



Enhancing Learning Systems using Interactive Intelligent components

Wilson Nwankwo¹, Chinecherem Umezuruike², Christian C. Njoku³

¹Associate Professor, Department of Computer Science, Edo University Iyamho, Nigeria, nwankwo.wilson@edouniversity.edu.ng

²Department of Information Technology, Kampala International University, Kampala Uganda, cumezuruike@kiu.ac.ug

³Department of Computer Science Education, Aminu Saleh College of Education Azare, Bauchi State, Nigeria, chrisnjoku821@gmail.com

ABSTRACT

Following the invention of the teaching machine in 1924 by Sidney Pressey, there has been an evolutionary trend in the creation and promotion of software systems categorized as Learning Management Systems(LMS). Functionally, a typical LMS could be on-premise or cloud-based, however, one common feature canvassed is increased effectiveness. This paper focusses on enhancing effectiveness using intelligent components (e.g. algorithms, logic, etc.) to ease decisions on vital academic fronts. Emphasized here is the role of academic heads of departments in HEIs in effectively drawing the course allocation plan for a learning period. In large departments with many academic programs, courses, and lecturers, drawing a plan that fairly reflects capacity as to 'which teacher teaches what course', is an issue. This paper presents a systematic design of an intelligent component from the grounds up using the rapid application development(RAD) approach. The system underscores a posteriori knowledge on the part of the authors as to having significant experience on the modalities around academic activities across the different horizons and explores such to drive home the interestingness in enhancing the traditional practice through the deployment of intelligent interactive software. The prototype system is web-based with an android option and implemented using C#, ASP.Net Core, Xamarin forms; to enhance multi-platform support. The intelligent engine is driven by naïve bayes algorithm. The system is validated using two human experts who are presented with the histories of two courses and a handful of lecturers in a given department in a Nigerian HEI. The result of the validation shows that the advice given by the system is in tandem with the best decision an expert could have taken but would outperform a human expert when faced with scenarios as in large academic highlighted above. The paper concludes that the solution portends immense potentials and would ease the pressures and stress encountered by decision makers in fathoming capabilities of teaching staff vis-a-vis course offerings and with regard to the dynamic nature of academic activities in a typical HEI.

Key words: Intelligent systems, intelligent software agents, Learning Management Systems, Machine learning, Fuzzy logic

1. INTRODUCTION

Education is about learning and its object is entrenched in some remarkable symbolisms:

“... a wish for less misery among the poor, less ignorance in schools, less bigotry in the temple, less suffering in the hospital, less fraud in business, less folly in politics.” [1]. As a remarkable symbol in shaping the society, most HEIs like universities often adopt one statement crafted as a ‘motto’ to reflect their stand on learning.

Having realized the need to foster education without boundaries, developments around virtual learning had remained ongoing for decades. These efforts have culminated into several species of virtual learning [2].

The 21st century HEIs are critical to the development of national and international economies. Consequent upon this truth, such institutions need sophisticated tools and platforms to remain competitive in the knowledge-driven human venture[3]. However, the aforesaid is an ideal situation that remains a vision of many HEI policy makers and decision takers[4]. Efforts have been made in several domains in evolving integrated learning and associated systems[5].

Majority of the popular commercial and open source LMSs though may incorporate some analytics(e.g. Canvas, Blackboard,etc.) are more oriented towards content provision and management and for which scopemay be extended by the licensee or purchaser following acquisition or deployment. Many of these solutions allow functionality extension through application programming interface(API), whereas others may not be extensible. With the covid-19 pandemic more HEIs are adopting the cloud-based LMS as the most viable option to continue their academic operations. As governments in developing countries are indicating interests in ensuring academic calendars of HEIs are not distorted, the acquisition and deployment of LMSs may assume a different direction in the near future. Public private partnership [6], [7] model of acquisition and deployment may become a vital option for most institutions.

