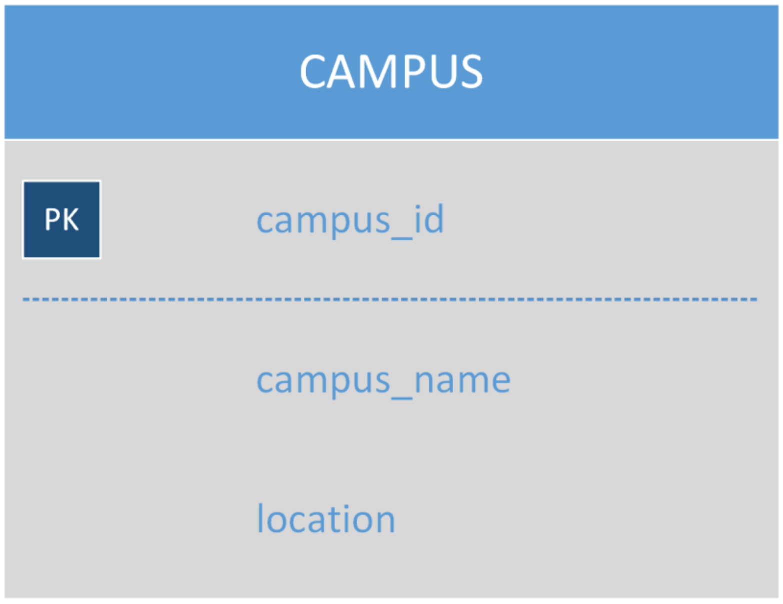


Diagram 2

This table enables us to incorporate our system to other institutions by extension as well. For now we are currently deploying it in NUST H-12 only. Hence we have made only one entry in this table i.e. NUST H-12.

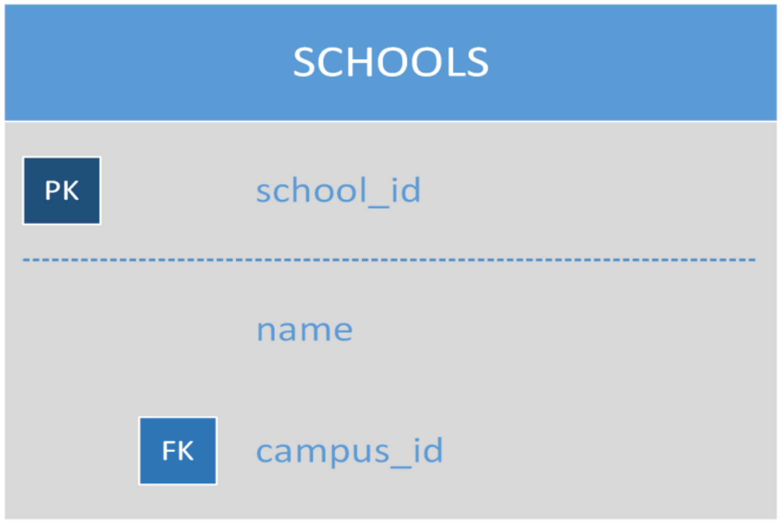


Diagram 3

This table enables us to distinguish one school's document from other school's documents. Hence it has a campus id as a foreign key.

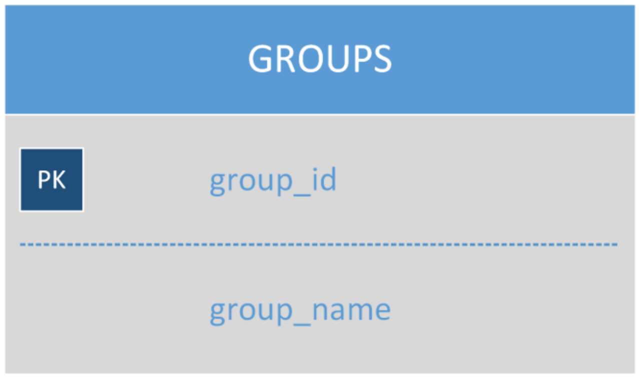


Diagram 4

Above table enables us to create and manage groups in our infrastructure. Admin user has the ability to add a new group type.

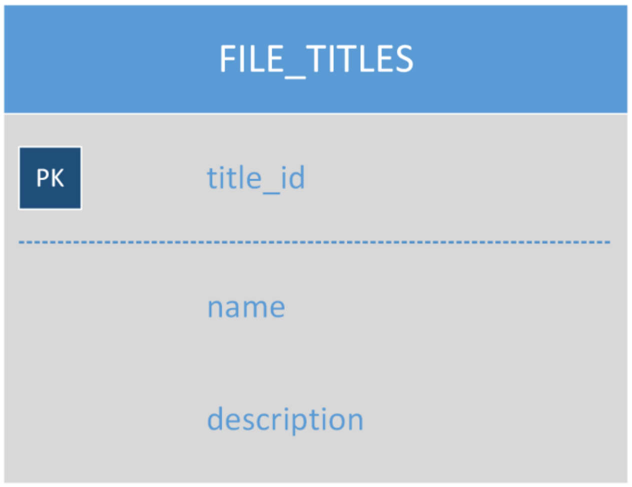


Diagram 5

Above table allows us to manage all the different kinds of files by adding a name and a brief description. Our system describes files as the physical folder in which the official documents are attached for storage. Admin user has the permission to add a new file title type. E.g. misc., transport etc.

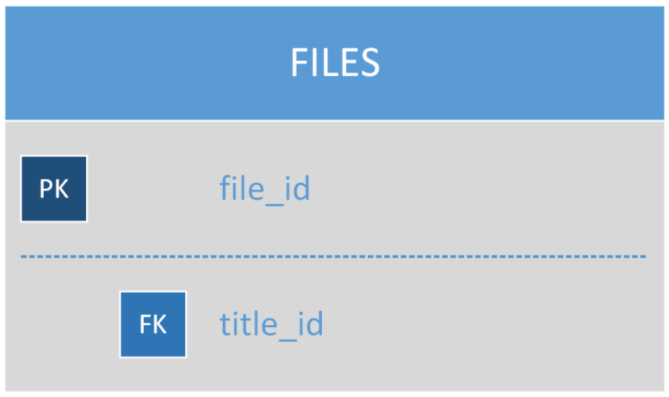


Diagram 6

This table creates a new file e.g. transport, misc. file coming from file_type table as foreign key.

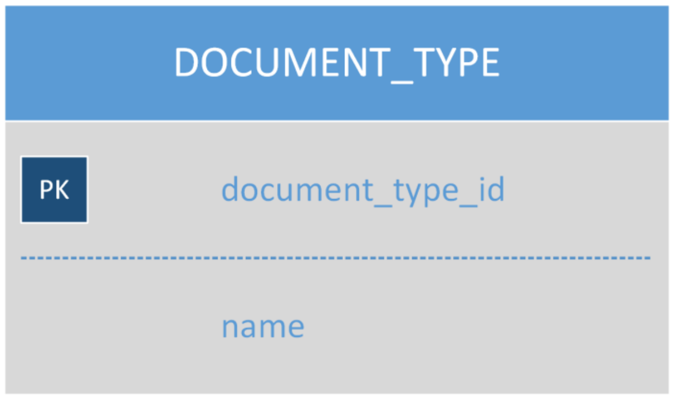


Diagram 7

This table permits us to store the different types of documents, officers will be using in their daily official business. For e.g. letters, applications, minute sheets etc. Admin user has the authorization to add a new type.

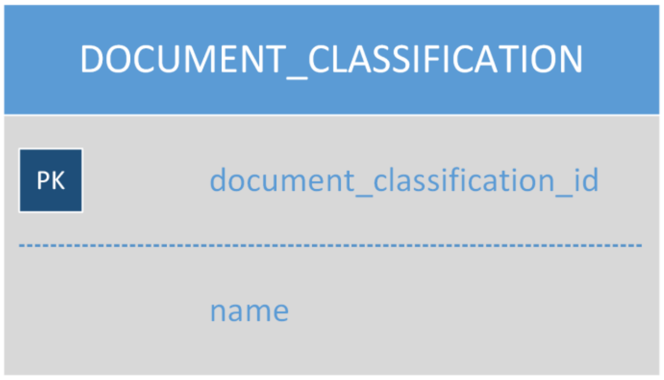


Diagram 8

This table tells about a single document's privacy. Admin is not allowed to add another document classification. This table is not currently being used but is for added feature.

