

# Development Of An Automated Mail Management And Tracking System For Tertiary Institutions, A Case Study Of University Of Uyo

**Akpasam Joseph Ekanem<sup>1</sup>**

Department of Electrical/Electronic Engineering,  
Akwa Ibom State University Mkpata Enin, Akwa Ibom  
State, Nigeria

*gentlejay@yahoo.com*

**Obinna Chukwuma Joseph<sup>2</sup>**

Department of Information Technology Imo  
Polytechnic Umuagwo, Imo State Nigeria

**Eze Irene F.<sup>3</sup>**

Department of Computer Science Imo State  
Polytechnic Umuagwo Imo State Nigeria

**Abstract**— In this paper, an automated platform for managing and tracking mails of all categories of students, staff and people outside the University community in a seamless and interactive manner is presented. The design technology adopted for the implementation is a 3-tier architectural technology, with HTML, CSS and JavaScript as the client technology, PHP as the server technology, and MYSQL as the database. The system was locally hosted using Apache web server. The software development methodology adopted is a participatory incremental process model (PIP Model). The top down stepwise functional decomposition, flowchart and dataflow diagram of the system are provided to explain the major functionalities of the system. Screenshots of various system functionalities from the test run of the functional system are also presented. This system facilitates efficient service delivery and provides added advantage in academic records management.

**Keywords**— *Mail Management, Mail Tracking, Interactive System, Web Application, Auto Responder, Software Process Model*

## I. INTRODUCTION

Development of an automated mail management system (MMS) is to replace the current paper records [1]. While the Universities in developed countries already have a well-established administrative management system, some of their counterparts in the developing countries like Nigeria still adopt the

manual or semi-automated method which provides only some basic elements of mail management and is time consuming, demanding and are often prone to varieties of errors and disaster[2,3,4,5]. Majority of MMS do not have web interfaces and others do not integrate well with the other University wide information system [1]

In this paper, a sample MMS is presented. Department of Electrical/Electronic and Computer Engineering, University of Uyo is used as a case study. The system functional decomposition, flowchart and dataflow diagram is presented. This will afford the stakeholders in the University the opportunity to participate effectively in the development process. Numerous screenshots of the developed MMS are presented to further demonstrate the possible interface of the system.

## II. METHODOLOGY

### A. PARTICIPATORY INCREMENTAL SOFTWARE PROCESS MODEL

The software development process model adopted in this research is the participatory incremental process model (PIP Model) [ 6,7,8,9,10]. This methodology has been adopted because of the nature of the work which requires incremental development and deployment of software increment through an iterative process as presented in Figure 1. This paper is tailored to organize and manage mails in the various departments/units in the University and mails from outside the University using Electrical/Electronic and Computer Engineering department as a case study.

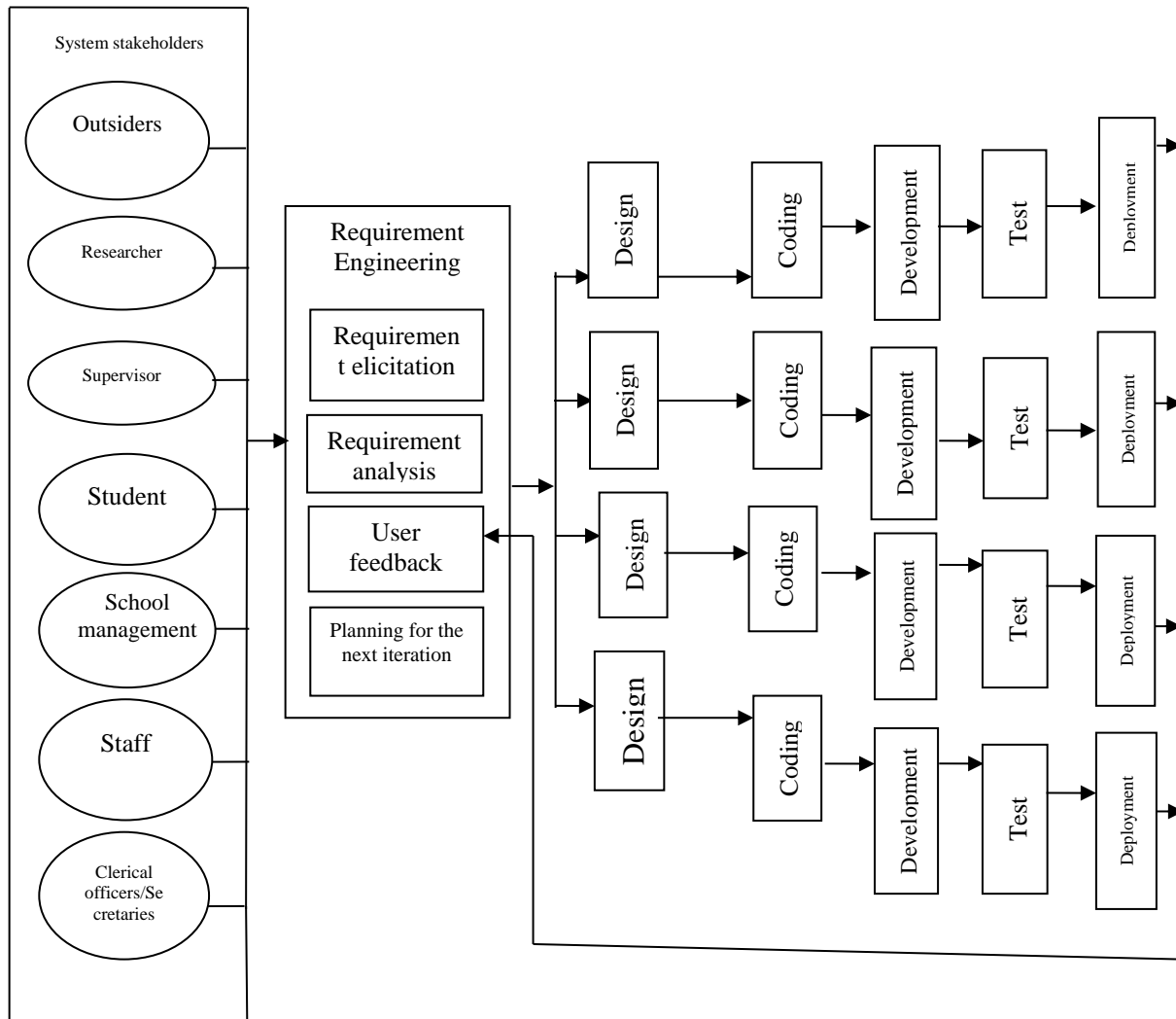


Figure 1 Participatory Incremental Process Model (PIP Model)

The system stakeholders' include the software developer, supervisor, students, clerical officers, school management, staff and people outside the university. The PIP model begins with requirement Engineering and Participatory Requirement Engineering Process (PREP) Model was adopted for this work. The PREP Model as presented in Figure 2 embodies all the activities carried out by the developer in understanding, eliciting, analysing, validating and properly replicating the user's mental model of the system. Interviews, questionnaires, stakeholders' group discussion, social analysis, evaluation and inspection of documents are used in the elicitation requirement stage. The requirements are analysed to ensure that the requirement specification is free of inconsistencies, omission, ambiguity and redundancy. Conflicts in the system were reconciled through negotiation.