Amid the consistent advances in the development in electronic learning solutions, users are more concerned on particular software quality parameters [8] that increase their productivity and security. Relevant in this aspect are: interactiveness[9], flexibility [10], security[11][12] [13][14], maintainability[15], speed[16], platform-support or independence[17], reliability[18], mobility[19],

extended support and intelligence[20], robustness, size, etc. As modern tools evolve and pervade academic environments, experienced academics in leadership positions such as heads of academic departments(HODs) in medium to large HEIstend to be attracted to tools that exhibit some higher level of precision and extensive versatility in such a manner that does not pose issues in their usage. Course assignment also called allocation falls within the responsibilities of these HODs. Course allocation is resource allocationfunction and may be tasking. The traditional rule in many HEIs is governed by specialties, experience, availability, and convenience of lecturers. Issues would certainly arise in the following circumstances:

- a. Where there are many specialists in a field within a departmental program e.g. BSc-Computer Science
- b. Where there are many supposedly qualified persons by experience and academic credentials which may reasonably raise questions that border on which of the supposedly qualified academic would ‘do justice’ to the course in question. In such a scenario that decision-maker is constrained to adopt a rule of thumb(perhaps based on some psychology *or a posteriori* knowledge). The result may be enshrouded in bias and for which outcome may be detrimental in the long run.
- c. Where the *academics* are relatively new in the system wherein there is no knowledge as to their previous performances.
- d. Findings have shown that courses are assigned in many scenarios based on the available human capacity. In many scenarios the determining factor is convenience i.e. academic’s *domain of comfort*. In other words, course(s) for which an *academic* is ‘at home with’. This does may not often reflect performance in the said course. This is common in all the departments and schools in many HEIs.

In such scenarios painted above, interactive and intelligent decision support(IIDS) tools may provide the decision-maker with better alternatives to support their adventures hence this paper is aimed at showcasing the design and implementationof an intelligent interactive component(IIC)that extends the functionality of an LMS. The proposed component would effectively assist a HOD of a large academic department in automatically scheduling and allocating courses for programs to resources(teachers) for a given period. The specific objectives are:

- a. To use a case HEI in Nigeria to showcase the traditional course assignment requirements and ascertain challenges and problems.
- b. Present structured equivalent of the course assignment processes using an enhanced system. This is realized through the deployment of a web/mobile softwarecomponent that uses the Naives Bayes model to predict and justify the choice of suitable academic resource(s) for a courses offered.

2. METHODOLOGY

The Rapid Application Development approach, a child of the object-oriented paradigm [21] [22] [23] is used in evolving this system. The approach is divided into three phases: requirements planning(engineering), rapid design, and implementation.The software used include: Microsoft SQL Server 2016, and Microsoft Visual Studio 2019 (community edition). This database server is chosen for this project because it integrateseasily with the development environment, and offers good object-relational capability, machine learning(ML)support, business intelligence support, multi-platform data integration capability, multi-threaded database engine, and portability. C# is the mainstream development language and support the direct integration of ML model codes such as Naïve Bayes used in this work. The hardware include:

- i. A HP Elitebook G2Intel core-i7 system @2.7 GHz, 16GB RAM, 1TB storage;
- ii. A connection to the Internet.

2.1 Requirements planning and analysis

Case study: Edo University Iyamho(EUI) located in Auchi axis of Edo State Nigeria

Since the autumn of 2018, EUI has been using Canvas LMS(<https://edouniversity.instructure.com>) for its academic operations. To extend the the functionalities of the said LMS, it has developed an academic information system(AIS)(<https://euiiais.ng>) and embedded same into the LMS through API. Through its AIS, various business functions of academic departments are conducted. However, there is need for further extension of the functionality of these systems and this paper is aligned with that quest.

The various requirements for buiding the proposed component are captured herein. The application boundaries for which the proposed system must satisfy and lie within are highlighted using use cases. The validation and verification of these requirements are done through a *posteriori knowledge*. In this context, the first stage is identifying the actors in the domain. The domain in question is resourceallocation i.e. fair assignment of courses based on parameters that could produce the best results.The first concern on requirements is actor identification. The actors in the course assignment process are:HOD, Academics, and Administrative officers.The sequence of activities in this subsystem is represented by the activity diagram in figure 1 and summarized thus:

- a. The HOD reviews all courses on offering in the present circumstance and perhaps previous allocations. Where quality assurance is implemented, the HOD further reviews feedbacks received from previous course undertakings by the various academic staff. The feedback may emanate from students or assessors superior to the academic staff.
- b. The HOD drafts of all academics attached in the department, assigns courses based on areas of expertise and number of academics available;
- c. The HOD informs their secretary/officer to prepare a document conveying the allocation, and communicate same to the concernedacademics.

- d. The Department prepares lecture timetable and communicates same to academics, students and other authorities.

