GLOBAL JOURNAL OF ENGINEERING, DESIGN & TECHNOLOGY
(Published By: Global Institute for Research & Education)
www.gifre.org

Creation of Loan Automation Application

Adewale O Adebayo, Aibangbee Zandra, Lufadeju Oluwatosin & Maradesa Adepeju
School of Computing and Engineering Sciences, Babcock University, P.M.B.21244 Ikeja, Lagos, Nigeria

Abstract

A business objective of providing efficient loan process using technology as an enabler in order to give the company a competitive advantage within the industry suggests automation. Although a number of lenders have a certain kind of technology infused into their loan processing system, there are noticeable insufficiencies. Finding a guide to the design and creation of a loan automation system is also elusive. There is therefore the need for the creation of a loan automation application which this research addressed. Design and creation research strategy was adopted and data collection was through existing documents and structured interviews. Waterfall software development method was adopted. The outcome provides a loan automation application that saves paper works throughout the life of the loan, and its design is also available as a contributory guide towards creation of similar system.

Keywords: *Loan System, Loan Automation, Loan Application, Design Loan System, Design Loan Application.*

1.0. Introduction

A loan is an amount of money given to an individual or institution on the condition that it will be paid back over a given period with interest, which serves as payment for the use of money. There are various types of loans such as credits, finance, and mortgages. Sources of loans range from banks, credit unions, pawnbrokers, and savings institutions to trust companies. The manual process of obtaining and granting loans is stressful, inconvenient and time wasting for the lender and the borrower as well. A lender having geographically dispersed branch offices requires more effective and efficient means to standardize and control the loan procedure and processes. The business objective of providing efficient loan process using technology as an enabler in order to give the company a competitive advantage within the industry suggests automation (Bank System and Technology, 2013). There is the need to automate the loan processing procedure as much as possible. A loan automation system is one that enables processing of loans from application to final confirmation without the physical presence of both the applicant and the lending institution (Christeson, 2009; WAUSAU, 2010). Although a number of lenders have a certain kind of technology infused into their loan processing system, there are noticeable insufficiencies. Finding a veritable guide to the design and creation of a loan automation system is also elusive.

The goal of this project is, therefore, to design and create loan automation application software that can capture the required loan data elements only once, keep such information secure throughout the loan process, convert the previous hand to hand system of obtaining and granting loans into a computerized less stressful form, keep track of individuals and institutions involved in the process, monitor and track loans given out to allow better workflow and enhance compliance to conditions, ensure security of information, reduce lending lifecycle times, apply appropriate interest to loans, indicate and penalize defaulters, and notify the concerned individual or institution on the progress of their loan**.**

The research strategy is design and creation (Oates, 2009). The software development model was waterfall, which was chosen because the project requirements were clearly stated and understood before design and development started and prescribes a systematic approach to software development which starts with a well-defined, understood client's specification of requirements and moves through to deployment in a linear form (Green & DiCaterino, 1998; Hughes & Cotterell, 2009; Night, Steinbach & Kellen, 2001; Whitten, Bently, & Dittman, 2001; Yin, 2003). The old systems of four banks were examined and the requirements to develop a better system that will eliminate or reduce the noted short comings of the old system were determined. These requirements served as a guide for the design phase of the proposed system, which flowed through initial system investigation, feasibility Study and system analysis to system design. Fact – finding techniques were employed during system analysis. Adequate sample data were collected on the available files, decision points and transactions handled by the present systems. Interviews and on-site observation were complimentary data collection methods used. In addition, secondary data reflecting on loans already granted by the lending or financial institution to borrowers and the repayment patterns were utilized. Personal interviews were conducted with the operators of the Entrepreneurship Development Program (EDP) in order to assess the impact of the scheme on the lending policy of the lending institution. Furthermore, a review of previous articles on this research subject and related issues (Claude, 2009; Croft, 2010; John, 2006; Srinivas, 2005; William, 2006; Witman & Roust, 2008) were also examined to help reveal clearly the areas of concern, and help create the proposed application. The software development environment for the creation of this Loan automation application software include Hypertext Markup Language (HTML), WAMP server (Windows Apache Microsoft PHP), Javascript, CSS (Cascading Style Sheet), PHP (Hypertext Preprocessor), and MySQL. Web browser: The platform on which the application will run includes windows 7 operating system though the software can run on any operating system, and a web browser such as Mozilla Firefox, Opera, and Internet explorer.

Loan automation application software would boost a lending institution's customer service in many ways, while making their operations more efficient and transparent along with adhering to the myriad of federal mandates and regulations. It would decrease wait time, improve customer service level, increase consistency, increase accuracy**,** optimize workflow, achieve compliance and combat fraud. The Loan Automation Application Software is being created to assist rather than replace the human decision maker during the computation of relevant data, and integrating decisions made immediately

on it to be effected and communicated. There has to be some sort of partnership between this application and individuals that intend to use it for it to yield its best anticipated results. The computational speed, accuracy and storage capabilities of the computer would be merged with the creativity and intuition of the human.

