Development and Deployment of VoiceXML-Based Banking Applications

Azeta A. A., Ikhu-Omoregbe N. A., Ayo C. K., and Atayero A.A.
College of Science and Technology, Covenant University, Ota, Nigeria
{azeta_ambrose@yahoo.com, omoregbe@ieee.org, ckayome@yahoo.com, atayero@ieee.org}

ABSTRACT

In recent times, the financial sector has become one of the most vibrant sectors of the Nigerian economy with about twenty five banks after the bank consolidation / merger exercise. This sector presents huge business investments in the area of Information and Communication Technology (ICT). It is also plausible to say that the sector today is the largest body of ICT services and products users.

It is no gainsaying the fact that so many Nigerians now carry mobile phones across the different parts of the country.

However, applications that provide voice access to real-time banking transactions from anywhere, anytime via telephone are still at their very low stage of adoption across the Nigerian banking and financial sector.

A versatile speech-enabled mobile banking application has been developed using VXML, PHP, Apache and MySQL. The developed application provides real-time access to banking services, thus improving corporate bottom-line and Quality of Service (QoS) for customer satisfaction.

Keywords: ATM, E-Banking, GSM-based, UML, Web, Phone, VoiceXML, Voice recognition and Voice authentication, QoS

1.0 INTRODUCTION

The need to improve customers' services and provide efficient banking services has resulted in a dramatic increase in the use of Information and Communications Technology (ICT) based solutions in the financial institutions. The banks in Nigeria have to a good extent adopted recent advances in information technology for e-banking [1]. E-banking is a means whereby banking transactions are carried out using automated processes and electronic devices such as personal computers, telephones, fax machines, Internet, card payment, Automated Teller Machine (ATM) and other electronic channels. Today, the development of banking is no longer focused on the growth of basic services (money transactions), but rather on distribution channels via which self-service platforms are extended to the clients. Clients are looking for easy and

inexpensive access to information on their banking transactions. Electronic access via the Internet and ATM are only few options, which is attracting an increasing number of users, saving them the time spent waiting at bank counters. Other types of e-banking include: PC banking, telephone banking, mobile banking, etc. Some of the services provided through e-banking are: Utility bill payments, setting-up standing orders, funds transfer from one account to another, confirmation of account balances and electronic statement of account and account history [2]. Some of these solutions are available via, ATMs, Mobile (GSM-based) banking, Smart Card Technology, etc. The advent and proliferation of telephones (particularly mobile phones) in Nigeria has further challenged the Nigerian financial institutions to provide banking services while on the move. The telephone cannot only be used for presenting visual application and data, but can also support speech-enabled transactions. With speech, bank customers are capable of sending debit requests to their banks through the use of mobile or land phones. Users of public utilities such as Power Holding Company, Nigerian Telecommunication Plc, Water Board, etc, can give a standing order for funds to be transferred from their account to the account of any of the utility providers.

This paper discusses the addition of voice functionalities enabled by the VoiceXML technology to the design of mobile banking applications for enhanced QoS and wider acceptance in the application domain. The application developed allows a caller to access the system over the phone where he/she will be prompted for user name, password and account information for authentication. Once authenticated, the caller will be presented with a selection of banking services such as balance inquiry, funds transfer, etc.

The remaining part of the paper is organized as follows: Section 2 presents VoiceXML development infrastructure, in section 3 we present a UML model and the system design architecture. In section 4, the development and deployment of the application is discussed and section 5 concludes the paper in section 5.