2.2 Reasoning/Intelligence requirements

With respect to a course offering, an *academic* is graded into either of 3 categories: *most qualified, qualified, less qualified*. This is the dependent variable and technically represented by $P(\text{academic})$. To computer $P(\text{academic})$, various independent variables are defined (See Table 1). The reasoning logic is the Naïve Bayes (NB) model [24][25]. NB classification may be crude but exhibits several merits over some more-sophisticated options like neural network, logistic regression, support vector machine, etc. NB is quite simple, easier to implement, and also scales impressively well to extremely large datasets. NB could also be extended to problem scenarios requiring more than three independent variables (multinomial classification).

The attribute domain given as:

X such that $X \rightarrow (\text{course}, \text{score}, \text{pscore}, \text{grating}, \text{crating})$. The attributes are defined in Table 1. To predict that academic Y could take on course c ; the algorithm is thus:

- Define the training data set from the case data set on Y ;
- Compute the probabilities for each attribute conditional on the class value;
- Compute joint conditional probability for the attributes using the product rule while discarding attribute with missing values;
- Where an attribute value does occur regularly with the class value, insert a probability of zero (0);
- Use Bayes rule to calculate the conditional probabilities for the class variable
- Compare the probabilities and compute the mean and standard deviation of the set
- Return class with the highest probability
- Apply results to a testing data set

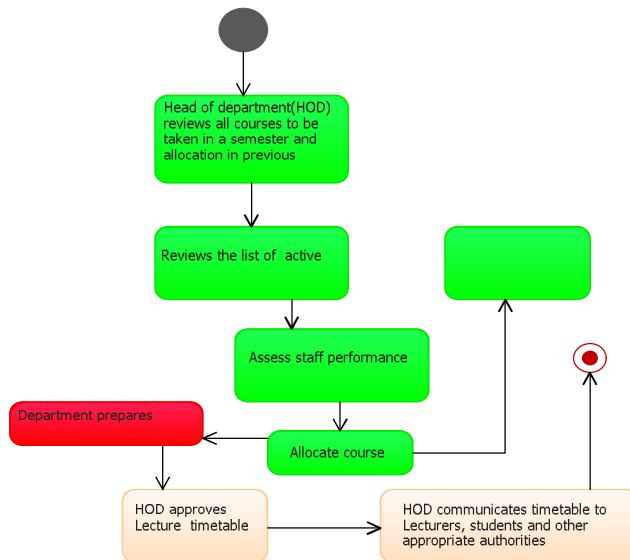


Figure 1: Activity diagram of a departmental course allocation process

The next phase is use case analysis i.e. identifying the functionality to be desired in the proposed system [12]. Figures 2-4 show the use case diagrams. Figure one represents the actual course assignment process whereas Figure 3 reflects the attendant student performance review which may be subsumed into the course assignment process. Similarly, Figure 4 shows previous performance reviews on academics prior to course allocation.