2.0. The Loan Automation System

2.1. Requirements

Requirements, functional and non-functional, describe what the proposed system should do to solve identified problems (Stair & Reynolds, 2007).

2.1.1. Functional requirements

The functional requirements details must haves of the proposed system. These are statements of services the system should provide. All functionality requirements are related to the information content. The functional requirements of the Loan automation application software include provision of online avenue for secure loan application, application processing, and acceptance notification, links to accept loan repayment via Online banking and Automatic Teller Machine (ATM), capability to enabled multiple users view different fields of the services at the same time, and avenue for administrative update and insertion of information in the database tables.

2.1.2. Non-functional Specification

Non-functional requirements are good to haves. The non-functional requirements of the Loan automation application software include Portability (access from various software and hardware platforms, and deployment to various hardware platforms), Efficiency (loan contents managed efficiently and content structure well organized and precisely represent the system; interface concise and precise for all users), Security (identities of each registered client kept private; each registered client logs-in with username and password; without appropriate permissions, clients cannot access any data in the system), Reliability (system will not crash on software bugs; platform failures must be recoverable by the program developers), Speed (system function must not be seen as slow), and Fault tolerance issues (back-up and clean-up).

2.2. Design of the proposed system

2.2.1. Features of the system

The propose system is web based software having features as depicted in Figure 1, and as further detailed as follows:

i. Home page, News page, Service Page, About Us page and the Contact Us page.
ii. Fast internet connection speed
iii. Login page where clients will log in and make loan, pay loan, change of password and also see messages from the administrators.
iv. Allows administrators to easily authorize users to manipulate certain pages on their web pages as checking their account balance.
v. Service page which contains all information on the loan types, categories and payment plans, among others
vi. On the administrators page the Admin will approve each client loan status after the request of loan has been sent by each client.

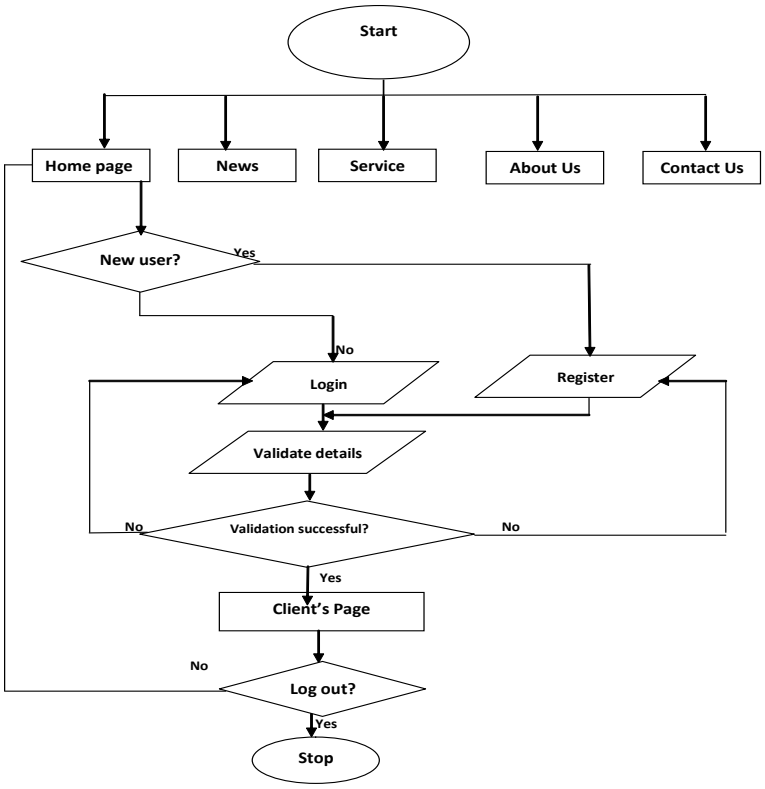

Figure 1: The Flow Chart of the System

2.2.2 Hardware Features

The minimum hardware requirements of the system include computers configured as follows:

i. **Speed:** Personal Computers and workstations should have processor speed of 500MHZ
ii. **Main Memory:** client computers should have a main memory of 128MB and the server computer 512MB
iii. **Secondary Memory:** The server computer should have 120GB of hard disk and the client 10GB.

The system also requires a highly available and quality 60kbps broadband internet connection.

2.2.3 The Application Software

The application software has two modules, Administrator and User. The Administrator module, depicted in Figure 2, contains all the backend functionalities that the administrator requires. The User module, depicted in Figure 3, has functionalities to cater for loan applicants.