2.0 VOICEXML DEVELOPMENT INFRASTRUCTURE

VoiceXML (also known as VXML) technology allows a user to interact with the Internet through voice-recognition technology by using a voice browser and/or the telephone [3]. Using VoiceXML, the user interacts with the

voice browser by listening to audio output that is either pre-recorded or computer-generated by submitting audio input through the user's natural speaking voice or through a keypad, such as a telephone. A VoiceXML platform is a foundation for developing and operating voice automation applications [4]. During the human computer interaction, it executes the commands and logic specified by applications written in VoiceXML. It also provides the speech processing capabilities (speech recognition, speech synthesis, voice authentication, etc). VoiceXML platform architecture is based on HyperText Transfer Protocol (HTTP) protocol, and uses both phone and Internet. The web server is often connected to a database which the user can query and update. See Figure 1 for a telephony web system architecture. The components of the telephony web system architecture are described as follows [5]:

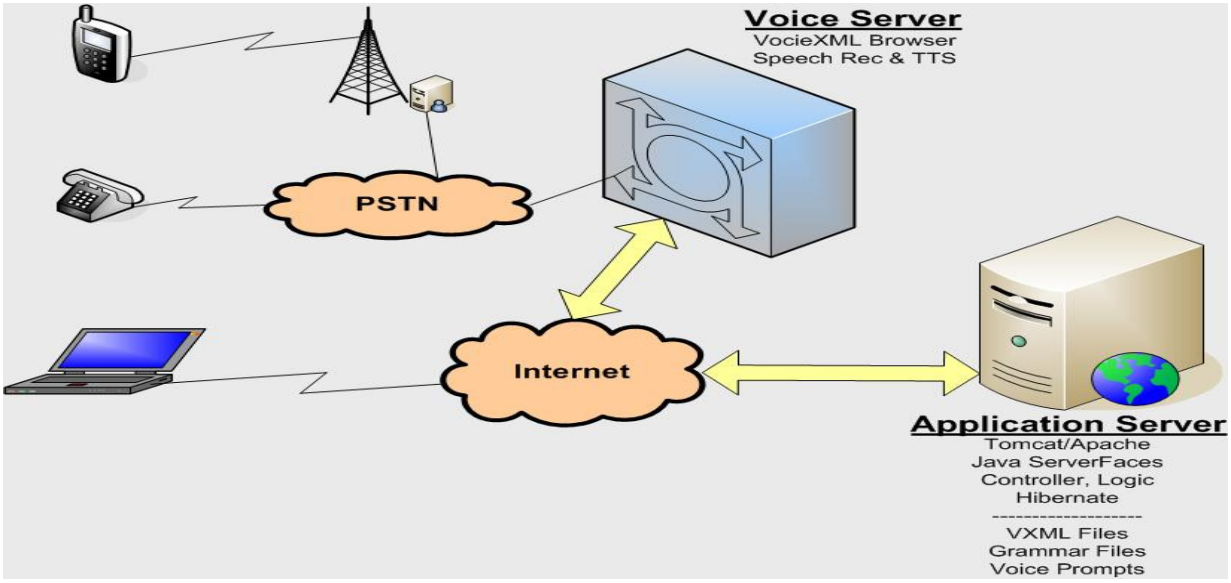

Figure 1: Telephony Web System Architecture Sources [5]

Voice Server: The voice server (also called VoiceXML gateway) is responsible for communicating with the user over the telephone and performs telephony tasks. The VoiceXML gateway pulls VoiceXML, grammar and ".wav" files from the web server over the Internet or depending on a dedicated link application requirements and implementation architecture. The voice browser in the VoiceXML gateway collects speech input from the user over the telephone and presents to the inner components of the VoiceXML gateway for processing. The voice browser also

presents it the output to the user in the form of spoken words over the telephone. VoiceXML code itself is neither generated nor stored by the VoiceXML gateway. The VoiceXML gateway fetches the VoiceXML code from the web server over the Internet. In some cases, voice server and application servers can be co-located in order to reduce dependency on the Internet link between the two servers.

VoiceXML Browser: A VoiceXML browser is the human machine interfacing component of web-based speech driven voice application. A voice browser provides an interface between the caller and the different components of the voice server. The voice browser is an equivalent of the HyperText Markup Language (HTML) browser (Internet browser). Just like an HTML browser interprets the application server code and presents to the user and accepts inputs (HTML forms) from the user for processing by the server, the voice browser interprets the voice server code and presents to the caller over the phone and accepts speech input (VoiceXML forms) for further processing by the server.

