

Enhanced Email Attachment Reminder (Ear) Plugins

AJAYI, Olusola Olajide

Department of Computer Science, Faculty of Science,
Adekunle Ajasin University, Akungba-Akoko, Ondo State

olusola.ajayi@aaua.edu.ng

+2347056433798

&

AKINDELE, Temitope Clement

Department of Computer Science, Faculty of Science,
Adekunle Ajasin University, Akungba-Akoko, Ondo State

topson47online@gmail.com

+234806 223 3145

ABSTRACT

The need for an attachment reminder system prompted this research work, looking at the existing email reminder system in Gmail and Outlook, its limitations does not make the system to work in an efficient way. We therefore, developed an enhanced system for email attachment reminder which will rectify and solve certain imperfection noted in the attachment reminder of some email service provider that are using attachment reminder in their email system using the keyword matching methodology. The system can be highly useful through the uses of the instant reminder as well as the reminder alert that popup immediately send button is clicked, assisting user to remember attaching a file and to some extent save user from embarrassment the forgotten attachment cause email users most times. The application is web- based and allows all tasks to be carried out using the personal computer and the Internet. The implementation of email attachment reminder system was done using PHP and JavaScript as frontend and MySQL for the database design.

Keywords: Email Attachment, Reminder, Attachment Detector, Forgotten Attachment, Keyword, Text Editor, Email Client.

1. BACKGROUND TO THE STUDY

Email is one of the most popular forms of communication today. The acceptance of this communication medium is best exemplified by the sheer number of current users. This form of communication has the simple advantage of being almost instantaneous, intuitive to use, and costing virtually nothing per message.

One of the many features of an email is an Attachment Reminder that checks to see if a user have not attached a file to a message, it will provide the user with a prompt, and warn the user before the message is sent; to enable the user to attach necessary documents to its message before sending it. This have save a lot of email users from embarrassment and wasting of time and resources caused by forgotten to attach a document.

Historically, the term *electronic mail* was used generically for any electronic document transmission. For example, several writers in the early 1970s used the term to describe fax document transmission. As a result, it is difficult to find the first citation for the use of the term with the more specific meaning it has today.

Most commonly referred to as email or e-mail since 1993, is a method of exchanging digital messages from an author to one or more recipients. Email may operate across the Internet or other computer networks (Shlomo, 2006). Some early email systems required the author and the recipient to be both online at the same time, in common with instant messaging.

Today's email systems are based on a store-and-forward model. Email servers accept, forward, deliver, and store messages. Neither the users nor their computers are required to be online simultaneously; they need connect only briefly, typically to a mail server, for as long as it takes to send or receive messages.

2. STATEMENT OF PROBLEM

The major purpose of email system is for information sharing. Sharing information through the email system can come in different ways like sending and receiving of plain text prepared by a text processing layer and this message may likely include an attachment, such as a word processing document, a spreadsheet, a photograph, a scan, a music file, a video file, etc. In an environment wherein electronic messages may be sent with or without an attachment, Email has seen explosive growth of usage in the last twenty-five years. As such, there are issues and problems facing a system that has not kept abreast of technological developments in which forgotten attachment reminder is one.

In determining whether an electronic message suggests that there should be an attachment, the message having a subject line and a body, the method comprising: inspecting a message to determine whether the message already has an attachment; if the message does not already have an attachment then inspecting the subject line of the message or the body of the message to determine a likelihood that the message should have an attachment; and if the subject line of the message or the body of the message indicates that the message should have an attachment then providing a suggestion to a user that the message may need an attachment.

However, inspecting the subject line of the message comprises: searching the subject line of the message for presence of at least one predetermined keyword, if at least one predetermined keyword is present, then providing the suggestion, and if no predetermined keyword is present, then inspecting the body of the message, and wherein inspecting the body of the message comprises: parsing the message to identify at least one sentence within the message, inspecting the at least one sentence for at least one of presence of at least one predetermined word, for presence of at least one statistically significant phrase, indicating the likelihood that the message should have an attachment, and the user provide the attachment.

The literature reviewed as well as common e-mail applications that have the forgotten e-mail attachment plugin proved that many of such are designed with limited or predefined keywords. Mark Dredze et al (2006) in his reviewed work however submitted that, some new enhanced framework can be proposed on the basis of the work already done by researchers in the area of forgotten attachment reminder plugins.

