DOCUMENT MANAGEMENT SYSTEM

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Aman Sharma

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Program:
Software Engineering

December 2012

Fargo, North Dakota

North Dakota State University

Graduate School

Title

Document Management System

By
Aman Sharma

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Kendall E. Nygard
Chair

Simone Ludwig

Kenneth Magel

Limin Zhang

Approved by Department Chair:

12/21/2012
Date

Dr. Kendall Nygard
Signature

ABSTRACT

The debate of using paper documents over computer documents has been a long topic of interest, there has been a lot of research done on the topic. Also, experiments have been done to evaluate the performance of users working with computers v/s working manually with no definitive conclusions (Chris Anderson, 2010) (Askwal, 1985) (Noyes and Graland, 2008). However, one cannot deny the time it takes to store and go through papers manually.

Every year, new policies help the user understand the guidelines and operational procedures that they have to follow while being employed by a company. Keeping track of such documents as well as assigning tasks to certain users for revising such are tasks that are lengthy and cumbersome, and are not automated. The software proposed in this paper would help an organization maintain documents, reducing unnecessary tasks and the disk space on the machine.

ACKNOWLEDGMENTS

I would like to express my gratitude and to say thanks to everyone who helped me and supported me during the completion of this project and my research paper. First, I would like to express my thanks to Dr. Kendall Nygard for helping, guiding, and supporting me throughout my master’s degree and my research completion. I would also like to thank my committee members, Dr. Kenneth Magel, Dr. Simone Ludwig, and Dr. Limin Zhang, for being on my committee. I would like to thank my department for providing the courses that helped complete different chapters of this paper. I would also like to thank my supervisor, Amber Randall, for letting me do something that I liked and my company, Axio Research, for allowing me to present this software as part of my master’s research. Last but not least, I would like to thank my family members for their constant and unrelenting support towards my education and for their impartial love for me. I would also like to thank my friends and the most important person, without whom it would have been impossible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

QA	Quality Assurance
FDA	Food and Drug Authority
AUDITOR	Regulators from companies that review Process
CRO	Contract Research Organization
DMS	Document Management System

CHAPTER 1. INTRODUCTION

These days, a lot of company documents are stored on computers and servers which can be easily accessible to users at any place as long as they get permission to view the documents. Companies use these documents in a number of ways: to simply store them at one location, to train users, or to maintain training documents and logs.

At Axio Research, the same need has come along since the plans for expansion came into being. This needled to developing a Document Management System which can be used to store the company's policy documents for the standard operating procedure used to train employees and to reduce the paperwork that needs to be maintained.

1.1. Motivation

Axio Research is a CRO that works in the pharmaceutical Industry in Clinical Trials. The main work that is part of Axio Research is helping the Data Monitoring Committee's with providing statistics on the Clinical data and helping them in making key decisions regarding the future of a trial. Being in such a rapid and fast paced environment requires a Quality Assurance Specialist who takes care of the quality of the documents produced in the company.

It is the task of Quality Assurance specialist to maintain all the documents necessary for later audit trails by either the Company or the FDA. These documents can be Protocols, Statistical Analysis Plans (SAP's) or Standard Operating Procedures (SOP's). Maintaining such documents as manual paper work can result in reduce performance and even lead to errors. Hence a need for a system that can be used for storing such documents and can also be used for assigning tasks to users for making revisions to such documents arise.

According to studies done in the past (Noyes and Garland, 2008), software which would automate such processes and would help the QA for keeping track and for minimizing her workload have been proposed to the company.

1.2. Aim of Software

This software is developed in order to help the Quality Assurance Specialist at Axio Research to keep track of the document changes and to reduce the paperwork in the process. The Document Management System allows the users to view and save different versions of the same document, and allows the Admin to assign tasks for users to make changes to the current version. The system users are able to view any tasks assigned to them for document revision and to upload the changed documents. The Document Management System is being built for Axio Research and should have the scope of being extended for additional features in the future.

1.3. Paper Organization

The rest of the document is divided into three parts: OBJECTIVE, IMPLEMENTATION, and TESTING. The Objectives chapter lists the need for building the system. It provides use cases to help the business and technical users understand. It also gives a detailed explanation of each use case to help with design and implementation, and outlines the constraints regarding the software. The Implementation chapter contains the detailed design of the system, including the Class Diagram, Activity Diagram, and Component Diagram. The chapter also includes a detailed explanation for each component as well as the interaction of each class and its components with each other when carrying out certain tasks. This chapter also includes the software’s mock screen shots.

CHAPTER 2. OBJECTIVES

This chapter gives the overall description of the Document Management System. Then, we address the specific product requirements. The document has details about the Product Function, User Characteristics, Constraints, and Assumptions and Dependencies for the Document Management System.

2.1. Product Perspective

The Document Management System is a web-based system and can be accessed using: Internet Explorer8.0 and above, Mozilla Firefox 2.0, and Google Chrome.

2.1.1. User Interface

The two interface types in the Document Management System are as follows:

1. **User Interface**: Users shall be able to view the tasks assigned to them and to upload the new/revised documents to the system.
2. **Admin Interface**: The Admin shall be able to assign tasks to users for creating new documents/making changes to revised documents, approving/rejecting the new/changed documents, and viewing the detail summary for each document history.

Both interfaces will be described in detail in the External Interface Requirement section under Chapter 3.

2.1.2. Hardware Interface

The Document Management System shall provide minimum hardware requirements. The following hardware configurations are required for a PC using the Document Management System:

- Pentium processor

- 32 MB of free hard-drive space
- 128 MB of RAM

2.1.3. Software Interface

This section list the requirements of the system that would be needed to run the System efficiently. It would list the Operating System required that would allow the system to run effectively, the interface that would be required to run the application, the driver required for running .NET web applications and the third party tool that would be used for editing purposes.

1. Operating System: Windows (Vista/Windows 7) or MAC OS
2. Web Brower: Internet Explorer (8.0 and above), Mozilla Firefox (3.0 and above), or Google Chrome
3. Drivers: Java Runtime Environment
4. Third-Party Tool: Microsoft Word

2.2. Product Function

The Document Management System shall provide the following basic functions:

1. The system shall display all the documents stored in the system.
2. The system shall allow the Admin to add comments to his or her feedback.
3. The system shall allow users/Admin to upload documents.
4. The system shall allow the Admin to assign tasks to users for creating/updating current documents.
5. The system shall allow the Admin to accept/reject the documents.

2.3. User Characteristics

The users of the Document Management System, based on their roles, are employees (users) and the Admin (Quality Assurance Specialist). These users are identified based on their experience and technical expertise.

1. **Admin:** The Admin is the Axio employed Quality Assurance Specialist. He or she must have a basic understanding of computers and the internet. He or She should have some prior knowledge in task assignment and standard Axio operating practices. The Admin would be responsible for maintaining all the training documents required for the system. The Admin should be able to perform the following functions:
    - Assign tasks to specific users for creating new documents/making revisions to the current documents.
    - Accept/reject uploads from the user.
    - View the entire history of the documents' revisions and changes.
2. **Users:** The company management would select a group of individuals with enough experience creating guidelines and policies who would be responsible for making changes to the current organizational guidelines and practices. They would be employees of AXIO Research. They must have basic understandings of computers and the internet. They should have some prior knowledge in policy/guideline creation. The users should be able to perform the following functions using this system:
    - View tasks assigned to them.
    - Download required documents for making modifications.
    - Upload new/updated documents

2.4. Constraints

1. **Hardware Limitations:** The minimum hardware requirement for the system would be 128 MB of Ram and a 32-MB hard-disc drive.
2. **Regulatory Policies:** For implementing any system in AXIO, proper legal permission has to be obtained from the Board of Directors and the IT department. If approval is not granted, a desired system cannot be implemented.
3. **Accessibility:** Initially, the software should be available as a desktop application for a small set of users to test.
4. **Others:** The application should be built using .NET and should be accessible through the World Wide Web.

2.5. Assumptions and Dependencies

The assumptions and dependencies are as follows:

1. Users and Admin are accustomed to the paper-based system and would require training for using the Document Management System.
2. The system is completely dependent on the availability of an Internet connection.
3. We assume that the IT department has enough disk space to store all the documents.
4. We assume that users of the system adhere to the system’s minimum software and hardware requirements.
5. This system will use third-party software, and it is assumed that users of the system are familiar with the third-party software.

2.6. Specific Requirements

This section contains details about all the software requirements that will be sufficient for designers to create a system to satisfy those requirements and for testers to test the given requirements. This section contains the interface description of each GUI for the different users involved with the system. This section also gives a description of all the system inputs, all the functions performed by the system, and all the output (responses) from the system.

2.6.1. Functional Requirements

This section contains the requirements for the Document Management System. The Functional Requirements, as collected from the users, have been categorized to support the types of user interactions the system shall have.

1. **Login to the system:** The system shall recognize the user based on the login information (i.e., username and password), and based on the user's role, the system will show a different interface.
    - **FR01:** The Document Management System shall have two types of access/login:
        i. User
        ii. Admin
    - **FR02:** The Document Management System shall only be accessible to specified users and Admin with a valid username/password.
2. **User: View Documents:** The users shall be able to see their designated page after login and to select the view task from the tab interface. The user shall be able to view their assigned tasks.
    - **FR03:** The users shall be able to view the documents assigned to them.
3. **User: Upload/Download Document:** After logging into the system, the users shall be able to see their designated pages. After viewing their task, the users should be able to view the documents added to the system. The users shall be able to download the document they need to update. The Admin shall be able to upload a new document or a revised document.