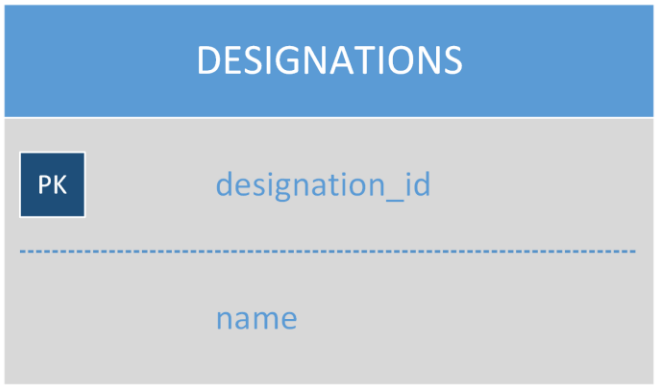


Diagram 9

This table is for adding a new type of designation for the user. For e.g. professor, associate professor, lecturer Etc.

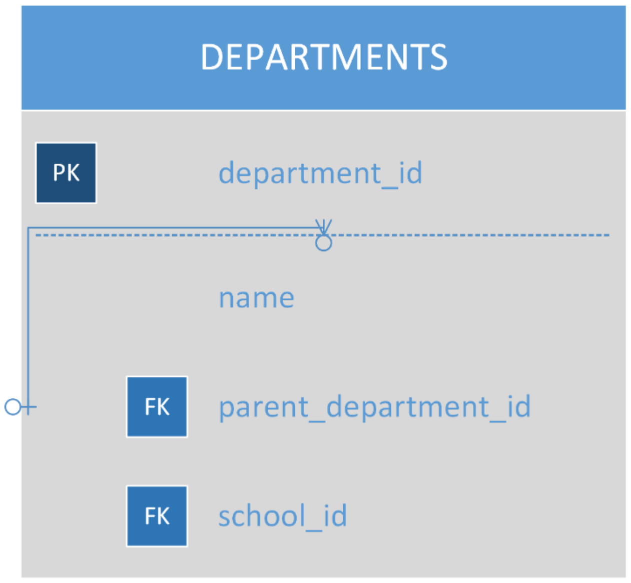


Diagram 10

This table allows for adding new type of departments that exist in a school. It has a relation to itself because there can exist sub departments within a department as well.

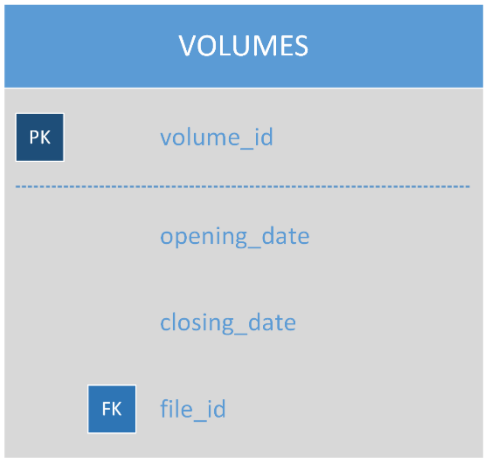


Diagram 11

Volumes table enables the documents to group in volumes. It has an opening_date and closing_date fields that tells the user when the volume was created and when it will be final. For each file there can be many volumes. In the currently deployed physical filing system, each volume can have a limited number of documents in it. But in our case, currently there is no limit but it can be implemented.

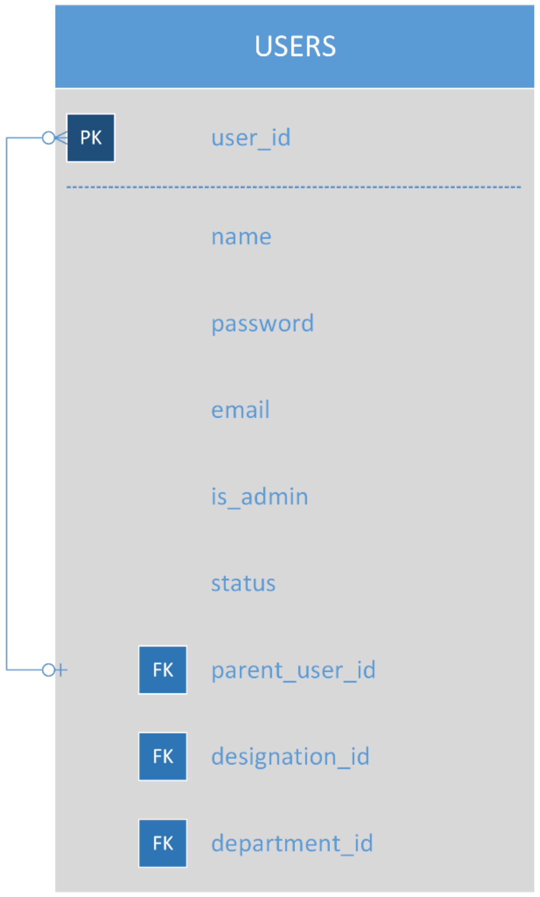


Diagram 12

This table manages the users of our system. There are two types of users, admin and normal user, is_admin property is used for this purpose. When new user is created he is assigned a password by the admin, to his email. Status tells if the user is active or not. This table has a relation to itself because there can be sub users as well. This table has foreign keys coming from the specific tables. The designation_id describes the designation of the user. Department_id tells us which department the user is from.

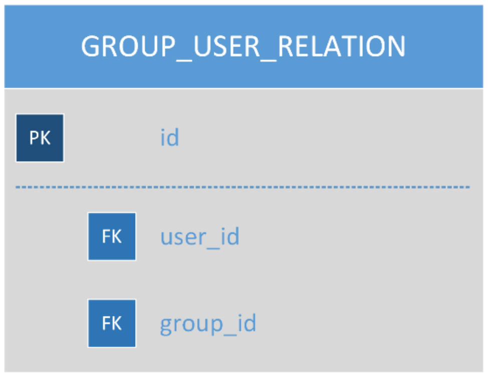


Diagram 13

Above table allows the many-to-many relationship between group and user tables. User_id identifies the unique user coming from the user table. And to which group this user belongs, the group_id tells us that.

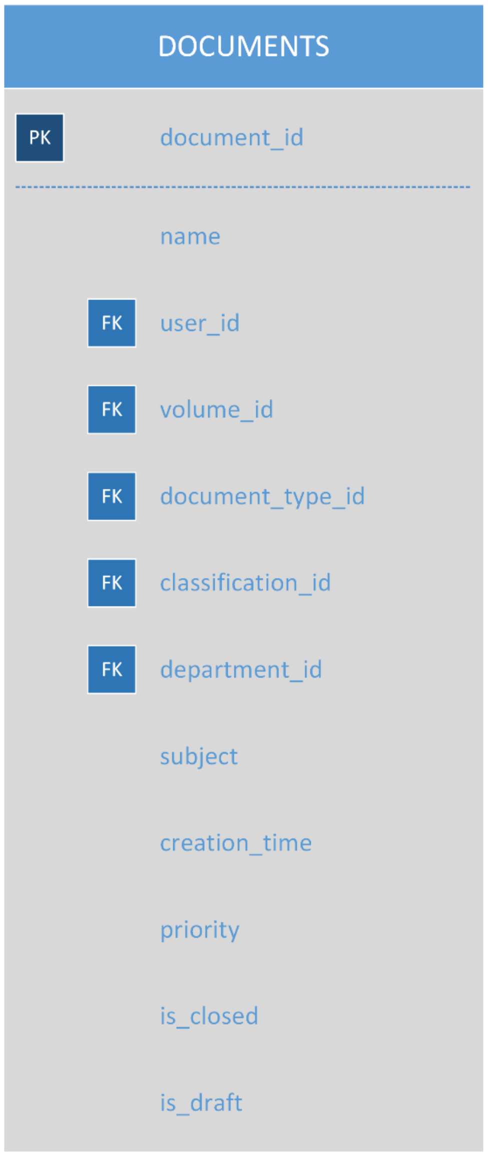


Diagram 14

Documents table is related to managing the documents within a file. It has a subject field which tells user the short description of the document. Creation_time, when it was created and has priority high, medium, low. Is_draft tells if document is rough draft or not. Is_closed means that the document is finalized and no further forwarding or editing will be done with it. After a document is closed, it goes into its file and stays there forever.

