VOL. 3, NO. 7, July 2013 ISSN 2225-7217
ARPN Journal of Science and Technology
©2011-2013. All rights reserved.

http://www.ejournalofscience.org

A Mailing List Web Service

[1] Eteng, Idongesit.E., [2] Osofisan, Adenike O, [3] Udomisor, W.S.
[1] Department of Mathematics/Statistics & ComputerScienceUniversity of Calabar, Calabar, Nigeria
[2] Department of Computer Science University of Ibadan, Nigeria
[3] Department of Mathematics/Statistics & ComputerScienceUniversity of Calabar, Calabar, Nigeria

[1] idongesitessien@yahoo.com, [2] mamoshof@yahoo.co.uk, [3] Pmuf4lyf@gmail.com

ABSTRACT

Platform and vendor specific software/applications have been a major set-back to the integration of heterogeneous applications. A mailing list web service can be used to provide ready-made functionalities to a traditional mailing list application. Clients can subscribe to the mailing list web service and they can also chose to consume any web service function of their choice. This research studyuses procedural programming, systems analysis and software engineering techniques to design, develop, implement and subscribe to the mailing list web service. The data responsible for all processing in the mailing list web service is managed by a Relational Database Management system. The end product of this project is a newsletter sending system which enables the mailing list application administrator to send newsletters to its subscribers. The mailing list web service can be consumed by any web based application.

Keywords: *Web Service,Relational Database Management System,Mailing List Service*

1. INTRODUCTION

Web services present a new architecture for creating applications that can be accessed from any computer. In recent years web services have been established as an important paradigm in building applications and business processes for the integration of heterogonous applications in the future. Web services are based on open standards, and focus on communication and collaboration among peopleand applications.

This technology connects programs to each other across distant points on the global map, transport large amount of data more efficiently and cheaply than ever before, thereby promoting software portability and reusability in applications that operate over the Internet.

Even though there is a limit to what web services can do, they are changing the way we think about distributed software systems. Web services provide a layer of abstraction above existing software systems such as application servers, messaging and packaged applications, and they are capable of bridging the operating system, hardware platform or programming language just as the web is.

On the other hand mailing list managers have a venerable history on the Internet and they are an excellent vehicle for distributing focused information to an interested receptive audience.

According to Coita (2007), mailing lists typically refer to four things - a list of email addresses, the people (“subscribers”) receiving mails at those addresses, the publications (email messages) sent to those addresses, and a reflector, which is a single email address that, when designated as the recipient of a message will send a copy of that message to all subscribers.

After building up a base of subscribers to a website, it is nice to be able to keep in touch with them by sending out a newsletter. A “mailing list manager” (MLM) software does this reasonably well.

There are an ever increasing number of mailing list managers available and almost all of them share the following common traits; the ability for users to subscribe and unsubscribe themselves from a distribution list, the ability of an administrator to manually remove a user, the ability to restrict posting to a small number of individuals and soon.

Possible uses for such a piece of software include a monthlynewsletter for customers of a business, a way to distribute information to users of a particular software package - for example, notification of a security fix, or a way for people who share common interests to communicate with each other.

Within the context of this research work, a mailing list manger software was developed, and the most basic features of a mailing list manager were incorporated into the software through the consumption of a“MLM” web services, an idea born from the concept of “software as a service” (SAAS). It should be noted that some of the features/functions of the software runs on a server/servers elsewhere on the Internet. When that server is updated, all clients worldwide see the new capabilities, no local installation is needed. The service is accessed through a browser - these are quite portable so you can run the same applications on different kinds of computers from elsewhere in the world.

1.1 Statement of the Problem

The rapid growth of the Computer and Information Technology world has resulted in the need for the development of platform-independent and vendor-independent software/applications that solve problems that cut across many fields of endeavour.

With the availability of the Internet and Computer Networks, attempts have been made in this research work to develop and deploy a Mailing List Web Service (a technology which depends on the Internet and Computer Networks) that provides a "subscribe" and "unsubscribe" function that is not limited to any platform or vendor.

1.2 Aim of Research Work

This project is aimed at designing and implementing a web-basedmailing list manager.

1.3 Objectives of the Study

The objectives of this research work are as follows:

- To establish an understanding of what the underlying concepts of web services and mailing list mangers are.
- To study and analyze the general structure of a mailing list manager.
- To produce design artifacts for mailing list manager software.
- To build a mailing list manager by integrating some of its basic functions through a web service, and making it possible for these functions to be consumed and integrated into any other application.

1.4 Scope of the Study

This research work has its focus on implementing a mailing list manager software and some of its basic features as a web service such that anyone or group of functions can be integrated, used or consumed by any other application provided the developer of that application intends to do so.

1.5 Lımıtatıons of the Study

Due to the complex nature of web services and mailing list managers, this research was limited to the building and implementation of a newsletter sending system. A few features that are not needed for the proper functioning of the mailing list manager were not included in this research work.

2. LITERATURE REVIEW

In this section, a brief description of the relevant literature in web services and the mailing list manager is is given.

2.1 Web Services

There exist quite a number of different ideas/opinions on the meaning/definition of a web service.

The World Wide Web Consortium (W3C) Architecture Working Group defines a web service "as a software system designed to support interoperable machine-to-machine interaction over a network". In their opinion it has an interface described in a machine-processable format (specifically Web Service Description Language). They also state that other systems interact with the web service in a manner prescribed by its description using SOAP messages, typically conveyed using Hyper-Text Transfer protocol (HTTP) with an Extensible Markup Language (XML) serialization in conjunction with other web related standards.

Richards, (2006) does not agree with such "strict definition" of the term above. Instead he prefers to define a web service "as an application that is assessed across the Internet using Standard Internet Protocols and that uses Extensible Markup Language (XML) as its messaging format.

