CLAIMS

1. A system to emulate manual filing system by storing and processing document that operates on Relational Database Management System (RDBMS), comprising;

a Electronic Document (eDoc) (11) having at least one Electronic Document Identifier (eDoc-Identifier), Section, Rowtype and Column;

a Program Module having at least one Electronic Form (eForm) to capture data entry based on set of instructions and data fields that pre-defined in at least one Electronic Dictionary (eDict);

a Virtual Memory for storing the Electronic Document (eDoc) (11); and

a Web-Read Module (4) for retrieving the Electronic Document (eDoc) (11) from the Virtual Memory based on at least one identifier of Electronic Document (eDoc) (11), in which the identifier is extracted based on the captured data entry of Electronic Form (eForm),

wherein the Web-Read Module (4) for retrieving the Electronic Document (eDoc) (11), further comprising;

a Paging Module (7) having the Electronic Document (eDoc) (11) append into at least one Electronic File (eFile) in the RDBMS according to a predefined Page limit;

a Index Module (8) having at least one Index for the Electronic File (eFile) based-on document identifier, date, end sequence number, document status, document offset and document length; and

a Read Module (9) to obtain the Index and at least one Data Relative Page of the Electronic File (eFile) from the Index Module (8) based on the identifier, in which the Electronic Document (eDoc) (11) retrieved from the Paging Module (7) based on the retrieved Index and Data Relative Page to be stored in the Virtual Memory and update the Index Module (8).

2. A system according to claim 1, wherein the identifier of Electronic Document (eDoc) (11) comprising the Electronic Document Identifier (eDoc-Identifier), Section, Rowtype and Column.

3. A system according to claim 1, wherein the identifier of Electronic Document (eDoc) (11) comprising document identifier, date, end sequence number, document status, document offset and document length.

4. A system according to claim 1, further comprising; a Enquiry Module (14) for retrieving a pluralities of Electronic Document (eDoc) (11) information based on at least one Information for the Electronic Document Identifier (eDoc-Identifier), Section, Rowtype and Column of Electronic Document (eDoc) (11), in which the retrieved Electronic Document (eDoc) (11) information having at least one file history display into at least one list form.

5. A system according to claim 4, wherein the list form having at least one predefined information for each document.

6. A system according to claim 4, wherein the Enquiry Module (14), further comprising a Editing Module to load the retrieved Electronic Document (eDoc) (11) for updating the retrieved Electronic Document (eDoc) (11) and store at least one Updated Data to the Virtual Memory.

7. A system according to claim 4, wherein the Enquiry Module (14), further comprising a Viewing Module to load the retrieved Electronic Document (eDoc) (11) for viewing the retrieved Electronic Document (eDoc) (11).

8. A system according to claim 4, wherein the Enquiry Module (14) further includes a Searching Module, wherein the Searching Module retrieves the Electronic Document (eDoc) (11) using the Web-Read Module (4) based-on at least one Index, in which the index is retrieved from the identifier of Electronic Document (eDoc) (11) comprising document identifier, date, end sequence number, document status, document offset and document length.

9. A system according to claim 1, wherein the Web-Read Module (4) further includes a Uploading Module to upload the Electronic Document

(eDoc) (11) based the identifier of Electronic Document (eDoc) (11), in which the Uploading Module establish connection to at least one server having RDBMS and update the RDBMS with the uploaded Electronic Document (eDoc) (11).

10. A method to emulate manual filing system by storing and processing document that operates on Relational Database Management System (RDBMS), comprising steps of;

retrieving predefined Program (eProgram) using a Program Module from database ;

loading the predefined Program it into at least one Electronic Form (eForm) to enable data entry of at least one user input (101);

verifying if eDoc available to load the eDoc data in the eForm based on the user input (102);

extracting the data entry of user input from the eForm (104);

verifying the extracted data entry using a Program Module based on a set of rules that assist user on entering valid data and a verification tools that assist manipulating eForm and eDoc (104);

saving the verified data into a eDoc (105);

verifying if the saved eDoc has at least one Flow Control program defined by Electronic Business Processing Module (eBPM);

submitting the saved eDoc to eBFS to define at least one Flow Control (108);

transferring the defined eDoc to a Transaction Processing Module to regulate the eDoc to be in Sarbanes–Oxley (SOX) compliance; and

storing the eDoc in to a database.

11. A method according to claim 10, wherein verifying if eDoc available to load the eDoc data in the eForm based on the user input (102), further comprising steps of;

retrieving the eDoc using at least one Web-Read Module (4) based on the user input; and

loading the data from the retrieved eDoc into the loaded eForm.