From an architectural point of view, the main difference between a web and a telephone application is that in a web application, the browser is provided by the client machine. As a result of that, application designer can rely on client resources in using technologies like Javascript and Flash to present and accept the data. The voice browser on the other hand is a part of the VoiceXML gateway and does not reside on the client, which is the calling device (telephone) for voice applications.

Speech Recognition: This is also called Automatic Speech Recognition (ASR) or Speech Recognition Engine. The main purpose of ASR is to convert human speech into the text form that application can understand. It interprets callers' spoken words and makes decisions based upon the spoken language and application rules defined in the grammar files.

Text to Speech: Affectionately called TTS, a Text-to-Speech system provides the reverse of the speech recognition process: that is converting text to speech. A TTS system announces text provided by the application to the caller in the form of spoken words.

Application Server: The Application server houses the database and necessary codes that are used to communicate with the user of the application and performs tasks related to business logic. The

application can either be resident on the web server or hosted on a separate database server.

VoiceXML grammar: A VoiceXML grammar identifies different words or phrases that a user might say, the recognized vocabulary and syntax are decided *a prioi by* the programmer. The VXML grammar is the core of a VXML application, since it determines the recognisable inputs. Grammars can range from a single word to phrases. A voiced grammar could be given in the body of the VoiceXML script (inline grammar), or in separate file (external grammar). Here, the important thing to understand is that a user can only pronounce a sentence predefined in the VoiceXML application's grammar. For example, it means that it will be impossible for the computer to understand "Nicholas" or "Azeta", for instance, if they are not available in the grammar.

2.0 UML MODELLING AND DESIGN ARCHITECTURE

The Unified Modelling Language (UML) was used to capture and model some of the functionalities in the application. The UML is a visual language that provides a means to visualize, construct and document the artefacts of software systems [6]. The sequence diagram for accessing information in the application is contained in Figure 2.

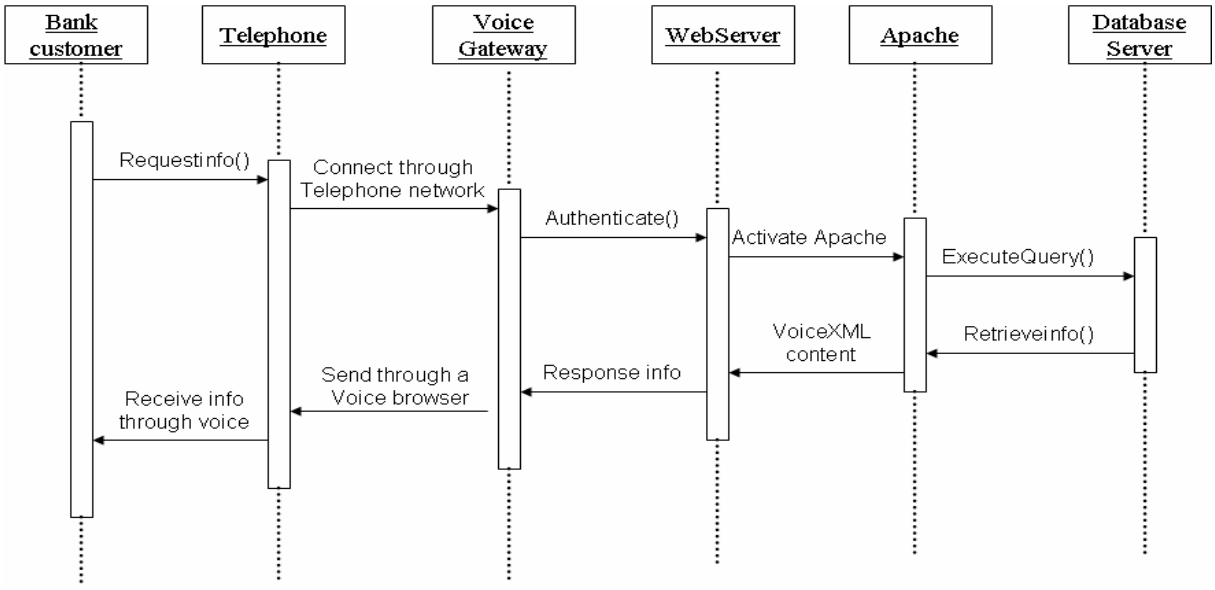

Figure 2: Sequence Diagram for Access to the System

The architecture presented allows easy access to banking transaction information from any point in real-time using either a mobile phone or land phone. Further more, we present client-server architecture for the physical infrastructure and a 3-tier, client-server architecture for the logical implementation and deployment. The two architectures are depicted by Figure 3 and 4 respectively.