- **FR04:** The user shall be able to view the documents added to the system.
- **FR05:** The user shall be able to download existing documents.
- **FR06:** The Admin shall be able to upload new/revised documents.

4. **Admin: Assign Document**

- **FR07:** The Admin shall be able to view all the users registered in the system.
- **FR08:** The Admin shall be able to select users and to assign them a document.

5. **Additional Functional Requirements**

- **FR 09**: The Admin shall be to view the entire history for a document's revisions.

6. **User: View/Send Messages:** After logging into the system, the users shall be able to see their designated pages. The system shall allow users to view messages from other users who are enrolled in the system. The system shall allow users to send messages to other users enrolled in the system and to attach files to those messages.

- **FR10:** The user shall be able to view the messages sent to them.
- **FR11:** The user shall be able to send messages to other users.
- **FR12:** The system shall allow users to attach files to a message.

7. **User: Edit Profile/Add/Delete Users:** After logging into the system, the users shall be able to see their designated pages. The Admin should be able to view all the users enrolled in the system. The system shall allow the Admin to edit/add users enrolled in the system. The system shall allow all the users to edit their personal profiles.

- **FR13:** The system allows the users to edit their personal profiles.
- **FR14:** The system allows the Admin to edit user profiles.
- **FR15:** The system allows the Admin to create new users.

2.6.2. Performance Requirements

This section would list the Performance Requirements expected from the system while in using within AXIO

1. **PR01:** The user shall be able to login to the system in less than 4 seconds.
2. **PR02:** The navigation between pages shall take less than 5 seconds.

2.7. Design Constraint

This section would list the design requirements to be followed by the system as part of Axio Software development guidelines.

1. **DC01:** If the logged in user does not perform any operation for more than 30 minutes, the system shall guide the user to the login page to re-login to the system.
2. **DC02:** The User Interface (UI) must have specific fonts and font sizes. The System shall match the fonts and font sizes used in all Axio applications. The default type size is 12, and the default font is Times New Roman.
3. **DC03:** The UI shall have a help section to guide users (Admin and users).
4. **DC06:** The system shall ask users for the location of the file from which they want to browse the file for upload and the location to which they would like to save the file.

2.8. Software System Quality Attribute

1. **Integrity:**
    - **QA01:** The authorized user shall be allowed to access the Document Management System.
    - **QA02:** Based on the type of user, the Document Management System shall provide a user-specific interface.

2. **Correctness:**

   - **QA03:** The assigned task should be received by the specified user.

3. **Availability:**

   - **QA05:** The system shall be made available to the User/Admin year round.

4. **Robustness:**

   - **QA07:** The system shall be able to save the documents. If the database server goes down, the system should generate an error message for the users (Admin and Users).

CHAPTER 3. IMPLEMENTATION

This chapter includes the detailed design used to build the Document Management System. The system's mock screen shots are also shown.

3.1. Detailed Scope

This project is supposed to be delivered in three main phases, with each phase being an add-on to the project that makes it more usable and more acceptable.

1. In the first delivery, the application must be able to add a customer profile and case.
   - Add users.
   - Update user profile
   - Add documents.
   - Give document rights to users.
2. The application must be able to send messages to users.
   - Send and view messages.
   - Tag messages together.
   - Upload/download documents through messages.
3. After the first two phases of the project have been completed and thoroughly tested after a joint decision by the management team and the IT division, the system would be enhanced into a joint Document Management and User Training System.

3.2. Static Decomposition and Dependency Description

This section contains the System Use Case diagram for the Document Management System (DMS) and also has a detailed explanation for each use case in the System Use Case.

3.2.1. High-Level Use Case Diagram

The System Use Case shows the user a detailed view of the system and how the actors would interact with each other and with the system. The explanation for each use case is then provided below the System Use Case (Figure 1) and helps the user understand who the actors areas well as giving the description of each use case along with its pre and post conditions that should be satisfied once the use case is implemented in the software.

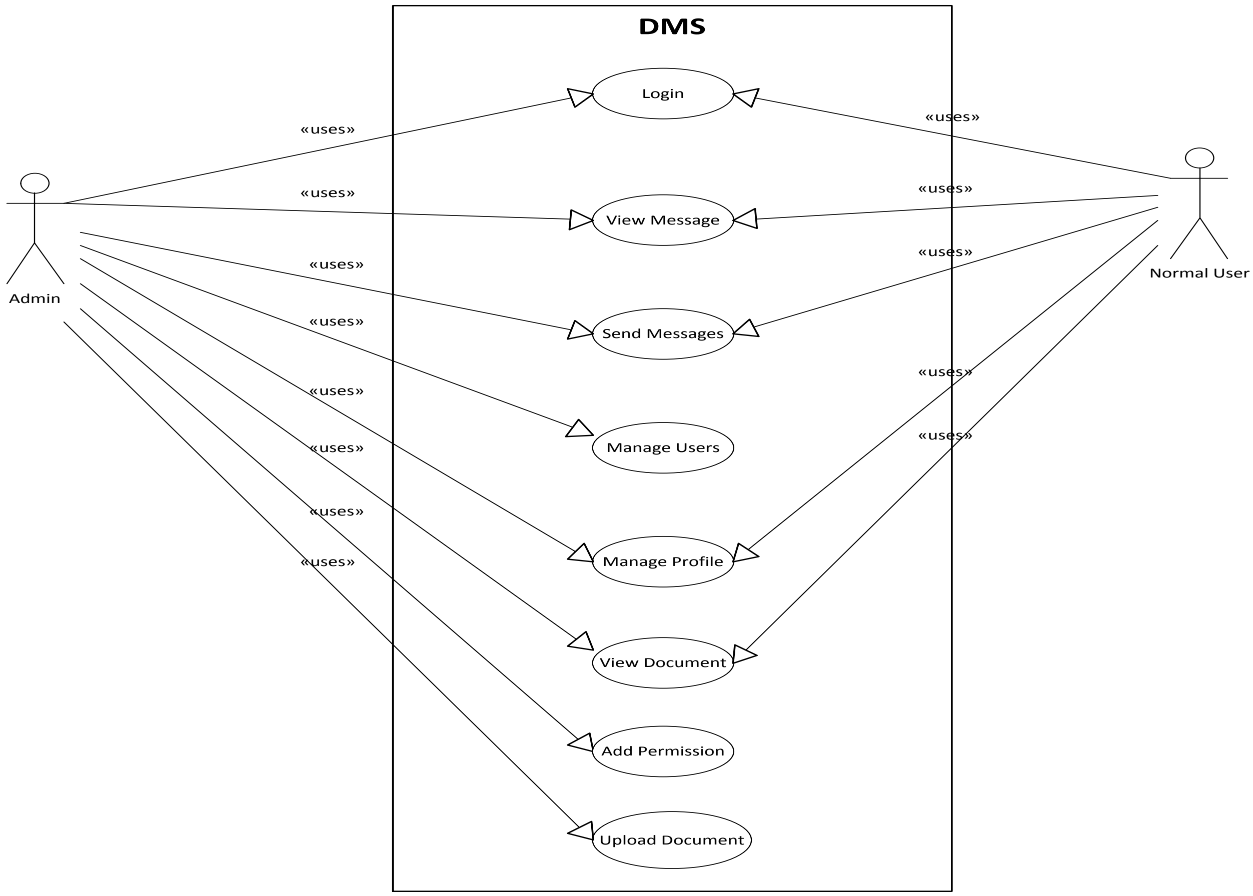


Figure 1. Document Management System (DMS) System Use Case Diagram.

Below is the explanation of the different Use Cases in the System Use case and the Actors associated with each Use Case. The Description is used for a novice user to better understand the working of the System and the pre-conditions that should be satisfied before invoking each use case.

1. **Use Case Number:** US-001
   - **Application:** DMS
   - **Use Case Name:** Login
   - **Use Case Description:** This details the login process of the system to access it. The user should login to the system using a username and password.
   - **Primary Actor:** Admin/User
   - **Precondition:** The user is already assigned a username and password to access the system
   - **Postcondition:** The user logs in to obtain access to the system.
   - **Basic Flow:**

     i. Browse the website

     ii. Select the Login

     iii. Enter the username and password

     iv. Submit

     v. Show the welcome screen
   - **Exceptional Flow:**

     i. Browse the website

     ii. Select the login

     iii. Enter the username and password

iv. Show the login failed message and recovery options

2. **Use Case Number:** US-002

- **Application:** DMS
- **Use Case Name:** Upload Document
- **Use Case Description:** This use case will help the user to browse documents in the system.
- **Primary Actor:** Admin/User
- **Precondition:** The user is already assigned a username and password to access the system
- **Postcondition:** The user successfully uploads the document..
- **Basic Flow:**

i. Log in to the system

ii. Select Document Management from the options

iii. Browse for a selected file

iv. Upload the document

- **Exceptional Flow:**

i. Log in to the system

ii. Select Document Management from the options

iii. Browse for a selected file

iv. Click upload without selecting the file, hence the error message

3. **Use Case Number:** US-003

- **Application:** DMS
- **Use Case Name:** Add Permission
- **Use Case Description:** This use case will help the Admin to add permission for the documents in the system.
- **Primary Actor:** Admin
- **Precondition:** The Admin is already assigned a username and password to access the system and has selected Document Management.
- **Postcondition:** The user is successfully able to add permission to the document.
- **Basic Flow:**

  i. Log in to the system

  ii. Select Document Management from the options

  iii. Click on the file name

  iv. Select the user to give permission

- **Exceptional Flow:**

  i. Log in to the system

  ii. Select Document Management from the options

  iii. Click on the file name

  iv. Users do not exist in the system

4. **Use Case Number:** US-004

- **Application:** DMS

- **Use Case Name:** Manage Profile
- **Use Case Description:**

  i. This use case will help the Admin to Edit/Add/Delete a user's profile.

  ii. This use case also allows the users to manage their personal profiles.