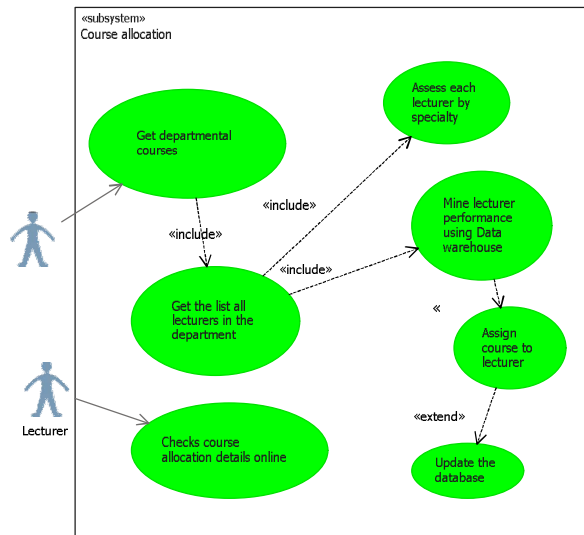


Figure 2: course assignment use case diagram

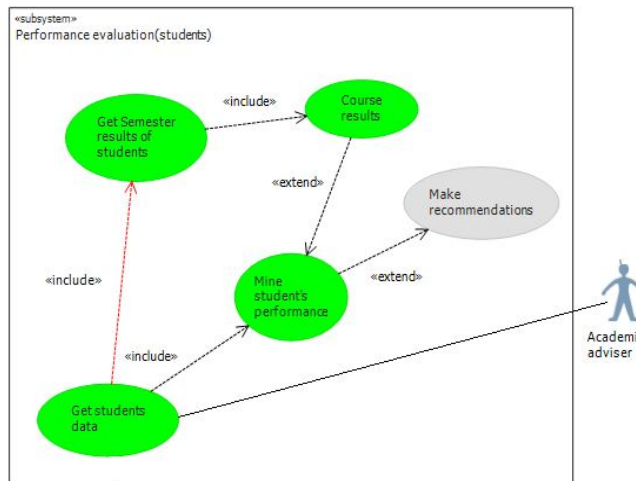


Figure 3: Performance use case diagram

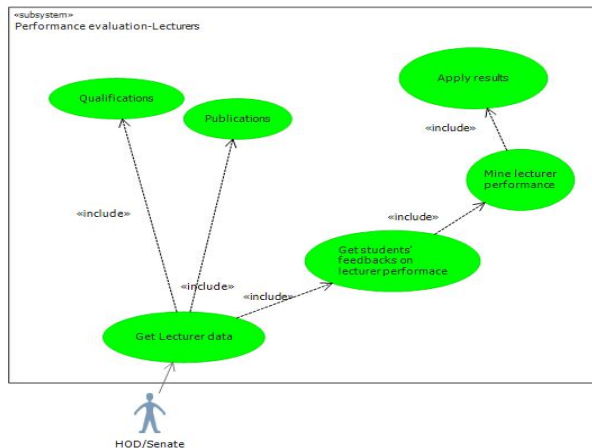


Figure 4: Performance review on academic staff

3. RESULTS AND DISCUSSION

The subsystems in the proposed system are grouped into three (3) components: database, logical reasoning agent, frontend. Inputs, transformation and outputs revolved around the aforementioned components. It is assumed that the database exists as the proposed system is only a component to be embedded in an existing LMS. The frontend envelops the various user interfaces and underlying application/business logic. All inputs are captured through the application frontend. The operational database (the backend) stores all data created by the application. The product of the design phase was a set of models that described the system structure, made necessary trade-offs, provided a blueprint for implementation and allowed the iterative evolution of the system under controlled conditions.

2.2 Inputs to the design

The vital inputs to the design were:

- i. The use cases;

- ii. The activity diagrams, which were used to represent the model of the behaviour of the component objects within the system;

We used the system sequence diagram (SSD) (see Figure 5) to present an interactive and more detailed equivalent of individual use cases to clarify how tasks are performed between individual users and the automated system. The following are shown: external actors; messages or methods that are invoked by these actors; return values (if any) associated with previous messages; areas or points where loops or iterations exist. The SSD contains:

- i. The action line - indicates the actions performed by an actor/system; when connected to a lifeline, it shows the existence of interaction between the actor and the system;
- ii. The lifeline - long dotted line attached to the actor or the system;
- iii. The messages - show the details of an action;
- iv. The objects.