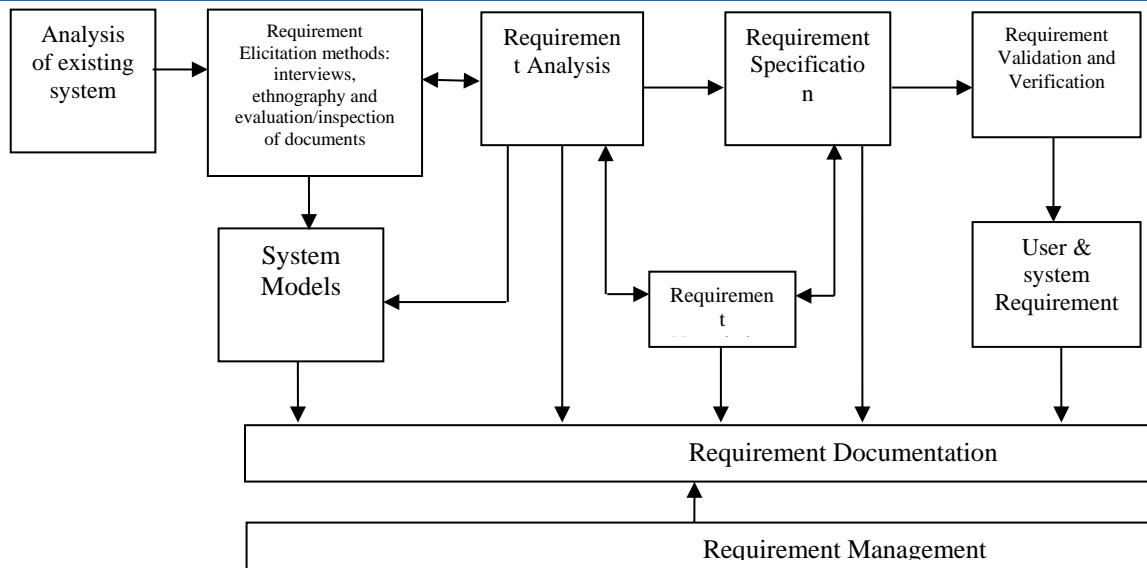


Figure 2 Participatory Requirement Engineering Process (PREP) Model

**B. FUNCTIONAL DECOMPOSITION OF MMS**

The functional decomposition [11, 12, 13, 14, 15] of the mail management system is presented in Figure 3 and the modules are further decomposed into its respective sub-modules as shown in the subsequent figures.

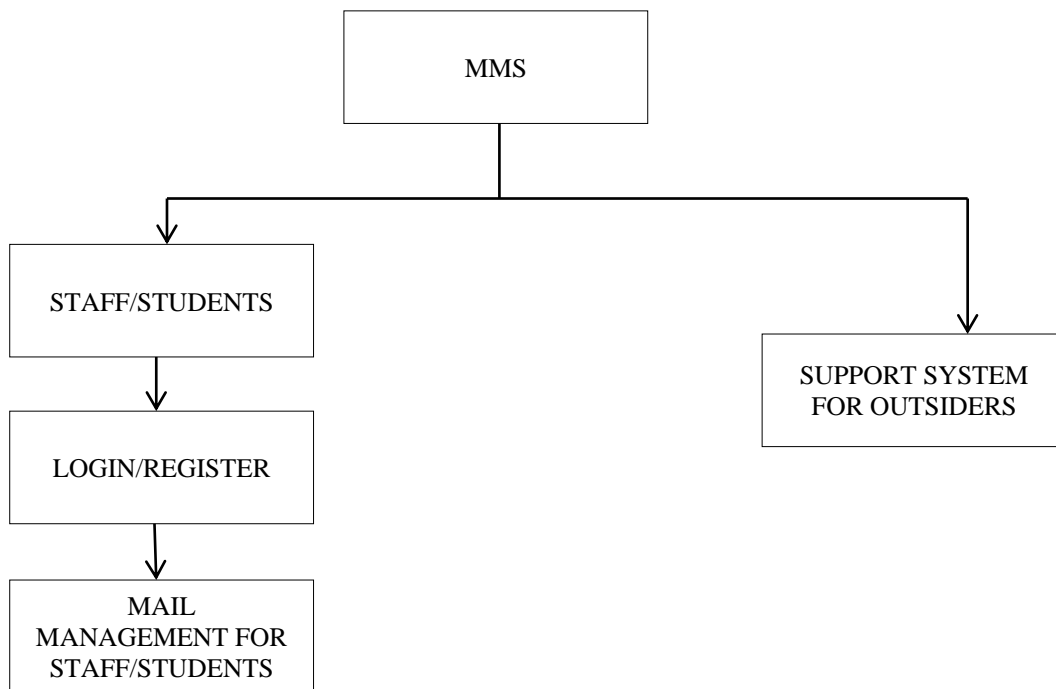


Figure 3 Functional decomposition of MMS

As presented in Figure 4, the system can be accessed by students, staff and people outside the University. The staff/students must first register on the system (by obtaining registration code from the system admin upon verification of his/her student/staff status) and then logon to access mail dashboard. The staff access privileges depending on the role the staff plays in the system. Clerical staff and secretaries can manage incoming and outgoing mails in the system. Other staff and students can only send and track mails. Details of how outsiders access the system is explained in section 2.5 of this paper

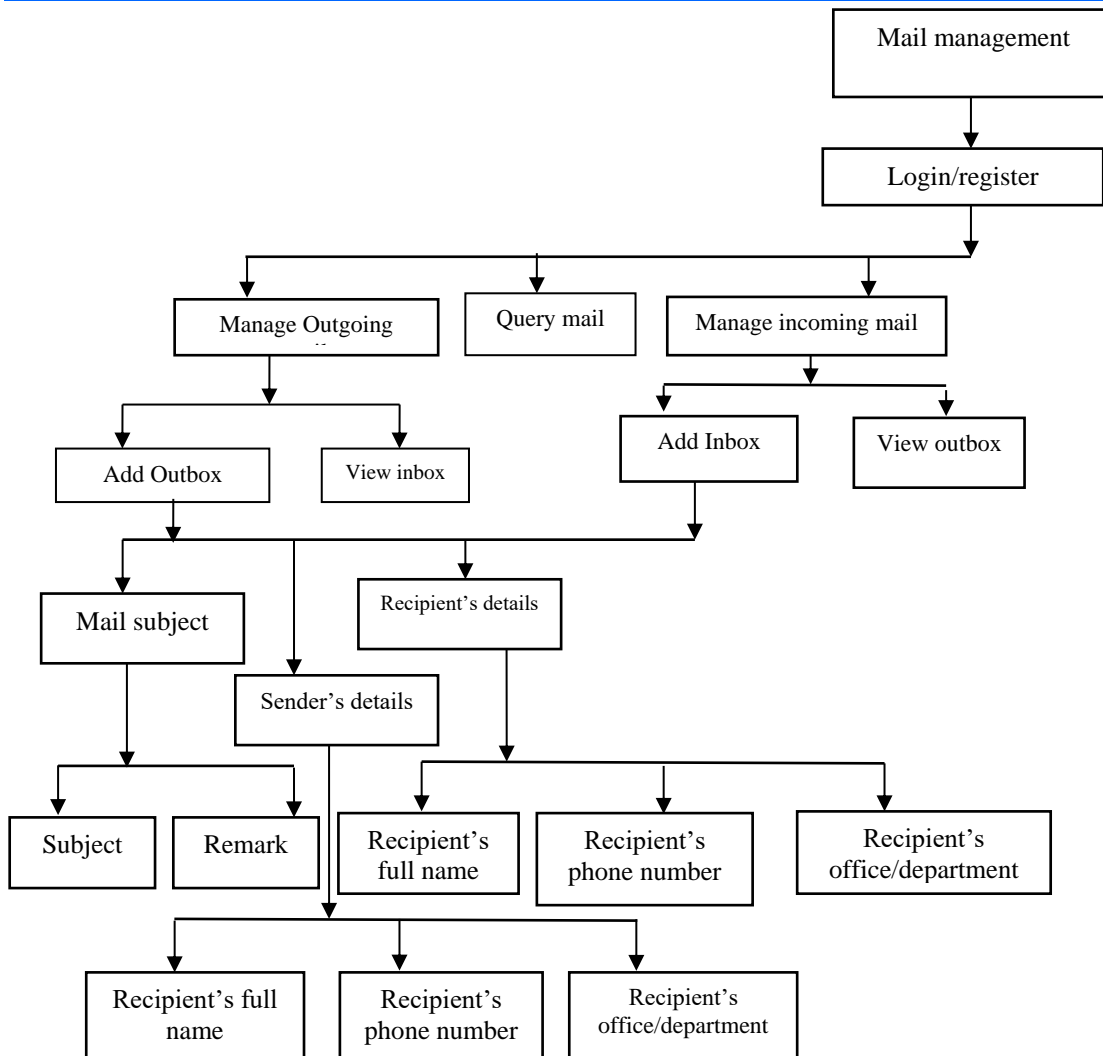


Figure 4: Mail management sub system

The query mail sub-system as presented in Figure 5 uses internal search engine function to track mail status using the mail subject, sender's name, recipient's name, date or remark.