Consequent upon this, we shall look into some of these other methods and keywords that could enhance forgotten attachment reminder and as observed in Gmail and Outlook, the major keyword used is attach/attached. This study therefore seeks to incorporate other keywords into the detector database e.g. enclosed, file, append, embed, find etc.)

3. OBJECTIVE

The study seek to achieve the following objectives To develop an email attachment reminder system that will be reducing email composition errors by reminding the user of possible attachments. To present a high precision classifier that could trigger a warning when an outgoing message is missing a needed attachment.

4.0 RELATED LITERATURES

Method of Email attachment confirmation by Larky et al (2005) find a method and system for confirming that attachment are made to relevant email messages before they are sent. The employ database comparison approach techniques will work only when the sender is sending message to email in "attachment-prone" database.

In 2006, Mark Dredze et al used logistics regression for email content classification method which includes numerous features for email classification and before a message is sent, the system will be able to check whether or not a document is attached to the message but in a situation where an email did not directly reference attachment in its content, the system will be unable to detect if an attachment is forgotten.

Marco et al (2001) in their work address how to recognize missing email attachment by using machine learning. The work combine both Naïve Bays' algorithm and Graham algorithm and uses conventional machine learning techniques to classify the content of an email message.

Brun et al (2013) use linguistically based detector that looks for a given set of keyword pattern. The study was based on email text recognizing natural language. But, triggering set of keywords may have to be defined explicitly within a list of the user and this may involve entering all inflected forms of the keyword.

A method of detecting and reminding an email user of forgotten attachment using "regular expression" was developed by Caldwell et al (2014). The method employ is analysing an email message layer using regular expression analysis, a tokenizer and unique body identification. The solution was using a list of predetermined keyword, and phrases to identify words in an email message indicating that a message should have an attachment. But the detector could only be triggered if some set of predetermined keyword is found in the message of the email.

However, in this study, we shall look to enhance and improved on the work done so far by other authors and as noticed in Gmail and Outlook, the keyword used in implementing the forgotten attachment system is "attach/attached". Therefore this study will include more keyword like find, document, see, file, etc. for the system to be able to detect effectively and detect forgotten attachment appropriately.

5. METHODOLOGY

In order to achieve the aim and objective of this research work, we adopted the Keyword matching methodology approach. This methodology will help to maintain a list of keywords which are indicative of an email that requires an attachment. A specified algorithm then simply walks through all the words

5.1 Design of the Proposed System

Systems design is concerned with how to provide an efficient (economical) and effective (relevant and useful) system. It is an integral part of software development process which is embarked upon after a detail analysis of the system has been done, and the project feasibility study undertaken.

The various design specifications of the new system are examined. The functional and non-functional requirements of the proposed system have been analyzed and outlined. This forms the basis for the design specified in this research work. The design is divided into various parts which include interface design, database design and logic design.

The major goal of the design of this system is to design a system that fulfils requirements and at the same time, be friendly to the end-users.

Below entails the design for the proposed Email Attachment Reminder Plugin. The designs, which entail the interface design, database design, and logic design, were specified using appropriate design tools