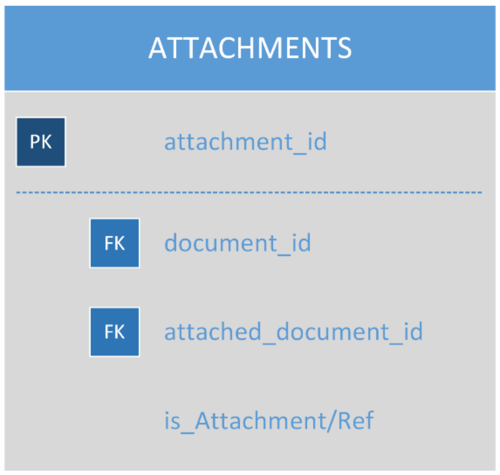


Diagram 15

This table is used to manage the attachment to each document that the user creates. Document_id tells us the document we want to attach the previous documents to and attached_document_id tells us the document we want to attach. Is_attachment attribute tells if the attachment is an attachment or reference. This column is not being used right now. For reference, this table is used as it is in case of attachment.

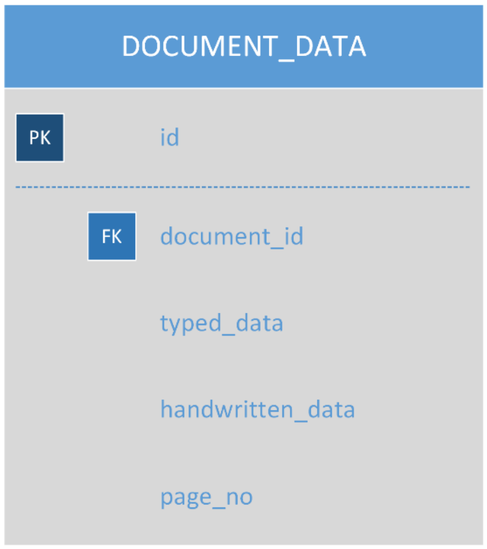


Diagram 16

Typed_data saves the text that was typed for this specific document and this specific page. Same goes for handwritten_data, except that it saves handwritten data as an image in bytes. Page_no field tells the page number of document. For every page of every document, there is a row in document_data.

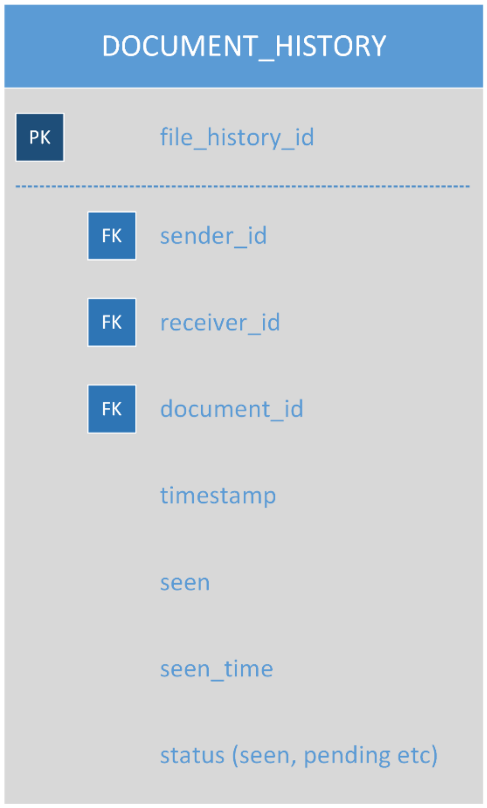


Diagram 17

This table is used for tracking the document. Timestamp attribute is the time when the document was sent. Seen, tells sender if the document was viewed by the receiver. Seen_time, at what time the document was opened by the receiver. Status means if the document has been viewed by the receiver or whether it's still pending. For every sent, there is a row here.

Use Case Diagram

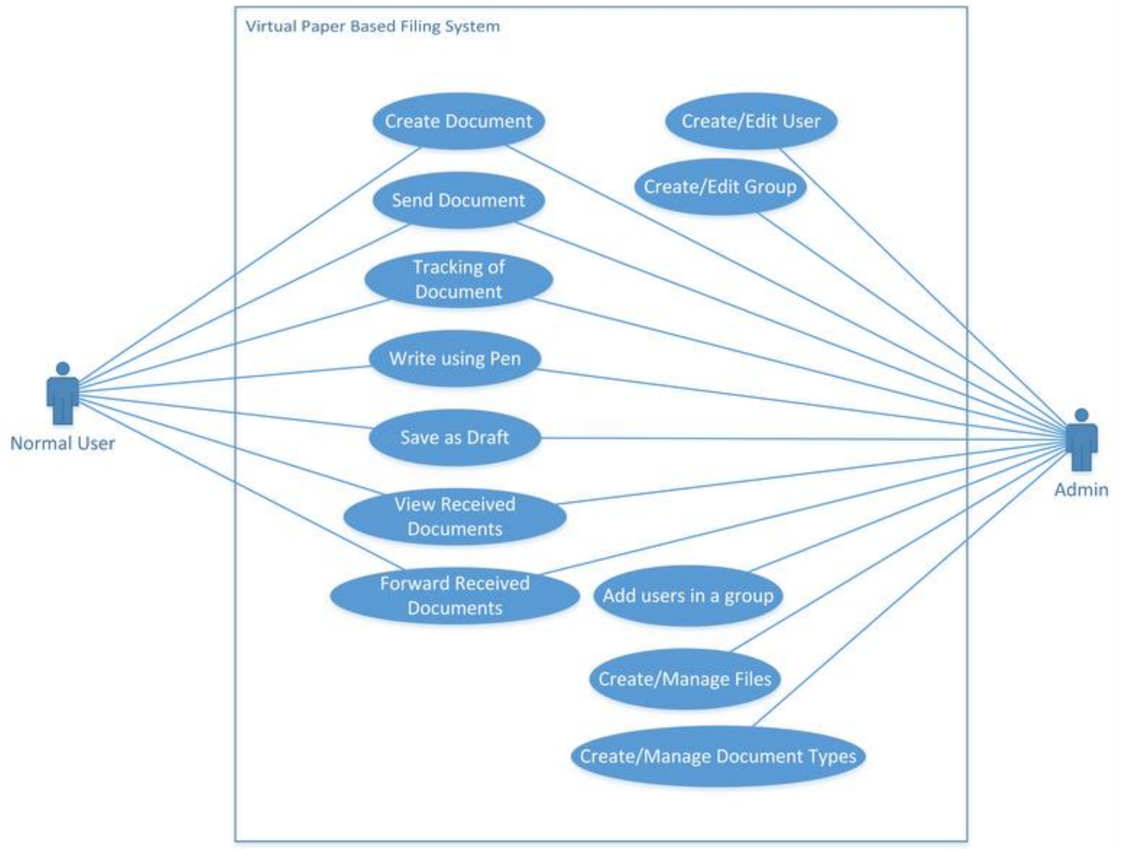


Diagram 18

Use Case Diagram explains the functionality of the software and the users (actors) associated with each function. It is used to familiarize the system with its users. Admin user has the administrative responsibility of adding, creating, deleting and managing users, and group and document types. The admin has all the functions a normal user has plus extra functionality as shown in the diagram above.

Data Flow Diagram

Login

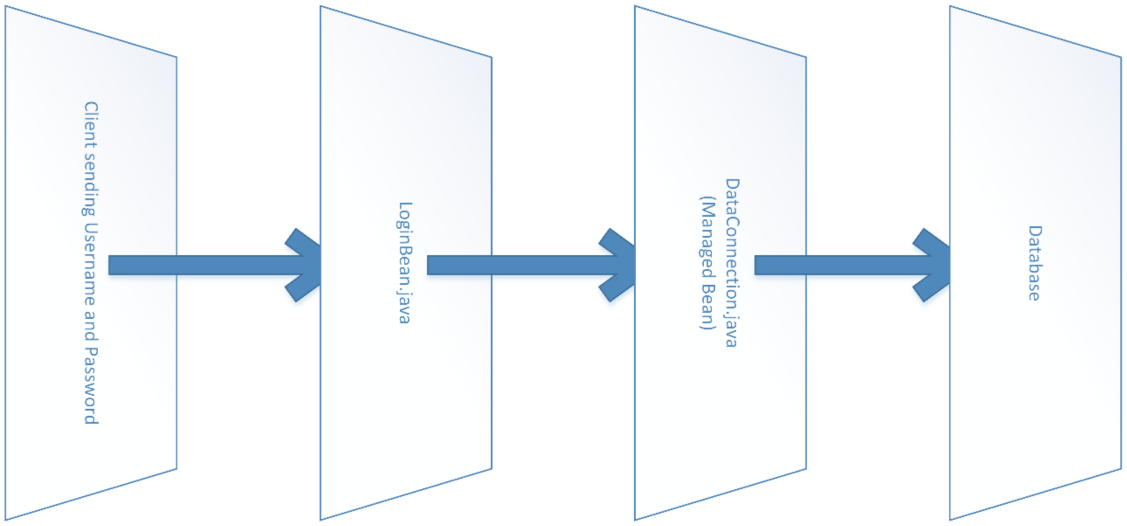

Client sending Username and Password
LoginBean.java
DataConnection.java (Managed Bean)
Database

Diagram 20

Login is the first screen the user is directed to. Here, the user enters his email and password that the admin has assigned him.