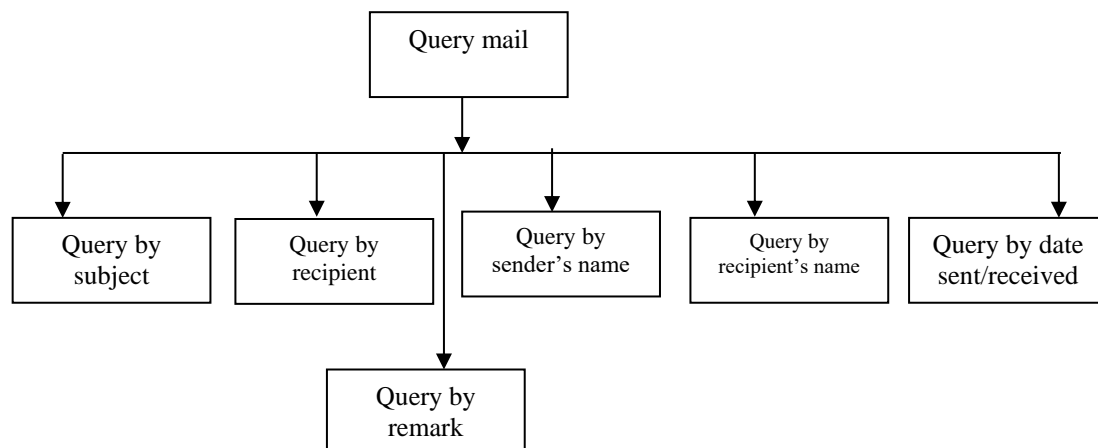


Figure 5: Query Mail Sub-system

### C. FLOWCHART OF THE MAIL MANAGEMENT MODULE

As presented in Figure 6, the clerical officer or secretary receives a mail request. If it is a new incoming mail, he will record the mail details by clicking on "Add Inbox". Once the mail is ready to be forwarded to another unit or department, he will click the "add outbox" option to record outgoing mail details. If the clerical officer or secretary wishes to track mail status at any time, he will use the "query mail" option.

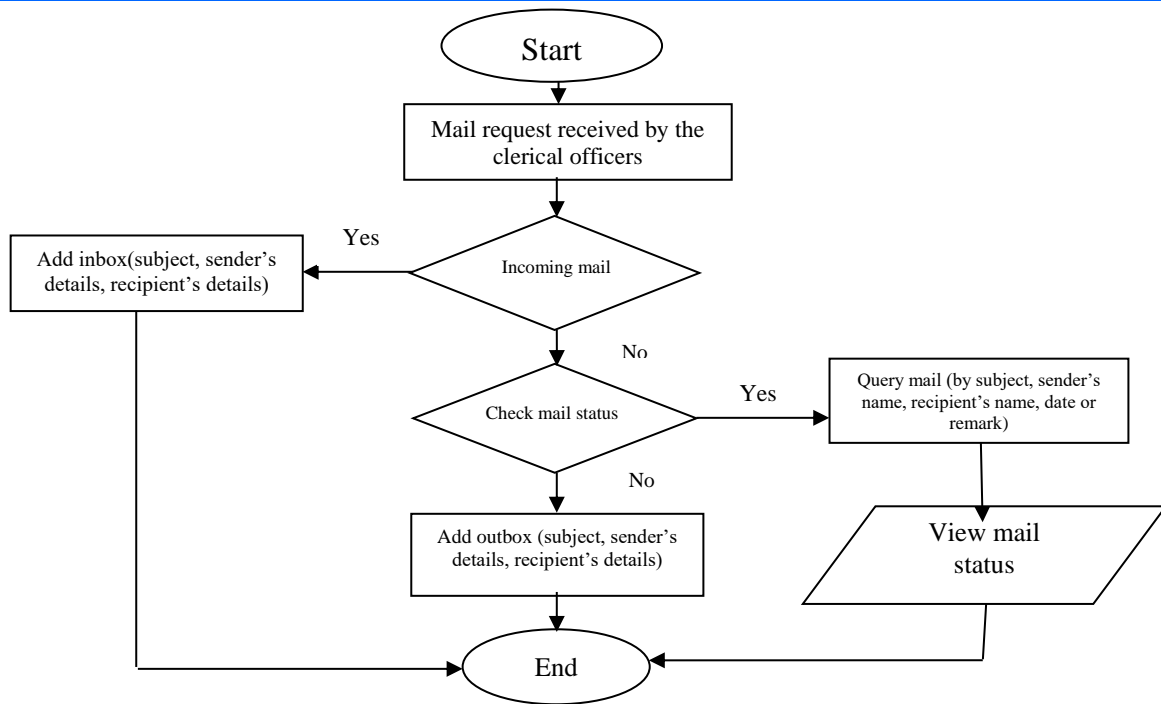


Figure 6: Flow chart of the proposed mail management system

**D. DATAFLOW DIAGRAM OF MMS TRACKING MODULE**

The DFD presented in Figure 7 illustrate the data flow in the system when staff or students' in the department sends mail to the Vice Chancellor (VC) through the Dean of the faculty, through the Head of Department (HOD). The departmental secretary or clerical officer receives the mail from staff or student and records the mail in the mail management system. When the mail leaves the Department to the Dean, the Dean's Secretary will use the automated tracking ID generated for the mail at the department to update the status of the mail. Also as the mail leaves the Dean's office to the VC's office, the status of the mail is updated by the VC's Secretary. The student or staff can track the status of his/her mail from anywhere using either a mobile phone or computer system that is connected to the internet.