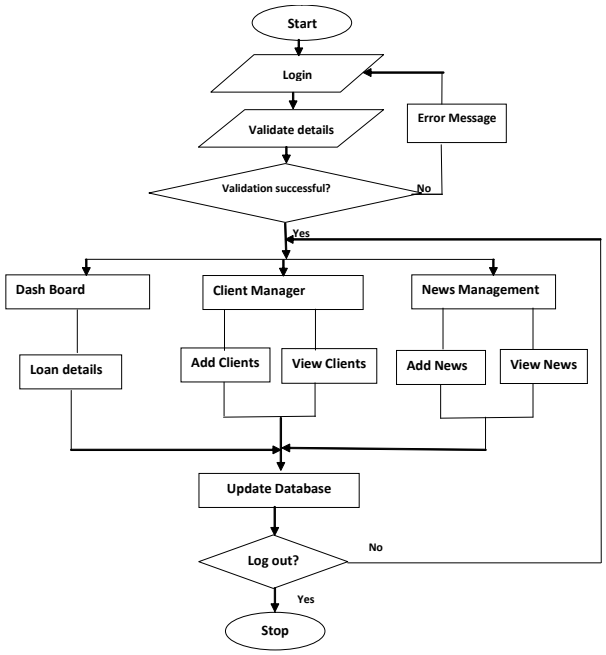

Figure 2: Flowchart of the administrator module

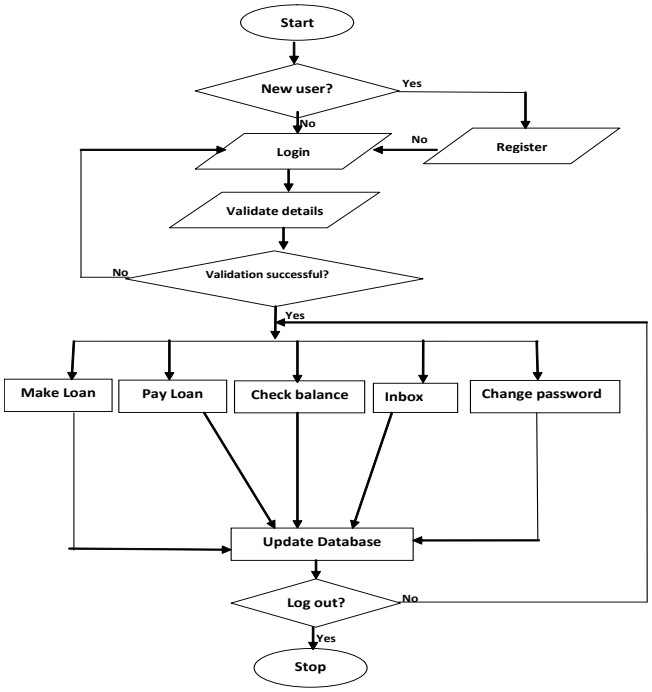

Figure 3: Flowchart of the user module

2.2.4 The Database

The database includes the Admin, Client's, Bank, Contact_Us, Inbox_Mail, Loan_Detail, and New_Update tables. The Admin, Client's, Bank, Inbox_Mail, and the Loan_Detail tables are presented in Tables 1 to 5, respectively. The Entity Relationship diagram of the database is shown in Figure 4.

Table 1: The Admin Table

Field	Type	Collation	Attributes	Null	Default	Extra
Id	Int (9)			No		Auto_increment
username	Varchar(90)	Latin1_swedish_ci		No		
password	Varchar(90)	Latin1_swedish_ci		No		
Name	Varchar (90)	Latin1_swedish_ci		No		

Table 2: The Clients' Table

Field	Type	Collation	Attribute	Null	Default	Extra
Id	Int(9)	Latin1_swedish_ci		No		Auto_ncrement
Full_name	Varchar(150)	Latin1_swedish_ci		No		
Residential_address	Varchar(200)	Latin1_swedish_ci		No		
telephone	Varchar(50)	Latin1_swedish_ci		No		
e_mail	Varchar(50)	Latin1_swedish_ci		No		
date_of_birth	Varchar(50)	Latin1_swedish_ci		No		
nationality	Varchar(50)	Latin1_swedish_ci		No		
State	Varchar(50)	Latin1_swedish_ci		No		
religion	Varchar(50)	Latin1_swedish_ci		No		
gender	Varchar(50)	Latin1_swedish_ci		No		
marital_status	Varchar(50)	Latin1_swedish_ci		No		
Name_of_employer	Varchar(100)	Latin1_swedish_ci		No		
Address_of_employer	Varchar(200)	Latin1_swedish_ci		No		
Occupation	Varchar(50)	Latin1_swedish_ci		No		
Grade_of_	Varchar(50)	Latin1_swedish_ci		No		