- **Primary Actor:** Admin
- **Secondary Actor:** Admin/User
- **Precondition:** Admin/normal users are already assigned a username and password to access the system and to select Profile Management.
- **Postcondition:** The Admin successfully Edits/Adds a user profile.
- **Basic Flow:**

  i. Log in to the system

  ii. Select Profile Management from the options

  iii. Click on the user's name

  iv. Update the user's email address/password

- **Exceptional Flow:**

  i. Log in to the system

  ii. Select Profile Management from the options

  iii. Click on the user's name

  iv. Enter the wrong email address/password

5. **Use Case Number:** US-003

   - **Application:** DMS

- **Use Case Name:** Message Management
- **Use Case Description:** This use case will help the Admin/User to view/send messages.
- **Primary Actor:** Admin/User
- **Precondition:** Users are already assigned a username and password to access the system and select Message Management.
- **Postcondition:** Users are able to view/send messages
- **Basic Flow:**

i. Log in to the system

ii. Select Message Management from the options

iii. Click on Inbox (to view messages)/Compose (to send new messages)

iv. View/Compose message

- **Exceptional Flow:**

i. Log in to the system

ii. Select Message Management from the options

iii. Click on Inbox (to view messages)/Compose (to send new messages)

iv. View/Compose message

3.2.2 Activity Diagram

This section would list the Activity diagram and would describe the flow of Activities in the System. A detailed description is than followed after the figure for each activity. Figure 2 provides the overview of the Document Management System

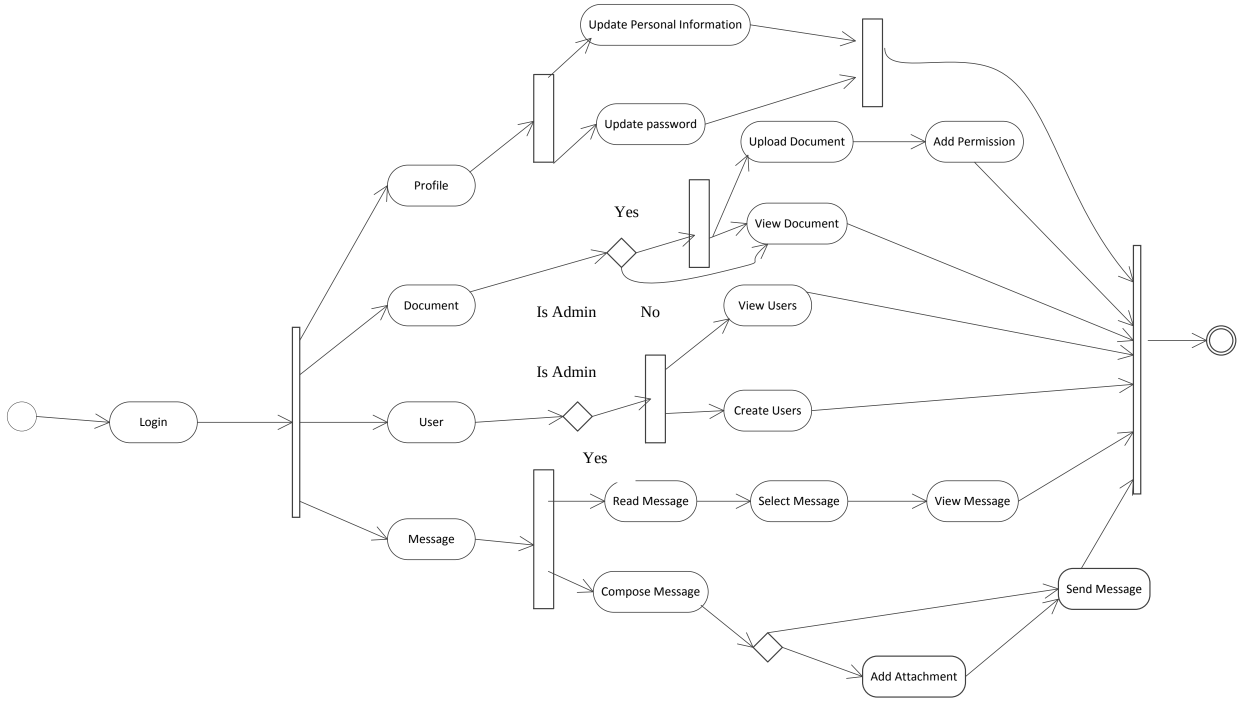


Figure 2. Activity Diagram for DMS.

The flow begins when the user starts the web desktop application. A login screen appears in the web browser, prompting the users to enter their ID and password. After the user logs in to the system, the system decides whether the user is an Admin or a normal user and then shows the appropriate user webpage. Users have a defined set of options for navigation once they log into the system:

1. **Message:** Under the Message tab, the users have three options, Compose/Inbox/Sent folders, to which they can navigate based on their choices.
    - **Compose:** The "compose" section of the message allows the user to send private messages to other users in the system via either their ID or email ID. The section also has a separate button to allow the users to access their computer folders and to add any attachments they want.

- **Inbox:** This section has all the messages sent to the user by other system users. It lists the number of unread messages in the system. It also tags all the messages exchanged between the system user and others based on their ID and message header. It also shows as one conversation.
- **Sent:** This section contains all the messages sent by the user to the other users with date and time stamp.

2. **User:** This tab only appears to the Admin and contains two sections: List/Create Users.
   - **List:** This section shows the Admin all the users enrolled in the system and also gives the Admin privileges to edit/delete the users.
   - **Create:** This section allows the Admin to add new users to the system.
3. **Document:**
   - **When the user is the Admin:** This tab contains the button to browse the folders on the Admin local machine and allows him or her to upload a document. Once a document is uploaded, the system would display a link to the document with a time stamp attached to each document. (Exception: No two documents with same name are allowed to be uploaded into the system.) If the Admin chooses to click on the link for the document created, the Admin would then have the option to select a user and give the user the permission to view and download the document.
   - **When the User is Normal User:** This tab contains the link to all the documents for which the Admin has given the user permission to view and download. When the user clicks the link, he or she has the option to save the file anywhere on his or her local machine.

4. **Profile:** This tab appears for all the users and allows the users to edit their personal profiles with their first/last names and passwords.
5. **Category:** This tab appears for the Admin and lets the Admin assign the documents uploaded into the system to a category created in the system. It also allows the Admin to assign a user to a category, allowing the user to view and download all the documents in that category; and the user would also have an option to download the documents in Zip files.
6. **Department:** This tab appears for the Admin and lets the Admin assign a category to the user, allowing the user to view and download all the documents and categories in that department, and the user would also have an option to download the documents in Zip files.

3.2.3. Class Diagram

1. **User Management:** This class is used to get the user information from the database and is also used for authenticating the users. Figure 3 shows the methods that are used in this class.
   - **Authenticate User:** This message is used to authenticate a particular user who has provided the login credentials and wishes to login in the system. This method checks the credentials from the database.
   - **Check User Name:** This method is called in the Admin section, when the Admin creates a user. Before creating the user, this method checks to see if the provided user name is already in use.
   - **Create User:** This method will save the details provided in the database. The user login will be functional immediately after the execution of this method.
   - **Get All Users:** This method displays the complete list of available users in the system.

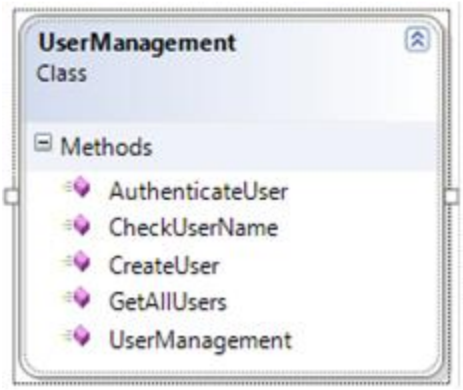


Figure 3. User Management Class Diagram.

2. **Role Management:** The class manages the user roles. It contains methods such as GetAllRoles and Role Management as shown in Figure 4

    - **Get All Roles:** This role fetches all the available roles in the system and populates the dropdown menu in the Admin section while creating the user

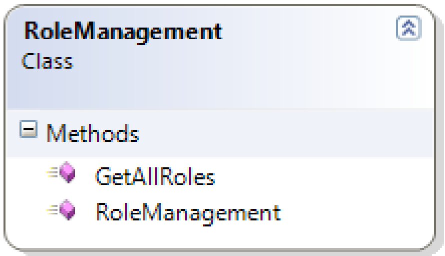


Figure 4. Role Management Class Diagram.

3. **Message:** This class is used for getting information from the database and is also used for updating the database with the MessageId and information used to process the message. Figure 5 shows the methods that are in this class.

    - **Get Completion List:** This role fetches the user name starting from the characters provided in the textbox. It is consumed by a web service that caters the data to Ajax Auto Complete Extender.

    - **Get Message by User Name:** This role populates the inbox for the logged-in user.

- **Get Message Details by ID:** This role populates the details of a selected message.
- **Insert Message:** This role saves a message sent to one or more recipients.
- **Insert Reply:** It is used to insert a reply to the message.