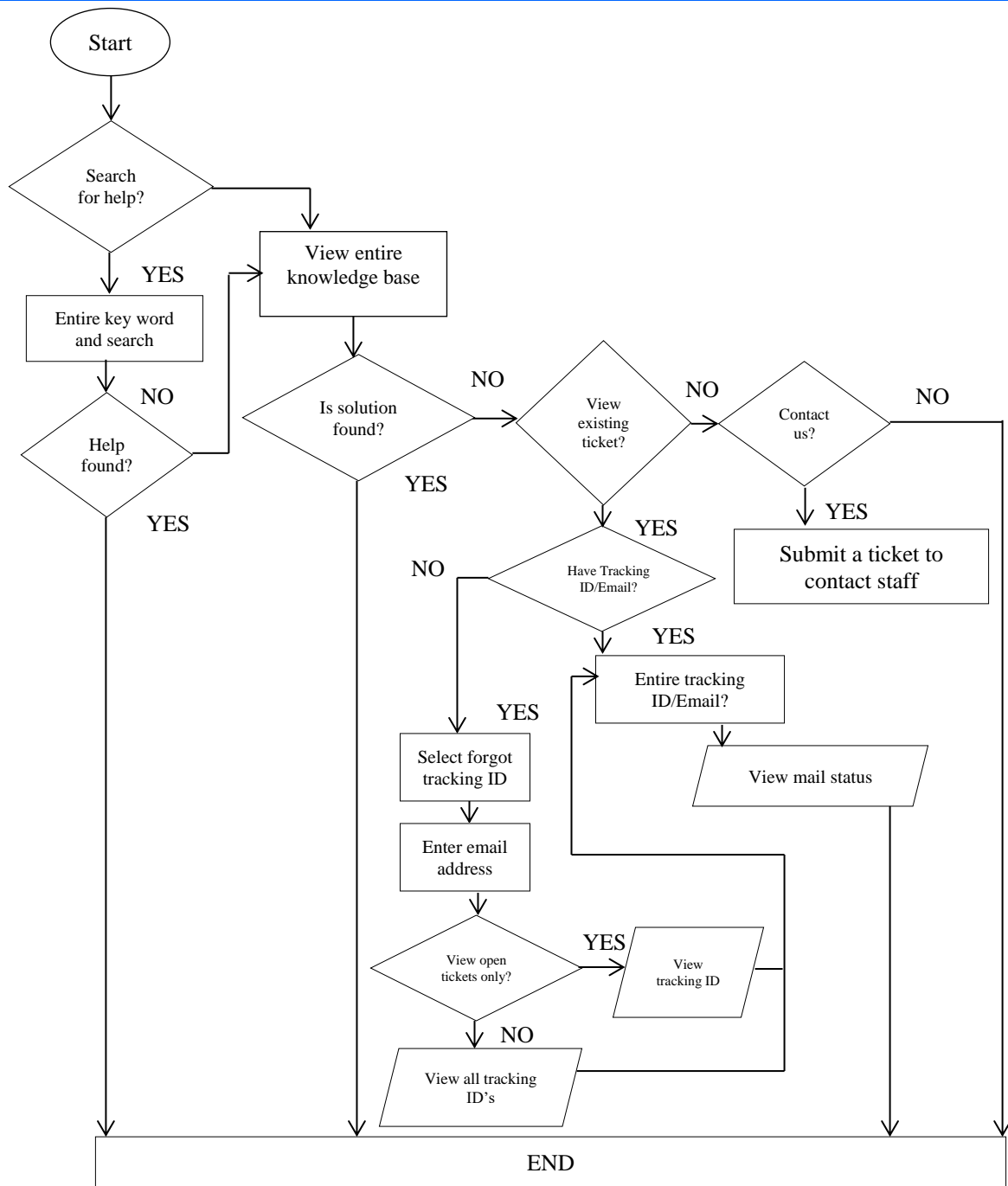


Figure 8 The flowchart for the support module

### III. RESULT AND DISCUSSION

The screenshot in Figure 9 shows how a staff can track the status of his/her mail by first logging into his/her account and clicking on “my mail”. Once the mail sub-module opens, the staff must enter his/her mail tracking ID to track the status of his/her mail as shown in Figure 9. The mail tracking id is generated when a staff of student submits mail.

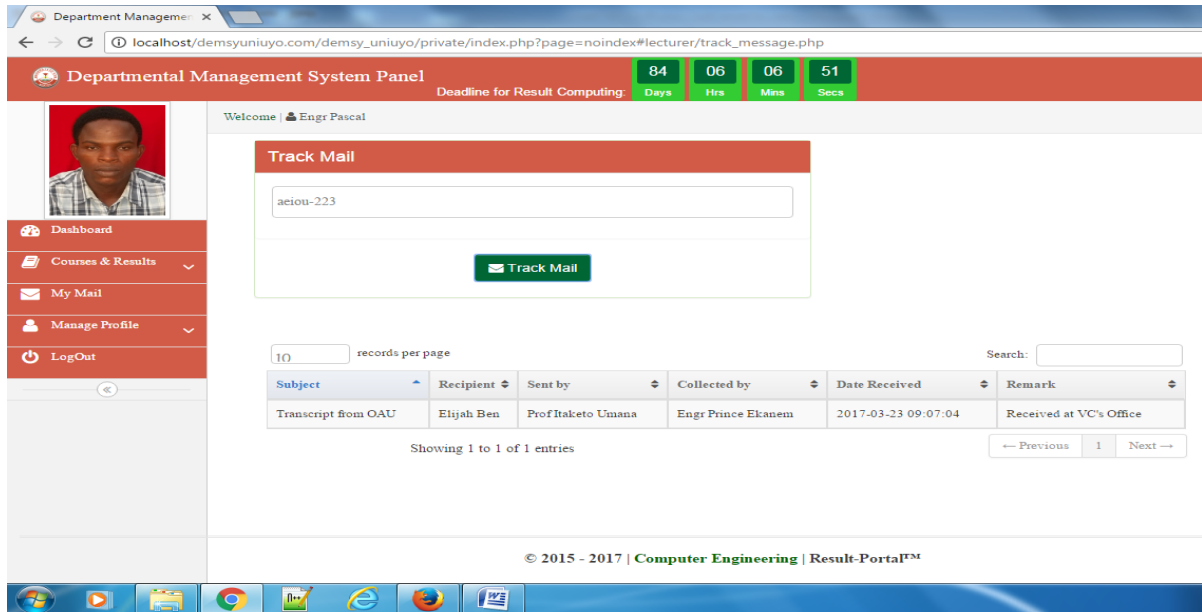


Figure 9: Staff mail tracking sub-module

#### A. CLERICAL OFFICER OR SECRETARY

Figure 10 shows how clerical officer or secretary manages mail. The inbox represent the incoming mails while outbox represent the outgoing mails.



Figure 10: Manage mail

Figure 11 depicts how the clerical officer or admin manages incoming mails. The incoming mail box can be queried using the search box shown in Figure 11.



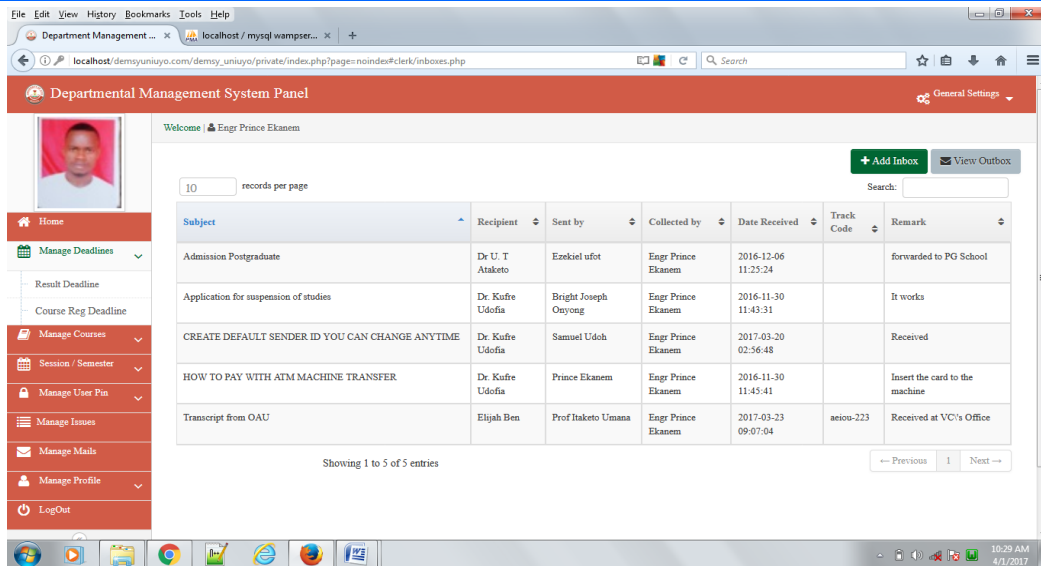


Figure 11: Managing incoming mails

Figure 12 shows how the clerical officer or secretary manages incoming mails. The clerical officer or secretary clicks on “add inbox” and will be presented with two options to either track mail received from other units/departments or record a new incoming mail. If the mail received is coming from another department/unit and have been recorded in the system, the clerical officer or secretary will enter the tracking id in the space provided and click “track”, then the mail will open. He/she can update the status of the mail received as shown in Figure 13.