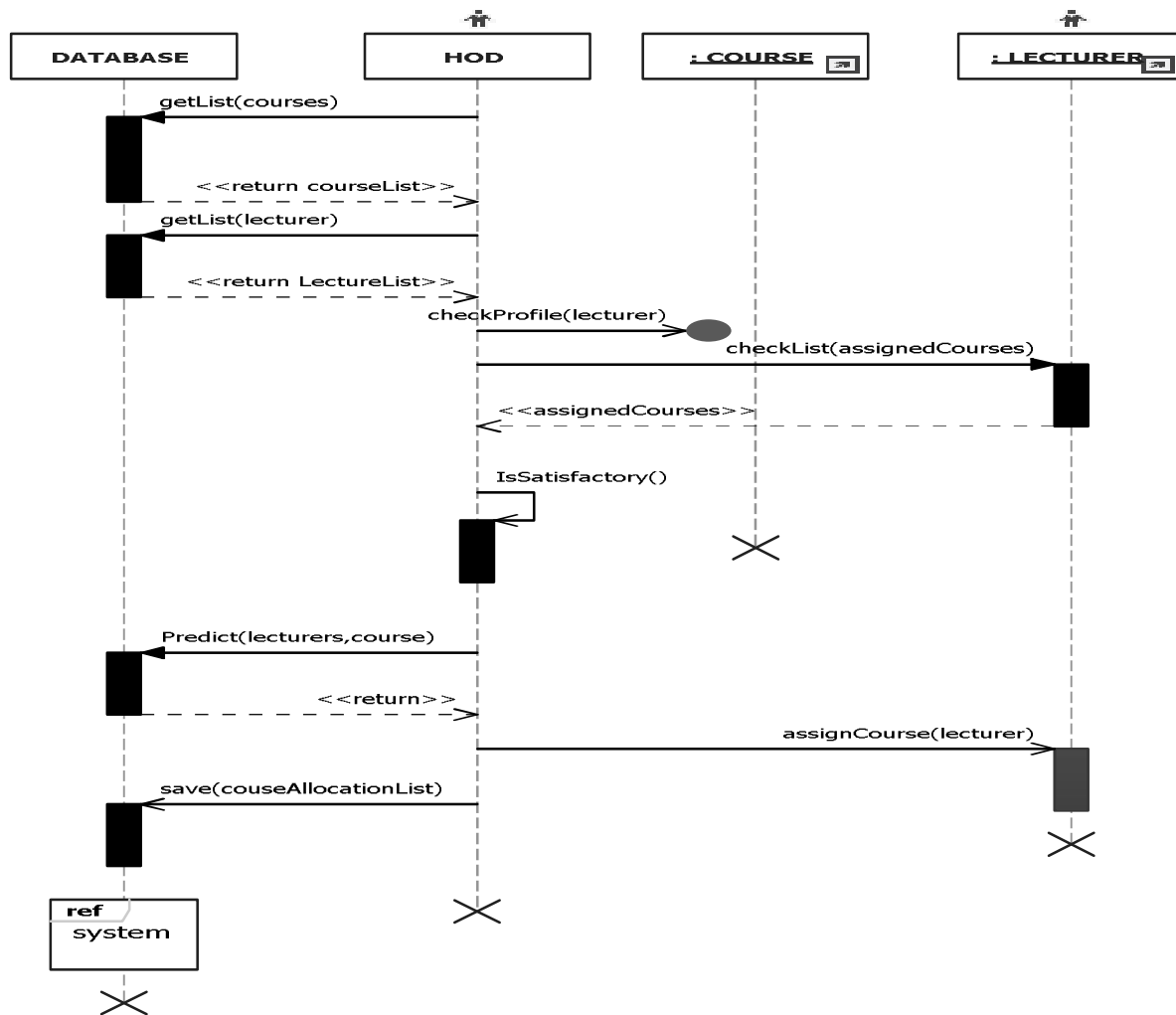


Figure 5: Course assignment SSD

3.1 Modeling the system structure

Modeling the system structure is done using class diagrams (Figure 6-7). They are employed to model the structure of the proposed component at the lower level. The lower level underscored herein is the logical model. Included in the logical model are rules that operate on the

attributes, how these rules are grouped into interfaces and classes, and how the various objects interact among themselves to solve macro level requirements. In the logical model, the data types of the attributes of the objects are clearly specified and the relationships established. Figure 7 is used to show the various components of feedbacks. These objects capture feedback histories on various academics on courses taught. Figure 8 is a class

diagram containing the objects associated with an academic in a department.