12. A method according to claim 10, wherein verifying if the saved eDoc has at least one Flow Control program defined by Electronic Business Processing Module (eBPM), further comprising steps of;

logging –in using predefined details in at least one Login Module (1101);

retrieving the log in details (1102);

validating the login detail with the login details defined in the database (1103);

retrieving a security matrix control setting having user access privileges (read, write, update and delete) on the Document Identifier of the eDoc by using Web-Read Module (4) (1104); and

defining at least one task to the eDoc using at least one predefined program (1105).

13. A method according to claim 12, wherein defining at least one task using at least one predefined program (1105), further comprising steps of;

Retrieving at least one predefined task by using Web-Read Module (4) (1202);

Selecting the predefined task based on a predefined task list by using Web-Read Module (4);

Verifying if there is a Document Identifier for the eDoc from previous task;

generating a new Document Identifier (1203);

loading the predefined task from Web Processing program system (1204);

retrieve the eDoc from Web Processing program (1205); and

saving the predefined task to the eDoc(1206).

14. A method according to claim 12, wherein saving the predefined task to the eDoc(1206), further comprising steps of;

retrieve the master process flow information by using Web-Read Module (4) (1301).

saving the predefined task by using a Transaction Processing Module (1302).

validate the predefined current task by using Web-Read Module (4) (1303).

create a new operational workflow ID for this process by using Web-Read Module (4), If System identified current task is new task in the workflow (1304);

updating current task in operational workflow metering status by using Updating Module, if System identified current task is not new task in the workflow(1305);

saving the current task to database by using Transaction Processing Module (1306);

verifying if the current task is completed (1307); and

updating the completed tasks in operational workflow to master Ledger by using Transaction Processing Module (1308).

15. A method according to claim 14, wherein updating current task in operational workflow metering status by using Updating Module, if System identified current task is not new task in the workflow (1305), further comprising steps of;

logging –in using predefined details in at least one Login Module (1401);

retrieving the log in details (1402);

validating the login detail with the login details defined in the database (1403);

retrieving a security matrix control setting having user access privileges (read, write, update and delete) on the Document Identifier of the eDoc by using Web-Read Module (4) (1404);

performing at least one metering enquiry to the eDoc using at least one predefined enquiry program (1405);

retrieving at least one parameters to enquiry the targeted result based on at least one user input (1501);

parsing the input by using Retrieval Module then send to enquiry program (1502);

generating at least one task metering result based on the enquiry parameters using Read Module (9) (1503);

validating if the user selects any task management for assign or unassign task (1504);

loading at least one predefined Manage Job program, if user selects task management to assign or unassign task (1505);

retrieving at least one user role based on the task (1601);

retrieving the selected task information by using Web-Read Module (4) (1602);

verifying, if the user would like to update task status (1603);

performing a task status update when user create, execute, update and delete the task (1604);

verifying if user would like to assign or unassign task, if user does not update the task status (1605);

verifying the user role based on predefined user role ranking (1606);

updating at least one task setting and assign the task to at least one user by using Updating Module, then save the task setting to database by using Transaction Processing Module (1607);

verifying if user would like to edit task, if user does not assign or unassign task (1608); and

updating the task edit details such time, duration and User ID by using Updating Module, then save the task to database by using Transaction Processing Module (1609).

16. A method according to claim 14, wherein saving the current task to database by using Transaction Processing Module (1306), further comprising steps of;

logging –in using predefined details in at least one Login Module (1701);

retrieving the log in details (1702);

validating the login detail with the login details defined in the database (1703);

retrieving a security matrix control setting having user access privileges (read, write, update and delete) on the Document Identifier of the eDoc by using Read Module (9) (1704);

performing at least one control setting to the eDoc using at least one predefined control program (1705);

retrieving at least one user role based on the setting (1801);

verifying if the current user role is Administrator (1802);

performing user sign up and security matrix setting;

verifying if user selected User Control (1803);

loading Sign Up program, If user selected User Control (1804);

verifying if user selected Document Identifier Access Privileges (1805);

retrieving selected user security matrix setting by using Web-Read Module (4), if user selected set user Document Identifier Access Privileges (1806);

setting at least one access privileges for Document Identifier Read, Write, View and Delete by using Web-Read Module (4); and

saving the access privileges to database by using Transaction Processing Module (1807).

17. A method according to claim 14, wherein loading Sign Up program, If user selected User Control (1804), further comprising steps of;

retrieving at least one User ID(1901);

verifying the User ID in the eDoc by using Web-Read Module (4) (1902).

verifying if the user would like to perform sign up by determining User ID in submitted account eDoc is empty;

verifying if the user would like to perform update user account process, if submitted account eDoc is not empty (1903);

submitting the User ID in the submitted account eDoc to perform User ID existence checking by using Transaction Processing Module, If User ID in submitted account eDoc is empty (1904);

creating at least one new user account for the User ID, If the User ID does not exist (1905);

generating the User ID for the new user account and saved in submitted eDoc (1906); and

updating the submitted account eDoc in the user account Ledger by using Transaction Processing Module, If User ID in submitted account eDoc is not empty (1907).