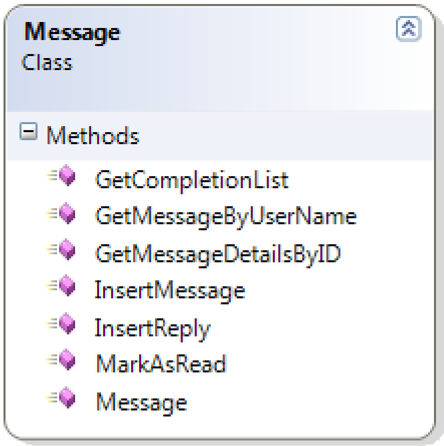


Figure 5. Message Class Diagram.

- **Mark as Read:** It is used to mark any message as Read.

4. **Document:** This class is used to process all information regarding the documents. Figure 6 shows the method involved in this class**.**

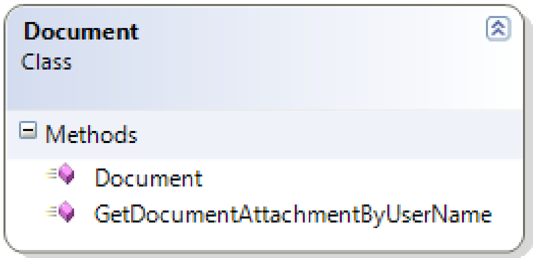


Figure 6. Document Class Diagram.

3.2.4. Component Dependency Diagram

This section lists the Components that are used while creating the Document Management System. It would describe the Components the detailed description of each component and the dependency of each component in the Diagram.

The Figure 7 below gives the overall perspective of the Component diagram, where each component represents the structure that was used while creating the Document Management System. It also represents how each component is liked with each other and the flow of information that is shared between them.

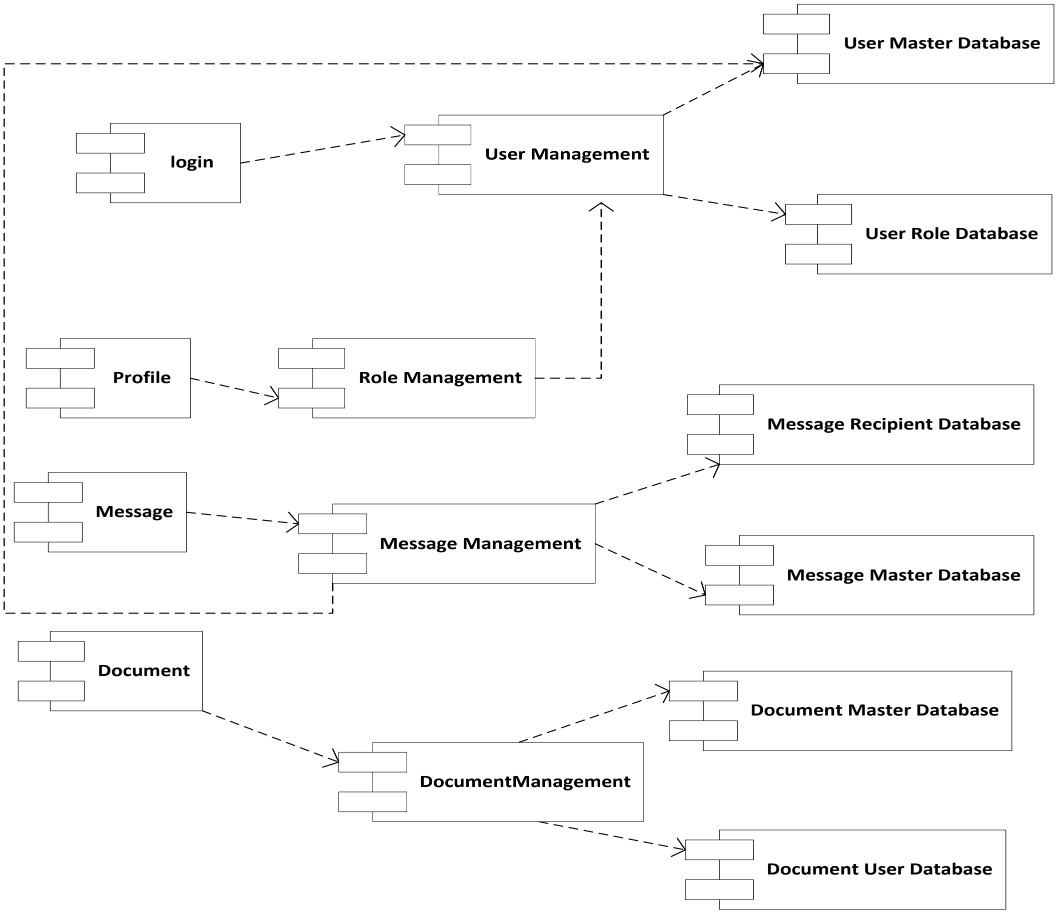


Figure 7. Component-Level Diagram.

3.2.4.1. Component Dependencies

This section provides an overview of the components in table with a short description of all the Component Links in Table 1

Table 1. Component Link Description.

Component Link	Component Link Description
1	The login component is dependent on the user component to authenticate user access and to manage necessary privileges for each level of user.
2	The user component authenticates a user using security infrastructure and the user database.
3	The user component authenticates a user by accessing necessary credentials from the user database.
4	The profile component is dependent on the user component and the role component. The profile component manages necessary profile updates for a new user or an existing user.
5	The role component depends on the user component to update the user-profile updates.
6	The document component is dependent on the Document Management component and manages the necessary documents and document privileges.
7	Document Update is dependent upon the Document Master and Document User to manage documents and Document Privileges.
8	The message component is dependent upon the message-management component and the user-management component which get the list of users and uses to manage the messages.

3.2.4.2. Component Description

This section provides the user with a detailed description of each component along with the requirement number to which the required data adhere, preconditions that need to be satisfied, and the database tables with which they interact.

1. **Login Component:** The login management component handles the role-based authentication of existing users and the access privileges for different system components.

   - **Requirement Number:** F001

     i. **Requirement description:** The system shall have two different roles for users, or two different access privileges: Admin and user.

- **Requirement Number:** F002

   i. **Requirement Description:** The system shall require a username and a password for authenticating different kinds of users.

- **Component Data:** This component maintains the user data for the purpose of verifying authentication credentials.
- **Preconditions:** To function properly, the login component should be able to communicate with the user component and should provide that component with the necessary information to manage user roles and access privileges.
- **Tables:** This component does not access any tables directly.

2. **Profile Component:** The Profile Component handles the Add/Edit feature for the user profiles. For example, it allows the Admin to change the last name of a user from "A" to "B" to change the user's email ID, or to add a new user to the system.

- **Requirement Number:** F003

   i. **Requirement description:** The system shall allow the Admin to add a new user.

- **Requirement Number:** F004

   i. **Requirement Description:** The system shall allow the Admin to update the profiles for existing users.

- **Component Data:** For data, this component accesses the user.
- **Preconditions:** To function properly, the Profile Component should be able to communicate with the user components and to update the user database component with up-to-date information for later ease of access.
- **Table:** This component does not interact with the System directly.

3. **Document Component:** The Document component handles the Upload/View/Add permissions of different documents. For example, normal users can view/upload documents shared with them, and the Admin can add permissions to documents for each user separately.

   - **Requirement Number:** F005
     1. **Requirement description:** The system shall allow the Admin to upload documents into the system.
   - **Requirement Number:** F006
     1. **Requirement Description:** The system shall allow the Admin to add permissions to a document for the users to view them.
   - **Requirement Number:** F007
     1. **Requirement Description:** The system shall all normal users to view and download the documents shared with them by the Admin.
   - **Component Data:** For data, this component accesses user, document management
   - **Preconditions:** To function properly, the Document Management component should be able to communicate with the user (to access the users added to the system), to store the documents in the system, and to add permission for the users and collect from those components the necessary information.
   - **Table:** This component does not have direct access to any table.

4. **Message Component:** The Message Component handles the View/Send messages to different users. For example, if the Admin wants a user to update a certain document, he or she can send the user a message, and when the user sees that message, the user can access the document and upload the updated document into the system as a message attachment.

- **Requirement Number:** F008
    1. **Requirement description:** The system shall allow the users to View/Send and Receive messages.
- **Requirement Number:** F009
    1. **Requirement Description:** The system shall allow the users to upload and download attachments.
- **Requirement Number:** F010
    1. **Requirement Description:** The system shall allow the users to view all messages with the same subject together.
- **Component Data:** For data, this component accesses user, message management component.
- **Preconditions:** To function properly, the message's Management component should be able to communicate with the user (to access the users added to the system), message components and collect from those components necessary information.
- **Table:** The component has direct access to the message table.

5. **User Component:** The User Component authenticates a user by utilizing the security infrastructure and the user database.

- **Requirement Number:** F011
    1. **Requirement description:** The system shall have two different roles for users, or two different access privileges: the Admin and the user.
- **Component Data:** This component maintains the user data for the purpose of verifying authentication credentials.

- **Preconditions**: To function properly, the User Component should be able to communicate with the User Database component.
- **Table:** The component accesses the User Master and User Role Database tables.

6. **Role Management Component:** The Component handles the Add/Update request made by the Profile Management Component for every new user or previous user in the system.