One can define a web service "as a collection of functions that are packaged as a single entity and published on the network for use by other programs (Conolly and Begg, 2010).Microsoft has a somewhat narrower definition and therefore defines a web service "as a small reusable application written in Extensible Markup Language (XML), which allows data to be communicated across the Internet and intranet between otherwise unconnected sources that are able to host or act on them.A web service is also defined "as a software component stored on one computer that can be accessed via method/function calls by an application (or other software components) on another computer over a network" (Deitel and Deitel, 2007).

2.2 What Web Servıces Look Lıke

Generally speaking, a web service is a software application that can be accessed remotely using different XML-based languages. Normally a web service is identified by a Uniform Resource Locator (URL), just like any other website. What makes a web service different from ordinary websites is the type of interaction they can provide.

Most websites are designed to provide a response to a request from a person. The person either types in the Uniform Resource Locator of the site or clicks on a hyperlink to create a request. This request takes the form of a text document that contains some fairly simple instructions for the server. The instructions are limited to the name of document to be returned or a call to a server-side program, along with a few parameters. Fig 2.1 shows how a browser interacts with a web server to make a request.

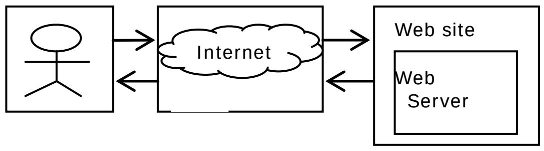


Fıg 2.1: A browser ınteracts wıth a webserver to make request

A web service is similar to a web site in that it is accessed via a Uniform Resource Locator but the difference lies in the content of what is sent in the request from the client to the service. Web service clients send an Extensible Markup Language (XML) document formatted in a special way in accordance with the rules of the SOAP specification. A message can contain a call to a method along with any parameters that might be needed. In addition, the message can contain a number of header items that further specify the intent of the client.

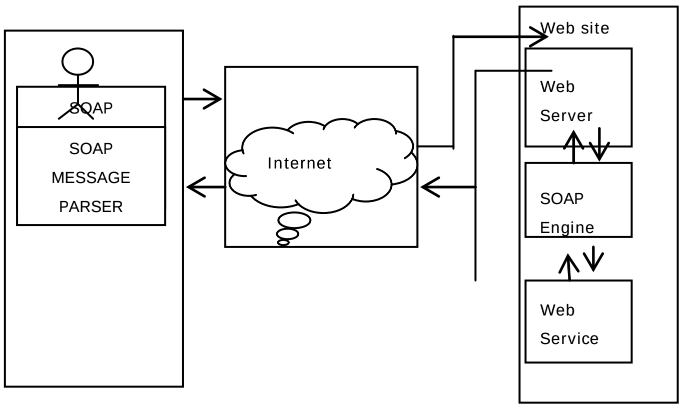


Fıg 2.2: A Clıent Interacts Wıth A Web Servıce Vıa A Web Server Such As Apache

Web services are a messaging framework. The most common form of a web service is that it must be capable of sending and receiving messages using some combination of standard Internet protocols. This is illustrated in fig 2.2. Web service call procedures running on a server, in which case the messages encode “call this subroutine with the arguments”, and “here are the results of the subroutine call” (Snell et al, 2000).

Due to the complex nature of web services, it is most times difficult to get a grasp of what they really are, unless there is a clean-cut and simple scenario to explain the concept.

2.2.1 A Scenarıo Descrıbıng the Nature of a Web Servıce

Using procedural programming in Hypertext Preprocessor 5 (PHP 5), an application is built using procedures/sub-routines. The procedures are well thought out, so each performs operations for specific areas of functionality. Another area of the same organization is working on a separate application and ends up needing to access functionality from the first application and on top of that this new application is not written in PHP and so cannot reuse any code. The brute force method would be to have this new application duplicate the logic the PHP application does. This, however, presents problems if the logic were to change In the PHP application. The other application would need to also change its logic or face the problem that it no longer works correctly, which could lead to a variety of problems within the company/organization, including data corruption.

Using service oriented Architecture (SOA), the PHP application can expose the functionality of its procedures via a service. Through a common protocol and descriptive messaging, the other application can access the functionality of the PHP application (Richards, 2006).

2.3 Characterıstıcs of Web Servıces

XML-based: Web services use XML at data representation and data transportation layers. Using XML eliminates any networking, operating system,, or platform binding.

Loosely Coupled: A consumer of a web service is not tied to that web service directly. The web service’s interface can change with time without compromising the client’s ability to interact with the web service. Adopting a loosely coupled architecture tends to make software systems more manageable and allows simpler integration between different systems.

Ability to be Synchronous or Asynchronous: Synchronicity refers to the binding of the client to the execution of the service. In synchronous invocations, the client blocks and waits for the service to complete its operation before continuing. Asynchronous operations allow a client to invoke a service and then execute other functions. Asynchronous clients retrieve the result at a later point in time, while synchronous clients retrieve their result when the service has completed.

Supports Remote Procedure Calls (RPC): Web services allow clients to invoke procedures, functions, and methods on remote objects using an XML-based protocol. Remote procedures expose input and output parameters that a web service must support.

2.4 Key Components of Web Servıce (Core Web Servıce Standards)

For a web service transaction to complete successfully, all of the components involved in processing the transaction must behave in ways that the other components expects them to.

The core web service standards as of this research study are briefly described below. The core components of web service architecture are shown in fig 2.4.

2.4.1 Extensıble Markup Language (XML)

XML has become the defacto standard for describing data to be exchanged on the web and it is the language that all web service languages are built on.

2.4.2 Soap

SOAP originally stood for Simple Object Access Protocol but SOAP is now considered a specification name and not an acronym. SOAP is a specification that defines an XML grammar for both sending messages and responding to messages that one receives from other parties. The goal of SOAP is to describe a message format that is not bound to any hardware or software architecture, but one that carries a message from any platform to any other platform in an unambiguous fashion. SOAP is also not tied to any particular protocol, although Hypertext Transfer Protocol (HTTP) is allowed.