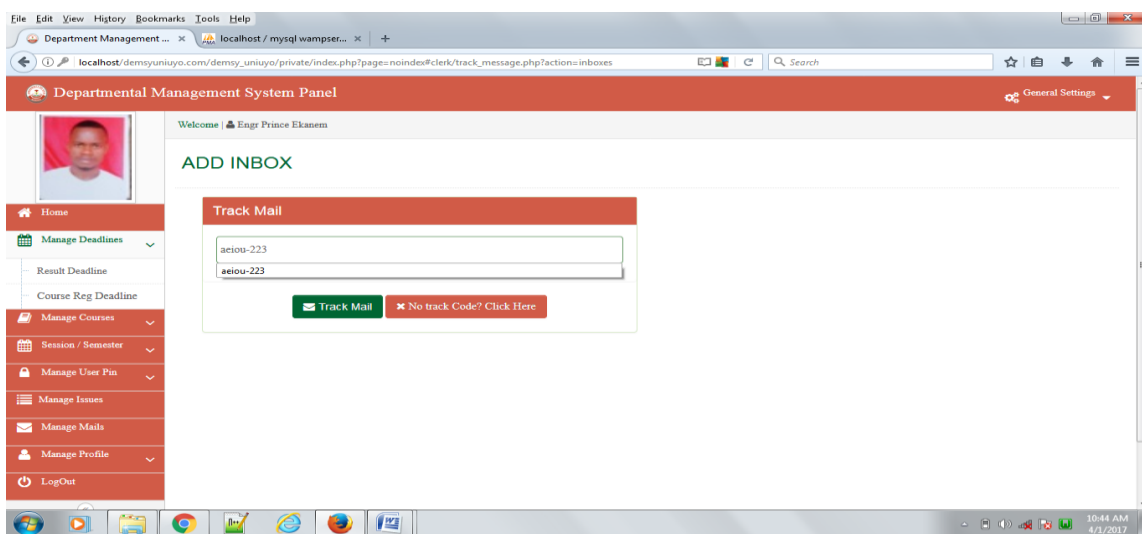


Figure 12: Tracking the status of mail received from other department/unit

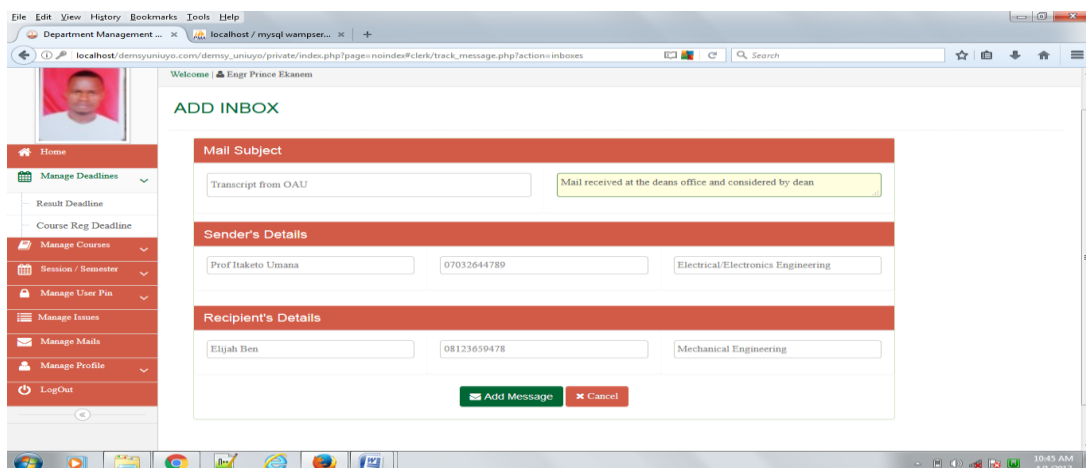


Figure 13: Updating the status of the mail received and tracked by secretary

Figure 14 shows how the clerical officer or secretary records a new incoming mail. The clerical officer or admin clicks on “inbox” presented in Figure 10 and then click on “add inbox” presented in Figure 11 and will be presented with two options to either track mail received from other units/departments or record a new incoming mail. To record a new mail, the secretary will click on “no tracking code” presented in Figure 13 and enters the mail subject, sender’s details and recipient’s details in the spaces provided as shown in Figure 14.

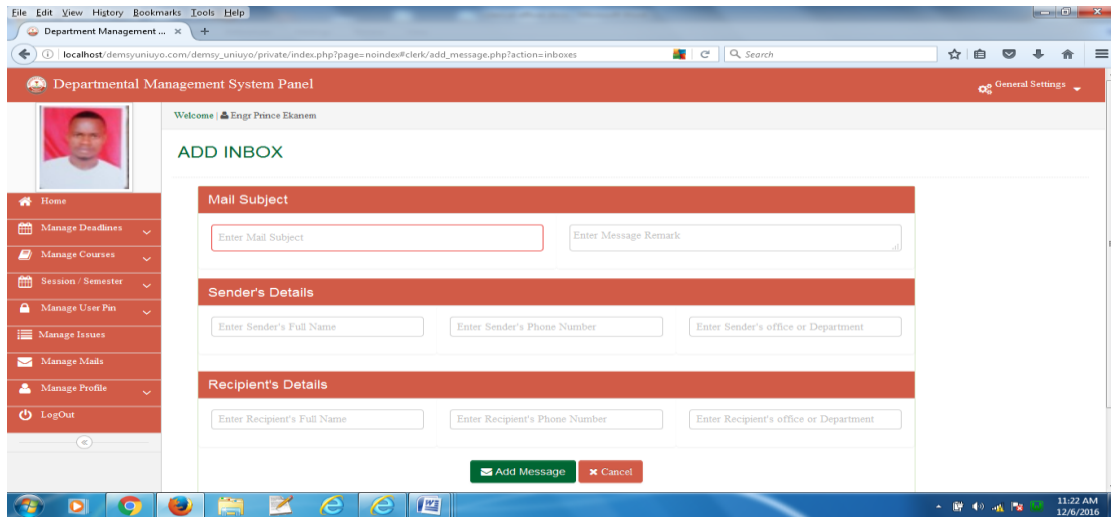


Figure 14: Recording new incoming mails

Figure 15 shows how the clerical officer manages outgoing mails. He/she can query mail using the search box.

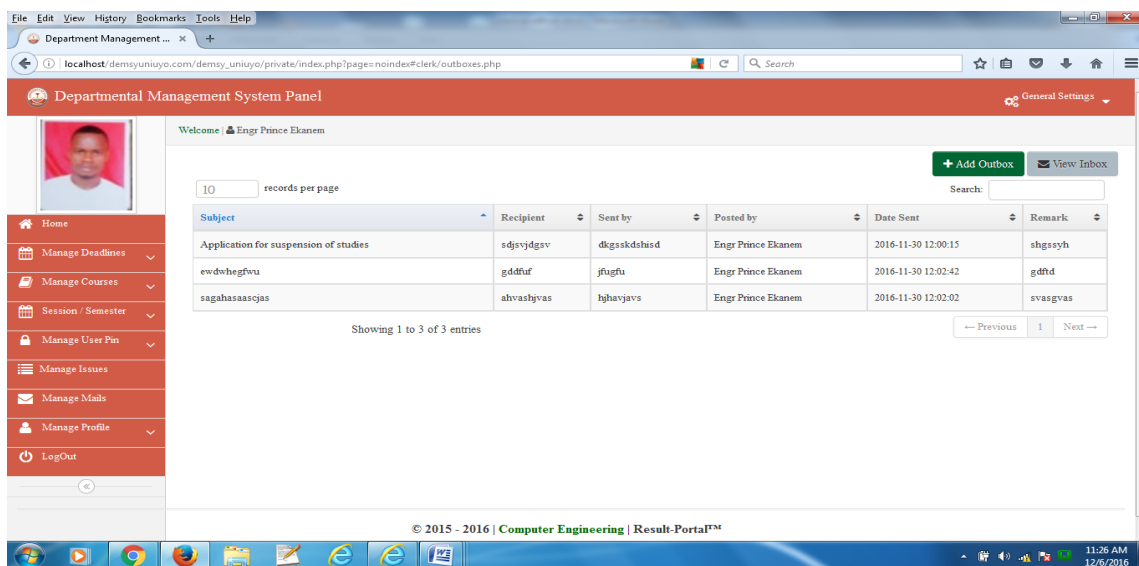


Figure 15: Managing outgoing mails

Recording of outgoing mails is presented in Figure 16. The clerical Officer enters the mail subject, sender’s details and recipients details in the box provided.