The flow of the login use case is in such a way that the user sends a request containing his username and password to the Login Bean Java class, which is a backing bean on the server side.

Backing Bean is a Java class that converts the data from HTML form into a Java object.

There is a function in this backing bean that obtains the object of DataConnection, which is also a Java class and is a Managed Bean.

A Managed Bean is a Java class that performs all the operations including connecting with Database.

An entry is created against the sent data in the database.

CreateDocument

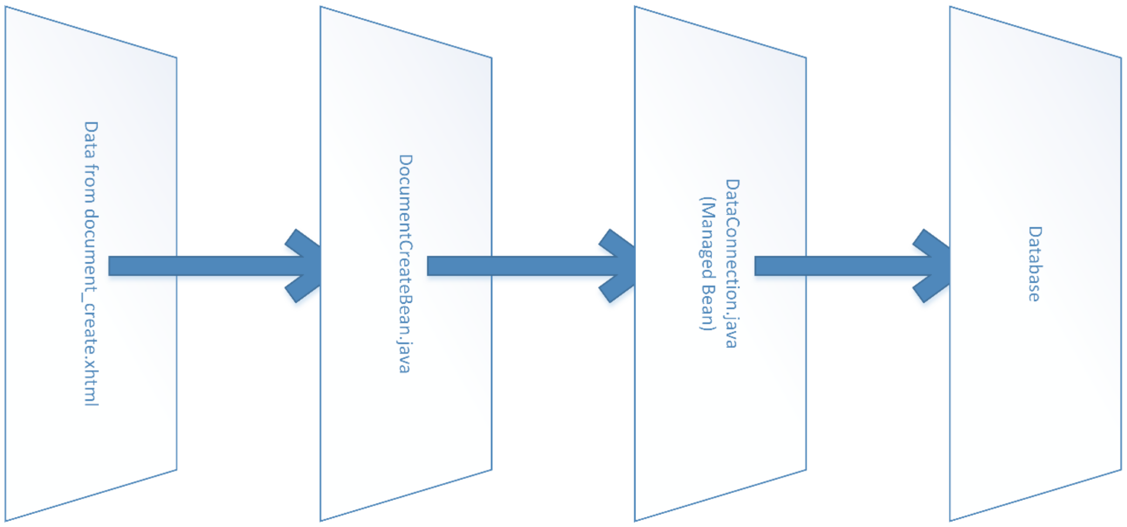

Data from document_create.xhtml
DocumentCreateBean.java
DataConnection.java (Managed Bean)
Database

Diagram 21

This diagram shows the flow for create document page. The page has a text editor and an HTML5 canvas.

The flow of the createdocument use case is in such a way that when the user forwards the document the saved text data is sent to the DocumentCreateBean Java class.

Backing Bean is a Java class that converts the data from HTML form into a Java object.

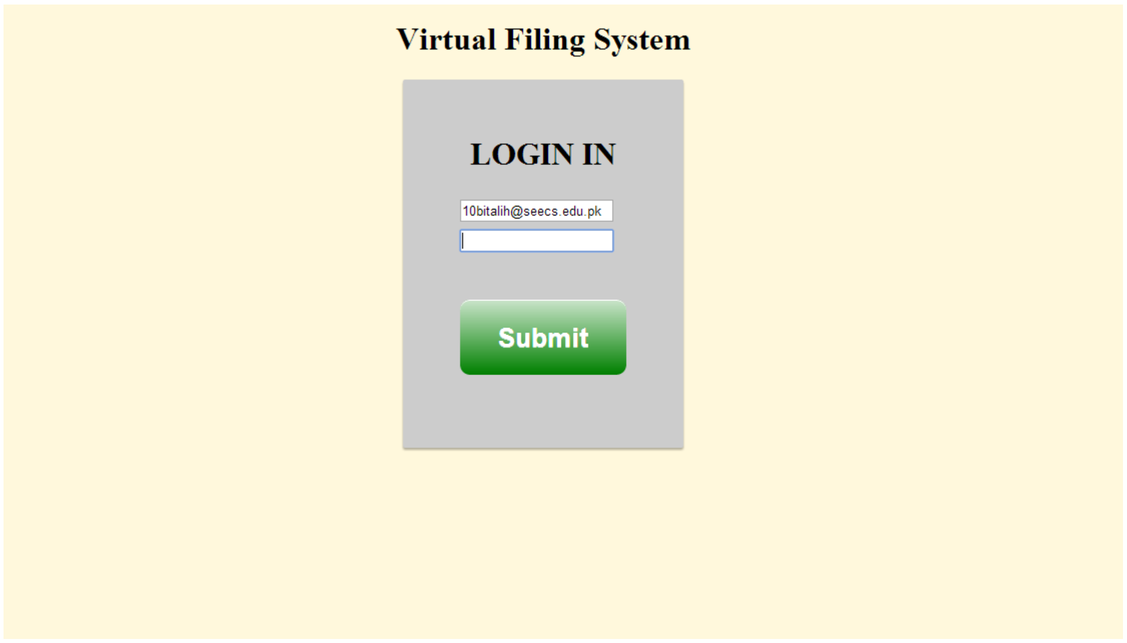
There is a function in this backing bean that obtains the object of DataConnection, which is also a Java class and is a Managed Bean.

A Managed Bean is a Java class that performs all the operations including connecting with Database.

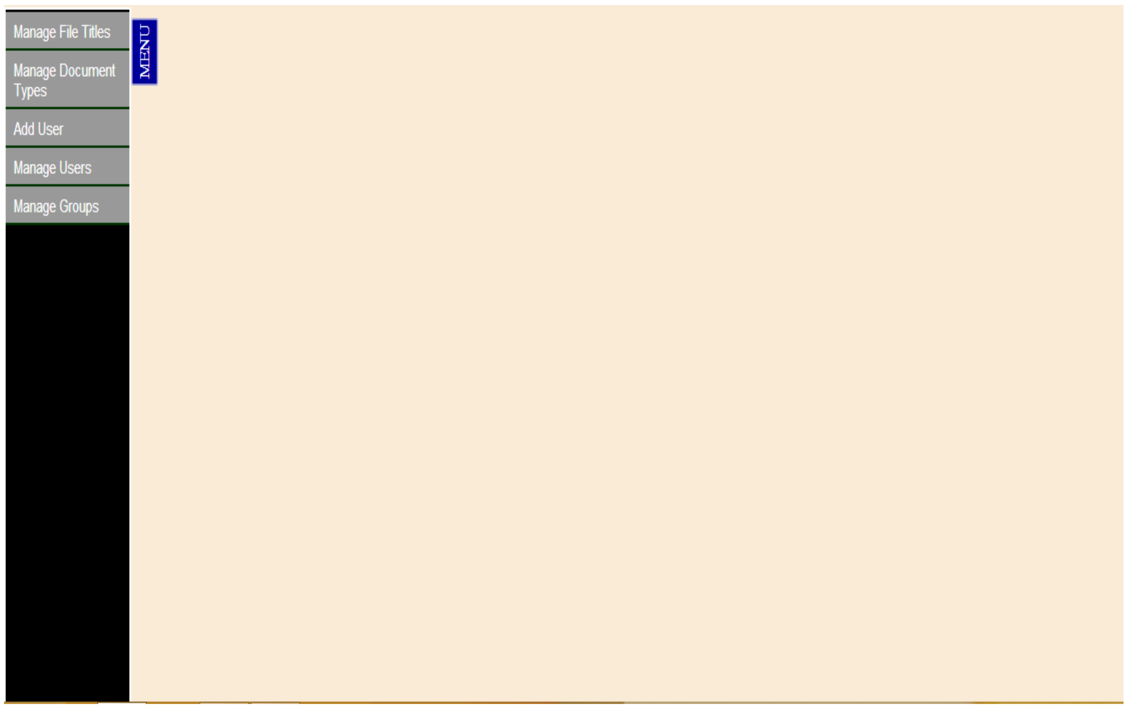
An entry is created against the sent data in the database.

