

Development of a Management Application for Micro-Finance Banks (BLIMB)

Tingir T. James¹, Awuhe T. Samuel², Yilwatda, M. Manji³, Tsenongo N. Stephen⁴

^{1, 2, 3}*Department of Electrical and Electronics Engineering, Federal University of Agriculture Makurdi, P.M.B. 2373, Makurdi, Benue State, Nigeria*

⁴*Bringfireh Technologies, Gboko, Benue State, Nigeria*

Abstract— The MicroFinance Management Application is a software developed to cater for the need of a Micro Finance Bank. The Application has the Administrator and Customer interfaces. The Administrator interface sets up loans, gives loan approvals, checks total saving and withdrawals for the organization, handles active and paid loans, handles SMS and other Company settings. The Customer interface creates customer accounts, obtains their details, handles customer savings, withdrawals, loan applications, and bookings, The application is run online and is used by Javascript based browsers. The Application was written in ASP.NET MVC and the GUI application was written using C-sharp programming.

Keywords—: Micro-finance, ASP.NET MVC, C-Sharp, Visual Studio. GUI

I. INTRODUCTION

The world of banking has metamorphosed from the manual mode of operation to a digital one. Modern banking no longer requires a client to be at a specific location to be able to do business or handle any transaction. The use of Finance Management applications has solved problems that made banking difficult in the past. Even though the banking sector has been infused with improvements by software, smaller banking institutions have largely been ignored when developers design Finance management applications. This research is targeted at smaller financial organizations and micro finance banks. Several works have been done on banking software in the past.

Uddin (2015) developed a banking account system which covered the basic functionality of a Bank Account Management System. The research was aimed at solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. The application was developed using PHP, HTML language and MYSQL for database connection. The system was designed as an interactive and content management system. The content management system dealt with data entry, validation confirms and updating while the interactive system deals with system interaction with the administration and users. The system was designed in accordance with specifications satisfied the requirements [1]

Mitropoulos *et al* (2013) proposed a state-of-the-art approach to the development of an effective and secure web banking system. It aimed at addressing the latest trends and advancements on web system development and operational requirements of e-banking environments. The application was designed as a prototype web banking system with its analysis and design based on the waterfall model through the use of workflow and UML diagrams, as well as its implementation approach based on web programming technologies such as active server pages and web services. After testing using a multi-perspective system evaluation, which is based on the balanced scorecard method, the system proved to have high applicability and significant value [2]. Gomathy and Rajalakshmi (2014) designed a software design pattern for bank service oriented architecture. The Self-Service and comprehensive venture design patterns and the function incorporation techniques, could be used to start executing solutions of banking industry using the service oriented architecture approach. The said approach was flexible, reliable and competent [3]

II. SYSTEM ANALYSIS AND METHODOLOGY

The system was designed using the incremental model. The system is designed step by step, debugged and other features are added as the development increases.

2.1 Software requirements.

1. The Software is compatible with Windows (2000, XP, Vista, 7, 8 and 10), Mac OS and Linux Operating Systems.
2. Java Script enabled Browsers.

2.2 Hardware requirements

1. Volatile Memory (Random Access Memory, RAM) of at least 512 MB.
2. Processor Speed of at least 500Mhz.
3. Storage Memory of at least 2 GB

2.3 System implementation and structure

Implementation is the carrying out, execution, or practice of a plan, a method, or any design, idea, model, specification, standard or policy for doing something. In development of software, implementation refers to actualization an objective or algorithm through coding and deployment of software.

This application is developed using several languages

1. ASP.NET MVC is a web application framework developed by Microsoft, which implements the model-view-controller (MVC) pattern. It is open-source software, apart from the ASP.NET Web Forms component which is proprietary. [4]
2. C# (pronounced *C sharp*) is a general-purpose, multi-paradigm programming language encompassing strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. The Graphical User Interface was built using this language. [5]
3. Microsoft SQL Server is a relational database management system developed by Microsoft. As

a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network. The database was built using this language. [6]

III. MODE OF OPERATION.

The system has ASP.NET (MVC) as its base which works with the other tools to bring to life. it uses the model view controller (MVC) framework of ASP.NET to incorporate all the necessary tools needed to implement this application. The MS-SQL tool creates the database which the system uses to store information for different functions and purposes. The Graphical User Interface (GUI), (written in C #) acts as the bridge between the user and the software system. The GUI of the system links all phases forming a cohesive unit. The Use Case Diagram of the System is shown in Figure 1.