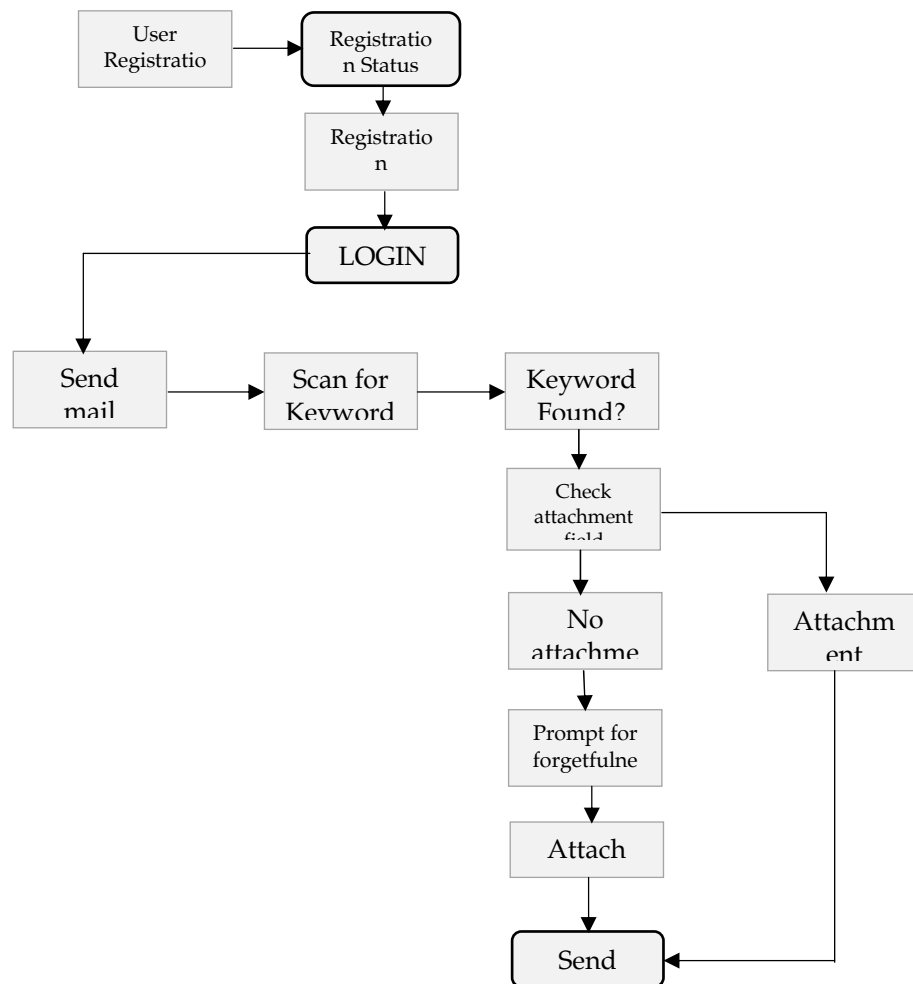


Fig 1: Flow Representation of the System
Source: Author

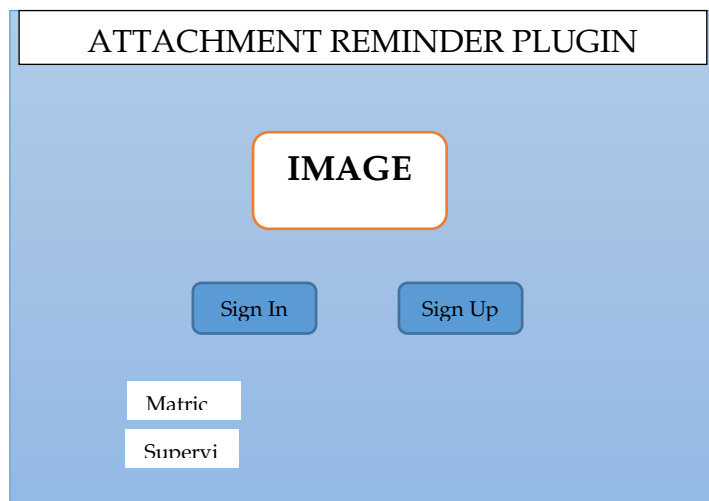


Fig 2: Interface design of the system
Source: Author

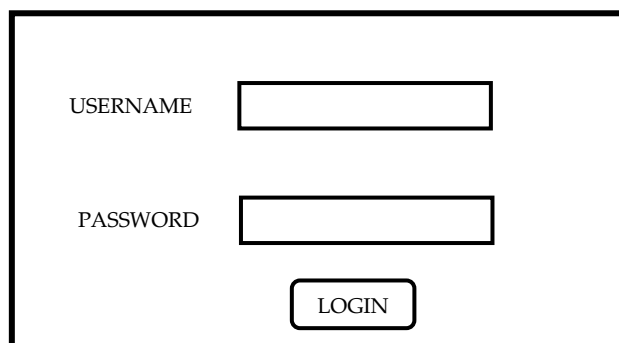


Fig 3: Login page prototype
Source: Author

6. PROPOSED SYSTEM MODELS

A model is a representation of an actual or something that is real in terms of system. In terms of system, it is the representation of actual system to be developed.

Model is the investigation of a system of interest under study. It can also be described as the pattern used to construct a system of interest. A model shows a picture of the construct of that system. According to the Object Modelling Group (OMG), “modelling is the designing of software applications before coding.”

In model-based software design and development, software modelling is used as an essential part of the software development process. Models are built and analyzed prior to the implementation of the system, and are used to direct the subsequent implementation.