2.4.3 Web Servıce Descrıptıon Language (WSDL)

This is a platform independent XML-based protocol for defining a web service. It specifies the location of a web service, the operation the web service exposes (i.e. it describes a piece of software in terms of the method/function call it responds to), the soap messages involved, and the protocol used to talk to the web service.

2.4.4 Unıversal Dıscovery Descrıptıon Integratıon (UDDI)

The UDDI specification describes how a potential customer of a web service could learn about its capabilities and obtain the basic information needed to make initial customers obtain the basic information needed to make initial contact with the site. Normally contact includes a download of the web service description language (WSDL).

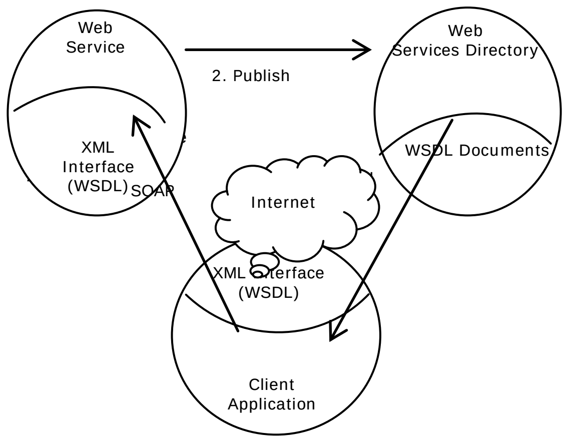


Fıg 2.4: A typıcal web servıce archıtecture

2.5 How Automated Maılıng Lısts Work

Automated mailing lists or electronic mailing lists are usually automated through the use of special mailing list software and a reflector address that is a setup on a server capable of receiving email.

Incoming messages sent to the reflector addresses are processed by the software, and depending on their content, are acted upon internally (in the case of messages containing commands directed at the software itself) or are distributed to all email addresses subscribed to the mailing list. Depending on the software, additional addresses may be setup for the purpose of setting commands.

2.6 Brıef Hıstory of Maılıng Lıst Manager

Back in the mid - 1980's, the system of interconnected computers known as the Internet was not yet around. While in the United States of America, there was some interconnection between colleges and Government's "Advance Research Projects Agency Network" (ARPAnet), the only way any other machine; such as those of different Universities communicated was over a system called "Because It's Time Network" (BITNET). The BITNET machines let messages for each other pile up, and when they would call each other on the phone lines and send messages (Blackman et al, 2002).

BITNET had a central post, a Network Information Centre (NIC) called 'BITNIC'. BITNIC kept a number of distribution lists for BITNET users. However the BITNIC's lists were setup in the primitive way (i.e. setting up a single address on an email domain one had control, and forwarding it to a number of people; a single address with no way for users to add or remove themselves. If one wished to be on the mailing list, he/she had to contact the BITNIC staff and have them add him/her by hand (Blackman et al, 2002).

Unfortunately, as BITNIC grew larger, managing the lists by hand was no longer feasible. Additionally since all the mails for BITNIC was affected by the traffic of the lists – which were at that point quite large – even private BITNET emails were affected. This caused the entire system to be so slow that emails took a week to arrive their destinations.This indicated clearly that something needed to be done.

Since the source of the problem was the traffic of BITNIC's mailing lists, a computer science student named Eric Thomas decided to write a piece of software to replace the manually –managed mailing lists. When it went online on July 1996, a piece of software was managing a mailing list for the first time-Eric Thomas had created the first mailing list manager. Soon thereafter, others created similar packages for other systems, such as the LISTERV imitation for UNIX, Listproc (Blackman et.al,2002).

2.7 Types of Maılıng Lıst Managers

- **Announcement List (Newsletter mailing list)**: One type of mailing list is an announcement list which is primarily used as a one-way conduit of information and can only be “posted to” by selected people. Newsletters and promotional mailing lists are employed in various sectors of direct marketing campaigns

- **Discussion List**: Another type of electronic mailing list is a discussion list, in which any subscriber may post. On a discussion list a subscriber uses the mailing list to send messages to all other subscriber, who may answer in similar fashion. Thus actual discussion and information exchanges can happen. Mailing lists of this type are often topic-oriented (for example politics, religion, scientific discussion, jokes etc), and the topics can range from extremely “harrow” whatever you think can interest us”. In this, they are similar to “User’s Network” (USENET) newsgroup and share the same aversion to off-topic massage. The term discussions encompass both these type of lists and newsgroups. (Wikipedia 2013).

2.8 The Need for the Maılıng Lıst Manager Web Servıce

One of the challenges for organization individual is to discover opportunities in which web service can be used to solve today’s problem that are ever present, but never solve. These areas include the cost of doing business, in effect, the cost of software development, and the time it takes to react to new market opportunities.

Now we will take a look at a fairly extensive list of why we need a mailing list manager web service.

2.8.1 Remote System Management

There is virtually no limit to the amount of remote management that can be done using this technology. Most of the mailings occur when a maintenance personnel can manage, maintain, repair and correct a problem arising from the use of the mailing list manager from his own desk without incoming any travel over head or expense (Stephen and Mike. 2003).

2.8.2 Lower Software Development Cost

The “Holy Grail” of software development has been the concept of code reuse. The idea of code reuse in as simple as it is intuitive. Instead of reinventing the wheel a thousand times, you invent it once and use it a thousand times. The fact that web services can interoperate regardless of the programming language that each of them is written in a tremendous development in the code reuse world (Stephen and Mike 2003).

2.8.3 Faster System Developent

Better tools mean faster development, which in turn leads to lower development cost. Web service are based on both a service and a format description of the service called the Web service Description Language (WSDL). This document contains enough information about the service to allow a client to be generated. Instead of starting with a blank sheet of paper, the programmer of web service client starts with a generated client that can successfully access a web service. He/she can then enhance that client code to interact with the other system on the client side. The time and learning curve saving from this approach is significant. Even for complex requirement, the generated part of the web service code can get the project started. From there the programmer can hardcode the unique part that are beyond the capabilities of the tools (Stephen and Mike, 2003).