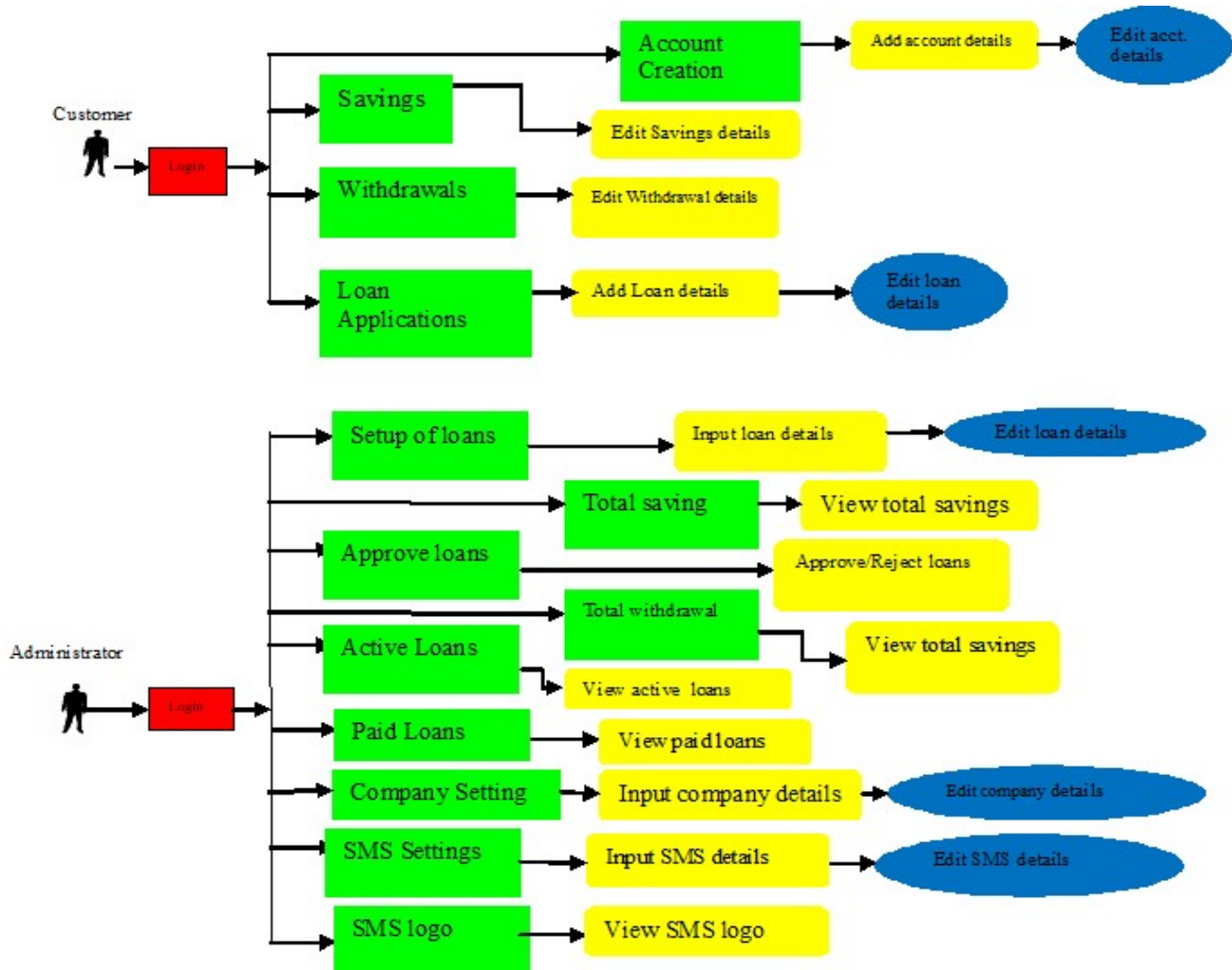


Figure 1. Use case Diagram

The system has the Administrator login and the Customer login. The Administrator handles the company's administrative functions such as setup of loans, total savings, total withdrawals, active loans, approve/reject loans, paid loans, company settings, SMS (Short Messaging Service) setting etc while the Customer section handles functions that concern the customers of the business. Functions like customer account creation, customer saving, customer withdrawals, customer loan applications are handled by these sections. A screenshot of the login page is shown in Figure 2