Table 3: The Bank Table

Field	Type	Collation	Attribute	Null	Default	Extra
Id	Int(6)			No	None	AUTO_INCREMENT
Username	Varchar(90)	Latin1_swedish_ci		No	None	
Password	Varchar(90)	Latin1_swedish_ci		No	None	
Current_amount	Int(9)			No	None	
Full_name	Varchar(100)	Latin1_swedish_ci		No	None	
Acc_num	Varchar(16)	Latin1_swedish_ci		no	None	

Table 4: The Inbox_mail Table

Field	Type	Collation	Attribute	Null	Default	Extra	
Id	Int(9)			No	None	AUTO_INCREMENT	
Msg_sender	Varchar(90)	Latin1_swedish_ci		No	None		
Msg_reciever	Varchar(90)	Latin1_swedish_ci		No	None		
Msg_subject	Varchar(200)	Latin1_swedish_ci		No	None		
Msg_body	Text	Latin1_swedish_ci		No	None		
Msg_date	date			No	None		

Table 5: Loan_Detail Table

Field	Type	Collation	Attribute	Null	Default	Extra
Id	Int(9)			No	none	AUTO_INCREMENT
Loan_amt	Int(9)			No	none	
Loan_period	Varchar(60)	Latin1_swedish_ci		No		
Loan_type	Varchar(45)	Latin1_swedish_ci		No		
Loan_cat	Varbinary(45)			No	none	
Loan_owner	Varchar(90)	Latin1_swedish_ci		No	none	
Pay_back_amt	Int(9)			No	0	
Loan_date	date			No	None	
Paid_amt	Int(9)			No	None	
Approve_status	Varchar(20)	Latin1_swedish_ci		No	Pending	
Client_status	Varchar(90)	Latin1_swedish_ci		No		
Paid_status	Tinyint(4)			No	none	

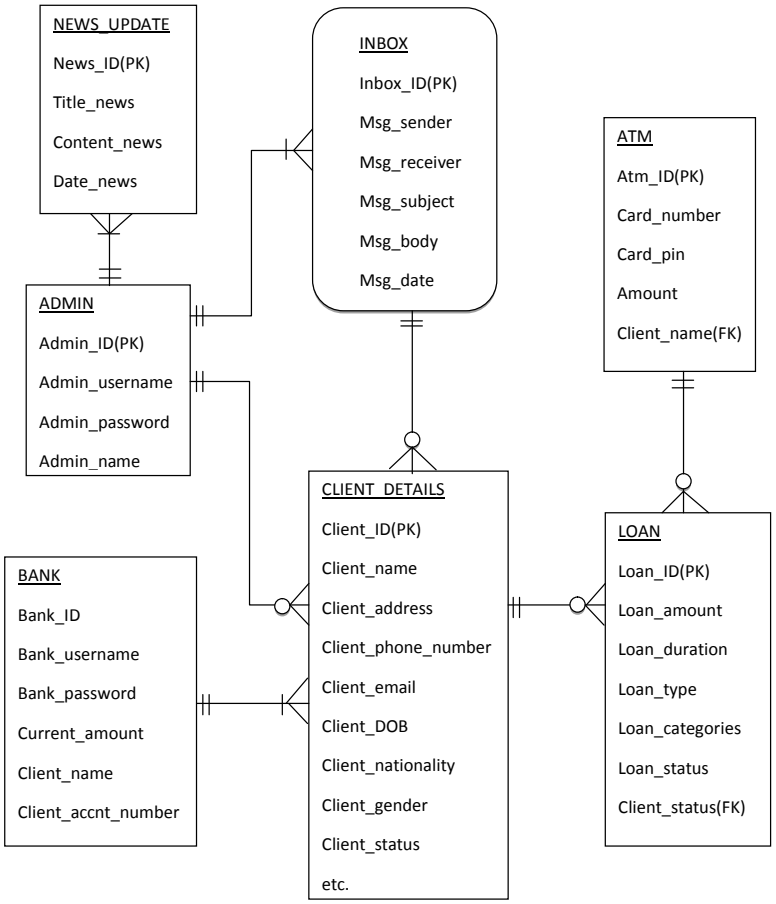

Figure 4: Entity Relation Diagram of the Database

2.2.5 System Implementation

This section covers a detailed documentation of how the system has been developed in a step-by-step manner. It clearly illustrates the implementation process, describes the interfaces and features of the entire system. It also covers the components and unit testing as well as the integration of the system. Software component testing involves integrating one or more system components functions or features and then testing this integrated system. The working process of the Loan automation applications software is also discussed.