3.1 Logical Architecture

Figure 3 gives a logical overview of the architecture of the banking application. The architecture shows the locations of each of the services in the system and consists of the client interface, middleware and database repository. The database is separated from the client by the middleware. The middleware concept helps to improve scalability, load balancing, transactional processing and interoperability issues by providing a means that allows for non-propriety/multi-platform banking services.

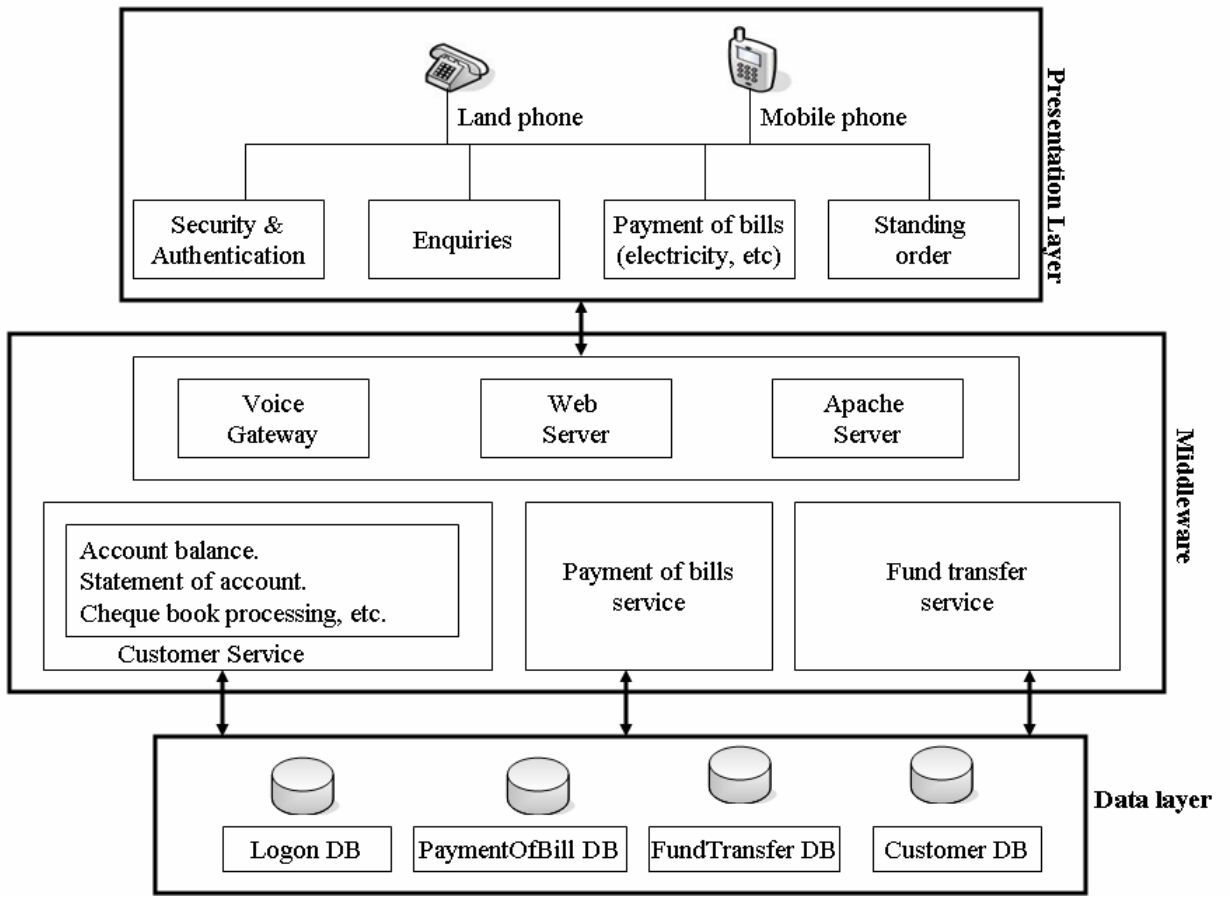

Figure 3: Software Architecture for the Application

The presentation layer provides client access to the banking application through the middle-ware. The solutions provided on this layer include Security and Authentication (SA), Enquiries, Payment of bills and Standing order. These services do not store or process any form of data. The phones deployed on this layer only serve the purpose of VUI. They only provide an interface for the middle layer and the data layer. Currently, server side-processing solutions offer the best alternative in most dynamic situations because they carter for the limited memory and processing power of many mobile devices [7]. It is important that any mobile access to the e-Banking enterprise application be strictly controlled and regulated to secure the system.

3.1.1 Middleware

The servers house all the application codes and are organized in three-tier architecture. The presentation tier communicates with the voice gateway component of the middle-tier through the voice browser. The middle-tier contains the voice gateway and the application logic.

Users access the application from various supported devices such as personal digital assistants and cell phones within and any where outside the bank provided there is network coverage. The application user interface allows users to access the application module. All users are authenticated before they can access any of the modules in the system. Once a user has been authenticated, the user's query is translated by the ASR to text and passed to the database server for execution. A user can only access the module for which s / he is authorised. The client application interfaces to the application layer using the voice gateway. The database server provides data services and data base management system function.

The application is developed for telephony and allows voice browsers (running on the voice gateway) to be used as the interface. The information from the database is presented in a form compatible with the client using the voice browser.

3.1.2 Data layer

The data layer is responsible for the storage, retrieval, maintenance and integrity of the data manipulation within the system. Though we used MySQL for the implementation of the data layer, the architecture presented in Figure 3 could allow other databases such as MS SQL / SQL stored procedures, Oracle Database, MS Access, etc, to be implemented. Stored procedures are complex queries stored in a compiled form inside the

database, which can be executed by the Database Management System (DBMS) on the server side to enhance the robustness and reliability of the system [8, 9].

3.2 Physical Architecture

The architecture employed for this prototype consists of a complete range of robust high performance client and server platforms with integrated enterprise application and data extendable to banking customers in real-time.