- **Requirement Number:** F012
  i. **Requirement description:** The system shall allow the Admin to create new users by providing their usernames, passwords, and email IDs. The system should also allow the Admin to update information about the current users in the system.
- **Component Data:** This Profile Update component accesses the user database to Add/Find information for any user.
- **Preconditions:** To function properly, this component should be able to communicate with the component accesses Users Management Component and provide that component with necessary information.
- **Table:** This component does not access any tables directly.

7. **Document Management Component:** This component uploads a new document and adds permissions to users for viewing the document.

- **Requirement Number:** F013
  i. **Requirement description**: The system shall allow the Admin to upload documents/add permission to documents and shall allow the user to view documents.
- **Component Data:** For data, this component accesses the Document Master and Document User Database Components.

- **Preconditions:** To function properly, the Document Management Component should be able to communicate with the login, Document Master, and Document User Components and to collect the necessary information to generate a report from those components.
- **Table:** The component accesses the Document Master and Document user database tables.

8. **User Master Component:** The User Master component has all of the information, such as email ID, username, password etc., for each user added to the system.

   - **Requirement Number:** F014

     i. **Requirement description**: The system shall have two different roles for users, or two different access privileges: Admin and user.

   - **Requirement Number:** F015

     i. **Requirement description**: The system shall require a username and a password for authenticating different users.

   - **Component Data:** This component maintains the user profile related information.
   - **Preconditions:** To function properly, the User Database Component must have SQL Server database connectivity and should be able to communicate with the software.
   - **Table:** This component accesses the user table.

9. **User Role Component:** The User Role component has all the users' information regarding their roles added to the system.

   - **Requirement Number:** F016

     i. **Requirement description**: The system shall have two different roles for users, or two different access privileges: Admin and user.

- **Requirement Number:** F017

  i. **Requirement description**: The system shall require a username and a password for authenticating different users.

- **Component Data:** This component maintains the user-role information.

- **Preconditions:** To function properly, User Role Database component must have SQL Server database connectivity and should be able to communicate with the software.

10. **Message Master Database Component:** The Message Master Database component has all of the information regarding a message such as Message root ID, Message ID, Message Text, sender info, Date and Time

11. **Message Recipient Database Component:** The Message Recipient Database component has all of the information such as Message root ID, Message ID, recipient ID

12. **Document Master Database Component:** The Document Master Database component has all of the information regarding the Document such as Document name, Category, Department, Document ID, Date and Time

13. **Document User Database Component:** The Document User Database component has all of the information regarding the permission on a document such as Document ID, category, Department and UserID.

3.3. Interface Description

This section describes the users the different interfaces of the System. It would contain a detailed description of each interface along with a screen shot of the Interface.

1. **Login Interface:** The login interface (Figure 8), is common to all the users/the Admin of the system. This interface shall be made available through the web application. At the time of logging in, based on the user ID, the system decides on the type of user.

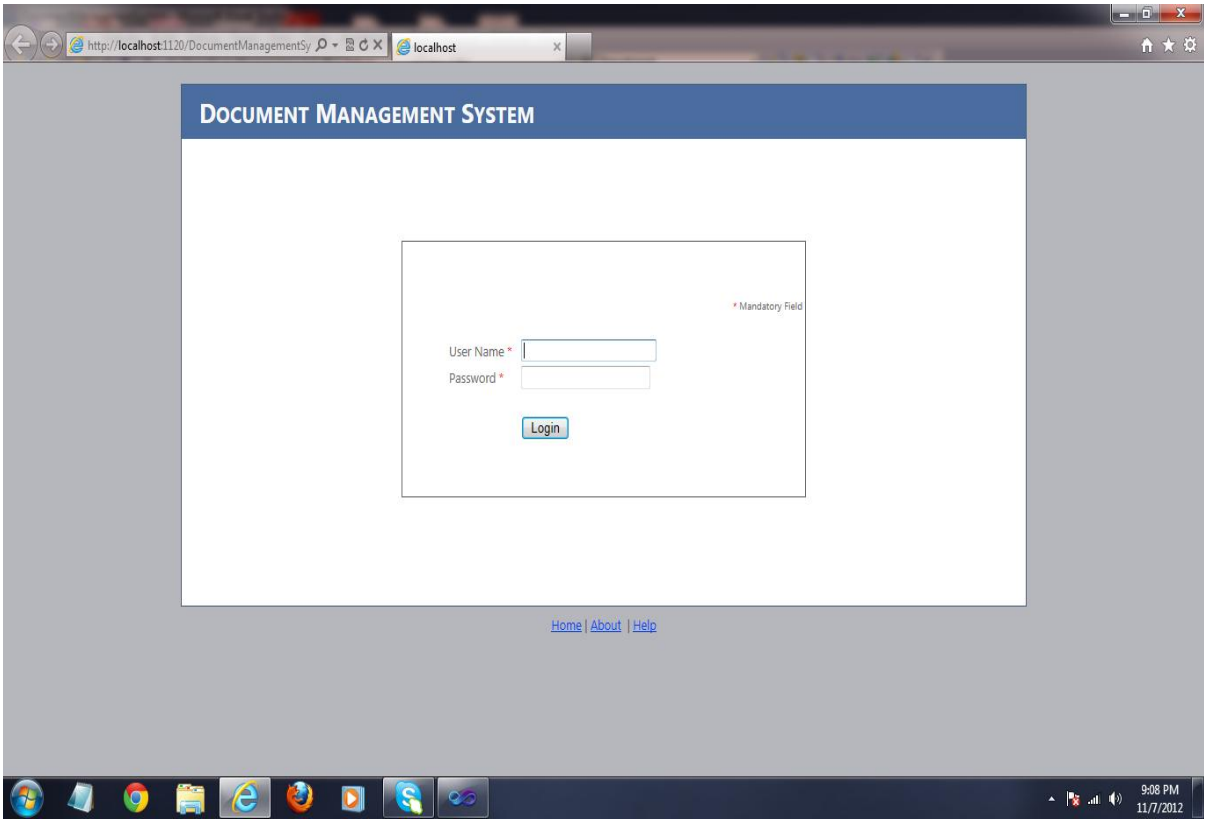


Figure 8. Screenshot of the Login Interface.

2. **Document Interface:** It constitutes two interfaces, i.e., one for the normal users and one for the Admin. The Admin interface has the option for the Admin to view all documents and to upload documents to the system. It also has a link to all the uploaded documents through which the Admin would be able to assign users to the documents. The interface for normal users would allow the users to see all the documents to which they have been assigned by the Admin.

- **Upload Documents:** The Upload Document Interface (Figure 9)is used for uploading the document.

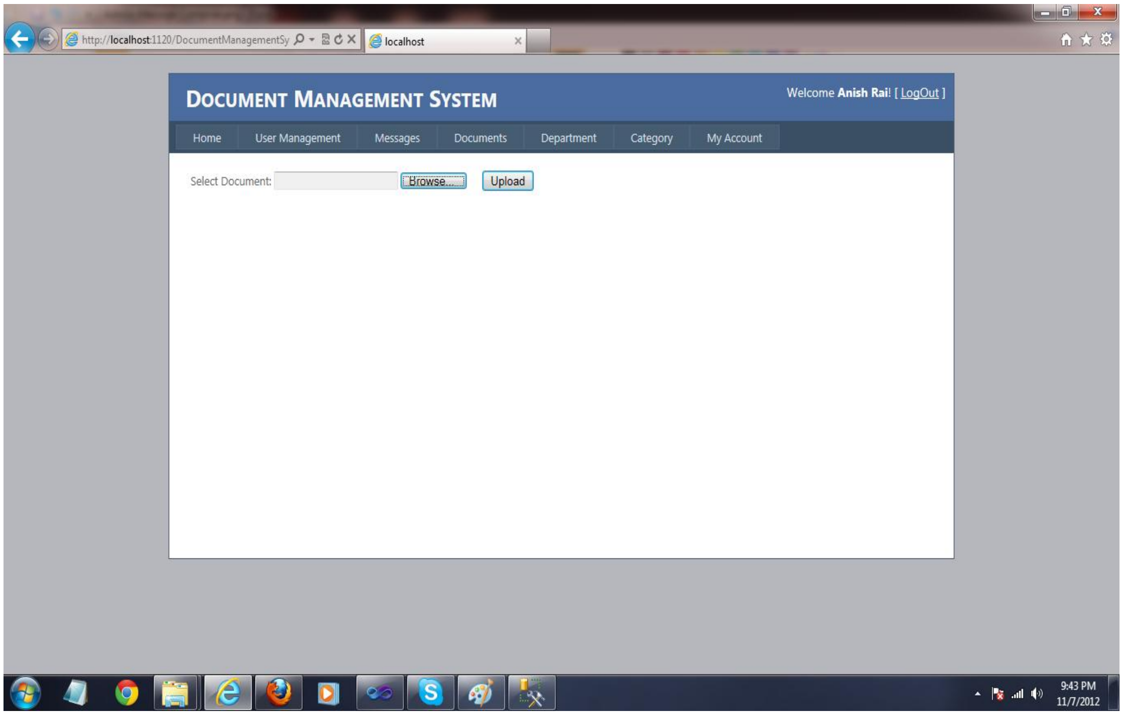
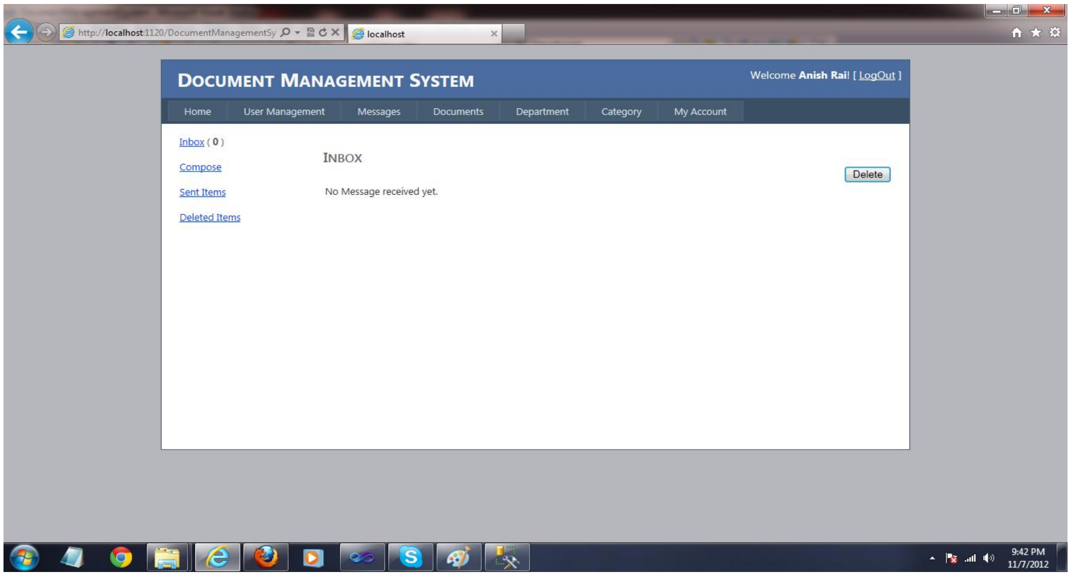