Component testing involves Validation and Verification. Validation is the test phase which answers the question 'did we build the right system?' to ensure that the end product suits the need of the customers. This phase employed the use of prototypes, which is a copy of an object, made on a smaller scale than the original. Prototypes have been used to gather requirements from the users and the prototype was subjected to tests till a functional system was evolved. Verification phase answers the question 'did we build the system right?' by subjecting the application to quality control activities throughout its life cycle to ensure that interim deliverables meet their input specification. The system has been verified to meet input specifications from users by verifying login details among others. The software has several components, which were integrated and tested for functionality. They include Home Page, News Page, Service Page, About Us Page, Contact Us Page, Make Loan Page, Pay Loan Page, Check Account Page, Change Password Page, Inbox Page, and Admin Content Management system Page.

After implementing the codes to meet the specification requirements, the entire system was tested. The system components that were tested include database, process, interface, and the server for the system. In testing the database, it was ensured that the database captures the specified fields according to their respective attributes, and that the storage and retrieval functions responded properly. All the tables carrying the bugs reports and projects were easily accessible by the system administrator. In process testing, the system was started and ensured to work acceptably well, all necessary links working and linking to intended locations. Other features such as the comments sending from the clients to the admin and vice versa were tested and ensured that messages sent were meaningful. In interface testing, it was ensured that there was a link for user who does not really know how to navigate around the system, and all links were ensured to lead appropriate page providing enriched user experience.

Almost all of the functions of the application run on the WAMP server, which is responsible for communicating with web browser. MySQL relational database server stores the information the application requires, and PHP was used as a middleware. Figure 5 represents the architecture of the system.

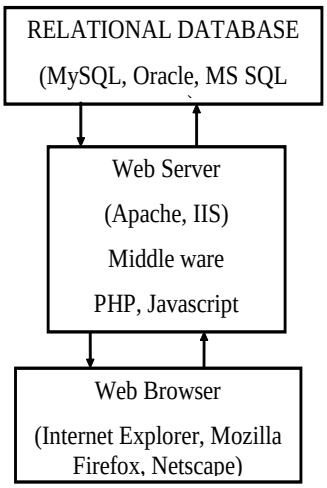

Figure 5: Architecture of Web application

The application starts running from the home page shown in Figure 6.

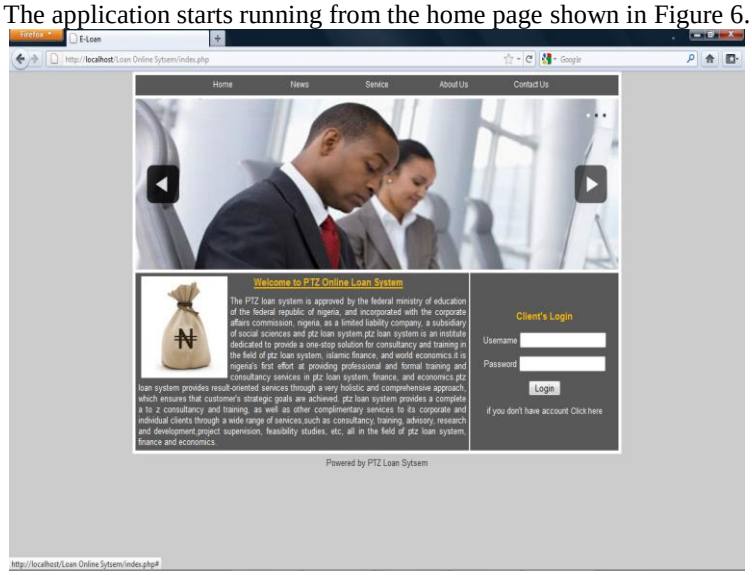

Figure 6: Homepage of Loan automation application software

After a client logs in, there are a number of options to choose from. The interface of the make loan page is shown in Figure 7. The user is prompted for some details after which the user is validated.

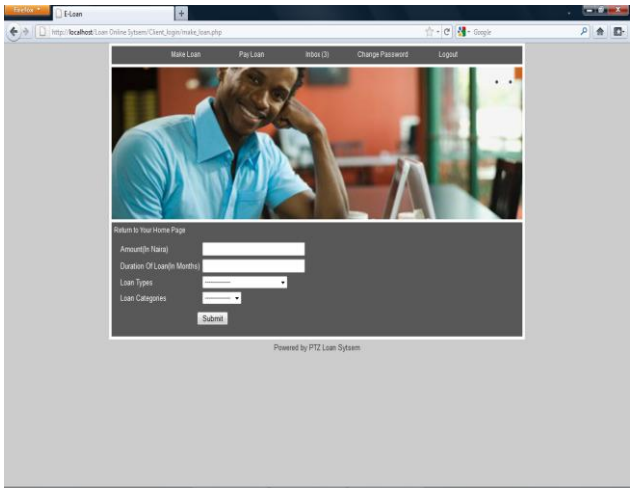

Figure 7: Make Loan page.