2.8.4 Better Integratıon Wıth Extenal Busıness Partners

Most companies are built on the fundamental of business partners - other companies have traditionally been classified as suppliers. The forward – thinking firms have concluded that the best way to lower cost is not to beat your suppliers down to get them to their margins, but to help those suppliers lower their cost. It is better to find a suppliers/partner that will allow a level of integration that will benefit both companies (Stephen. and Mike 2003). For instance, writing a newsletter to a number of the institutes of chartered accountants (ICAN) is essentially the same thing as writing a newsletter to a number of the Nigeria computer society (NCS), even though their daily work activities in both careers are very different (this is an example of outsourced human Resources) and for the outsourcing to be timely less expensive, the human resource vendors systems must be able to interact with the client’s system well.

Web services are ideal for this kind of interaction. One can interconnect two different systems without making significant alteration to either one. This can enable an outsourced company to act as a web service client in order to disseminate accurate information and gather personnel (Stephen and Mike, 2003).

2.9 The Pıtfalls of the Maılıng Lıst Manager Web Servıce

Web services are not a “magical bullet” solution for every issue, they do have limitations. Web services are plagued by pitfalls traditional to it as a technology, performance issues, lack of standards, staffing issues, and even newness of the web service technology.

No technology is perfect. Although web services do a great job at solving certain problems, they bring along issues of their own. Some of these pitfalls are inherent to the technological foundations upon which web services are based, and others are based on the specifications themselves (Potts and kopack, 2003).

2.9.1 Some of the Bıggest Issues of Web Servıces

- ❖ **Availability**: Web services which use the same infrastructure as websites are not 100% available. Even if the server is up and running, your Internet Service Provider (ISP) might not be available or the ISP hosting the other side of the transaction might not be available.

- ❖ **Matching Requirements**: Any time one creates a general service that handles a variety of customers; one will run into having to deal with specialized requirements. Some customers might require the one extra feature that nobody else needs. Web services are envisioned as a "one size fit many customers "technology.

- ❖ **Immutable interfaces**: If one invests in creating a web service for his/her customers, he/she has to avoid changing any of the methods that he/she provides and the parameters that his /her customers expect. One can create new methods and add them to the service, but if one is changing existing ones, his/her customers program will break.

- ❖ **Guaranteed Execution**: The whole idea behind having a computer program instead of doing a good job by hand is that the program can run unattended. Hypertext Transfer Protocol (HTTP) is not a reliable protocol in that it does not guarantee delivery or a response. If one needs this kind of guarantee he/she should either write the code to retry request or arrange to send request through an intermediary who will perform these retries for him/her.

- ❖ **Performance Issues**: The only performance guarantee that one has concerning the Internet is that performance will wildly fluctuate from one time to the next. Performance-critical systems are not suited to becoming web services. Web services rely on Hyper Text Transfer Protocol (HTTP), the HTTP communication transaction enables the server-side to handle many units, but this also means that a lot of time is wasted creating and terminating connections for clients that need to perform a large number of calls between the client and the server.

- ❖ **Lack of standards:** This issue centers on incomplete or non-finalized standards. If the guiding principle of Web services is to create open standardized interchange systems for remote program execution, the utilization of any vendor-specific solutions should be avoided. Current web service specifications and standards are lacking in areas such as security and privacy, authentication, non-repudiation (rock-solid proof that a communication took place transactions, billing and contracts, provisioning, scalability and testing.

- ❖ **Newness of Technology:** Most technologies are over hyped in their first two years because it typically takes that long for the platforms to mature to the point of fixing initial problems and the toolsets to become powerful enough to provide good solutions. Because web services are a relatively young technology, the standards and specifications are evolving rapidly.

- ❖ **Staffing Issues:** staffing can be one of the most frustrating parts of implementing any new technology. Because web services are a new technology, it can be somewhat difficult to find qualified and experienced staff to implement a workable solution.

3. DESIGN AND IMPLEMENTATION OF THE MLM

In this section, we will implement the mailing list manager by combining functions that traditionally belong to the mailing list application and functions that are exposed to the mailing list application through the web services. The mailing list application will be a newsletter sendingsystem in which only the lead administrator can send messages to subscribers.

3.1 Descrıptıon of the System

The mailing list manager (MLM) web service will lets an administrator create multiple mailing lists and send newsletters to each of those lists separately. This web service will use file upload to enable an administrator to upload text and HTML
versions of newsletters that they have created offline. This means that administrators can use whatever software they refer to create newsletters.

Users will be able to subscribe to any of the lists in the mailing list manager, and select whether to receive newsletters in text or HTML.

3.2 Requırement Specıfıcatıon

- Administrators shall be able to setup and modify mailing lists.
- Administrators shall be able to send text or HTML newsletters to all the subscribers of a single mailing list.

- Users shall be able to register in order to use the mailing list manager, and shall therefore be able to enter and modify their details.
- Users shall be able to subscribe to any of the mailing lists on the mailing list manager.
- Users shall be able to unsubscribe from lists they have subscribed to.
- Users shall be able to store their preferences for either HTML formatted or plain-text newsletters.
- For security reasons users shall be able to send mails to the lists or to see each other's e-mail address.