Figure 6: Feedback class diagram

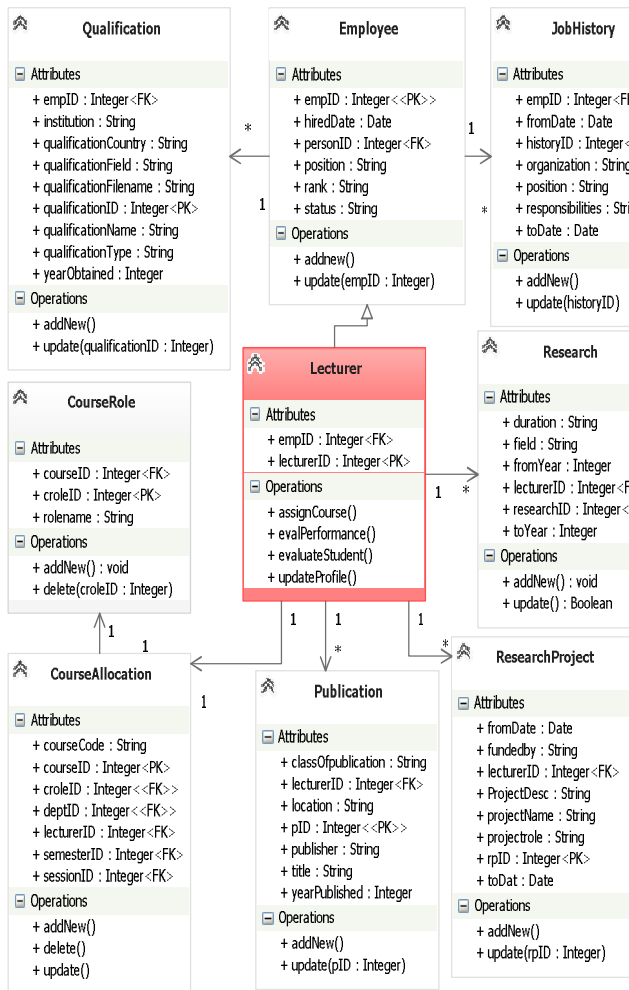


Figure 7: Logical model of the academic object

3.2 Application programming

The model-view-controller (MVC) approach was used for the program development. The Model-View-Controller design was implemented using the ASP.NET core MVC framework for cross platform support. In addition to mainstream coding, the C# language doesn't require any special libraries for implementing the Naïve Bayes logic.

3.3 Database implementation

The Database-first approach was used since there is an existing database. However, the SQL Server management studio was used to create additional database relationships, indexes, procedures, and functions.

3.4 The Reasoning engine

To address reasoning requirements, the Naïve Bayes algorithm was employed. The data specifications for the Naïve Bayes algorithm include:

- i. Single key field that contains only one numeric or text column that uniquely identifies each record;
- ii. Input fields that are either discrete or could be discretized. This ensures that the input values are independent of each other since prediction is involved. Note that if two closely related fields are used, the effect may multiply the influence of such fields potentially resulting in some obscurity that influences the outcome.
- iii. One or more predictable fields that is discrete or could be discretized, treated as input and relevant when when evaluating relationships among fields in a new dataset.

The *lecturer* table was used as the source data table and a subset of the data was selected, which include the fields defined in Table 1

Table 1: Dataset specification for Naïve Bayes model

Variable	Data type	Content	predict	input	function
course code	text	Discrete	no	yes	unique course in the course list
lecturer name	text	Discrete	yes	no	The academic's fullname
lecturerid	long	Key	yes	no	ID of an academic in a department
sscore	double	discretized	no	yes	aggregated score computed from all feedbacks from students on a course in the past
pscore	long	discretized	no	yes	Aggregated score computed against staff from previous assessment by superiors
qrating	long	discretized	no	yes	General rating based on qualification
crating	long	discretized	no	yes	Course rating based on familiarity with domain courses in the department

A Nave bayes module is created in the application. After the created model is trained using 20% of all historical whereas testing is done with 80% of the data, the result is stored as a set of patterns, which is explored to make predictions. Thereafter, stored procedures are created to return predictions about how new data relates to the

predictable attribute, or retrieve statistics that describe the correlations found by the model. Prediction is handled through a set of functions in the application.

3.5 User interface

The implemented component's user interface is shown in Figure 8. The user interface enables a HOD or anyone acting in such a capacity to obtain some intelligent insights interactively during the course assignment process. The interface presents courses on offering from the database. The HOD may wish to select several courses at the same time for which insights are to be provided upon. When the HOD hits the 'suggest lecturer' button, the intelligent agent responds with the name of the lecturer that the agent considers better placed for the course. The HOD may wish to probe further to know the reasons for the suggestion.

DEPARTMENT OF COMPUTER SCIENCE
COURSE ALLOCATION WINDOW

2019/2020 SESSION SECOND SEMESTER 2019/2020 SEMESTER Date: Sunday, June 14, 2020

Current Courses for Second semester 2019/2020

#	Course Code	Course Title	Unit	Status	Semester	
1	CSC 121	Principles of Compute Programming	2	C	second	<input checked="" type="checkbox"/>
2	CSC 122	Data Structures and Algorithms	3	C	second	<input checked="" type="checkbox"/>
3	CSC 123	Structured Programming	2	C	second	<input type="checkbox"/>