18. A method according to claim 10, further includes Communication Processing Module, further comprising steps of;

receiving eDoc to start the process (501);

identifying the process from row identifier that stated in the eDoc using Retrieval Module (502);

validating the process from row identifier (503);

triggering error message, If not valid request (504); and

directing the eDoc to the requested process, If valid process (505).

19. A method according to claim 10, further includes Online Processing Module, comprising steps of;

receive ledger identifier, document identifier, date and time to start the process (601);

delaying the process till the predefined date and time (602);

summing the data of predefined field(s) from eDocs from the predefined document identifier in Transaction eLedger (LT10) and Master eLedger(LM) based on the last processed date and time;

retrieving the eDoc using the Read Module (9) and each field(s) from eDocs are retrieved using the Retrieval Module (603);

compares the summed value (604);

validating if the summed value(s) from LT10 and LM is/are tally (605);

locating the missing eDoc in LM from LT10 and store and update the missing eDoc into LM through Transaction Processing Module; and

updating the process completion date and time to the process log , if summed value(s) tally, (607).

20. A method according to claim 10, further includes Batch Processing Module, comprising steps of;

obtain a batch of eDocs (701);

extract pre-summed balance(s) from the batch header for later comparison (702);

summing the data of predefined field(s) from eDocs. Each field(s) from eDocs are retrieved using the Retrieval Module (703);

comparing the summed value(s) with pre-summed balance(s) (704);

validating if the summed value(s) and pre-summed balance(s) is/are tally (705);

storing the entire batch of eDocs to Transaction Ledger (LT10) and Master Ledger (LM) using Transaction Processing Module, if summed value(s) and pre-summed balance(s) is/are tally (706); and

prompting output error, if summed value(s) and pre-summed balance(s) is/are not tally (707).

21. A method according to claim 10, further includes Storage Procecing Module, comprising steps of;

obtaining at least one Index and at least one Data Relative Page of a Electronic File (eFile) having document identifier, date, end sequence number, document status, document offset and document length from a Index Module (8) based on the identifier;

retrieving the Electronic Document (eDoc) (11) from a Paging Module (7) based on the Index and Data Relative Page in the RDBMS;

storing the Electronic Document (eDoc) (11) in the Virtual Memory; and

updating the Index Module (8).

22. A method according to claim 10, further includes Enquiry Processing Module, comprising steps of;

retrieving a pluralities of Electronic Document (eDoc) (11) information based on at least one Information for the Electronic Document Identifier (eDoc-Identifier), Section, Rowtype and Column of Electronic Document (eDoc) (11) using a Enquiry Module (14); and

displaying the retrieved Electronic Document (eDoc) (11) information having at least one file history into at least one list form.

23. A method according to claim 10, further includes Parellel Processing Module, comprising steps of;

receiving instruction either to create 10, 100 or 1000 databases and ledger identifier to be processed from a program (2201);

creating databases based on the input instruction (2202);

distributing eDocs from the defined ledger to databases created based last 2 or last 3 digit(s) of account number is used to determine which database the eDoc to be distributed using Paging and Index Module (8) (2203);

initiate parallel processing once all eDocs have been distributed into the designated databases (2204); and

updating the processed result to the predefined Control eLedger through the Mapping Module (2205).

24. A method according to claim 10, further includes Mapping Processing Module, comprising steps of;

receiving source eDoc from a program (2301);

parsing source eDoc for later operation using Extraction Module (2302);

identifying and loading destination eDoc from the source eDoc (for later updating) using Read Module (9) (2303);

loading a predetermined mapping instructions (2304);

validating if there are more mapping instruction (2305);

validating if the data of the predetermined source column fulfill the predetermined requirement(s) from the mapping instruction, if there are more mapping instruction (2306);

performing computation on the data of the predetermined source column from the mapping instruction, if the data of the predetermined source column fulfill the predetermined requirement(s) from the mapping instruction, then (2307); and

storing the updated destination eDoc back into database using Paging and Index Modules (2309).

25. A method according to claim 10, further includes Mapping Processing Module, comprising steps of;

receiving eDoc from a program (2401);

store received eDoc into Transaction eFile using Paging and Indexing Module (2402);

update received eDoc to Transaction eLedger using Paging and Indexing Module (2403);

verifying if Transaction eLedger updated successfully (2404);

storing the received eDoc into Master eFile using Paging and Indexing Module, if received eDoc updated successfully (2405);

updating received eDoc to Master eLedger using Mapping Module (2406);

verifying if Master eLedger updated successfully (2407); and

returning the update status, if Master eLedger updated successfully (2408).