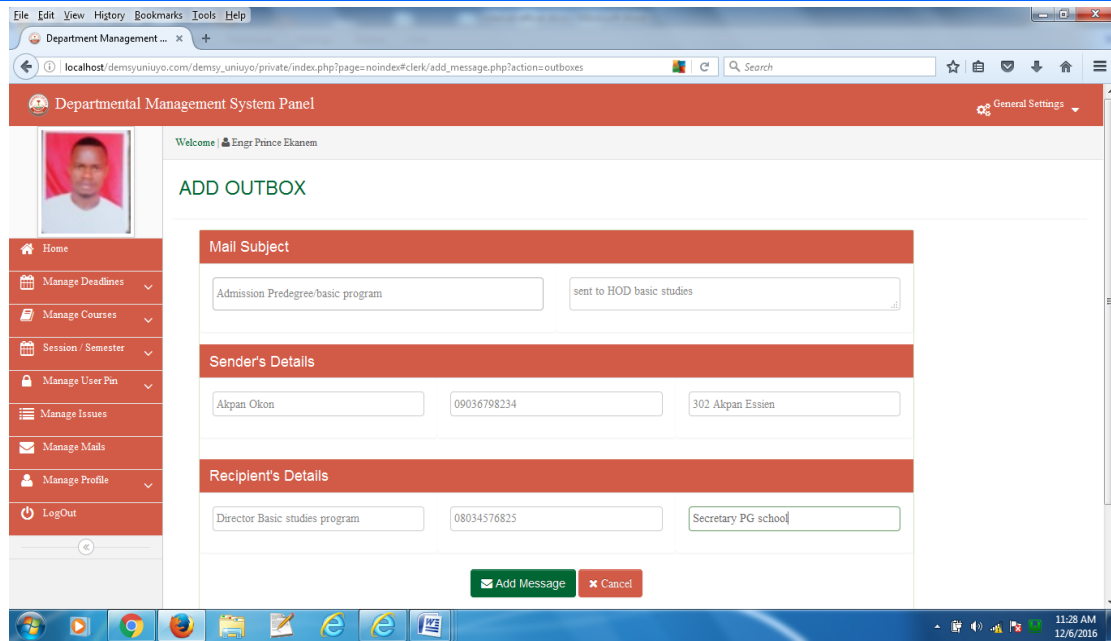


Figure 16: Recording outgoing mails

The student must enter his/her mail tracking id to track the status of his/her mail as shown in Figure 17. The mail tracking id is generated when a staff of student submits mail.

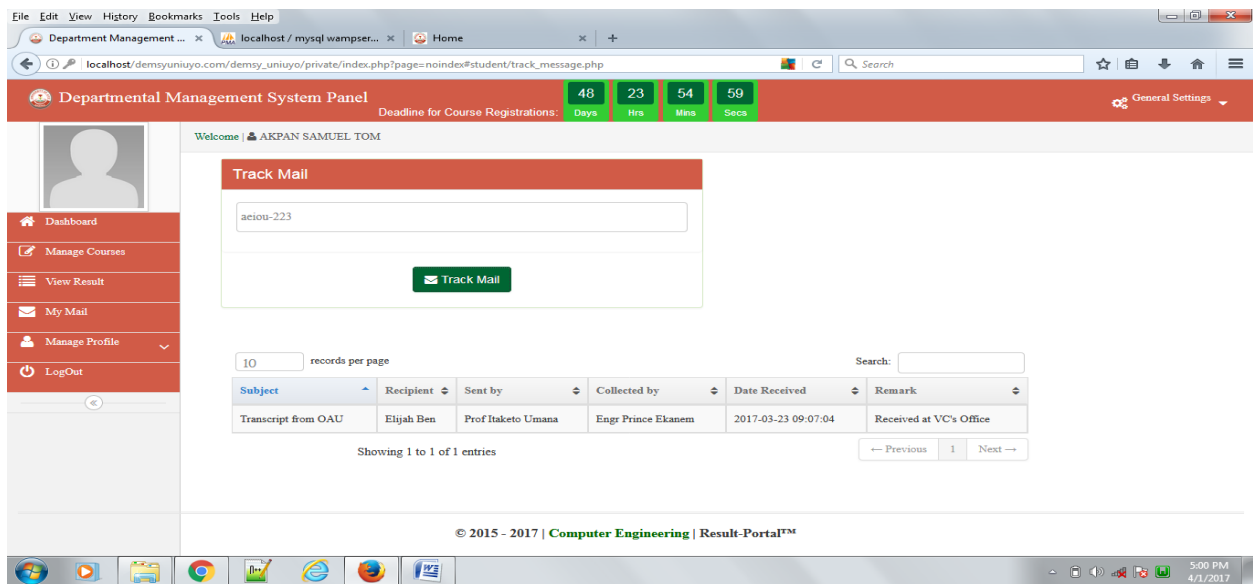


Figure 17: Student mail tracking sub-module

### B. SUPPORT MODULE

Figures 18,19, 20, 21, 22, 23 and 24 show the support module. This is a simple, fast, lightweight, secure and user-friendly PHP help desk platform. Through this platform users can contact any staff in the department and get his/her issues resolved with speed.