Lecturer
Daniel Jones

Course Role
Chief Lecturer

Suggest Lecturer

Provide Insights for Suggestions

Approve Suggestions Suggest another

Figure 8: Component user interface

3.6 Software security issues

Data security was implemented at the application and database levels. The user of the system is expected to be in control of the input data supplied to the system. As the system is network-based, there are bound to be network security issues that may affect the security of the data streams flowing through the network. At the level of the user, two types of checks and controls: client-side validation and server-side controls were implemented. Various client-side scripts were used to ensure that only valid data is entered at the client-side. These scripts are executed by the client (web browser) thereby preventing a round trip to the server. JavaScript was employed to ensure that the required fields are supplied with relevant data only. Field lengths and data types were appropriately defined for forms. Any submission of incomplete form is met with error messages. Some checks cannot be applied at the client end. Server side checks are necessary to save the system from: failing, data corruption, and displaying some undesired error messages. The checks imposed at the server side were:

- Database table constraints, deployed to check for validity of primary and foreign/composite keys;
- Inconclusive operations must be rolled back at the database;
- Various access control mechanisms were created at both table and database/data warehouse levels

to ensure that only users with the appropriate roles have access to the data.

B. Application security

A layered security model [11] is adopted in this work i.e. application layer and database layer. At the application layer, three forms of security modules are implemented: authentication using passwords, encryption, and roles. In other words, when a user is authenticated, the password is encrypted using the advanced encryption standard (AES) which is considered very effective in protecting against hacker attacks. Every password is encrypted before being relayed over the network. With the roles integrated into the system, a user can only access those features that are attached to the role automatically associated with the user's credentials. Each role has a unique operational boundary, that is, the extent of activity which a user with that role can perform on the system. The HOD is associated with the HOD role amid other roles. Each role has its restrictions, for instance, a user with a lecturer role would not have access to administrative modules in the application and vice versa. The rights and privileges associated with each role are activated on successful login to the system.

In addition, a user with any of the roles must be authenticated by the system prior to using the system. Authentication is through the use of credentials such as: user name and password. The credentials are stored in the database with the password encrypted using a 128-bit AES encryption key. The authentication/authorization mechanism ensures that only validated users are allowed on the system.

4. CONCLUSION

In this paper, we have highlighted the relevance of LMSs in academic institutions especially during a pandemic like covid-19. However, We have emphasized on the extended intelligent functions. To extend the functionality of a LMS, we have demonstrated a quick way of developing a component that would aid decision makers such as HODs perform vital functions such as course assignments. We conclude that integrating such a component into the LMS used by large departments in a HEI would save the decision makers time and effort in reviewing profiles of academic staff in determining 'which lecturer teaches effectively which course'. Future developments in this direction may include the use of a multi-algorithmic model wherein the reasoning component compares the results obtained by various algorithms at the background and showcases to the user the optimum result alone.

REFERENCES

- D.C. Gilman. **GILMAN'S Inaugural Address**. Johns Hopkins University, 1876. <https://www.jhu.edu/about/history/gilman-address/>
- W. Nwankwo. **Promoting Equitable Access to University Education through Online Learning Systems**, *World Journal of Engineering Research and Technology*, Vol. 4, Issue 2, pp. 517-543, 2018