This project work is modelled using combination of different type of models that best suit the concept. (Structural Modelling, Contextual/Conceptual Model, Behavioural Modelling and Architectural Modelling)

6.1 Conceptual Model of the Proposed System

Conceptual model is used to represent events/computing operations in accordance with the structure or nature of such events. The conceptual representation of the new proposed system is as shown below.

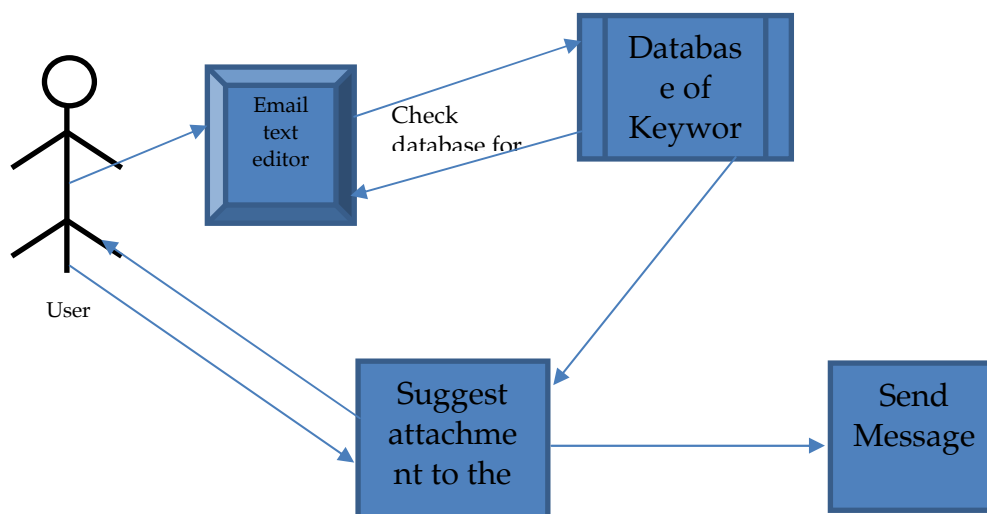


Fig 4: Conceptual model of the system
Source: Author

6.2 Used-Case Model

Used-Case model use to represent/model interactions between the system and its immediate or outer environments (i.e. internal and external). The diagram below show the Use-Case representation of the propose system.

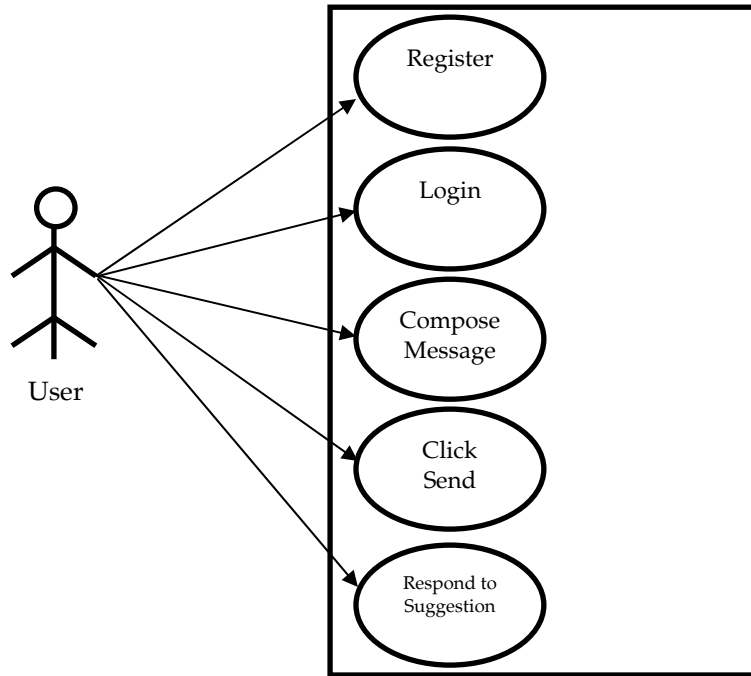


Fig 5: Used-Case model of the system
Source: Author

6.3 Process Model

This shows the representation of the activities happening between the user and the proposed email attachment reminder system.