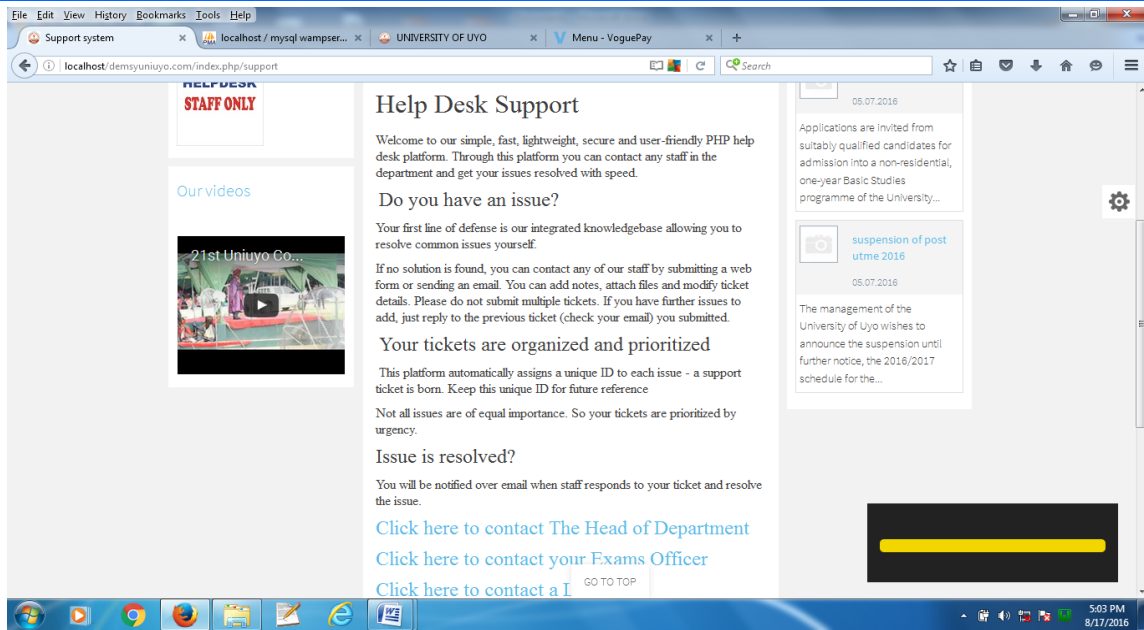


Figure 18: Support module top view

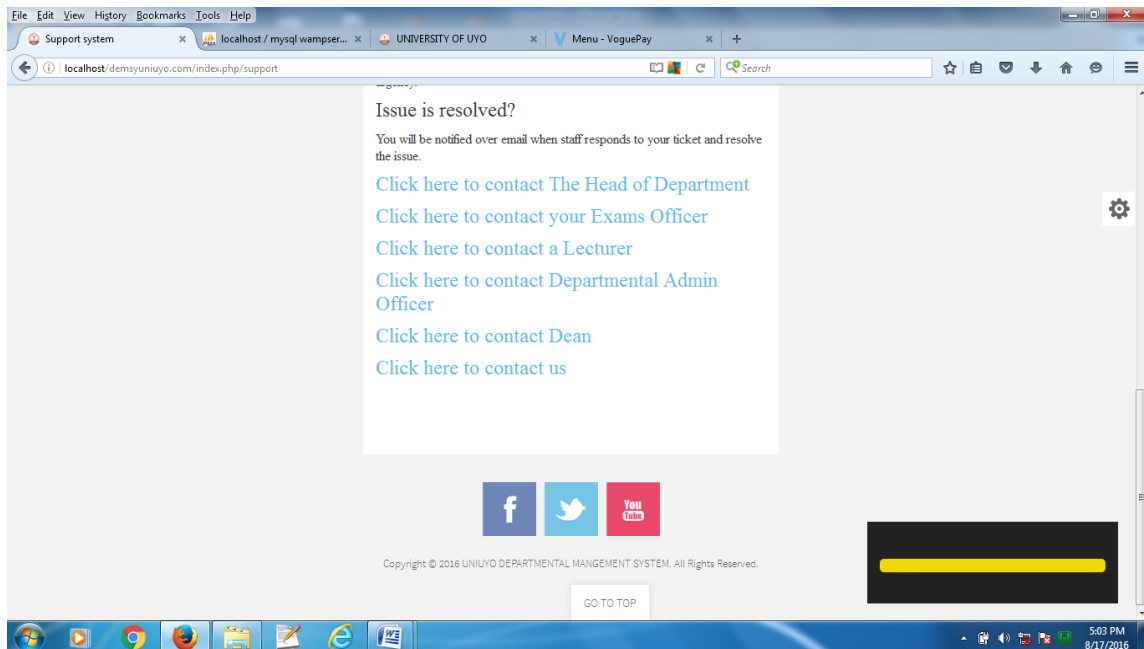


Figure 19: Support module bottom view

Figure 20 shows the support module showing submit ticket option, view existing ticket option and knowledge base

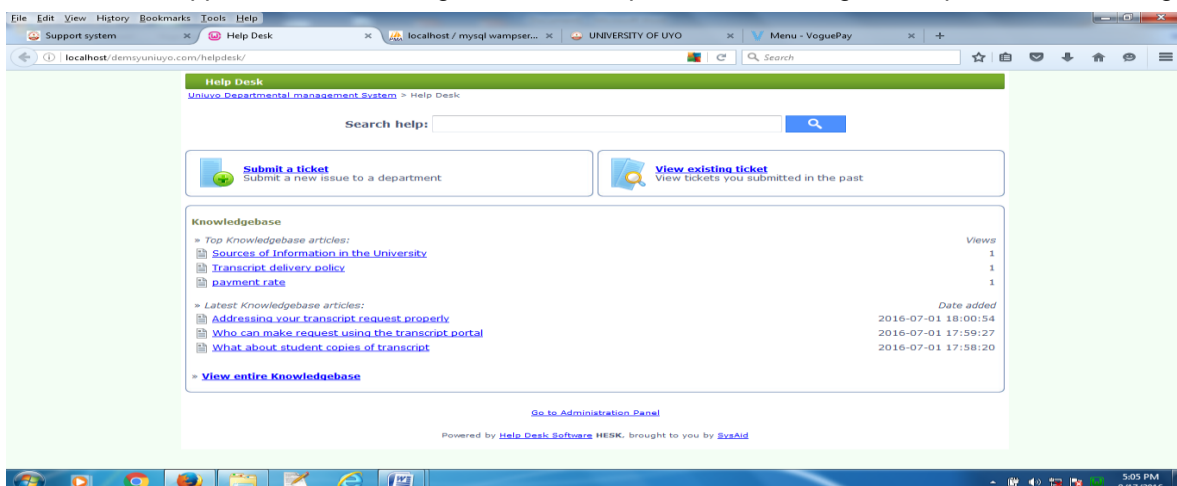


Figure 20: Support module showing submit ticket option, view existing ticket option and knowledge base.

Figure 21 is the support module showing submits ticket form. To raise a ticket, the user must supply his name, email address, priority of his/her issue (high, medium, and low), subject of the issue and the message. He/she can also upload reports concerning the issue.

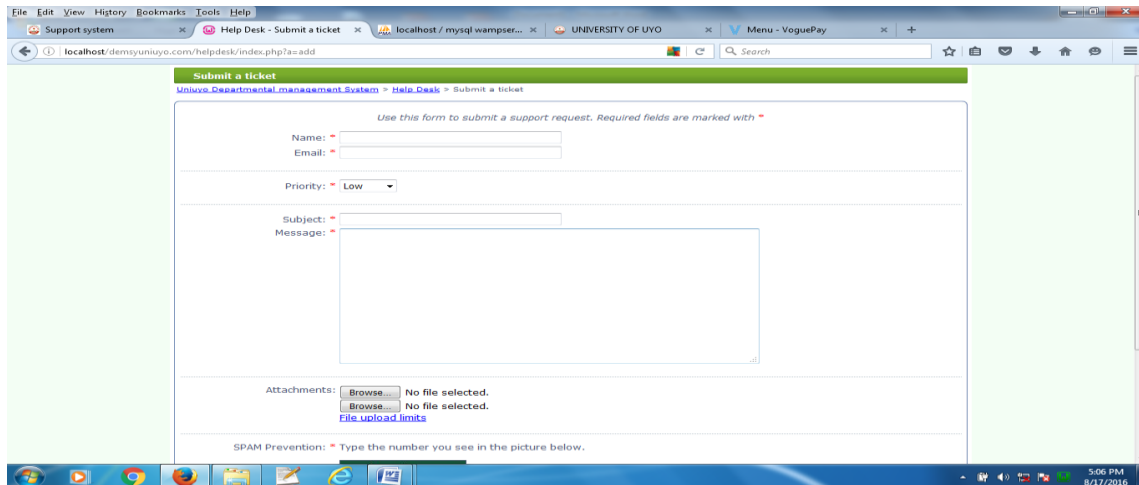


Figure 21: Support module showing submit ticket form

Figure 22 is the support module showing the knowledge base. The user can also utilize the system's knowledge base to find solution to common issues like how to process transcript, available programmes, admission processes and so on.