3. S.L. Martin, and R.G. Javalgi. **Explaining performance determinants: A knowledge based view of international new ventures.** *Journal of Business Research*, Vol. 101, pp. 615-626, 2019
<https://doi.org/10.1016/j.jbusres.2019.02.041>
4. P. Ramísio, L.M.C.Pinto, N. Gouveia, H.Costa, and D. Arezes. **Sustainability Strategy in Higher Education Institutions: Lessons learned from a nine-year case study.** *Journal of Cleaner Production*, Vol.222, pp. 300 -309, 2019
5. F. Jurado, M.A. Redondo, and M. Ortega(2012). **Blackboard architecture to integrate components and agents in heterogeneous distributed eLearning systems: An application for learning to program.** *Journal of Systems and Software*, Vol. 85, issue 7, pp. 1621-1636, 2012
<https://doi.org/10.1016/j.jss.2012.02.009>
6. W. Nwankwo and A. Kifordu. **Strengthening Private Sector participation in Public Infrastructure Projects through Concession Policies and Legislations in Nigeria: A Review.** *Journal of Advanced Research in Dynamical and Control Systems*, Vol. 11, Issue 8, 1360-1370, 2019
7. A. Kifordu, W. Nwankwo, and W. Ukpere. **The Role of Public Private Partnership on the Implementation of National Cybersecurity Policies: A Case of Nigeria.** *Journal of Advanced Research in Dynamical and Control Systems*, Vol. 11, Issue 8 (special issue), pp. 1386-1392, 2019.
8. R. Sidler (2002). **Software Productivity**, 2002.. http://www.umsl.edu/~sauterv/analysis/488_f02_papers/SoftwareProductivity.html
9. W.L. Johnson and A. Erdem. **Interactive Explanation of Software Systems.** *Automated Software Engineering*. Springer, pp.53–75, 1997.
10. L. Shen and S. Ren. **Analysis and measurement of software flexibility based.** *Software Measurement European Forum*. Italy, 2006. <https://pdfs.semanticscholar.org/587c/48afa6644316fb744d43d27b19cba4afa34f.pdf>
11. W. Nwankwo. **A Review of Critical Security Challenges in SQL-based and NoSQL Systems from 2010 to 2019.** *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, Issue 2, 2020
<https://doi.org/10.30534/ijatcse/2020/174922020>
12. W. Nwankwo and A.S. Olayinka. **Implementing a risk management and X-Ray cargo scanning document management prototype.** *International Journal of Scientific and Technology Research*, Vol.8, Issue 9, pp.93-105, 2019
13. W. Nwankwo and C.C.Njoku. **Adoption of Internet Voting Platform: Containing Data Injection Threats with Structured LINQ.** *Nigerian Research Journal of Engineering and Environmental Sciences*, Vol 4, Issue 2, pp. 724-739, 2019
14. N. Davis. **Developing Secure Software, 2005.** <http://www.sis.pitt.edu/jjoshi/devsec/securesoftware.pdf>
15. H. Alsolai and M. Roper. **A systematic literature review of machine learning techniques for software maintainability prediction.** *Information and Software Technology*, Vol. 119, 2020
<https://doi.org/10.1016/j.infsof.2019.106214>
16. J. Yip. **What does “speed” mean in software product delivery?** 2017. Retrieved March 18, 2020, from <https://medium.com/@jchyip/what-does-speed-mean-in-software-product-delivery-9bc95c1c36c3>
17. R. Mall. **Fundamentals of Software Engineering** (4th ed.). Delhi, India: PHI Private Learning Limited, 2014
18. M. Chemuturi. **Mastering Software quality Assurance: Best Practices, Tools and Techniques for Software Developers.** J. Ross Publishing, 2011
19. G.E. Nenad Medvidovic. **Software architecture and mobility: A roadmap.** *Journal of Systems and Software*, Vol. 83, No. 6, 885-898, 2010
20. A.E. Hassan and T. Xie. **Software Intelligence: The Future of Mining Software Engineering Data.** *18th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. Santa Fe, 2010
<https://taoxie.cs.illinois.edu/publications/foaser10-si.pdf>
<https://doi.org/10.1145/1810295.1810451>
21. W. Nwankwo and K.E. Ukhurebor(2019). **Small and Medium-Scale Software Contracts: From Initiation to Commissioning.** *International Journal of Scientific and Technology Research*, Vol. 8, Issue 12, pp.1554-1563
22. W. Nwankwo and K.E. Ukhurebor. **Web Forum and Social Media: A Model for Automatic Removal of Fake Media using Multilayered Neural Networks.** *International Journal of Scientific and Technology Research*, Vol. 9, Issue 1, 2020
23. C. Umezuruike, W. Nwankwo, P. Tibenderana,, John Patrick Assimwe, Muhirwa Ronald. **Corona Virus Disease (COVID 19): Analysis and Design of an Alert and Real-time Tracking System.** *International Journal of Emerging Trends in Engineering Research*, Vol. 8, No.5, 2020
<https://doi.org/10.30534/ijeter/2020/41852020>
24. B. J. Hutagaol, T. Mauritsius. **Risk Level Prediction of Life Insurance Applicant using Machine Learning.** *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, Issue 2, 2020
<https://doi.org/10.30534/ijatcse/2020/199922020>
25. A.A. Mahule, A.J. Agrawal. **Hybrid Method for Improving Accuracy of Crop-Type Detection using Machine Learning.** *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, Issue 2, 2020
<https://doi.org/10.30534/ijatcse/2020/209922020>