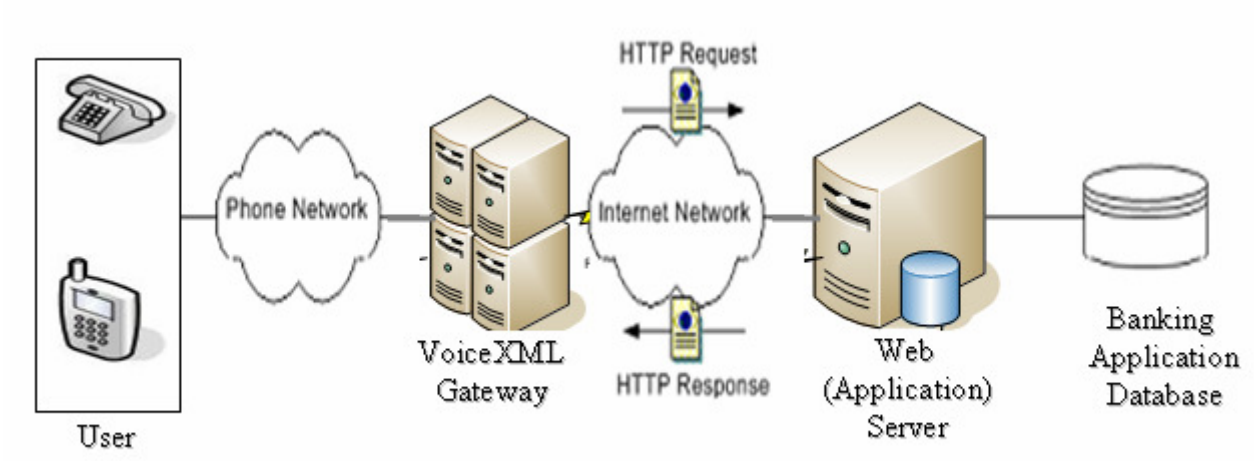

Figure 4: Hardware Architecture for the Application

The client systems include hand-held devices such as mobile phones and personal digital assistants and telephones. The servers are used to maintain connectivity to enterprise resources for the mobile banking solutions that include the customer's service, teller/cashier, standing order and payment of bills. The availability of an enterprise application and servers is crucial to banking services where customer records must be available 24/7.

A major benefit of the multi-tier architecture used is that it increases application scalability and performance by enabling several banking customers to be connected concurrently to the system. In a multi-tier architecture only application servers connect directly to databases. In this way, the application server can process multiple requests from many banking customers within the telephone network and Internet through a pool of pre-allocated database connection, thus reducing the database server load. Furthermore, the load on the application server tier is balanced by using multiple application servers. The multi-tiering of the servers support the implementation of thinner clients, since most of the logic

would be made to run in the application server and database tier [10, 11].

3.3 System Design Pseudocode

The pseudocode presented in Figure 5 was used to implement the prototype application discussed in this paper.

```
BEGIN
 SYSTEM PROMPT 'Welcome to secure
 Bank Nigeria Ltd, Automated Phone
  Banking System';
 //Caller suppler a username and password
 SYSTEM AUTHENTICATES A
  CALLER;
 //Caller suppler an Account Number
  SYSTEM AUTHENTICATES
  CALLER'S ACCOUNT NUMBER;
   WHILE NOT EOF DO
      SYSTEM request enquiry from
      caller;
      CALLER supplies enquiry;
      IF enquiry is in database THEN
         SYSTEM supplies answer from
         database;
      ELSE
         SYSTEM process enquiry;
         SYSTEM sends a notification to
         CALLER;
       ENDIF
     ENDDO
END
```