Figure 9. Screenshot of the Upload Interface.

3. **Message Interface:** It constitutes two interfaces, i.e., one for the user to read messages and the other for users to send messages. This interface helps the Admin and normal users to interact effectively and also, at the same time, allows the normal users to send documents to the other users and the Admin as an attachment.

- **Read Messages:** Utilizing this interface (Figure 10), users can read the messages sent to them by other system users.

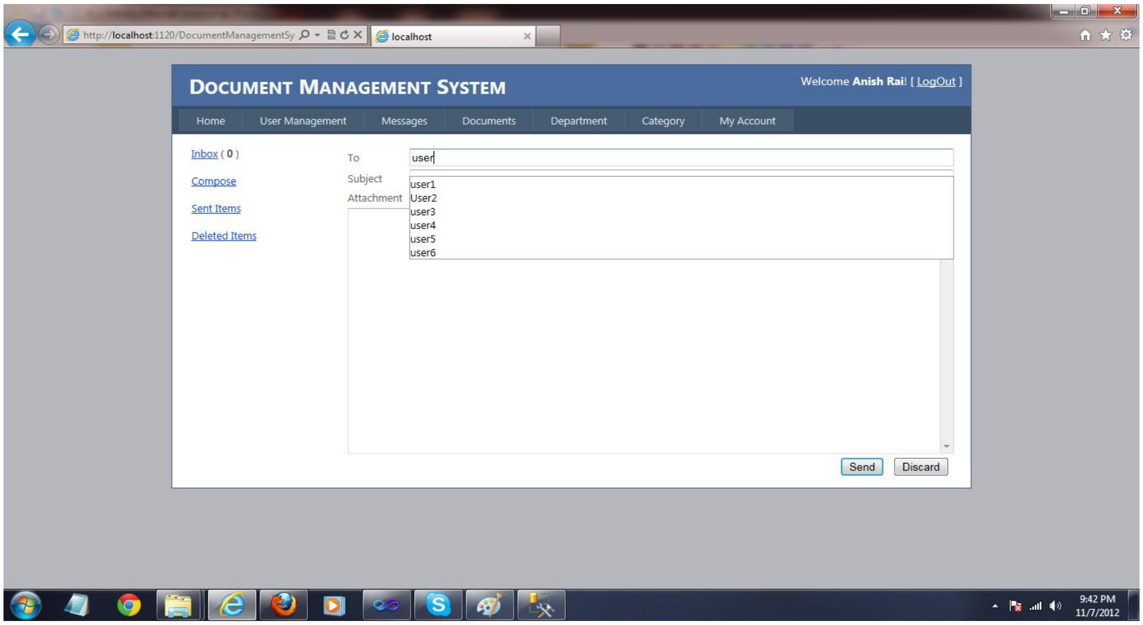


INBOX

No Message received yet.

Figure 10. Screenshot of the Message Interface.

- **Send Messages:** Using this interface (Figure 11), the user should be able to send a message to another user and should also be able to attach any file into the message before sending the message



To: user

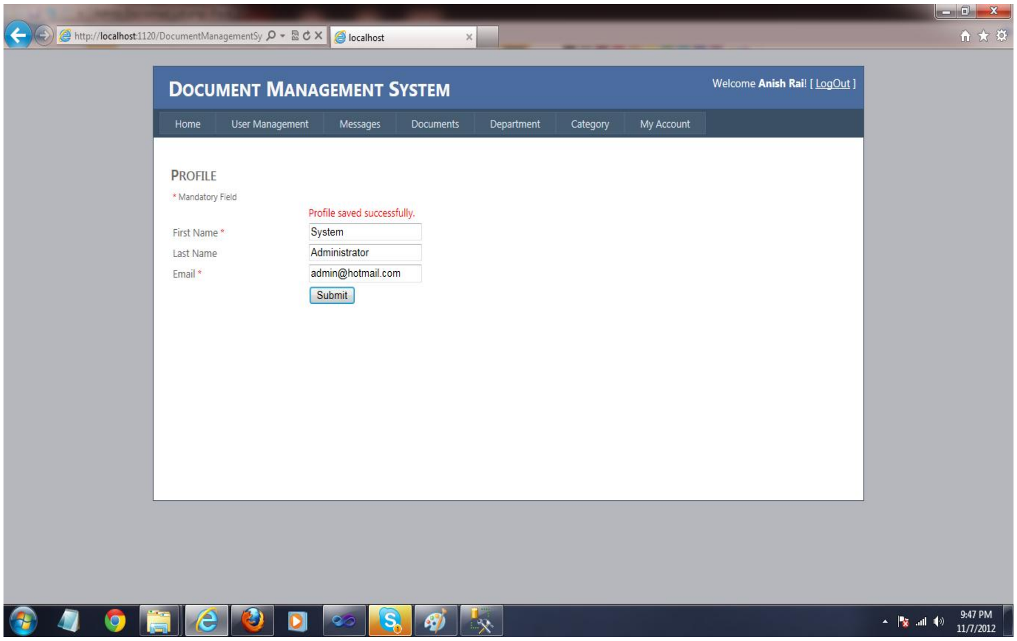
Subject

Attachment

Figure 11. Screenshot of the Send Message Interface.

4. **Profile Interface:**It constitutes two interfaces, i.e., one for the user to read messages and the other for users to send messages. This interface helps the Admin and normal users to interact effectively and also, at the same time, allowsnormal users to send documents to other users and the Admin.

    - **Edit Profile:**Using this interface (Figure 12), the users can edit their own profiles.TheAdmin can also edit other users' profiles.


http://localhost:1120/DocumentManagementSy
localhost
DOCUMENT MANAGEMENT SYSTEM
Welcome Anish Rai! [LogOut]
Home
User Management
Messages
Documents
Department
Category
My Account
Submit
9:47 PM
11/7/2012

PROFILE

* Mandatory Field

Profile saved successfully.

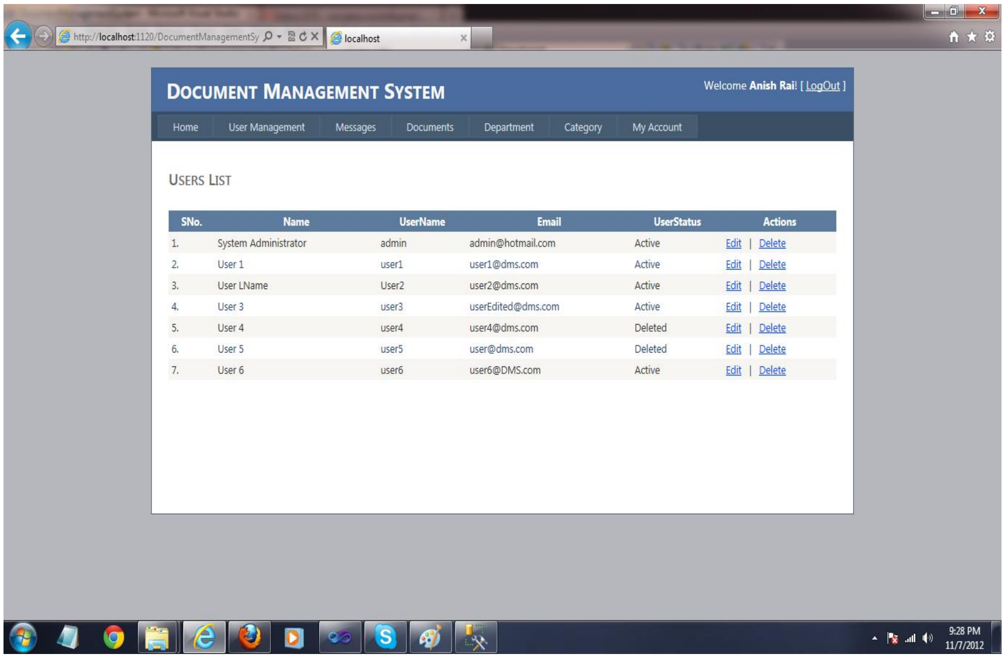
First Name *: System

Last Name: Administrator

Email *: admin@hotmail.com

Figure 12. Screenshot of the Profile Interface.