Figure 8 shows the interface from where messages in inboxes are viewed. This option is provided on logging in. Upon approval of loan, a notification is created and sent to the client's inbox.

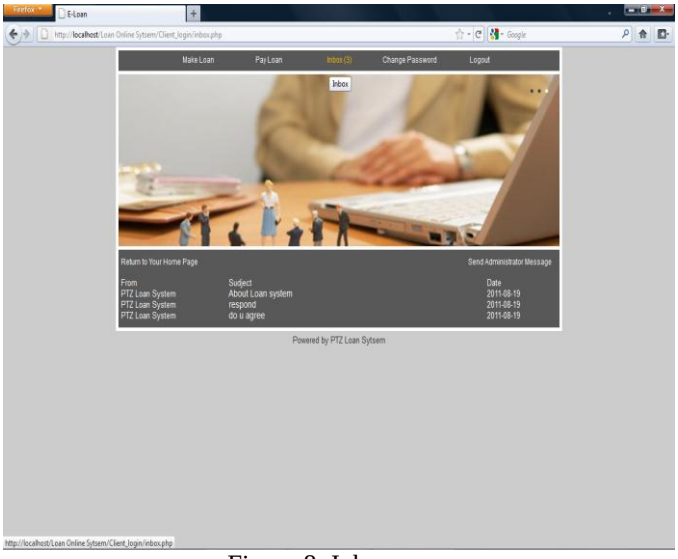

Figure 8: Inbox page

Figure 9 shows the Contact Us page, a form for users to type in any enquires or complaints they have. It holds textboxes for name, email and comment. A validation is also done to ensure that the email entered is in the right format and that none of the fields are left empty before the user is allowed to submit.

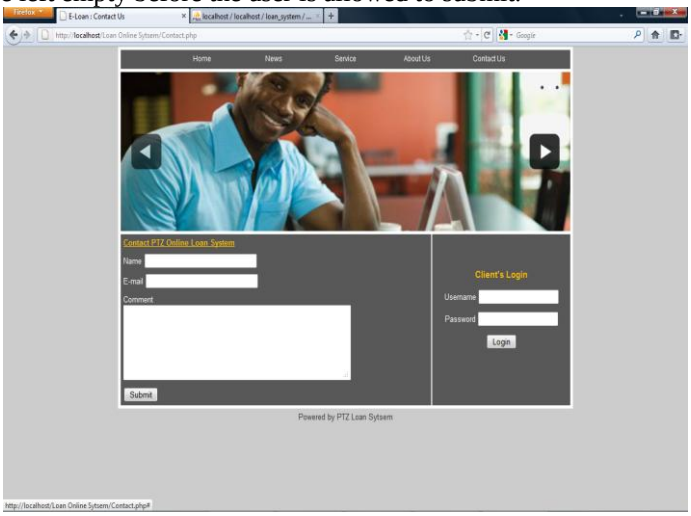

Figure 9: Contact page of Loan automation application software

Figure 10 shows the Admin login page. Administrator is the person in charge of the whole application. He runs it from a back-end. No one can modify or respond to users comments unless logged in with the administrator's details, which should be kept with utmost care.

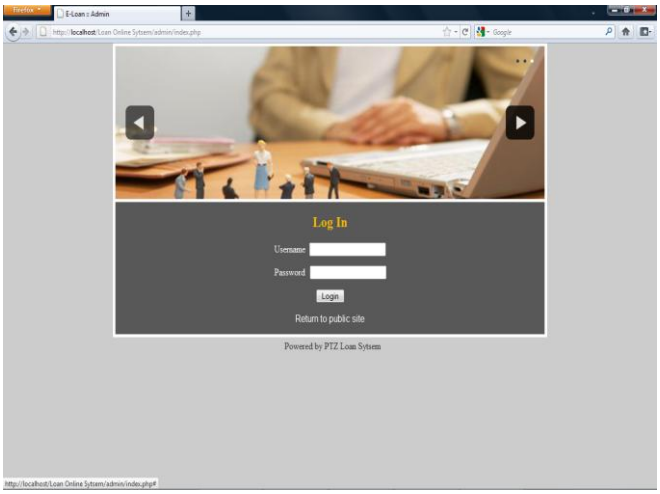

Figure 10: Admin login page

After the administrator is successfully logged on (Figure 11), he can now manage the entire application. He is able to manage clients, add news feeds, and check messages sent by clients and also log out. This interface is friendly and provides adequate functions.