3.3 Unıfıed Modelıng Language (UML) Dıagrams

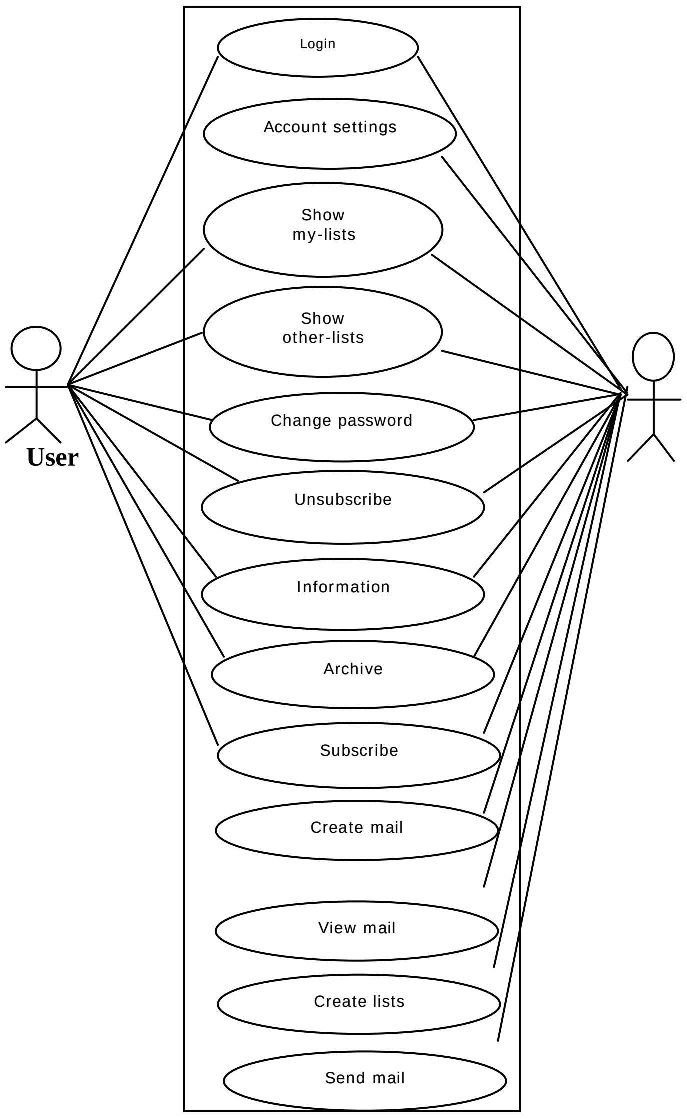


Fıg 3.1: A use case dıagram for mlm web servıce

Figure 3.1 shows the actions that both the users and administrators of the system can carry out after logging in. A user is allowed to change his password, change the lists he subscribed to, etc. On the offer hand administrators have additional options. Administrators can create mailing lists; create new messages for a mailing lists, etc.

3.4 The Database System

The user name, password and the lists they have subscribed will be tracked. We will also store each user's preference for receiving messages. The two options modeled in this research allow users to receive emails in either text or HTML format

A nice piece of functionality to have for a system like this is an archive of previously sent newsletters. Subscribers might not keep posting but might look something up. The archive can also act as a marketing tool for the newsletter as potential subscribers can see what the newsletters are like. For the MLM web service the database will need to store the following:

- ***Lists:*** this contains the mailing lists available for subscription.
- ***Subscribers***: this contains users of the system and their preferences.
- ***Sub_lists***: this contains a record of which users have been subscribed to which lists
- ***Email:*** A record of email messages that have been sent.
- ***Images:*** Because we want to send emails that contain multiple files, we also need to track images that go with the individual emails. For efficiency sake we will store these images outside the database. We will track the mails they are associated with, the path to the location where the image is actually stored, and the picture type of the image, e.g. jpeg/gif.

3.5 Entıty Relatıonshıp Dıagram (ERD)

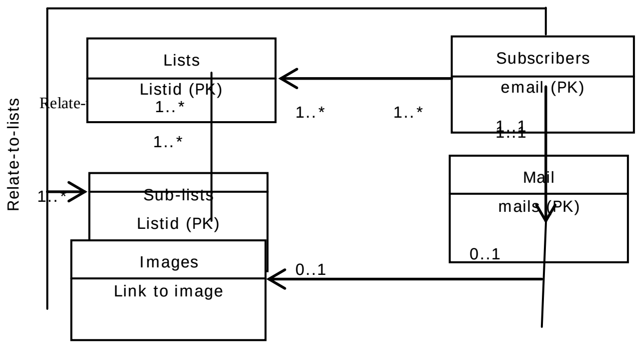


Fıg 3.2: An er dıagram for mlm database

Fig 3.2 shows an ERD for the database modeled in this paper.

Table 3.1: Possıble Actıons In The Mlm

ACTIONS	USABLE BY	DESCRIPTION
Log-in	Anyone	Log a user in and start a session.
Log-out	Anyone	End a session
New-account	Anyone	Creates new account for a users
Store-account	Anyone	Stores account details
Show-all-lists	Anyone	Show a list of available mailing lists
Show-archive	Anyone	Display archive newsletter for a particular lists
Information	Anyone	show basic information for a particular lists
Account-settings	Logged in users	Display user account settings
Show-other-lists	Logged in users	Display mailing lists to which the user is not subscribed to
Show-any-lists	Logged in users	Displays mailing lists to which user is subscribed to.
Subscribe (web service)	Logged in users	Subscribes a user to a particular list
Unsubscribed (web service)	Logged in users	Unsubscribe a user from a particular list
Change-password	Logged in users	Display the change password form
Store-change-password	Logged in users	Updates user's password in database
Create-mail	Administrator	Displays form to allow upload of newsletters
Create-list	Administrator	Display form to allow mailing list to be created
Store-list	Administrator	Store mailing list details in database
View-mail	Administrator	Displays uploaded but not yet sent newsletters
Send	Administrator	Send newsletters to subscribers.

3.6 Mailing List Application to Consume Mlm Web Service

The mailing list manager is implemented with functions that come with it by default, but it also subscribe/consumes two functions from the MLM web service – the subscribe and unsubscribe functions.

This approach is used in order to point to the fact that web services can be combined with traditional application programming to produce a fully functionalapplication.

