**Introduction:**

The Document Management System (DMS) is an application built using PHP, jQuery, and the CodeIgniter framework. It allows users to record, index and route documents to specific persons for immediate action. The DMS is designed to be efficient, user-friendly, and customizable to meet the specific needs of your organization.

Features:

1. Document Recording - The DMS allows users to upload and store documents in the system. Users can add various document types including Word documents, PDFs, images, and other file formats.
2. Indexing - The DMS allows users to add metadata to documents to make it easier to search and retrieve. Users can add tags, keywords, author, and other relevant information to help categorize documents.
3. Routing - The DMS allows users to route documents to specific persons for immediate action. Users can add notes and comments to documents before routing to provide additional context for the recipient.
4. Collaboration - The DMS allows users to collaborate with other team members. Users can share documents with other team members, leave comments and notes, and collaborate on document edits.
5. Email Integration - The DMS allows users to respond to requests via email if they are unable to access the system. When a document is routed to a user, the user can respond via email, and the response is automatically recorded in the system.

System Architecture:

The DMS is built using a client-server architecture. The client is a web browser, and the server is responsible for processing requests, retrieving and storing data, and rendering the user interface. The application is built using the following technologies:

* Frontend: HTML, CSS, JavaScript, jQuery
* Backend: PHP, CodeIgniter framework, MySQL

Deployment:

The DMS can be deployed on a cloud-based server or on-premise server. The application is built using the CodeIgniter framework, making it easy to deploy and manage the application. The application can be deployed on the following platforms:

* AWS Elastic Beanstalk
* Google App Engine
* XAMPP
* WAMP

User Interface:

The DMS user interface is designed to be intuitive and easy to use. The user interface consists of the following components:

* Document List - Displays a list of documents uploaded by the user
* Document Viewer - Allows the user to view and edit documents
* Routing Form - Allows the user to route documents to specific persons for immediate action
* Metadata Editor - Allows the user to add metadata to documents
* Comment Section - Allows the user to leave comments and notes on documents

**USE CASE**

Primary Actor: Administrative Assistants of the Division Chief of Pantawid Pamilyang Pilipino Program

Secondary Actors: All Pantawid Staffs, including those assigned in Field Offices

Goal: To efficiently manage and organize documents related to Pantawid Pamilyang Pilipino Program, and to easily route documents to the appropriate staff for immediate action.

Preconditions:

* The Document Management System (DMS) is installed and running.
* The administrative assistants and staffs have user accounts in the DMS.
* The administrative assistants have access to all documents in the system.

Basic Flow:

1. The administrative assistants log in to the DMS and access the dashboard.
2. To upload a new document, the administrative assistants click on the "Upload" button, select the file to upload, add metadata (if applicable), and click "Upload".
3. To index a document, the administrative assistants select the document from the document list, click "Edit", and add or modify the metadata as needed. They then click "Save" to save changes.
4. To route a document to a specific staff member, the administrative assistants select the document from the document list, click "Route", select the recipient from the list of Pantawid Staffs, add notes (if applicable), and click "Send".
5. To collaborate on a document, the administrative assistants select the document from the document list, click "Share", add the email address of the recipient, and click "Send". The recipient will receive an email with a link to the document.
6. To respond to a request via email, the recipient can reply to the email notification. The response will be automatically recorded in the DMS.
7. The staff member receives a notification of the routed document and can access the document through the DMS. They can also respond to the routing request via the system or email if they are unable to access the system.
8. The administrative assistants can track the status of the document and see if it has been viewed, replied to, or completed by the staff member.

Alternative Flow:

1. If the document is urgent and requires immediate action, the administrative assistants can route the document to the appropriate staff member with a higher priority level, such as "Urgent" or "High".
2. If the document is confidential or sensitive, the administrative assistants can mark the document as "Confidential" and restrict access to only authorized staff members.

Postconditions:

* The document is successfully uploaded, indexed, routed, and responded to if applicable.
* The administrative assistants and staff members can track the status of the document through the DMS.

Conclusion:

The DMS is an efficient and effective application that makes it easy for users to manage their documents. With its document recording, indexing, routing, collaboration, and email integration features, the DMS helps users stay organized, respond to requests quickly, and collaborate with other team members. Its customizable design, CodeIgniter framework, and intuitive user interface make it an excellent choice for organizations of all sizes.