Figure 5: Pseudocode for phone banking application

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version = "2.1">
  <form id="MainMenu">
     <field name="username">
        <prompt>
   Welcome to Secure Bank Nigeria Limited, Automated Phone Banking System.  What is your user name.
        </prompt>
         <grammar type="text/gsl"> [admin]  </grammar>
         <nomatch> The user name you entered is not recognized.<reprompt/>  </nomatch>
        </field>
        <field name="acctnumber">
         <prompt> What is your account number. </prompt>
        </field>
        <field name="choice">
          <prompt>
               for question one, What is my account balance, say one.
               for question two, Request for statement of account, say two.
           </prompt>
           <grammar type="text/gsl"> [one two]  </grammar>
         </field>
          -----
         <filled>
          <if cond="choice == 'one'">
              <submit next= "customerenquiry.php  method= "GET" namelist = "acctnumber"/>
          </if>
        </filled>
   </form>
</vxml>
```

Figure 6: Partial code for the application

4.0 SYSTEMS DEVELOPMENT AND DEPLOYMENT

We developed the application using VoiceXML to render voice response to the phone interface. Hypertext Pre-processor (PHP) and Apache web server were used for the application layer. MySQL database was used for the data layer. The VoiceXML was preferred amongst other competing tools for developing telephony application, such as Microsoft .NET Speech Technologies because it seems most likely to persist in the long term and is the most powerful, most general and popular [12]. The choice of PHP, Apache and MySQL is because they are free and open source[13], which makes it more economical to develop with at the prototyping stage.

In developing the application (see Figure 6 for the partial code), first, we downloaded the voxeo free phone emulator (prophecy) from [14] and install on a local computer to facilitate execution of the application as a localhost. Second, we deploy the application on a voxeo voice network so that it can be accessed anywhere, anytime via the Internet.

Simply dial 009-1-312-4360541 or 009-1-857-3628433 from any mobile or land phone to connect and execute the application deployed in voxeo voice server. The default username is admin and password is admin. The default account number is 1234, for testing purposes.

Figure 7 and 8 depict a live deployment of the phone banking application on a Nokia 6301 mobile phone. They show the screen shots representing the voice interaction between the caller and the system. The application is launched by dialling any of the recommended phone numbers. Once connected, the system will be prompted with a welcome message and go ahead to authenticate the user name, password and account number before any transaction can be processed.

Once a valid username, password and account number are provided, the system will ask for the services or enquiries demanded by a bank customer. Figure 8 shows a customer's voice request for account balance and the system's voice response.