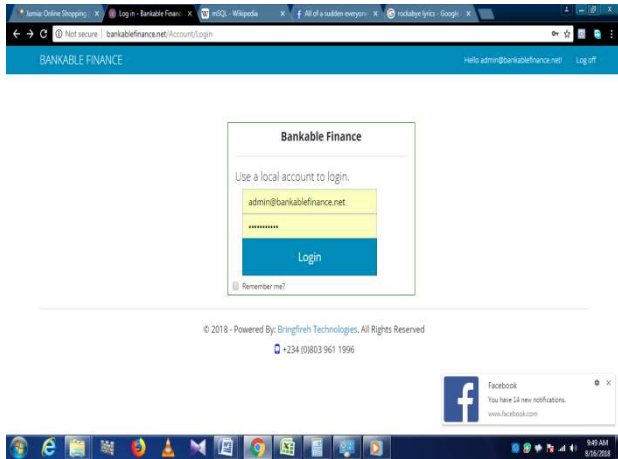


Figure 2 Login Page of the MicroFinance Bank application

3.1 The Administrator login.

The Administrator login has the following sections

- Setup of loans: requires loan details such as 'Loan name', 'Description', 'Category', 'Tenure type', 'Tenure', 'Interest rate', 'Maximum amount'. A screenshot of this interface is shown in Figure 3.
- Total savings; Displays a list of all savings amount made by the customers with the dates reference numbers.
- Total withdrawals: Displays a list of all withdrawals amounts made by the customers along with the dates of withdrawals and reference numbers.
- Active loans: Displays all customers with active loans and amount owed
- Approve/reject loans;: Displays a list of loan applications, with options for 'Approve' or 'Reject' loans
- Paid loans: Displays all customers who have completed loan payment and amount paid
- Company settings: requires company details such as Branch Name, Branch Address, Number of Branch Staff.
- SMS (Short Messaging Service) setting: requires details such as Phone Number, Withdrawals Messages, Savings Messages.

3.2 Customer login.

The Customer login has the following sections

- Customer account creation: requires details such as Acct. Number, Surname, Other Names, Sex, Email, Phone Number, Date of Birth, Branch Name, Address, Enable Minimum Balance, Minimum Balance. Screenshot is shown in Figure 4.
- Customer saving: requires details such as Reference Number, Date, Amount as shown in Figure 5.
- Customer withdrawals: requires details such as Reference Number, Date, Amount as shown in Figure 6.
- Customer loan applications: require details such as loan amount, loan duration, reference number yes/no drop list for active loans
The system is automated to send the customer an SMS if withdrawals or saving are made as well as any payments