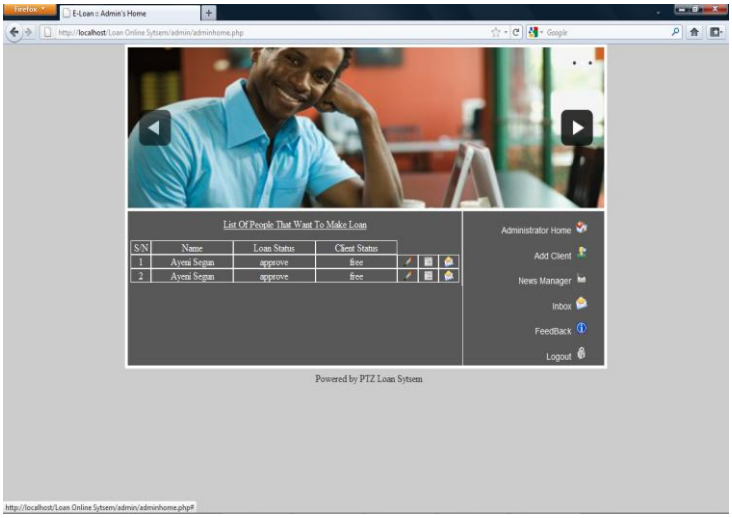

Figure 11: Admin Content Management System

3.0 Related Works

JEMS Loan Automation System was created for J.P. Morgan Mortgage Capital (JPMMC) to provide straight-through automated processing of commercial mortgages throughout the entire lifecycle of the loan. At its core, JEMS is an intranet solution that uses the latest technologies in service-oriented architecture (SOA) and object-oriented development to completely automate the loan process from front to back. The major strengths of this system include: Increased speed and richness built into the user interface; Use of BizTalk that allows JPMMC to more quickly adapt to changing business processing needs; and Web Services and .NET remotely used in the service-oriented architecture allowing for greater flexibility in scaling out individual pieces of the application. Windows Server 2003 was deployed for the release of JEMS. This proved helpful in providing enhanced security and speed. Many of the services were implemented using XML Web Services hosted in IIS 6.0. BizTalk 2002 was used to handle data integration to a legacy web application, as well as for automating several back end reporting and documentation processes. The database was implemented using Microsoft SQL Server 2000. Microsoft MapPoint 2002 was used to provide maps and directions for the real estate properties in the system. Microsoft Office XP and 2003 were used to provide spreadsheet functionality in several areas of the application. The spreadsheets include code that enables them to send data to a remote web service whenever a network connection is available. This provides the means for users to work offline and merge data at a later time (Morgan, 2006). The design of this application is not available for examination towards shared knowledge and possible stepwise systematic improvements.

ALERT is an examination tool developed to facilitate loan reviews of regulated financial institutions by examiners conducting safety and soundness examination. The program enables examiners to access electronic loan data provided by banks via e-mail, diskette, or compact disk. In addition to printing the loan data from electronic media, the program provides examiners with an analytical tool for assessing risk in bank loan portfolios. In particular, the application is used to evaluate compliance with loan concentration limits on bank loans to individual borrowers. ALERT is used by the Board, the Federal Deposit Insurance Corporation (FDIC), and most state bank regulatory agencies in US (Federal Reserve, 2007). The design of this application is not available for examination towards shared knowledge and possible stepwise systematic improvements.

WAUSAU Financial System Loan Automation is a paperless solution that integrates to existing platforms, speeding approvals while ensuring security and compliance. Originating and servicing loans requires a sophisticated alignment of content from various internal sources, including an organization's host platform, loan origination, and tickler and tracking systems. And external sources include content from credit reporting and appraisal agencies, title companies and borrower documents. Centralizing and simplifying all of those internal and external processes, WAUSAU's Optima3® IMS solution is particularly suited to becoming a core, underlying component of loan operation—with complete integration to existing platforms (WAUSAU, 2010). The design of this application is not available for examination towards shared knowledge and possible stepwise systematic improvements.

The InterLibrary Loan Internet Accessible Database (ILLiad) Automation System is a model, implemented in software, of the interlibrary borrowing process at Virginia Polytechnic Institute and State University (Virginia Tech). It is the software that examines the current state of each ILL borrowing request, and depending on the state of the request, it may perform an action, such as sending an overdue notice to the customer if the book is overdue. The ILLiad database (ILLData) was easily designed from the existing ILL request cards. The contents that were translated to database tables include Transactions, Users, Lender Addresses, Tracking and Invoices. ILLiad uses additional tables, which include Increment (where the Transaction Number is created), Inventory (where data is stored from a hand-held scanner during the inventory process), and OCLC Updates (where ILL numbers are stored awaiting an update, along with their status of received or returned), to manage its internal operations. (Virginia Tech, 2012). The major advantages of the ILLiad include the fact that it provides feedback services that enable customers confirm their loan status conveniently and generate statistical reports about the performance of the ILL service. On the staff side, it saves time and eliminates typing errors. Staff can also easily respond to inquiries since most information are available online. Major weaknesses include the issue of security of customer information when the ILLiad is accessed on a public workstation. ILLiad was developed on a Web-based user interface to accommodate customers on all platforms. Microsoft SQL Server was used as the relational database engine due to its integration into the Windows NT Server environment used on the campus network.