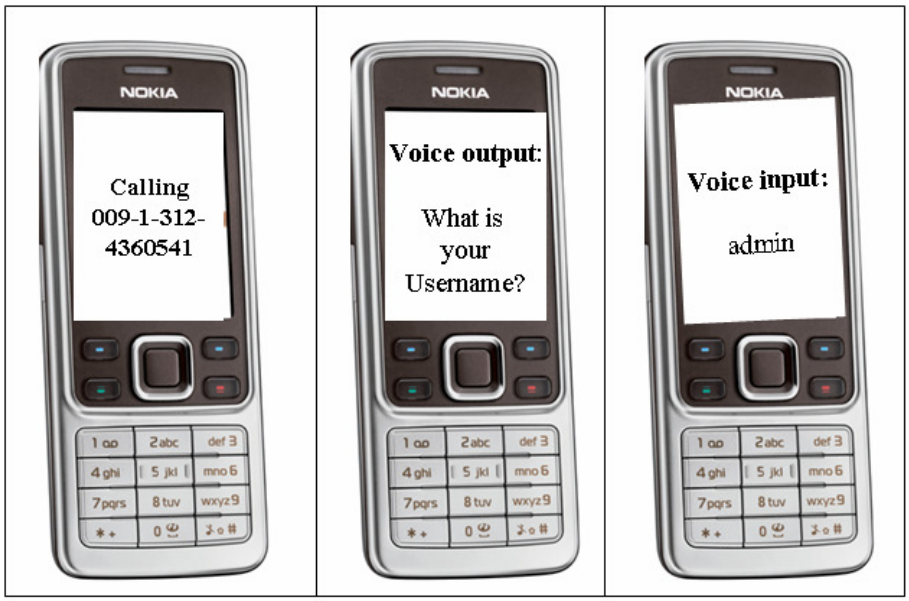

Figure 7: Dialling and Authentication session

Figure 8: Account balance request and response session

5.0 CONCLUSION

This paper has reported the design, development and deployment of phone banking application using VoiceXML to enhance the QoS derived from banks. The application ensures real-time availability of customers' transaction information by enabling access to authorised bank information systems via

telephones within and outside the banking premises. This will eliminate the long queues that usually result during pick periods in the banks. In addition, the integration of the e-banking voice functionality will assist the visually impaired, the aged and the pensioners to access their banking transactions at suitable locations without physically going to the bank premises or using the ATM stand.

Finally, the adoption of the product of this research in banking operations will offer a good contribution for banks and other financial institutions in providing better customer services thereby increasing the liquidity flow among individuals, small and medium scale companies, as well as big organisations, within a growing economy such as Nigeria.

7.0 REFERENCES

[1] Donli J. G. "The state and future of the Banking industry in Nigeria", available online at:http://www.ndic-ng.com/pdf/tsafobin.pdf, access 19th March 2008, p4.

[2] Nigerian ICT Handbook (2003), a report of ICT product and services usage in Nigeria, published by Compumetrics Solutions Ltd, Lagos. http://compumetricsolutions.com.

[3] Palo Wireless, VoiceXML resource center, available online at http://www.palowireless.com/voicexml/tutorials.asp, accessed 19th March 2008

[4] Rouillard J., (2007), 'Web services and speech-based applications around VoiceXML" Journal of networks, vol. 2, no. 1, February 2007, p27- 35.

[5] Qureshi R., (2007), "Building a Telephony Web Application" May 1, 2007. Available at:http://www.hunzatechnologies.com/telephony-web-tutorial/ accessed 23 January 2008

[6] Simeon Bennett, John Skelton, Ken Lunn (2005), *Schaum's Outlines UML,* 2nd Edition, McGraw-Hill International, UK

[7] Mark Ridgeway (2002), *.NET wireless Programming,* SYBEX, London.

[8] F. Lamberti, B. Montrucchio, A. Sanna, and C. Zunino (2003), "A Web-based Architecture Enabling Multichannel Telemedicine Applications", *Journal of Systemics, Cybernetics and Informatics,* Vol. 1, No. 1.

[9] Ramalho Jose Antonio (1999), *Learn Microsoft Server 7.0,* Wordware Publishing, Singapore.

[10] Paco Gomez and Peter Zadrozny (2000), *Professional Java 2 Enterprise Edition with BEA WebLogic Server*, Wrox Press, UK.

[11] "Client / Server and the N-Tier Model of Distributed Computing", Available at: http: //n-tier.com / articles / csoverww.html

[12] Coding a telephone enabled application, available online at: http://archive.devx.com/free/articles/2000/Phillips01/Phillips01-2.asp, accessed 19th March 2008.