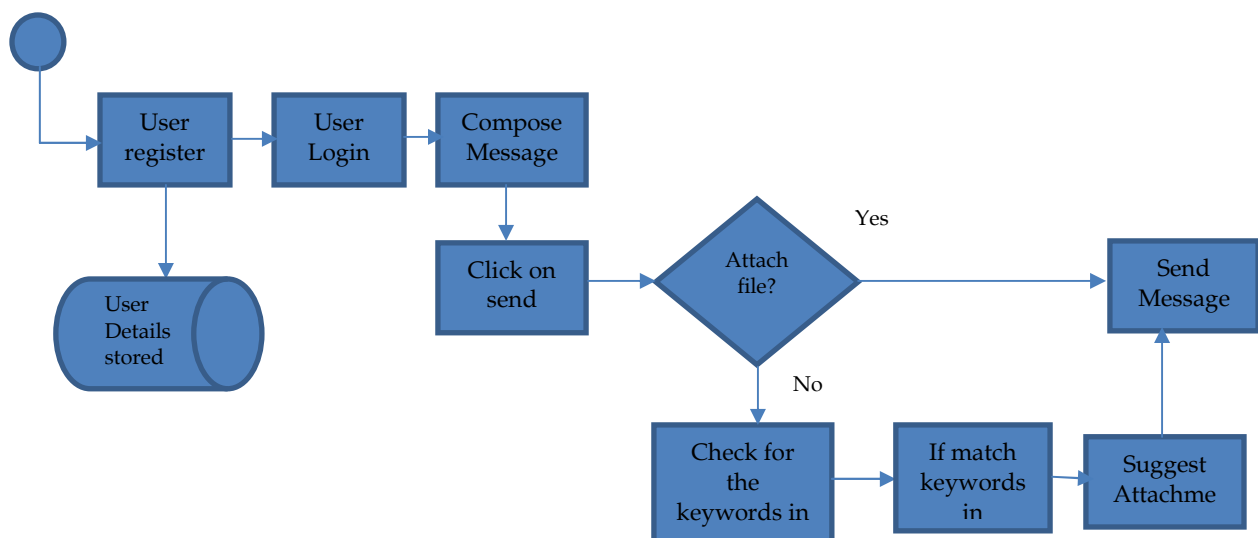


Fig 6: Process model of the system
Source: Author

7. IMPLEMENTATION APPROACH AND PROCEDURE

The implementation approach to be adopted for this study would be web-based development approach. For this, PHP will serve as the application interface, while MySQL will be the backend. Other tools for the implementation include HTML and JavaScript, for the scripting, CSS, for the front-end, and MS-Visio for the system modeling.

This study adopted the waterfall system development procedures/steps; spanning from analysis to testing and operation. Waterfall model is the implementation procedure used for this system. The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of requirement analysis and definition, system and software design, implementation and unit testing, integration and system testing, operation and maintenance. This model is highly structured, easily understandable, explainable and more disciplined.