Borland's Delphi was used as the development tool in order to reduce programming. The statistical reports displayed on the Web are generated through the Active Server Pages feature of Microsoft Internet Information Server version 3.0, which is the Web server included with NT Server. Concerning this project, the ILLiad model may not be fully adaptable for use in financial loan processes as it focuses mainly on book lending between libraries. The design of this application is not available for examination towards shared knowledge and possible stepwise systematic improvements.

Zia Consulting provides document automation solutions for companies across the mortgage industry, delivering significant cost savings while at the same time offering accelerated revenue opportunities and improving visibility into the process for enhanced compliance and business intelligence. Zia has experience providing document automation to a wide range of enterprises across the entire mortgage industry, which was used to create ZiaOne Loan Automation System (LAS) featuring: Automated Document Processing from any Source, in any form, of any type; Borrowers Supporting Documents to Closing Documents; Integration with LOS (Not Replacing it); Zia OneView for End-to-End Analytics (Zia Consulting, 2013). The design of this application is not available for examination towards shared knowledge and possible stepwise systematic improvements.

4.0 Conclusion

The outcome provides a loan automation application that saves paper works throughout the life of the loan, and makes available its design, a guide or foundation towards creation of similar or improved systems. The Loan Automation Application Software would help boost a lending institution's customer service in many ways, and make the lending operations more efficient and transparent. The computational speed, accuracy and storage capabilities of the computer would merge with the creativity and intuition of the human to make it all a success.

References

Bank System and Technology (2013). Automating Loan. Retrieved from www.banktech.com/management.../automating...loan...for-f/240163441.

Christeson E., (2009). "Banks Now Turning Towards Loan Automation Technology – Banking Frontiers Magazine" Retrieved from http://www. News_events Loan Automation_2.pdf

Claude E., (2009). Financial Services Solutions- Extended Loan Automation. Retrieved from http://www.4point.com/extended-loan-automation.

Croft S., (2010). Loan Processing, Lending, Workflow, Analytics, Decisioning Engine. Retrieved from Cypress Software Systems.info.criflendingsolutions.com/Loan-Automation-Software.

Federal Reserve (2007). Privacy Impact Assessment of Automated Loan Examination Review Tool. Retrieved from www.federalreserve.gov/files/pia_alert.pdf

Green D. & DiCaterino A., (1998). "A Survey of System Development Process Models". New York: New York University.

Hughes, B. and Cotterell, M. (2009). *Software Project Management*. London: McGraw-Hill Education.

John B., (2006). "Mortgage Loan Document Management"- Retrieved from http://www.xerox.com/globalservices.doc

Morgan J.P, (2006). SOA Enhancement to Loan Automation System for JPMorgan. Retrieved from http://www.rdacorp.com/wp-content/uploads/00130-SOAEnhancementToLoanAutomationSystem.pdf

Night L., Steinbach T., & Kellen V. (2001). "System Development Methodologies for web enabled E-business: A customizable paradigm". Retrieved from https://www.cms.gov/Research.../SelectingDevelopmentApproach.pdf

Oates, B. J. (2009). Researching Information Systems and Computing. London: SAGE.

Stair, R. M., & Reynolds, G. W. (2007). Fundamentals of Information Systems. Boston: Course Technology.

Virginia Tech (2012). ILLiad Automation System. Retrieved from http://www.ill.vt.edu/AboutILLiad.htm

WAUSAU (2010). WAUSAU Financial System. Retrieved from www.wausaufs.com/products/loan-automation/55/50

Whitten, J.L., Bently, L.D., and Dittman, K.D. (2001). *System Analysis and Design Methods.* New York: Irwin/McGraw-Hill.

William, S. (2006). Commercial Loan Automation/GonzoBanker.pdf. Retrieved from http://www.gonzobanker.com/article.aspx?Article=283

Witman, P.D., & Roust, T. (2008) "Balances and Accounts of Online Banking Users: A study of Two US Financial Institutions". International Journal of Electronic Finance, 2(2), 197-210.

Yin, R.K (2003) "Case Study Research: Design and Methods (3rd Ed.) Thousand Oaks, CA: Sage.

Zia Consulting (2013). ZiaOne LAS - Loan Automation System for Mortgage. Retrieved from http://vimeo.com/78174127