**Installation Guide**

Step 1: System Requirements

Before installing the Document Management System (DMS), ensure that the following system requirements are met:

* PHP 7.0 or later
* MySQL 5.6 or later
* Apache or Nginx web server
* CodeIgniter framework
* jQuery library
* Web browser (Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge)

Step 2: Download the DMS

Download the DMS from the website or GitHub repository.

Step 3: Configure the DMS

1. Unzip the downloaded file and copy it to the root directory of your web server.
2. Create a new MySQL database for the DMS and import the SQL file located in the /application/sql/ directory.
3. Open the /application/config/config.php file and set the base\_url and encryption\_key variables.
4. Open the /application/config/database.php file and set the database variables.

Step 4: Test the DMS

Open a web browser and navigate to the DMS URL. If the installation was successful, the login page should appear.

**User Guide**

Step 1: Logging In

To access the DMS, enter your username and password on the login page.

Step 2: Uploading Documents

To upload a document, click the "Upload" button on the dashboard. Select the file to upload, add metadata (optional), and click "Upload".

Step 3: Indexing Documents

To add metadata to a document, select the document from the document list, click "Edit", and add the metadata in the metadata editor. Click "Save" to save changes.

Step 4: Routing Documents

To route a document to a specific person, select the document from the document list, click "Route", select the recipient, add notes (optional), and click "Send".

Step 5: Collaborating on Documents

To collaborate on a document, select the document from the document list, click "Share", add the email address of the recipient, and click "Send". The recipient will receive an email with a link to the document.

Step 6: Responding to Requests via Email

To respond to a request via email, reply to the email notification. The response will be automatically recorded in the DMS.

Step 7: Logging Out

To log out of the DMS, click the "Log Out" button on the dashboard.

**DATA DICTIONARY**

A data dictionary is a structured description of the data elements or fields in a system or application. It provides detailed information about each data element, including its name, description, data type, length, and constraints. In the context of a Document Management System (DMS), the data dictionary defines the data elements used to store and manage documents.

|  |  |  |
| --- | --- | --- |
| Table: users | | |
|  |  |  |
| Column Name | Data Type | Description |
| id | int | User ID |
| username | varchar | User's username |
| password | varchar | User's password (encrypted) |
| email | varchar | User's email address |
| first\_name | varchar | User's first name |
| last\_name | varchar | User's last name |
| role | varchar | User's role in the organization (admin, user, etc.) |
|  |  |  |
| Table: documents | | |
|  |  |  |
| Column Name | Data Type | Description |
| id | int | Document ID |
| title | varchar | Document title |
| file\_name | varchar | Name of the uploaded file |
| file\_path | varchar | Path to the uploaded file |
| description | text | Document description |
| metadata | text | Document metadata (stored as JSON string) |
| status | varchar | Document status (active, archived, deleted, etc.) |
| created\_by | int | User ID of the user who uploaded the document |
| created\_at | datetime | Timestamp of when the document was uploaded |
| updated\_at | datetime | Timestamp of when the document was last updated |
|  |  |  |
| Table: recipients | | |
|  |  |  |
| Column Name | Data Type | Description |
| id | int | Recipient ID |
| document\_id | int | Document ID |
| user\_id | int | User ID of the recipient |
| notes | text | Notes (comments, instructions, etc.) from the sender |
| status | varchar | Status of the recipient (new, viewed, replied, etc.) |
| created\_at | datetime | Timestamp of when the recipient was added |
| updated\_at | datetime | Timestamp of when the recipient was last updated |
|  |  |  |
| Table: AuditTrails | | |
|  |  |  |
| Column Name | Data Type | Description |
| audit\_id | int(11) | The unique identifier of each audit trail entry. |
| user\_id | int(11) | The user ID of the person who performed the action. |
| username | varchar(50) | The username of the person who performed the action. |
| date\_time | datetime | The date and time when the action was performed. |
| document\_id | int(11) | The ID of the document that was affected by the action. |
| document\_name | varchar(255) | The name of the document that was affected by the action. |
| document\_type | varchar(50) | The type of document that was affected by the action. |
| activity\_type | varchar(50) | The type of activity performed, such as document upload, indexing, routing, or access. |
| activity\_result | varchar(50) | The result of the activity performed, such as "Successful" or "Failed". |
| ip\_address | varchar(50) | The IP address of the user who performed the action. |
| description | text | A brief description of the action performed. |