5. **View Users:** Utilizing this interface (Figure 13), the user should be able to send a message to another user and should also be able to attach any file to the message before sending it.



http://localhost:1120/DocumentManagementSy
localhost
Document Management System
Welcome Anish Rai! [LogOut]
Home
User Management
Messages
Documents
Department
Category
My Account
9:28 PM
11/7/2012

Users List

SNo.	Name	UserName	Email	UserStatus	Actions
1.	System Administrator	admin	admin@hotmail.com	Active	Edit | Delete
2.	User 1	user1	user1@dms.com	Active	Edit | Delete
3.	User LName	User2	user2@dms.com	Active	Edit | Delete
4.	User 3	user3	userEdited@dms.com	Active	Edit | Delete
5.	User 4	user4	user4@dms.com	Deleted	Edit | Delete
6.	User 5	user5	user@dms.com	Deleted	Edit | Delete
7.	User 6	user6	user6@DMS.com	Active	Edit | Delete

Figure 13. Screenshot of the View Users Interface.

CHAPTER 4. TESTING

This chapter follows up with the testing methods that were used during the validation of the system. The Conclusion and the Future Work for the software are also given.

4.1. Methodology

The method used while testing this software was different than the conventional testing route followed in the software industry (Antonio Bertolino, 2005) (Bochmann, G.V,. Petrenko, 1994) (Braind, Labiche, 2002) (Binder, 1999). This testing approach was valuable for the software and was easier because the company was familiar with the methodology.

In this approach, as the specs were ready for a prototype to be shown, the tester started writing his or her code and saw if he or she could obtain the same results as the specs mentioned. This way, the specs were tested on each prototype, and continuous testing was applied. This also helped in minimizing the testing that would have to be implemented at the end of the software lifecycle. In the process, all aspects of the software were tested. Steps to follow while implementing the methodology:

Start with a base functionality that you want to implement.

Create a document with the detailed requirement definition, an activity diagram with a description of the flow, database tables that would be used and component diagram and description of each component with precondition and tables that would be affected by the component.

Give the document to the tester, and work with the tester while he or she writes the code to check if the steps in the document can be implemented and if the result of each use case can be achieved.

If the tester finds a step difficult to implement or thinks he or she is missing additional information to implement the functionality, then go to step 2; otherwise, go to step 3.

Ask the tester to log all the errors and difficulties he or she encountered while working on the prototype implementation.

Once the prototype is done and the results match between the developer’s prototype and tester’s prototype, work on the other requirement, and expand the prototype to final software.

When the testing approach was implemented the following pros and cons regarding the testing approach were realized.

Pros of using the methodology:

Helps give a better understanding about the requirements

Better design at the end of the cycle

Reduced testing to be performed at the end of the cycle

Documents produced would be of higher quality.

Cons of using the methodology

The person working on the document should be experienced.

There are increased time and money involved in testing.

Different viewpoints for the same problem can lead to different results.

4.2. Interface Testing

This section lists the functional requirements that were used for creating the test cases table, the test cases that were used for verifying the Interface table and the results for the test cases table.

The table below list the Functional requirements for the Interface built for the Document Management System, along with a short description of each requirement

Table 2. List of Functional Requirements.

Functional Requirement Number	Functional Requirement Short Description
FR01	The Document Management System shall have two types of access/login: User/Admin
FR02	The Document Management System shall only be accessible to specified users and Admin with a valid username/password
FR03	The users shall be able to view the documents assigned to them
FR06	The Admin shall be able to upload new/revised documents
FR07	The Admin shall be able to view all the users registered in the system
FR08	The Admin shall be able to select users and to assign them a document
FR09	The Admin shall be to view the entire history for a document's revisions
FR10	The user shall be able to view the messages sent to them
FR11	The user shall be able to send messages to other users
FR12	The system shall allow users to attach files to a message
FR13	The system allows the users to edit their personal profiles
FR14	The system allows the Admin to edit user profiles
FR15	The system allows the Admin to create new users

4.3. Test Cases

The Table 3 below shows the Functional requirements that were used to write the test cases along with the test case numbers for each test case and a short description of the test cases

Table 3. List of Test Cases.

Functional Requirement No.	Test Case No.	Test Case Short Description
FR01	TC01	To test the Login Interface for Admin
	TC01	To test the Login Interface for User
FR03	TC02	To test users can to view the documents assigned to them
FR06	TC03	To test Admin can upload new/revised documents
FR07	TC04	To test Admin can view all the users registered in the system
FR08	TC05	To test Admin can select users and assign them a document
FR09	TC06	To test Admin can view the entire history for a document's revisions
FR10	TC07	To test users are able to view the messages sent to them
FR11	TC08	To test users are able to send messages to other users
FR12	TC08	To test system allow users to attach files to a message
FR13	TC09	To test system allows the users to edit their personal profiles
FR14	TC10	To test system allows the Admin to edit user profiles
FR15	TC10	To test system allows the Admin create new profile

The following list include the steps that should be taken by the user, the conditions that should be met for the successful execution of the test case and the end result that should be met for the test cases to pass.

1. TC01.1: To test the Login Interface
   - Input: Username and Password
   - Output: Valid Destination Page
   - Valid Range: User Name →Alphanumeric, Password → Alphanumeric
   - End Messages/Result:
     i. If (User == Valid User) , a message should be displayed Welcome, <User Name> (This shall be displayed on the respective screen.)
     ii. If (User != Valid User), an error message shall be displayed on the Login interface.
2. TC02: To test users can to view the documents assigned to them

- Description of Purpose: The system shows all the saved documents. The user can choose a document and download it.
- Input: User select a document under Category
- Output: Browser or third part tool (i.e., MS Word)
- End messages/Result:
    i. If (Selection == Documents and document == exists): the user will be able to download and view the document
    ii. If (Selection == Documents and document != exists): message should display “there are no documents in this category”

3. TC03: To test Admin can upload new/revised documents
    - Description: For the Admin, an upload button is assigned and allows the Admin to upload a document into the system
    - Input:
        i. User Name →Alphanumeric, Password → Alphanumeric
        ii. User==Admin
        iii. Selection==Documents
    - Output: System
    - End messages/Result:
        i. If(login type == “Admin” & uploadbutton.clicked = ‘true’ and document!=existing) then display “Document saved Successfully”
        ii. If(login type == “Admin” & uploadbutton.clicked = ‘true’ and document==existing) then display “Document already exists”
4. TC04: To test Admin can view all the users registered in the system

- Description: For the Admin, a tab I created under User Management called “List” to list all the users in the system.
- Input:
  1. User Name →Alphanumeric, Password → Alphanumeric
  2. User==Admin
  3. Selection==User Management → List
- Output: User List
- End messages/Result:
  1. If(login type == “Admin” & UserManagement.clicked = ‘true’ and list.clicked=true and userlist.exists=true) then display users
  2. If(login type == “Admin” & UserManagement.clicked = ‘true’ and list.clicked=true and userlist.exists=false) then display message “No users Exist”

5. TC05: To test Admin can select users and assign them a document

- Description: For the Admin, a link to each document is created and allows the admin to assign a user to a document.
- Input:
  1. User Name →Alphanumeric, Password → Alphanumeric
  2. User==Admin
  3. Selection==Permission → Document
- Output: Message
- End messages/Result:

i. If(login type == “Admin” & Permission.clicked = ‘true’ and Document.clicked=true and userlist.exists=true and name.clicked=true) then assign user and display message “task assigned”

ii. If(login type == “Admin” & Permission.clicked = ‘true’ and Document.clicked=true and userlist.exists=false) then display message “No users Exist”

6. TC06: To test Admin can view the entire history for a document’s revisions

- Description: The system shows all the documents saved in the system. The user can choose a document and download it.
- Input:

i. User Name →Alphanumeric, Password → Alphanumeric

ii. User==Admin

iii. Selection==Category → Document

- Output: System
- End messages/Result:

i. If(login type == “Admin” & Category.clicked = ‘true’ and Document.clicked=true) then display document with Date and time stamps

ii. If(login type == “Admin” & Category.clicked = ‘true’ and Document.exists=false) then display message “No Documents exists”

7. TC07: To test users are able to view the messages sent to them

- Description: The system shows all the messages sent to the user. The user can select any messages and should be able to view them.
- Input:

i. User Name →Alphanumeric, Password → Alphanumeric

ii. User==Admin/Users

iii. Selection==Message → Inbox

- Output: System
- End messages/Result:

i. If(login type == “Admin”/”User” & message.clicked = ‘true’ and inbox.clicked=true and message.exists=true and message.clicked=true) then display message

ii. If(login type == “Admin”/”User” & message.clicked = ‘true’ and inbox.clicked=true and message.exists=false) then display “No message received yet”

8. TC08: To test users are able to send messages to other users

- Description: The system shows all the users to send messages to other users
- Input:

i. User Name →Alphanumeric, Password → Alphanumeric

ii. User==Admin/Users

iii. Selection==Message → Compose

- Output: System
- End messages/Result:

i. If(login type == “Admin”/”User” & message.clicked = ‘true’ and compose.clicked=true and message.edited=true and send.clicked=true) then send message

ii. If(login type == “Admin”/”User” & message.clicked = ‘true’ and compose.clicked=true and message.edited=false) then display message “Please enter recipient name”

iii. If(login type == “Admin”/”User” & message.clicked = ‘true’ and compose.clicked=true and message.edited=true and attach.clicked=true) then

   a) If (file.browsed=true and send.clicked=true) then attach file and send message

   b) If (file.browsed=false and send.clicked=true) then do not attach file and send message