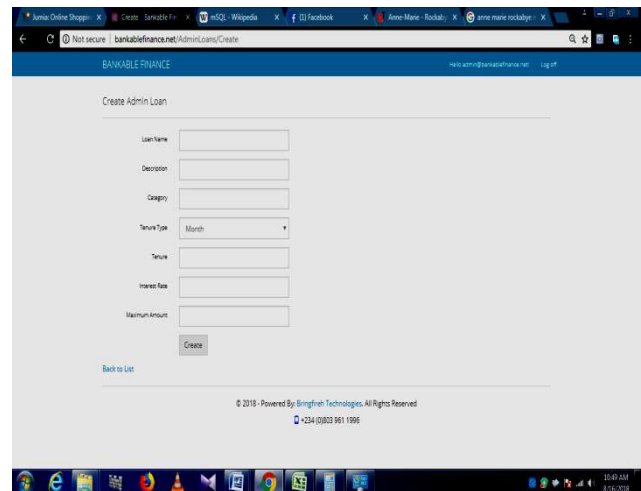


Figure 3. Screenshot of Admin Loan Setup

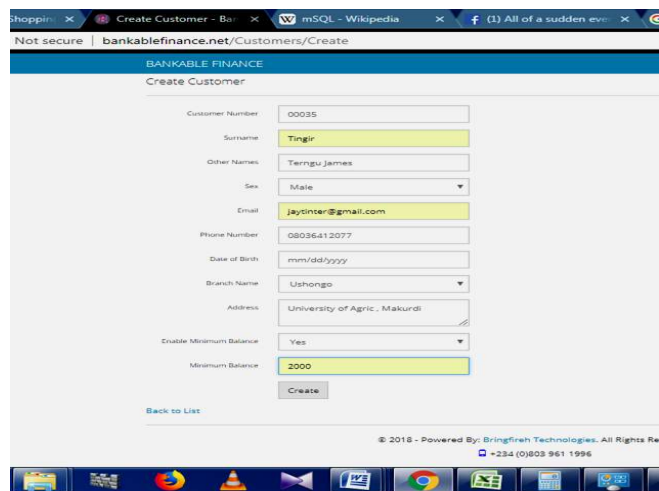


Figure 4. Screenshot of Customer Account Setup

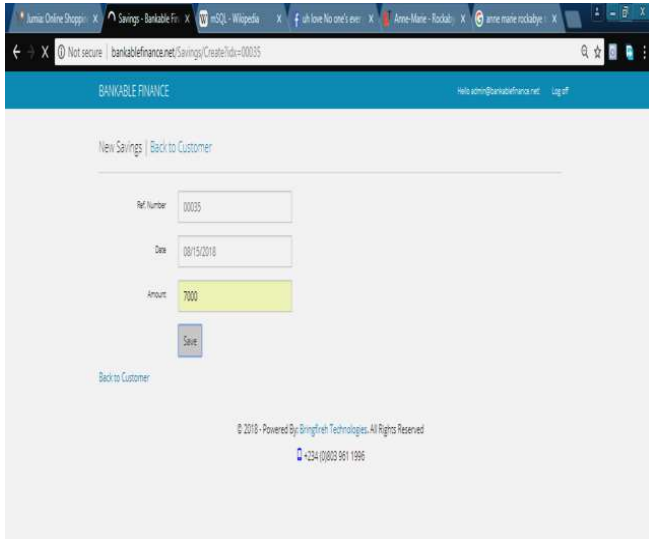


Figure 5. Screenshot of Customer Savings

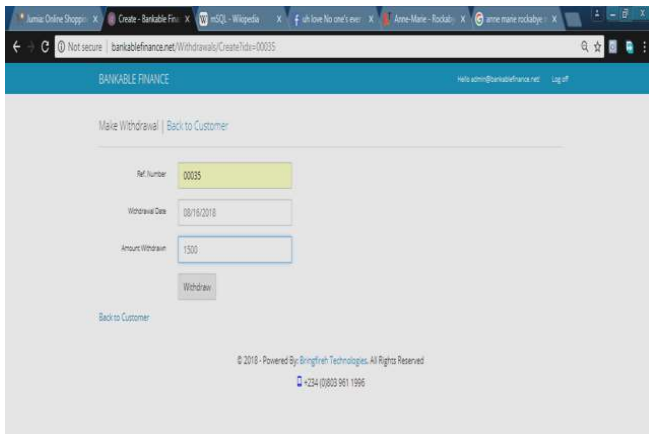


Figure 6. Screenshot of Customer Withdrawals.

IV. SOFTWARE TESTING AND RESULTS

Testing is done at two stages namely:-

1. Testing in section, where each of the sections are tested step by step before integration. This helps to fix the bugs to avoid complexities when integrating.
2. System Testing, where the system is tested after all sections have been integrated to see if it conforms with requirements and to see if the objectives desired are met.

4.1 Graphical user interface testing

The Graphical User Interface (GUI) was tested to ensure that the graphic component of the system links all functions in the system contained by the different sections. In this system, the administrator and customer sections are tested to ensure that all functions in their respective sections work according to their functions. The sequences in operation in each of the sections were checked to see if they were in the right order.

4.2 Testing for ease of use.

The system was tested for usability to see its user friendliness. Three users were provided with Login details and were asked to perform tasks like to Create customer account (for Customer section) and setup admin loan (For Administrator Section) and to Add Customer Details and Supplier Details. All users were able to access and execute the tasks with little difficulty.