3.7 Code Snıpet

3.7.1 Web Servıce Clıent Makıng Soap Request

```
/*The Mailing list web service client begins here*/
case 'subscribe' : {ini_set("soap.wsdl_cache_enabled", "0"); // disabling
WSDL cache $client = new SoapClient("http://localhost/web_service.wsdl");
$return = $client->subscribe(get_email(), $_GET['id']);
@$client->__soapCall("subscribe", @$email, @$_GET['id']); display_items('Subscribed Lists', get_subscribed_lists(get_email()), 'information', 'show-archive', 'unsubscribe');
break;}case 'unsubscribe' :
{ini_set("soap.wsdl_cache_enabled", "0"); // disabling WSDL cache
$client = new SoapClient("http://localhost/web_service.wsdl");
 $return = $client->unsubscribe(get_email(), $_GET['id']);
 @$client->__soapCall("unsubscribe", @$email, @$_GET['id']);
display_items('Subscribed Lists', get_subscribed_lists(get_email()),
'information', 'show-archive', 'unsubscribe');
break;}/*The Mailing list web service client ends here*/
```

3.7.2 SOAP Server to Handle Incoming SOAP Request from WebService Client

```
<?php
include_once 'mlm_web_fns.php';
ini_set("soap.wsdl_cache_enabled", "0"); // disabling WSDL cache
$server = new SoapServer("web_service.wsdl");
$server->addFunction(array("subscribe", unsubscribe"));
$server->handle();
?>
```

3.7.3 Maılıng Lıst Web Servıce Descrıptıon Language

```
<?xmlversion='1.0'encoding='UTF-8'?>
<definitionsname='mlm'
targetNamespace='urn:mymlm'
xmlns:tns='urn:mymlm'
xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
```

```
xmlns:soapenc='http://schemas.xmlsoap.org/soap/encodin
g/'
xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'
xmlns='http://schemas.xmlsoap.org/wsdl/'>

<messagename="subscribeRequest">
<partname="email"type="xsd:string"/>
<partname="listid"type="xsd:int"/>
</message>
<messagename="subscribeResponse">
<partname="result"type="xsd:boolean"/>
</message>

<messagename="unsubscribeRequest">
<partname="email"type="xsd:string"/>
<partname="listid"type="xsd:int"/>
</message>
<messagename="unsubscribeResponse">
<partname="result"type="xsd:boolean"/>
</message>

<portTypename="mlmPortType">
<operationname="subscribe">
<inputmessage="tns:subscribeRequest"/>
<outputmessage="tns:subscribeResponse"/>
</operation>

<operationname="unsubscribe">
<inputmessage="tns:unsubscribeRequest"/>
<outputmessage="tns:unsubscribeResponse"/>
</operation>
</portType>

<bindingname="mlmBinding"type="tns:mlmPortType">
<soap:bindingstyle="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
<operationname="subscribe">
<soap:operationsoapAction='urn:xmethods-delayed-
quotes#subscribe'/>
<input>
<soap:bodyuse='encoded'namespace='urn:xmethods-
delayed-quotes'
encodingStyle='http://schemas.xmlsoap.org/soap/encodin
g/'/>
</input>
<output>
<soap:bodyuse='encoded'namespace='urn:xmethods-
delayed-quotes'
encodingStyle='http://schemas.xmlsoap.org/soap/encodin
g/'/>
</output>
</operation>

<operationname="unsubscribe">
<soap:operationsoapAction='urn:xmethods-delayed-
quotes#unsubscribe'/>
<input>
<soap:bodyuse='encoded'namespace='urn:xmethods-
delayed-quotes'
encodingStyle='http://schemas.xmlsoap.org/soap/encodin
g/'/>
</input>
<output>
<soap:bodyuse='encoded'namespace='urn:xmethods-
delayed-quotes'
encodingStyle='http://schemas.xmlsoap.org/soap/encodin
g/'/>
</output>
</operation>
</binding>
<servicename="mlmService">
<portname="mlmPort"binding="tns:mlmBinding">
<soap:addresslocation="http://localhost/web_service.php"
/>
</port>
</service>
</definitions>
```

3.8 Sample Outputs

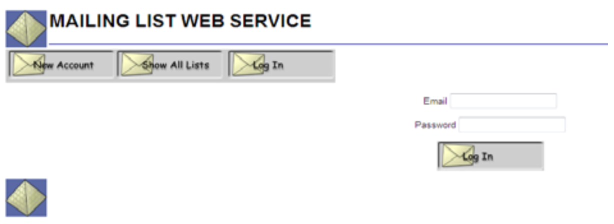
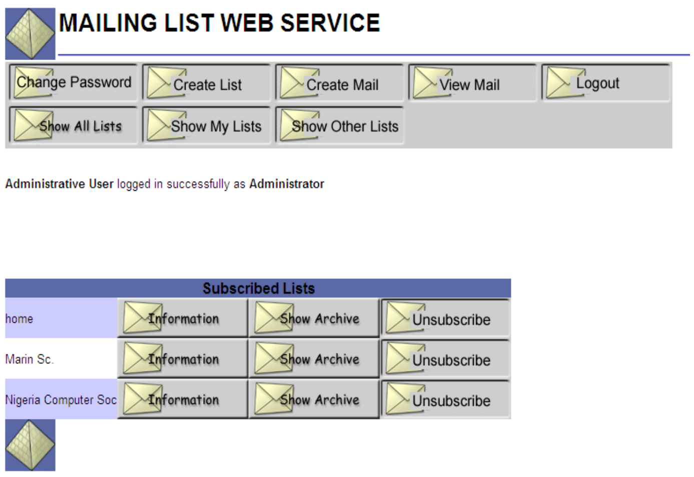