9. TC09: To test system allows the users to edit their personal profiles

- Description: The system shows all the users to edit their own personal profiles.
- Input:

i. User Name →Alphanumeric, Password → Alphanumeric

ii. User==Admin/Users

iii. Selection==My Account → Profile

- Output: System
- End messages/Result:

i. If(login type == “Admin”/”User” & myAccount.clicked = ‘true’ and Profile.clicked=true and field.edited=true and submit.clicked=true) then display message “Profile saved successfully” and update the database

ii. If(login type == “Admin”/”User” & message.clicked = ‘true’ and inbox.clicked=true and field.edited=false and submit.clicked=true) then display “Mandatory fields should be filled” and donot update the system

10. TC10: To test system allows the Admin to edit user profiles/create new profile

- Description: The system shows all the Admin to edit user profile and create new users.
- Input:

i. User Name →Alphanumeric, Password → Alphanumeric

ii. User==Admin

iii. Selection==UserManagement → Users or UserManagement → Create

- Output: System
- End messages/Result:

i. If(login type == “Admin”/”User” & UserManagement.clicked = ‘true’ and Users.clicked=true and users.exists=true)

a) If (edit.clicked=true and

ii. If(login type == “Admin”/”User” & message.clicked = ‘true’ and inbox.clicked=true and field.edited=false and submit.clicked=true) then display “Mandatory fields should be filled” and do not update the system

4.4. Results

This section lists the results that were produced by running the test cases. The table below lists the test cases that were used while testing the Interface along with the expected result and the actual results for each test case.

Table 4. List of Test Case Results.

Test Case Number	Expected Result	Actual Result
TC01	Pass	Pass
TC02	Pass	Pass
TC03	Pass	Pass
TC04	Pass	Pass
TC05	Pass	Pass
TC06	Pass	Pass
TC07	Pass	Pass
TC08	Pass	Pass
TC09	Pass	Pass
TC10	Pass	Pass

CHAPTER 5. CONCLUSION/COMPARISON/FUTURE WORK

This Chapter would include the Conclusion reached after creating the current version of the Software in regards to the Objectives of the System. The Comparison that was done between the System built and another system that is available online in regards to the features available. It would also list the future work that is intended to be accomplished in the later versions of the Software.

5.1. Conclusion

The following results have been achieved after the completion of the System which relates back to the Objective for the System.

1. **Should allow users to view and save different versions of the same document:** This is achieved when the Admin gives the permission to a certain user to view the files. The selected files then appear to the user under the Document Tab in the user interface.
2. **Save different versions of the same document:** The users can upload the files through a message and send them to the Admin, the admin then can change the version of the file and upload them to the System.
3. **Should allow the Admin to assign the users of tasks to make changes to the current version:** This is achieved when an Admin realizes the need for a Document Update or need for creating a new Document and sends a message to the user to Create/update a Document
4. **The users of the system should be able to view any tasks assigned to them for Document revision and upload the changed documents:** This is achieved when an Admin sends a message to the user with the task of creating/updating a document and the message arrives in the users Inbox.

5. **Approve/Reject the new/changed documents:** This is achieved when a user has created /updated a Document and sent a reply to the message from the Admin with an attachment. The Admin, can then download the attachment and review the document. They would then reply to the user with their comments and let them know if he/she have accepted the document or not.
6. **View detail summary for each document history:** This is achieved when a document is uploaded to the System by the Admin a date and time stamp is attached to the document automatically by the system and is in tabular form when the Admin tries to view the Documents in the System.
7. **Upload Files in compressed format:** This is achieved by the inbuilt function that compresses the files uploaded to the system either by the Document upload or by help of attachment under Message.

5.2. Comparison

This section list the difference between the system developed and another system called ETQ. ETQ is a Document Control Management System which allows users to manage the creation, approval, distribution and archiving of all controlled documents.

After briefly working with the Document Management System and working on a trial version of the ETQ system a comparison of the two systems were done on the basis of the common functionalities offered by both the systems and the results were noted in the table below.

Table 5 shows the difference between the two System and lists the features that DMS and ETQ offers and features that lack and should be implemented in DMS.

Table 5. ETQ vs DMS.

Feature	Description	DMS	ETQ
Import Documents	Feature that allows users to import documents into the system	Yes	Yes
Assign Document	Feature used to assign documents to other individuals	Yes	Yes
Auto Message	Feature that sends auto generated messages to assigned users	No	Yes
Electronic Signatures	Feature that allows users to sign documents with electronic signatures	No	Yes
Convert to PDF	Feature that converts word documents to PDF	No	Yes
Archive Old Version	Feature that archives old versions of documents after expiration date	No	Yes
Integrated Training System	Feature that allows users to get trained	No	Yes
File Compression	Feature that compress uploaded files for memory management	Yes	Unknown
Messaging	Feature that allows users to interact with each other within the system.	Yes	No
Message Threading	Feature that tags all the messages in a thread together	Yes	No

After the comparison of the online available ETQ system with the DMS developed for Axio research certain features have been realized of importance and were proposed to be included in the next release of the system and will be discussed in detail in section 5.3

5.3. Future Work

The following section discusses the future work that would be implemented in the future releases of the Software.

1. **Dashboard:** After much discussion with the QA another need for the system has been realized. A Dashboard has been proposed that would be included in the next release of the system. It would allow the Admin to view all the documents in the System by Category and

Department. It would also show the Admin then permissions that have been given to all the users of the system for each document along with the date and time stamp when permission was given.

2. **Search Function:** To make it easier for the admin to select particular user from a big list, a search function has been proposed to be included in the later release. So that the Admin can search the users listed in the System and would not be required to browse the whole list of user which would further help enhance the performance of the System and the Admin.
3. **Delete/Store Expired Documents: Another requirement has been realized after the** reviewing the current system, which would be to tag an expiration date with each document, so that when that expiration date arrive the document automatically get removed from the Document lists and gets stored in the Expired Section. This would help keeping track of all the Expired Documents and making sure that the Admin does not assign that document to a user and the user should not be able to see the document he/she been assigned previously.
4. **Automatic Message Generation for assigned task:** This has been proposed to save work redundancy and is expected to save extra effort for Admin. An Automatic Message generation by the system has been proposed for future release so that the system sends out an email to all the Users that have been given the permission to view the Document, hence saving time and effort for the Admin to type a message every time a permission to view the document is assigned to a user.
5. **Employee Training System:** The current system is expected to be extended to a complete Employee training system for which the specs are yet to be discussed but would be implement once the above future work has been accommodated to the system.

REFERENCES

Anderson, C. (2012, December 7). Is document control really that important? *Bizmanuals OnPolicy*. Retrieved September 23, 2012, from http://www.onpolicy.com/2010-12/is-document-control-really-that-important.html.

Askwall, S. (1985, April). Computer supported reading vs reading text on paper: A comparison of two reading situations. *International Journal of Man-Machine Studies*, *22,* 425–439.

ETQ. (2012). Document Control Software. Retrieved on July 15, 2012, from http://www.etq.com/documentcontrol/.

Oliver, L. File Hold Numbering and Versioning. Retrieved on July 10, 2012, from http://www.filehold.com/resources/whitepapers/numbering-versioning.

Noyes,M. J., Garland,J.K. (2008, September). Computer- vs. Paper-Based Tasks: Are They Equivalent?. *www.princeton.edu,* 51,1352–1375 . Retrieved on August 25, 2012, from http://www.princeton.edu/~sswang/Noyesa_Garland_computer_vs_paper.pdf.

Bertolino, A. (2003). Software Testing Research Practice. Retrieved on August 14,2012, from http://www.cis.upenn.edu/~lee/05cis700/papers/Ber03.pdf.

Bochmann, G.V., Petrenko, A. (1994). Protocol Testing: Review of Methods and Relevance for Software Testing. Proc, Int. Symp. On Software Testing and Analysis (ISSTA), 109-124.

Braind, L., Labiche, Y. A. (2002)UML-Based Approach to System Testing. Software and Systems Modeling. 1 (1), 10-42

R. V. Binder. (1999, October). Testing Object-Oriented Systems Models, Patterns, and Tools. Addison Wesley, Reading, Massachusetts

Saru, D. (2012, February). Impact of UML Techniques in Test Case Generation. *International Journal of Engineering Science & Advanced Technology*, 2(2), 214 – 217. Retrieved on September 15, 2012, from http://ijesat.org/Volumes/2012_Vol_02_Iss_02/IJESAT_2012_02_02_12.pdf.

Kundu, D., Samata,D. (2009, May-June). A Novel Approach to Generate Test Cases from UML Activity Diagrams, Journal of Object Technology, 8(3). Retrieved on September 25, 2012, from http://www.jot.fm/issues/issue_2009_05/article1.pdf.

MSDN (2001). File Compression: How to: Compress and Extract Files. Retrieved on October 7, 2012, from http://msdn.microsoft.com/en-us/library/ms404280.aspx.

Stackoverflow. (n.d.). .Net + Email Threading. Retrieved on August 4, 2012, from http://stackoverflow.com/questions/10942373/net-email-threading.

AdminSystem Software. (2001). Send Mass Email in C# with Multiple Threads. Retrieved on August 5, 2012, from http://www.emailarchitect.net/webapp/smtpcom/developers/netpool.asp.