Graphical User Interface (GUI)

Admin Panel



After logging into the system, the admin is directed to the admin panel.



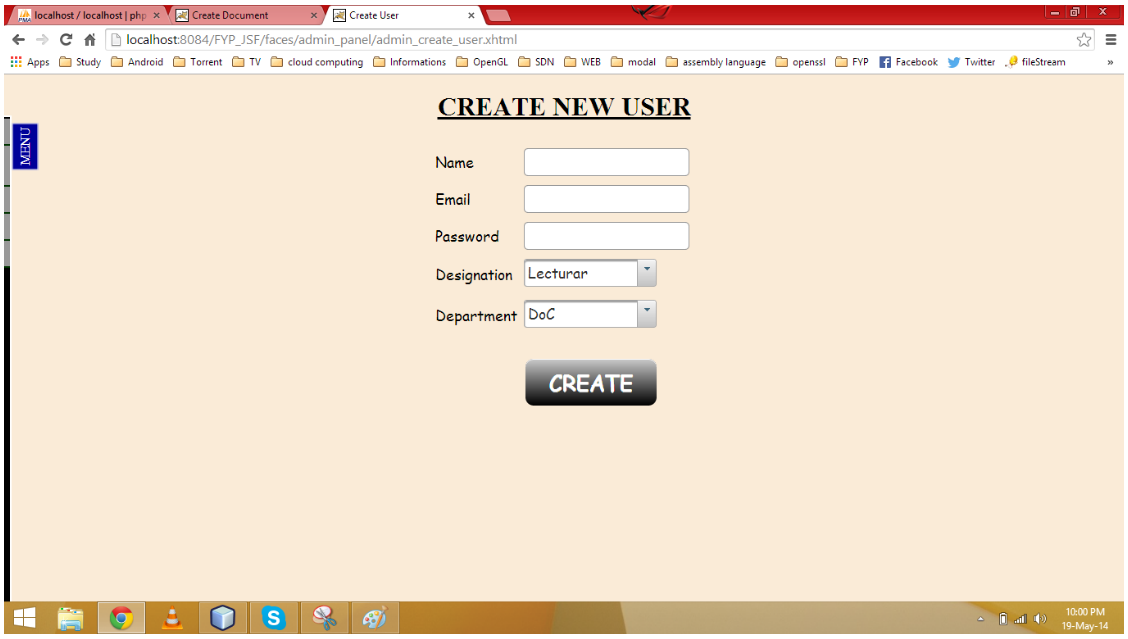
The admin panel provides functionality to the admin of the system and is available to the admin only, and not the normal user. We have made the design as simple as possible. Initially, the user is shown a simple screen with no extra menus or divisions, so the excess information does not confuse him.

There is one menu, but it is out of screen. Although a small part of it is visible. After the admin drags his mouse pointer over the menu, the menu translates into view, showing all the administrative tasks that the user can perform.

This menu has the following options:

- Manage File Titles
- Manage Document Types
- Add User
- Manage Users
- Manage Groups

Add User



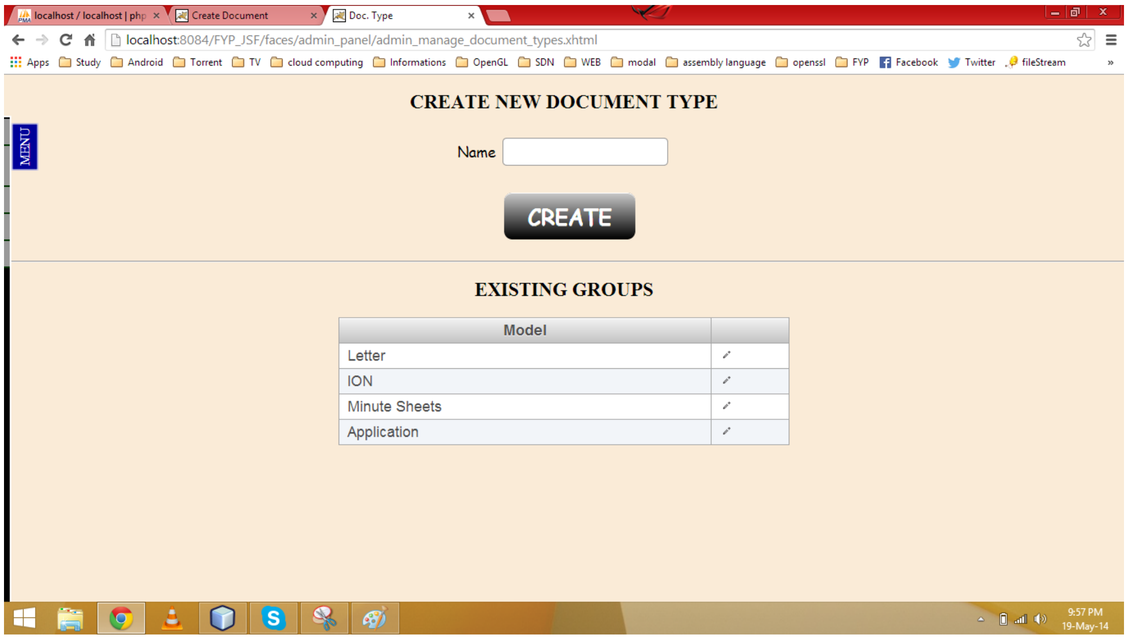
CREATE NEW USER

Name	
Email	
Password	
Designation	Lecturar
Department	DoC

If the user wants to use the system, he must send a physical request form to the admin, requesting to register him as a user, by providing him a valid email address. The admin then activates the user and sends him his password. The user will then use his email id and password to login to the system. Admin will add him according to his designation and department.

When the user fills his form and clicks submit button, it is sent to the server. There is a backing bean named LoginBean.java, lying against this form which will get the data and make loginBean object. The managed bean will make an entry of the new user in the database.

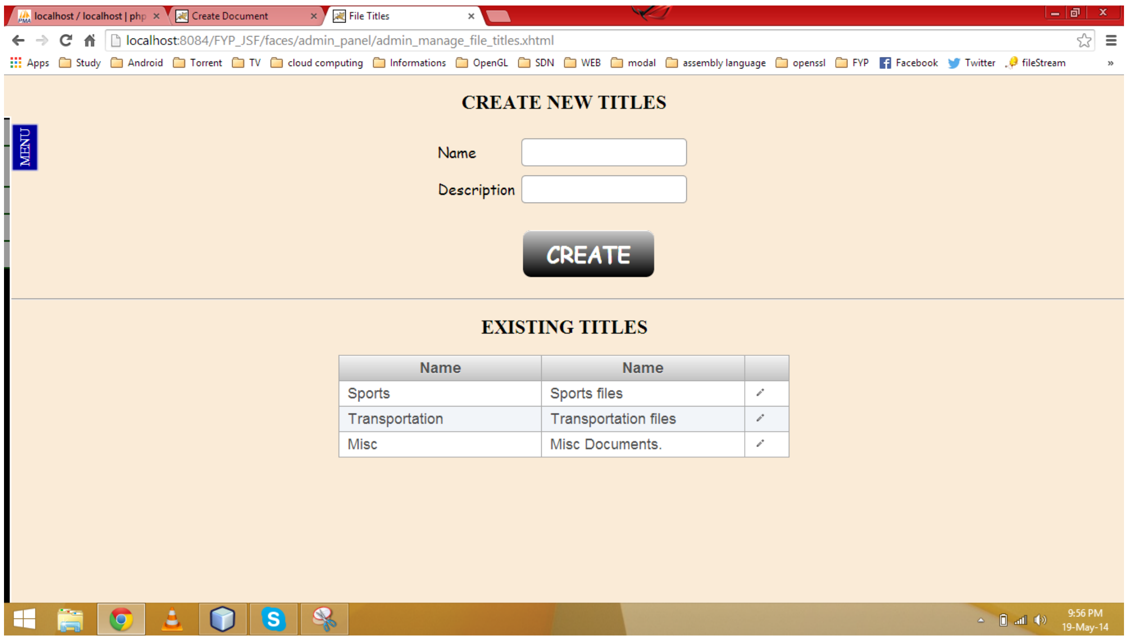
Manage Document Types


localhost / localhost | php
Create Document
Doc. Type
localhost:8084/FYP_JSF/faces/admin_panel/admin_manage_document_types.xhtml
MENU
CREATE NEW DOCUMENT TYPE
Name
CREATE
EXISTING GROUPS