Fig 3.3: Mailing List Manager Homepage

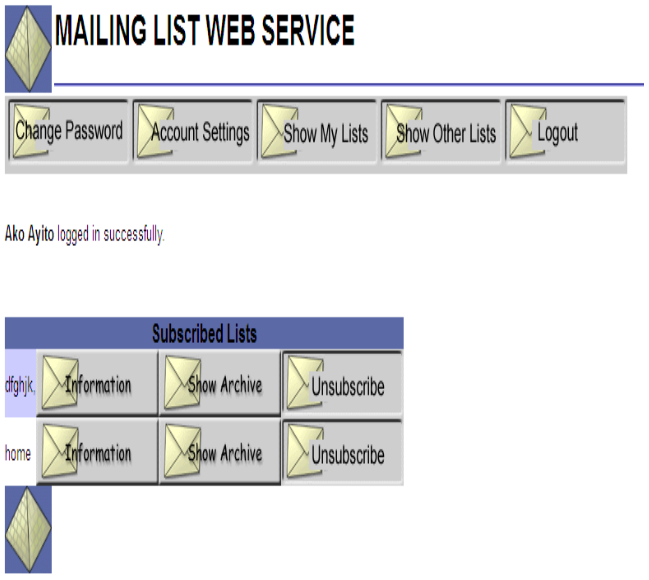
Fig 3.3 shows the homepage of the Mailing List Manager. From here a system user can carry out a few operations including opening a new account, showing all lists or loging in. Fig 3.4 shows the administrator page. Fig 3.5 shows the Mailing List Manager Visitor Page. Fig. 3.6 displays the Mailing List Manager Show All Lists Page while Fig 3.7 shows the Mailing List Manager Show My Lists Page



Administrative User logged in successfully as Administrator

Subscribed Lists			
home	Information	Show Archive	Unsubscribe
Marin Sc.	Information	Show Archive	Unsubscribe
Nigeria Computer Soc	Information	Show Archive	Unsubscribe

Fig 3.4: Mailing List Manager Administrator Page



Ako Ayito logged in successfully.

Fig 3.5: Mailing List Manager Visitor Page

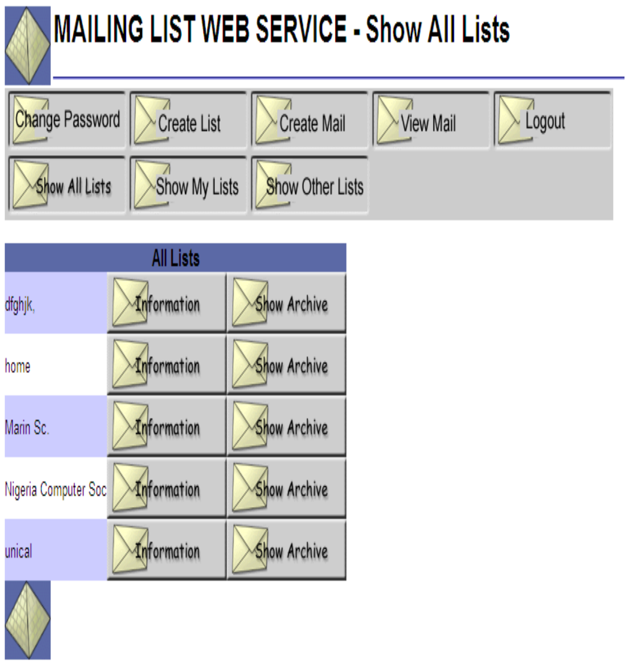


Fig 3.6: Mailing List Manager Show All Lists Page

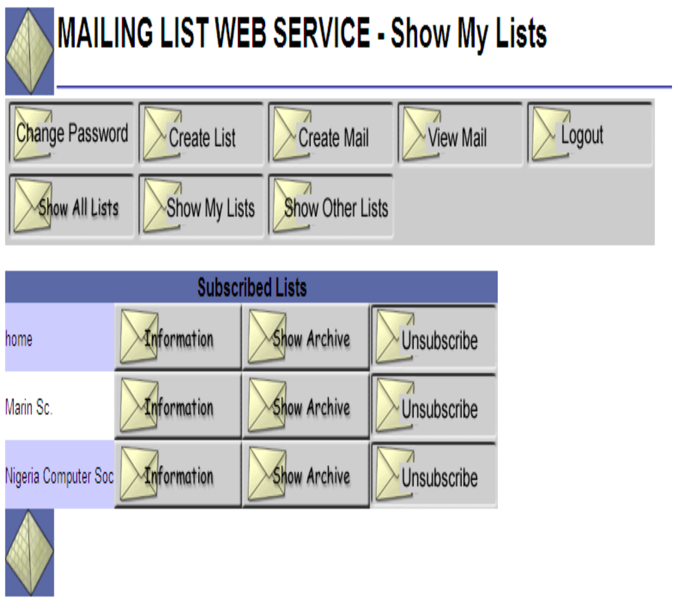


Fig 3.7: Mailing List Manager Show My Lists Page

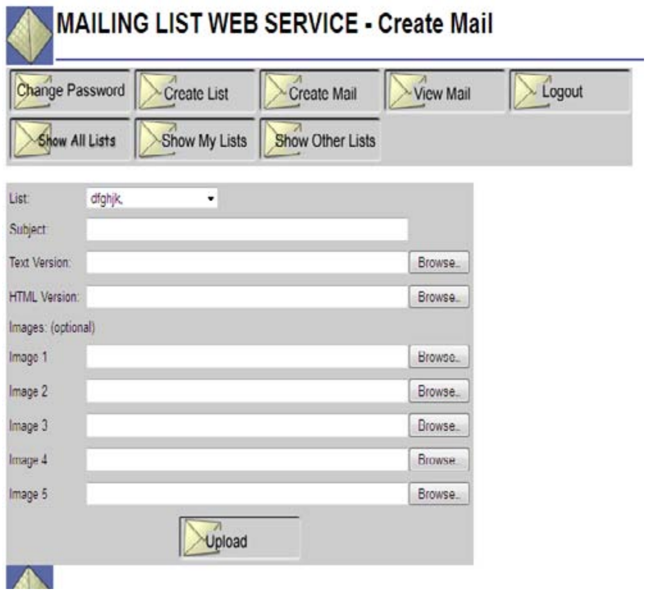


Fig 3.8: Create Mail Page

Fig 3.8 shows the paer where the mailing list web service page allows the user to create a mail page, fig 3.9 shows where all details of the mail pages can be viewed while fig. 3.10 shows the archive page.