8. EVALUATION RESULT/OUTPUT

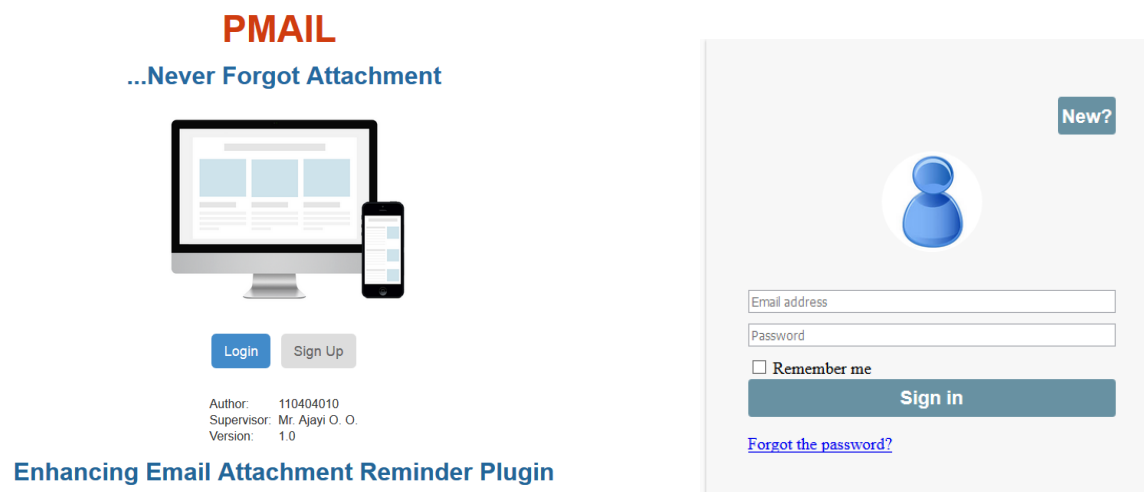


Fig. 7: Output Interface 1

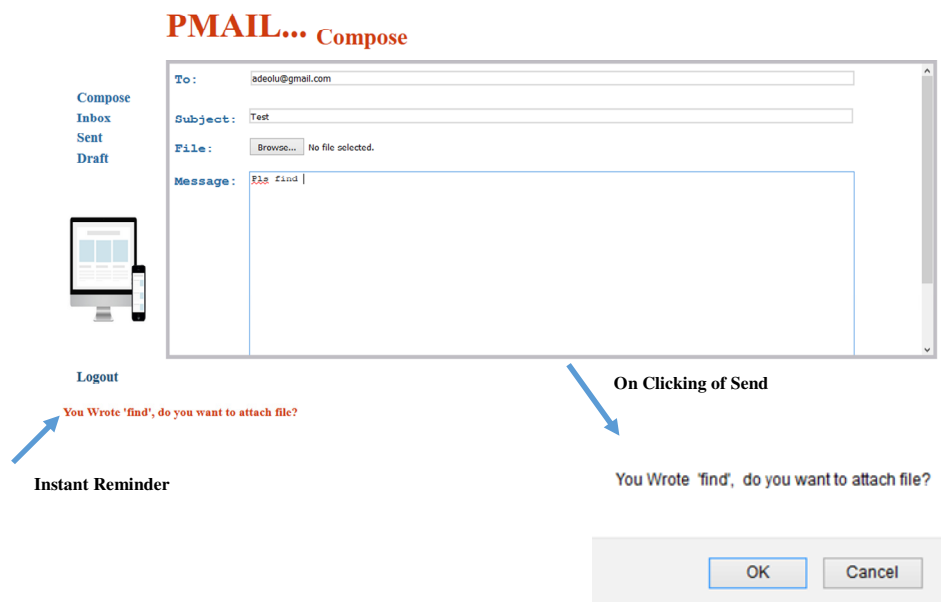


Fig. 8: Output Interface 2

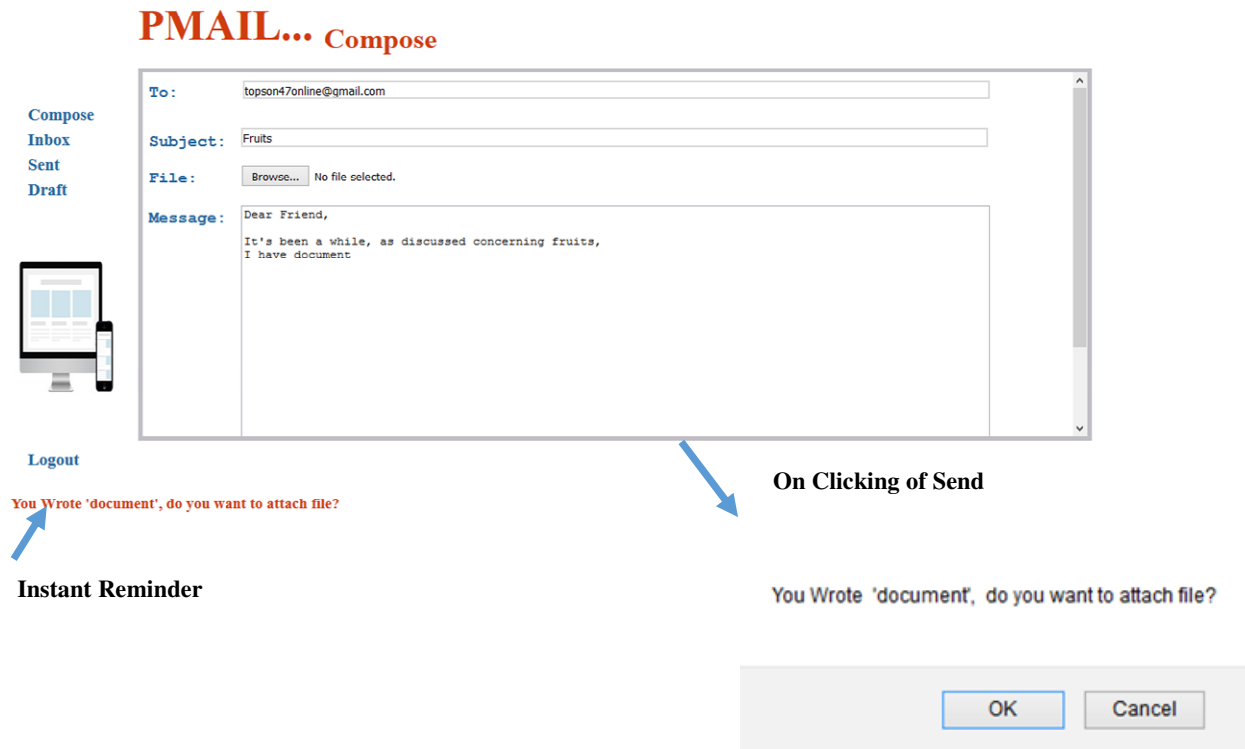


Fig. 9: Output Interface 3

5. CONCLUDING REMARKS

This research work covered an enhanced system for email attachment reminder which is to rectify and solve certain imperfection of the attachment reminder of some email service providers that are using attachment reminder in their email system. Email attachment reminder system application can be highly useful in assisting user to remember attaching a file and to some extent save users from embarrassment that do most times comes with forgotten a necessary/important attachment.

The new system is highly effective, efficient, simple, user friendly, and reliable; which offers more to email users and to email service provider at large.

6. RECOMMENDATION

Having seen the importance and the need to have a more enhanced email attachment reminder, it is therefore important to develop an attachment reminder system that will work and use/rely on more keyword other than "attachment" to function. This project work will therefore help big companies that provide email services like Gmail, Outlook, Yahoo etc. and provider of webmail services to have basis for developing and integrating a more robust email attachment reminder system into their email system and also to give email users another reason to use their email service.

This research work will also provide email user instant reminder of adding attachment to their message as they are typing and provide another reminder alert immediately they click the send button.

11. Future Work

Effectiveness and efficiency of this new system gives room for further research study. We suggest therefore that further research study be done on implementing this system on another platform (e.g. mobile) and making the new system to detect keyword on multilingual basis will be a welcome and improved approach.