Model	
Letter	
ION	
Minute Sheets	
Application	

The admin can also add a new document type for official use. The existing document types are letter, ION, Minute Sheets, Application etc. But once it is created and sent it can’t be deleted

Manage File Title



CREATE NEW TITLES

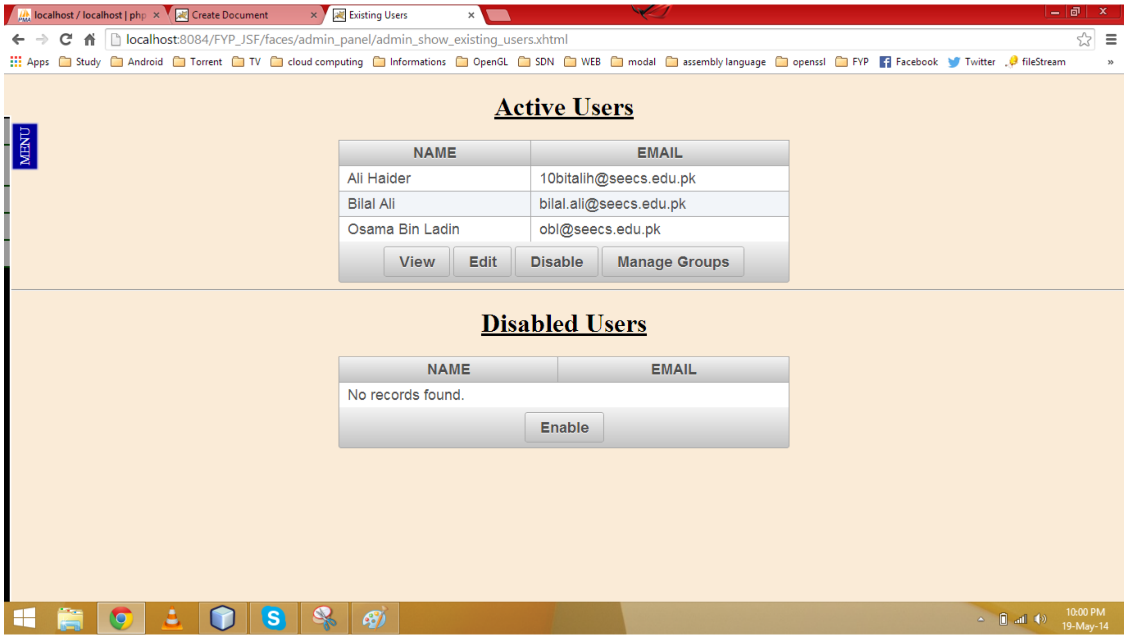
Name	
Description	

EXISTING TITLES

Name	Name	
Sports	Sports files	
Transportation	Transportation files	
Misc	Misc Documents.	

The admin can also add a new file type with a unique title. The file types can be transportation, fee, application etc.

Manage Users



Active Users

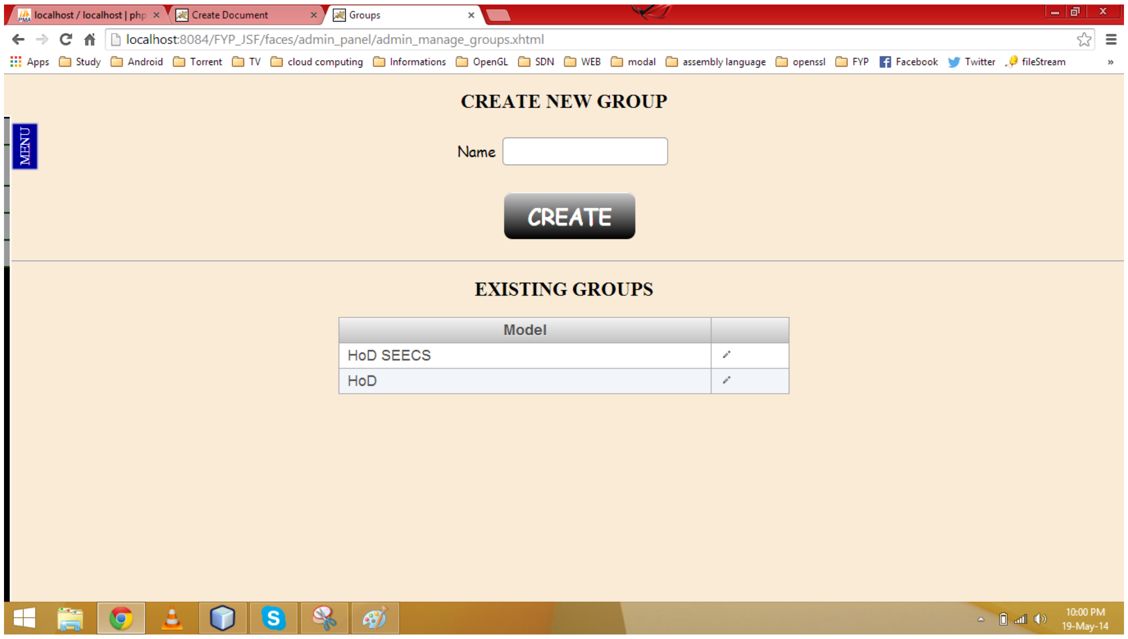
NAME	EMAIL
Ali Haider	10bitalih@seecs.edu.pk
Bilal Ali	bilal.ali@seecs.edu.pk
Osama Bin Ladin	obl@seecs.edu.pk

Disabled Users

NAME	EMAIL
No records found.	

It is an admin function that is used to manage users. In this panel, the admin can activate and disable users and view the active and disabled users. From this panel, the admin can edit users, and manage groups as well.

Manage Groups


localhost / localhost | php
Create Document
Groups
localhost:8084/FYP_JSF/faces/admin_panel/admin_manage_groups.xhtml
MENU
CREATE NEW GROUP
Name
CREATE
EXISTING GROUPS
10:00 PM
19-May-14

Model	
HoD SEECS	
HoD	

It is an admin function that is used to manage and create new groups. In this panel, the admin can view existing groups as well.

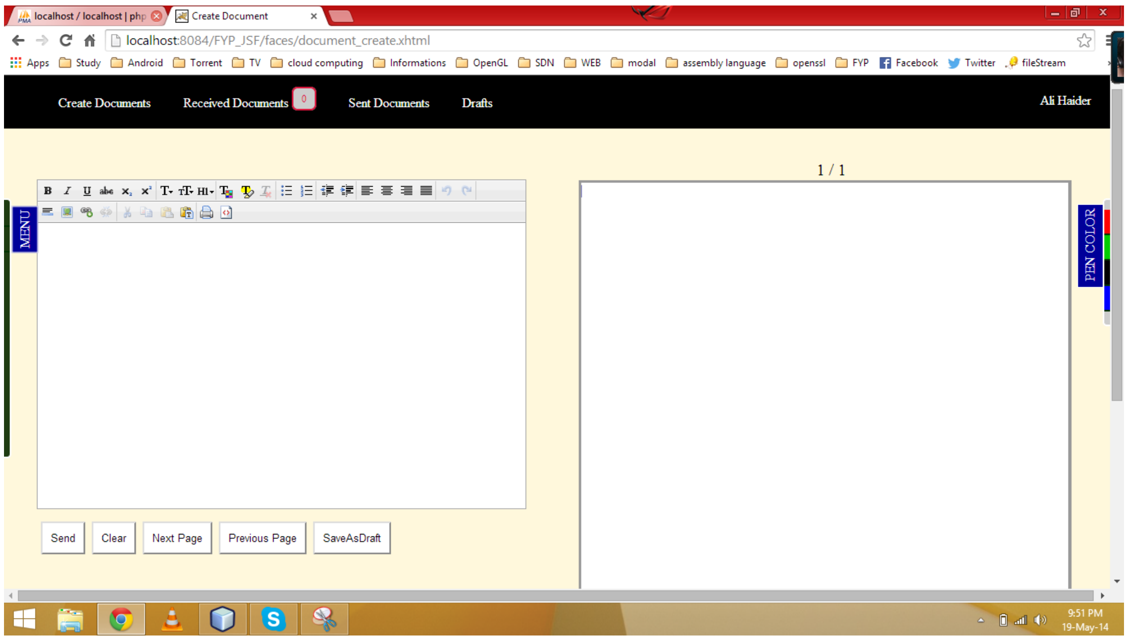
User Panel

This is the panel that the normal user can access. It has the following features:

- Create Document
- View Received Document
- Tracking of Document
- Forward Received Document

- Save as Draft
- Write using Pen
- Send Document

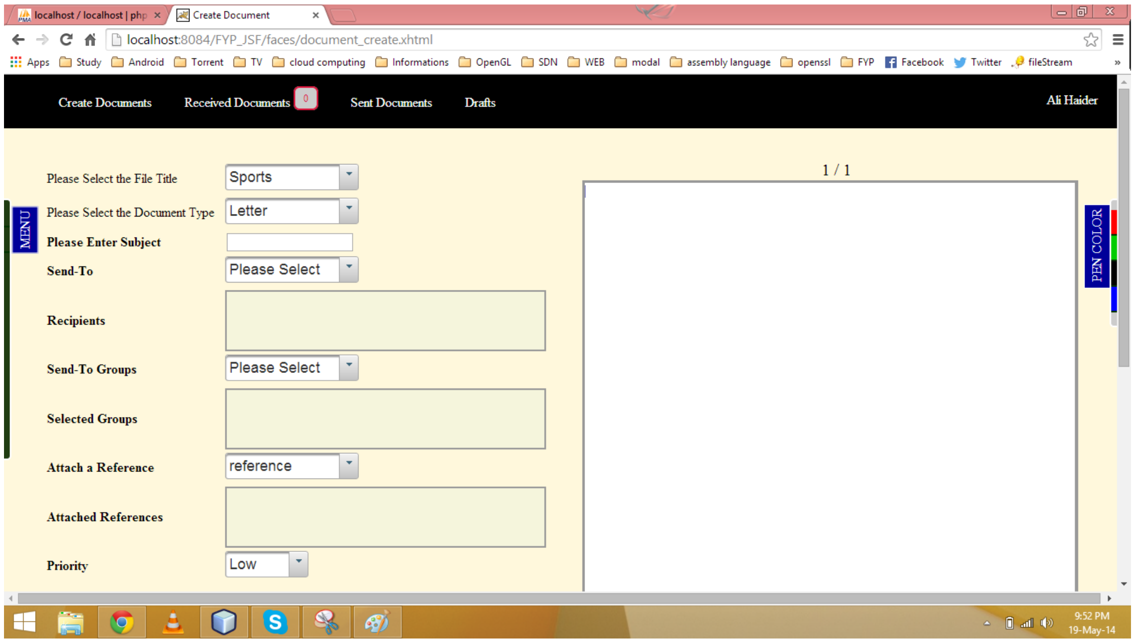
Create Document



This is the page that allows user to create document to be sent. We have added a JSF PrimeFaces text editor that has all features found in a standard text editor such as Bold, Italics text, highlighter, alignment etc.

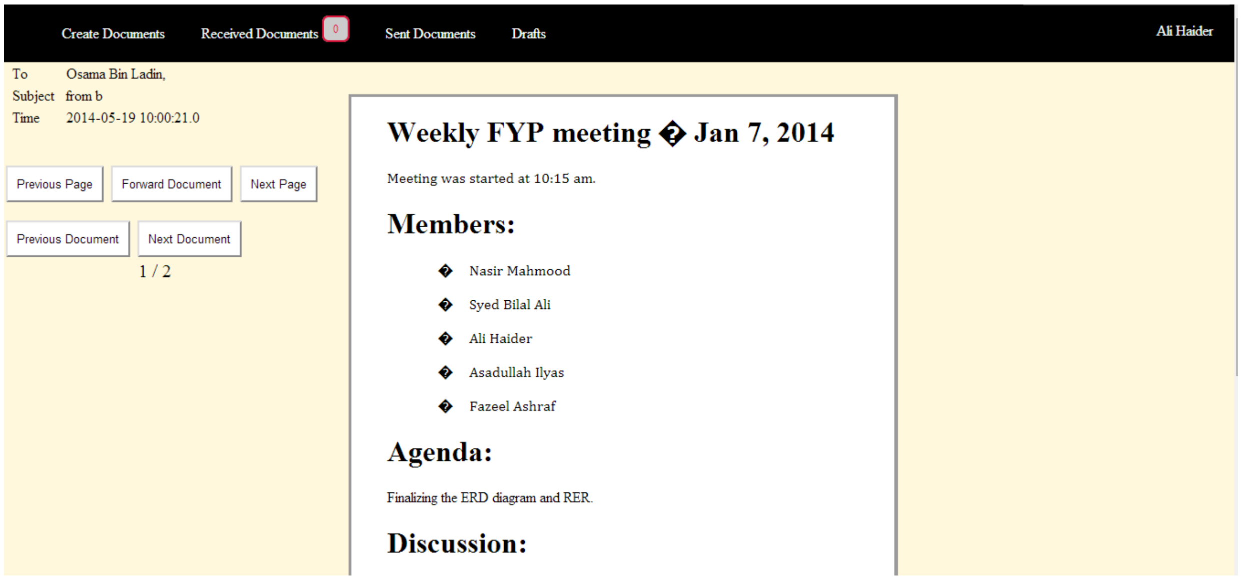
We type in the text editor and the text is copied into the page using a function. Our document is a canvas so we can write using pen input. Pen input is implemented using JavaScript function and the handwritten part is saved as image on the canvas. Clear button clears the page.

We can navigate the document by using the next and previous page buttons. We can save the document as draft as well to send later on. The menu button can be accessed when the user hovers his mouse over it, to show detail view of the document page. Here he can give reference of sender, attach attachments, set priority of document, enter subject and can even send in groups, as shown in the diagram below.


localhost / localhost | php
Create Document
localhost:8084/FYP_JSF/faces/document_create.xhtml
Apps Study Android Torrent TV cloud computing Informations OpenGL SDN WEB modal assembly language openssl FYP Facebook Twitter fileStream
Create Documents
Received Documents 0
Sent Documents
Drafts
Ali Haider
MENU
PEN COLOR
1 / 1
9:52 PM
19-May-14

Please Select the File Title	Sports
Please Select the Document Type	Letter
Please Enter Subject	
Send-To	Please Select
Recipients	
Send-To Groups	Please Select
Selected Groups	
Attach a Reference	reference
Attached References	
Priority	Low

View Received Document



To	Osama Bin Ladin,
Subject	from b
Time	2014-05-19 10:00:21.0

Weekly FYP meeting � Jan 7, 2014

Meeting was started at 10:15 am.

Members:

- Nasir Mahmood
- Syed Bilal Ali
- Ali Haider
- Asadullah Ilyas
- Fazeel Ashraf

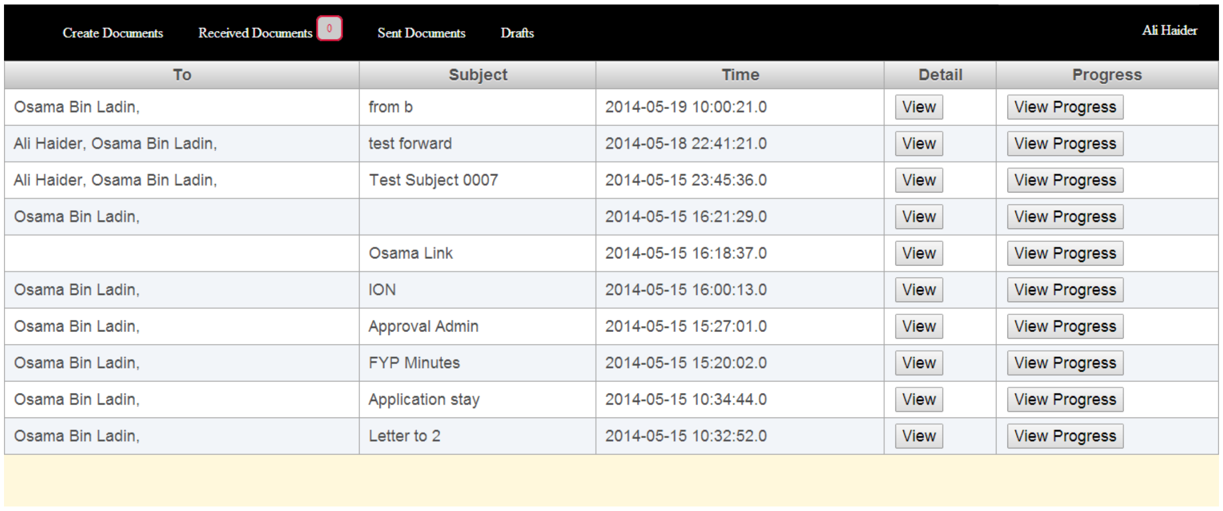
Agenda:

Finalizing the ERD diagram and RER.