VOL. 3, NO. 7, July 2013 ISSN 2225-7217
ARPN Journal of Science and Technology
©2011-2013. All rights reserved.

http://www.ejournalofscience.org

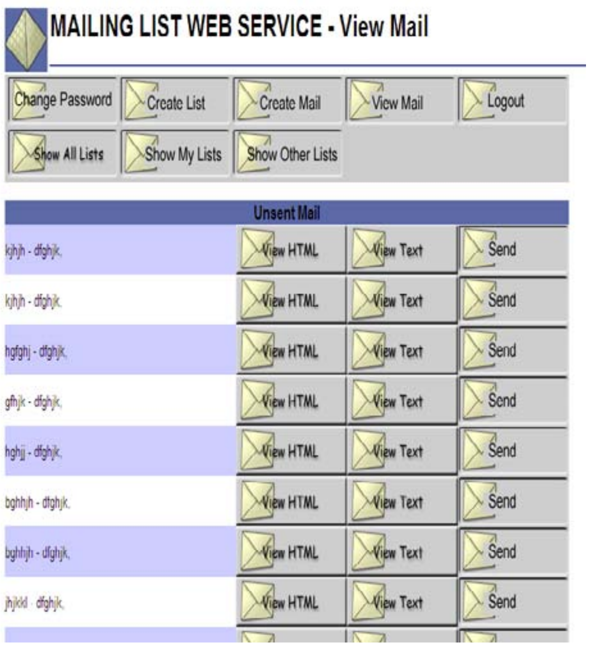


Fig 3.9: View Mail Page

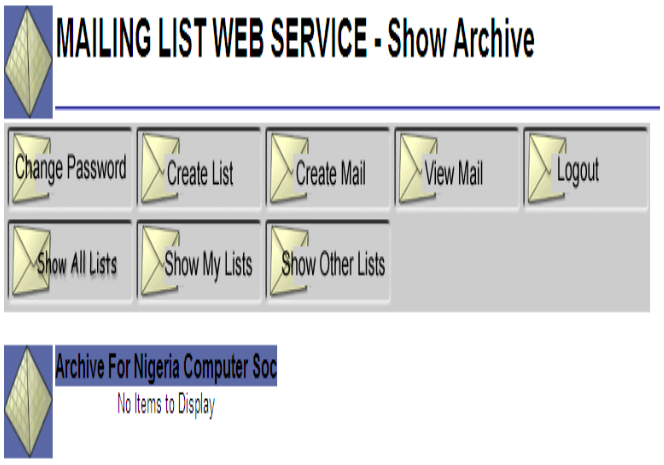


Fig 3.10: Mailing List Manager Show Archive Page

Table 3.2: Documentation

FILENAME	TYPE	DESCRIPTION
index.php	Application	The main script that runs the entire application.
include_fns.php	Functions	Collection of include files for this application.
data_valid.fns .php	Functions	Collection of functions for validating input data.
db_fns.php	Functions	Collection of functions for connecting to the MLM database.
mlm_fns.php	Functions	Collection of functions specific to this application.
output_fns.php	Functions	Collection of functions for outputting HTML.
user_auth_fns .php	Functions	Collection of functions for authenticating users.
web_service.php	Functions	Collection of functions for handling incoming SOAP request from client.
web_service.wsdl	Functions	Webservice description language for the webservice.
upload.php	Component	Script that manages the file upload component of the administrator role-separated out to make security easier.
create_mlm.sql	SQL	SQL to setup up the MLM database and setup a web server and an administrative user.
mlm_web_fns.php	Functions	Web service methods available in the webservice.

4. SUMMARY AND CONCLUSION

This research work has been centered on web services and their underlying concepts. Attempts have been made to analyze the strengths and weaknesses of we services, how to implement web service interface in a non-machine dependent process able format (WSDL).

The mailing list manger implemented in this research work has been used as a medium to show how web services can be consumed. It can be seen that applications with loosely coupled business processes can be developed using web services. Applications can be developed by the combination of object oriented programming and web services and this can be seen at the point where the mailing list manager application consumes the subscribe and unsubscribe functions exposed by the mailing list web services. It should be noted that the subscribe and unsubscribe functions are loosely coupled.A mailing list manager web service was developed from scratch and was deployed using a shared hosting sewer, and consumed by a mailing list manager web application.

4.1 Recommendatıon

In order to make this service rich with functionalities, highly automated and recognized, these are some recommendations:

- Most of its traditional functions should be exposed as web services so as to have a rich collection of functionality.
- Make the service platform independent (the SOAP extension of PHP in this research work works with only windows 7 and windows vista version of windows on the windows platform.
- Create activities logs
- Implementsecurity measures and provide a means of authentication.

REFERENCES

[1] Blackman, R. et al (2002). Ecartis Modular Mailing List Manager.

[2] Coita D. C. (2007) Electronic Mailing List And Internet Forums -Tools For Management and Marketing within Educational Organizations

[3] Connolly, T. and Bogg C. (2010). Database systems: A practical Approach to Design, Implementation, and management. Addison Wesley.

[4] Deitel, P. J. and Deitel, H. M. (2007). Java How to program 7th edition. Pearson Education.

[5] Potts, S. and Kopack, M. (2003). Sams teach Yourself Web Services in 24 Hours. Sam's Publishing

[6] Richards R. (2006). Pro PHP, XML and Web services. Après.

[7] Snell, J. et. al (2001). Programming Web services with SOAP. O'Reilly.

[8] Wikipedia (2013). Electronic Mailing Lists. Retriieved on June 6 ,2013 from https://en.wikipedia.org/wiki/Electronic_mailing_list