[13] Siemens G. (2003). *Open source content in education: Part 2 - Developing, sharing, expanding resources,* accessed April, 2007 from http://www.elearnspace.org/Articles/open_source_part_2.htm

[14] Voxeo voice network, available online at: http://www.voxeo.com/prophecy/ accessed 19th March 2008.

ABOUT THE AUTHORS

Azeta, A.A. is a Ph.D. student in the Department of Computer and Information Sciences, Covenant University, Ota, Nigeria. He holds B.Sc. and M.Sc. in Computer Science from University of Benin and Lagos respectively. His current research interests are in the following areas: Software Engineering, Algorithm Design and Mobile Computing. He currently lectures at Covenant University. He is a member of the Nigerian Computer Society (NCS) and Computer Professional Registration Council of Nigeria (CPN).

Ikhu-Omoregbe, Nicholas has a B.Sc degree in Computer science from the University of Benin, Benin City, an M.Sc degree in Computer Sciences from the University of Lagos, and a Ph.D. degree in Computer Science from Covenant University, Ota, Nigeria. His research interests include: Software Engineering, Mobile Computing, Mobile Healthcare and Telemedicine Systems, and Soft Computing. He currently lectures at Covenant University. He is a member of the Institution of Electrical and Electronics Engineers (IEEE).

Charles K. Ayo holds a B.Sc., M.Sc. and Ph.D in Computer Science. His research interests include: mobile computing, Internet programming, e-business and government, and object oriented design and development. He is a member of the Nigerian Computer Society (NCS), and Computer Professional Registration Council of Nigeria (CPN). He is currently the Head of Computer and Information Sciences Department of Covenant University, Ota, Ogun state, Nigeria, Africa. Dr. Ayo is a member of a number of international research bodies such as the Centre for Business Information, Organization and Process Management (BIOPoM), University of Westminister. http://www.wmin.ac.uk/wbs/page-744; the Review Committee of the European Conference on E-Government, http://www.academic-conferences.org/eceg/; and the Editorial Board, Journal of Information and communication Technology for Human Development.

Dr. Atayero, Aderemi Aaron Anthony holds a B.Sc. degree (summa cum laude) in Radio Engineering and M.Sc. in Satellite Communication Systems (1992, 1994 respectively) both from the Moscow Institute of Technology, and a Ph.D. in Technical Sciences (Speech Processing/ Satcom, 2000) from the Moscow State Technical University of Civil Aviation, Moscow, Russia. His current area of research interest is as follows: Speech Processing. FPGA Design, Architecture, and Applications: FPGA – Implementation of Digital Speech Processors. FPGA – Implementation of Digital Filters. FPGA System–on–Programmable Chip Design. He is a member of a number of academic and professional organisations including the Institute of Electrical and Electronic Engineers (IEEE), Nigerian Association of Inventors (NAI). Dr. Atayero is a Senior Lecturer in the department of Electrical and Information Engineering at Covenant University.