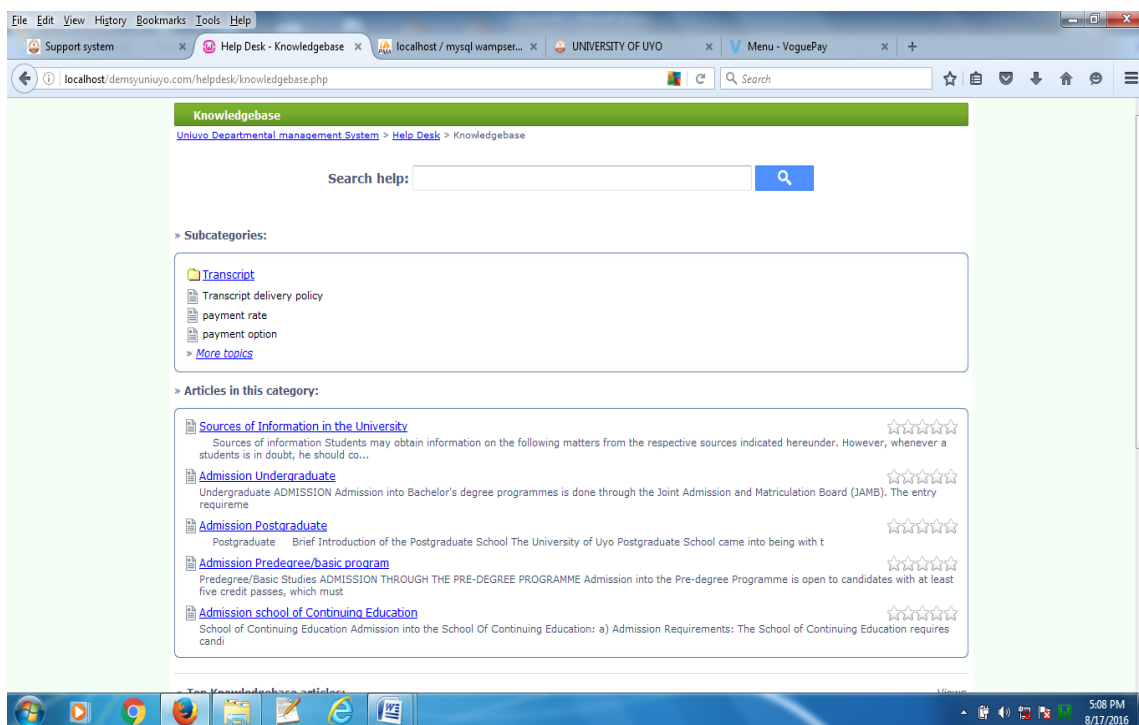


Figure 22: Support module-knowledge base

Figures 23 and 24 shows the administrative section of the support module showing how issues are resolved. Here the staff will be notified by mail when he has pending issue to resolve. The staff will login to his panel on the support system to view the issue, he/she will click on add reply to reply to the issue. Once the issue is resolved, the staff will change ticket status to "marked as resolved" as shown in Figure 24 and the user who raise the ticket will be notified automatically by mail about his/her issue. If the staff is not in position to respond to the issue, the staff

can move the ticket to another staff he feels can handle the issue using the “move ticket to” or “assign to” option. He can use the add note option to add notes to the staff he has moved/assign the ticket to.

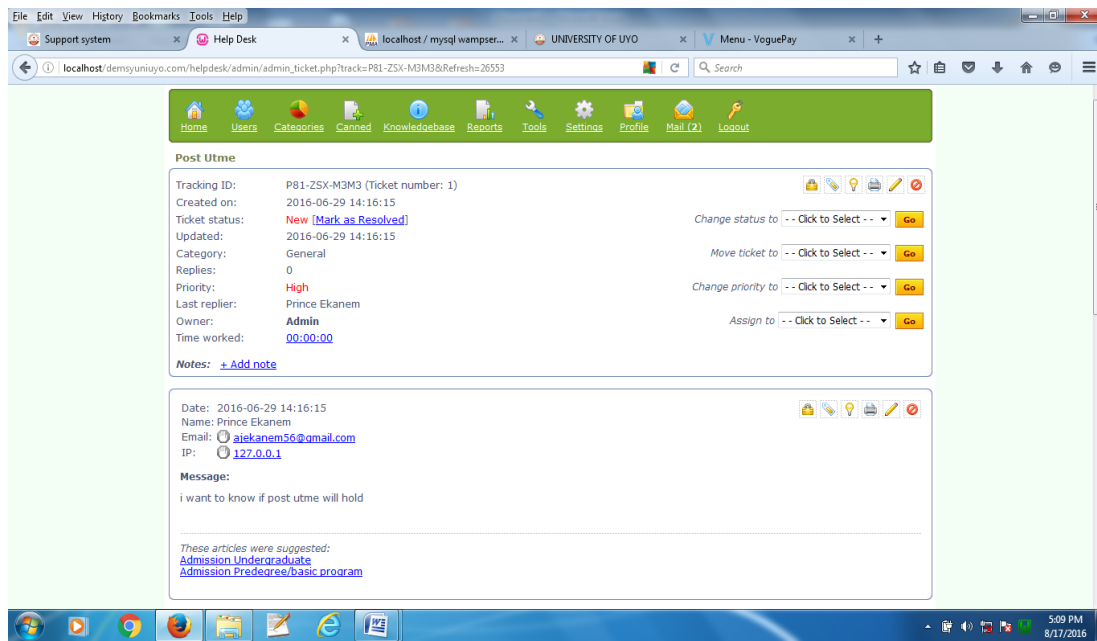


Figure 23: Administrative section of the support module (top view)



Figure 24: Administrative section of the support module (bottom view)

#### IV. CONCLUSION

In this paper, development of mail management system (MMS) is presented. The new system addressed the weaknesses of the current manual mail management system in the University of Uyo. The problems inherent in the current system include:

- i Time wastage in tracking of incoming and outgoing mails

- ii Loss of mails
- iii Errors in mail documentation [2,4,5]

The MMS is developed using HTML, CSS, JAVASCRIPT, PHP and MYSQL and is locally hosted using Apache web server. The software development methodology adopted is a participatory incremental process model (PIP Model). The top down stepwise functional decomposition, flowchart and dataflow

diagram of the system are provided to explain the major functionalities of the system. Screen shots of

various system functionalities from the test run of the functional system are also presented.

## REFERENCES

- [1] Zhibing, L. Wang, H. and Zan, H. (2010). Design and Implementation of student Information management system. Paper presented at the 2010 International Symposium on Intelligence Information Processing and Trusted Computing. IEEE Inc., TBD Huanggang, China
- [2] Eludire, A. A. (2011). The design and Implementation of Student Academic Record Management System. *Research Journal of Applied Science, Engineering and Technology* 3(8): 707-712, 2011
- [3] Ghatge, M.S.T., &Bhoite, S. D. (2016). Improving College Working Efficiency Through Design and Development of A Computerized Information System (CIS). *International Journal of Scientific Research*, 4(7), 521-524
- [4] Obasi, N., Nwachukwu, E. O. and Ugwu, C. (2013). A Novel Web-Based Student Academic Records Information System – AOL, *West African Journal of Industrial and Academic Research*, 7(1): 33-34
- [5] Bharamagoudar, S. R., Geeta, R. B. and Totad, S. G. (2013). Web Based Information Management System, *International Journal of Advanced Research in Computer and Communication Engineering* 2(6): 2342-2348
- [6] Siew, S. T., Yeo, A. W., & Zaman, T. (2013, July). Participatory action research in software development: indigenous knowledge management systems case study. In International Conference on Human-Computer Interaction (pp. 470-479).Springer Berlin Heidelberg.
- [7] Simonsen, J., & Robertson, T. (Eds.). (2012). Routledge international handbook of participatory design .Routledge.
- [8] Ferrario, M. A., Simm, W., Newman, P., Forshaw, S., & Whittle, J. (2014). Software engineering for 'social good': integrating action research, participatory design, and agile development. In Companion Proceedings of the 36th International Conference on Software Engineering (pp. 520-523).ACM.
- [9] Spinuzzi, C. (2005). The methodology of participatory design. *Technical communication* , 52 (2), 163-174.
- [10] Bano, M., & Zowghi, D. (2013). User involvement in software development and system success: a systematic literature review. In Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering (pp. 125-130).ACM.
- [11] Aron, R., Clemons, E. K., &Reddi, S. (2005). Just right outsourcing: understanding and managing risk. *Journal of Management Information Systems* , 22 (2), 37-55.
- [12] Bennett, K. H., & Rajlich, V. T. (2000, May). Software maintenance and evolution: a roadmap. In Proceedings of the Conference on the Future of Software Engineering (pp. 73-87).ACM.
- [13] Hawthorne, M. J., & Perry, D. E. (2005, May). Software engineering education in the era of outsourcing, distributed development, and open source software: challenges and opportunities. In International Conference on Software Engineering (pp. 166-185).Springer Berlin Heidelberg.
- [14] Keil, M., Cule, P. E., Lyytinen, K., & Schmidt, R. C. (1998). A framework for identifying software project risks. *Communications of the ACM* , 41 (11), 76-83.
- [15] Ekanem, A. J, Ozuomba, S., Jimoh, A. J. (2018). Development of Student Result Management System: A case study of University of Uyo. *International Journal of Mathematical and Software Engineering*, Vol. 3, No. 1(2017), 26-42. Varepsilon Ltd, Varepsilon.com