Discussion:

The user can view received documents. He can see at what time the document was sent to him, using the timestamp field. He can also access previous documents in the file and also navigate to the next or previous pages within a document. He can view the subject title as well. The number of pages contained in the received document are shown as well.

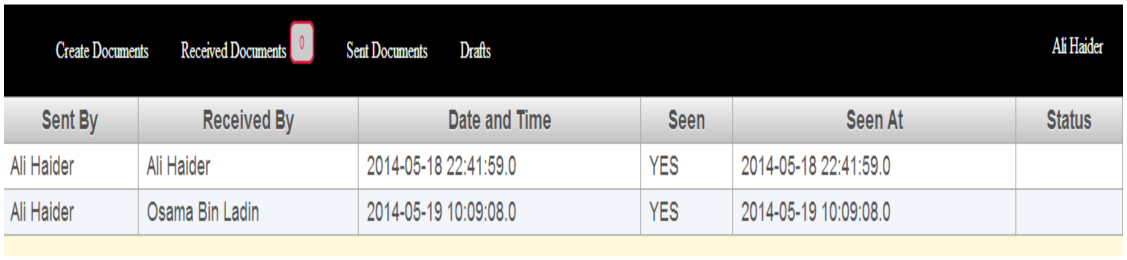
View Sent Documents



To	Subject	Time	Detail	Progress
Osama Bin Ladin,	from b	2014-05-19 10:00:21.0	View	View Progress
Ali Haider, Osama Bin Ladin,	test forward	2014-05-18 22:41:21.0	View	View Progress
Ali Haider, Osama Bin Ladin,	Test Subject 0007	2014-05-15 23:45:36.0	View	View Progress
Osama Bin Ladin,		2014-05-15 16:21:29.0	View	View Progress
	Osama Link	2014-05-15 16:18:37.0	View	View Progress
Osama Bin Ladin,	ION	2014-05-15 16:00:13.0	View	View Progress
Osama Bin Ladin,	Approval Admin	2014-05-15 15:27:01.0	View	View Progress
Osama Bin Ladin,	FYP Minutes	2014-05-15 15:20:02.0	View	View Progress
Osama Bin Ladin,	Application stay	2014-05-15 10:34:44.0	View	View Progress
Osama Bin Ladin,	Letter to 2	2014-05-15 10:32:52.0	View	View Progress

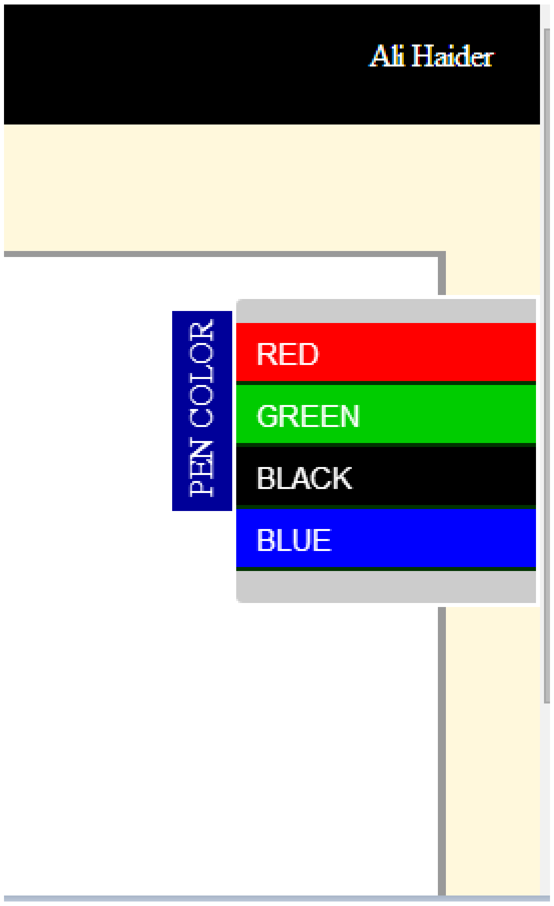
The user can see all the documents he has forwarded, in a table with details such as the subject line, the recipient, the time at which he sent the document, view button shows the detailed version of the sent document, and the view progress button helps to track the document, this page tells us that where our document has currently reached.

Document History View

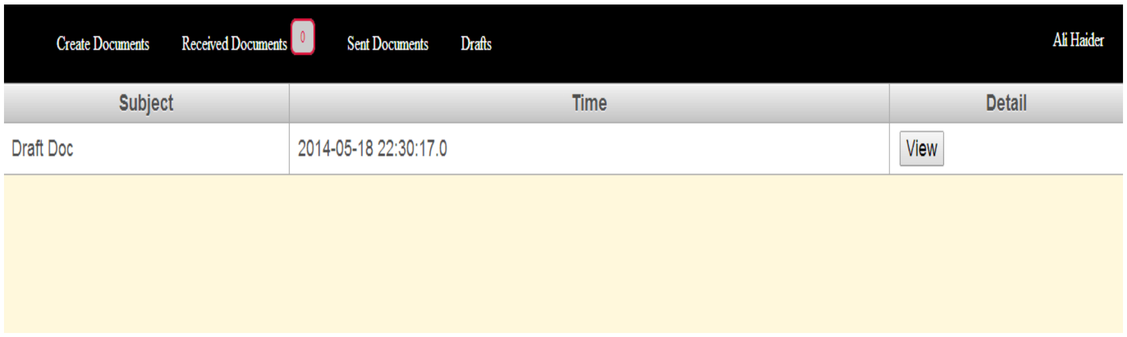


Sent By	Received By	Date and Time	Seen	Seen At	Status
Ali Haider	Ali Haider	2014-05-18 22:41:59.0	YES	2014-05-18 22:41:59.0	
Ali Haider	Osama Bin Ladin	2014-05-19 10:09:08.0	YES	2014-05-19 10:09:08.0	

The user can view the document details in the document history table. He can see the sent and received documents and the date and time associated time with each document. There is a seen field that tells sender if the document was opened by the receiver. Seen at field tells the sender the time at which the document was opened. Status field is not being used.



Red, Green, Black and Blue are the colors in which the user can write on the page using pen input.



Subject	Time	Detail
Draft Doc	2014-05-18 22:30:17.0	View

This is the draft document. The user write a document but does not forward it so it becomes saved as a draft.

REFERENCES

W3Schools Online Web Tutorials

http://www.w3schools.com

JSF Tutorial - Tutorials Point

http://www.tutorialspoint.com/jsf/

JavaServer Faces Technology – Oracle

www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html

PrimeFaces

primefaces.org/

ACKNOWLEDGMENTS

This project would not have been possible without the constant help and guidance of Sir Nasir Mahmood and Sir Bilal Ali, for giving us this amazing opportunity and guiding us along the way and always seeing the best in us. Their superior knowledge and extensive guidance helped us achieve our true potential and was the key factor in completing the project on time.

Next I would like to thank our parents who were able to give us the facilities that so few have, and always thinking positive about us and praying for our success.

Lastly and most importantly I would like to thank Allah who has blessed us with health and respect and the ability to pave our own destinies while always guiding us in difficult situations. He always gives us hope in dark times.

CONCLUSION

- We provided a cheap solution to an existing problem in an efficient way and in a timely manner.
- We learnt how to manage our time and how to work in a team
- We had a great learning experience
- We learnt many new things that we will use in our professional careers later on.