4.3 Database testing

Database used in this system was designed to contain a data structure that conform to Database Management System (DBMS) properties which include Atomicity, Consistency, Isolation, and Durability (ACID). The testing was conducted on the Customer Account with its data dictionary at hand so as to ensure that the edit/delete operations were as consistent as possible to avoid the possibility of occurrence of database crash as volume of data generated increases. The screenshot of the Customer Account arrangement is shown in Figure 7.

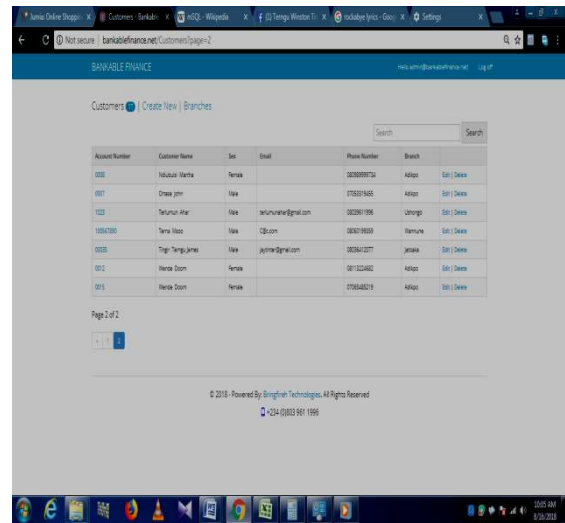


Figure 7: Customer account information shown in columns in the database

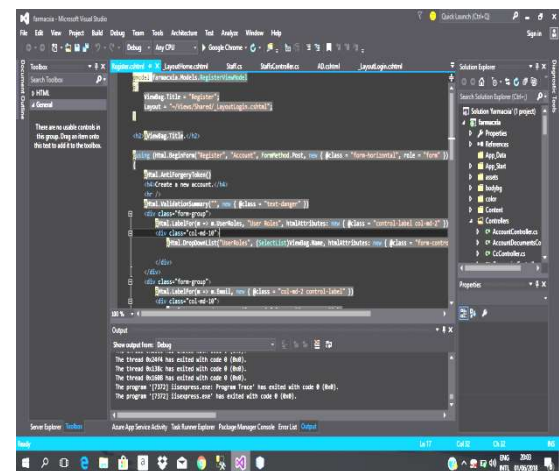


Figure 8: Screenshot of codes written in Visual Studio 2015 environment.

V. CONCLUSION

In the development of this Application for Management Microfinance Banks , the design was set up such that only authorized personnel can have access to the individual interfaces hence the inclusion of a login. The administrator's section allows only the administrator to the company's administrative functions such as setup of loans, approve/reject loans, company settings, SMS (Short Messaging Service) setting, It also gives the administrator access to view the detail of the total savings, total withdrawals, active loans and paid loans. Customer section handled functions that concern the customers of the business. Functions like customer account creation, customer saving, customer withdrawals, customer loan applications are handled. The system was tested and implemented successfully. All the objectives aimed at in the development of this software were achieved.

REFERENCES

- [1]. Uddin, M. J. and Nuruzzaman, M. (2015). Bank Account Management System. Thesis submitted to the Department of Computer Science & Engineering, City University Dhaka, Bangladesh.
- [2]. Gomathy, C. K. and Rajalakshmi S. (2014) A Software Design Pattern For Bank Service oriented Architecture. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* 3(4): pp 1302-1306
- [3]. Mitropoulos S., Othonos, C., Douligeri C., (2013) An effective and secure web banking system: Development and evaluation *International Journal of Business Information Systems* 12(3): pp 335-361 .
- [4]. Wikipedia: The Free Encyclopedia, 2018a, ASP.NET MVC (accessed February, 2019) [retrieved from https://en.wikipedia.org/wiki/ASP.NET_MVC]
- [5]. Wikipedia: The Free Encyclopedia, 2018b, C Sharp (programming language) (accessed February, 2019) [retrieved from [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))]
- [6]. Wikipedia: The Free Encyclopedia , 2018c Microsoft SQL Server (accessed March, 2019) [retrieved from https://en.wikipedia.org/wiki/Microsoft_SQL_Server]
- [7]. Microsoft Visual Studio (c) Microsoft Corporation . Version 2015