Inclusively, adding more keyword to the database could be made possible through the admin panel and not by opening the backend database before adding the keywords, this will make the application to be more robust.

REFERENCES

1. Adithya, G. (2007). *End user transparent email attachment handling to overcome size and attachment policy barrier*: UnitedState Patent Application Publication, WO2007133504 A2.
2. Brain, M., & Tim Crosby. (2007). *How E-mail Works*: HowStuffWorks.com, <<http://computer.howstuffworks.com/e-mail-messaging/email.htm>> 208 July, 2015.
3. Brownlow Mark. (2009). *Email and webmail statistics*: Email Marketing Reports, January 2009.
4. Brun, S, et al., (2003). *Normalization and paraphrasing using symbolic methods*: ACL: Second International workshop onParaphrasing, Paraphrase Acquisition and Applications.
5. Brun, S. et. al. (2013). *Linguistically enhanced email detector*: United State patent application publication, US8429741 B2.
6. Caldwell, et. al. (2014), *Forgotten attachment detection*: United State Patent Application Publication, US2014/0012923 A1.
7. Chanod, et al. (1995). *Tagging French text: tagset, dedicated lexicons and guesser*: Proc. From Texts to Tags: Issues inMultilingual Language Analysis, EACL SIGDAT workshop.
8. Carl, et al (2014) *Extending alignments with k-mismatches and l-gaps*. Theoretical Computer Science pp. 525, 80-88.
9. Chrabaszcz, M. N. (2000), *Electronic mail attachment verifier*: United State Patent Application Publication, US6073133 A.
10. Elleithy Khalid. (2008). *Innovation and advance techniques in systems, computing science and Technology*.
11. Ehsan Sherkat (2014) *A new approach for multi-pattern string matching in large text corpora*. 7th International Symposiumon Telecommunications (IST'2014), pp. 72-77.
12. Jun-Ichi Aoe. (1994). *Computer algorithm: string pattern matching strategies*: Computer Society Press, ISBN 987-0-8186-54622-6. June 1994, Wiley-IEEE Computer Society Press.