Unveiling Mapping Algorithm for a Computerized Academic Workload Sharing Model in Higher Institutions

J.A. Adebisi
Division of Engineering Tech.
University of West Alabama
United States

C.A. Chukwulobe
Electrical and Computer Engr.
University of Lagos,
Lagos, Nigeria

K.A. Abdulsalam
Electrical and Computer Engr.
University of Lagos,
Lagos, Nigeria

ABSTRACT
In modern day academic settings, one of the most challenging activities in ivory towers is academic workload sharing. It is a critical activity in most universities and other higher institutions of learning. It is more crucial with the rising effect of staff turn-over in academics. This activity is usually committed to dedicated staff members of the faculty who is in charge of distribution, allocation and de-allocation under the instruction of the headship. It is more challenging as staff members tends to associate workload stress and burden to those in-charge hence the need for a computerised model of allocating workloads to academic staff in a seamless and efficient manner. The new mapping algorithm proposed in this work will eradicate most existing manual approaches, which is connected to slow pace of allocation, inconsistency, mismatch of areas of specializations among others. This work developed a model, designed a software engineering approach and implemented a prototype system in form of an application uses a typical the university settings in Nigeria as a case study. The developed software system uses a suitable mapping algorithm to share workloads in a more user friendly and smart methodology. Analysis of the workload allocation results revealed that the accuracy depends on the number of available academic staffs qualified for the individual workloads and subject areas assignment. The developed algorithm was implemented and tested using lecturers and module data from a standard curriculum in which, most of the activities recorded an accuracy of 100% in most cases with a few margins of less 2% using randomization techniques.

Keywords
University Workload, Prospectus, Randomization, Course assignment, Higher Education.

1. INTRODUCTION
Higher institutions are institutions that offer tertiary education leading to award of an academic degree [1]. In other words, institutions of higher learning are regarded to as universities, colleges, institutes of technologies and other career institutions that awards qualification upon successful completion of a course of study. Workloads are allocated to academic staff for teaching students during their course of study. Each departments allocates these activities to lecturers and the workloads are delivered in courses every session. Although, there are several other workload activities but this work focuses on the course allocation algorithm. In every higher institution, there is always a course allocation system that distributes workload in terms of courses to lecturers [2-3]. Different frameworks have been provided for the development of workload allocation arrangements at the school or faculty level. For instance, in a broad-based university, there is an academic unit in charge of allocating courses to lecturers in various departments. The purposes of academic workload allocation are to ensure that courses are allocated in a manner that is consistent with faculty priorities, there is balance to activities assigned to staff, there is a transparent process of allocation with clear communication of the result and the need for efficiency of academic staff deployment being balanced with ensuring quality of research, teaching and service.

In [4], the efficiency of course allocation was considered in China which underscore the fact that, workloads are occasionally allocated in a manner that reflects the mission, vision and values of the institution and also takes into account all areas of research that are expected of staff. However, there are still some challenges in workload allocation. Reconciliation of differences in academic discipline, specializations and differences in perception of relative effort and/or value of various types of academic workload. This work demonstrated how courses and workload can be shared among lecturers in a way that would match their areas of specialization using a user-friendly web application. The problem with the current workload allocation method used by most academic units is not completely reliable for successful conduct of the courses because of its manual nature. Some of the new courses are allocated to lecturers that are not specialized in the area. This brings about the issue of lack of flexibility in the standard academic contract inhibiting the optimization of academic workload. Sometimes, when a session resumes a lecturer can be told to take a course and the lecturer only have limited time (one week) to prepare for the course which in turn impacts productivity from both the staff involved and the students [5].

1.1 Theoretical Framework
This section gives an insight into some mathematical models leveraged in this work and their relevance in how workload distribution and allocation is achieved. It further elucidates the technologies used eventually used to build the solution this research has to offer. The workload sharing application works by allocating courses to lecturers and storing all the information in a database [6]. The process by which a course is allocated to a lecturer is considered as mapping within the context of this work. Figure 1 presents a simple association of how courses are mapped to individual lecturers.

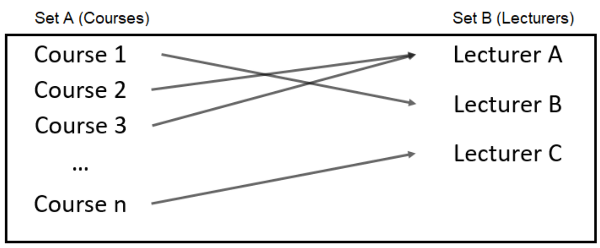


Figure 1: Course association with the lecturers

Mapping is a function that is associated with structures which are a set constituting the codomain [1], [6]. It is a special case of a relation in which there is only one output associated with each input. A map is a continuous function in topology whereas in linear algebra, it is a transformation. Mapping is done by assigning to each object in a set a particular object in another (or the same) set [7-8].

In computation, a map refers to the binary higher-order function that takes a function f and a list [v_0, v_1, …, v_n] as parameters and returns [$f(v_0)$, $f(v_1)$, … $f(v_n)$], where $n \geq 0$. For example, the higher order function, f1, used to map a list with the lower order associating function, f0, is illustrated in the code segment as follows[9];

```
f1(f0, list [])

{...return [f0(v0), f0(v1), ..., f0(vn)]}
```

To complement mapping is the randomization concept, which is very important to the work under study. Randomization is a deliberate haphazard arrangement of observations that stimulates chance according [8]. Machines and computers achieve this by generating random numbers using a random number generator (RNG). A random number generator is a device that generates a sequence of numbers or symbols that cannot be predicted fairly better than by chance at random [10], [11]. Different applications of randomness have resulted to the growth of several distinct techniques for generation of random data, some of which have existed since ancient times, among whose ranks are well known "classic" examples, including the coin flipping, rolling of dice, the shuffling of playing cards etc. The linear congruential generator, which utilizes the recurrence is indicated in equation 1;

$$X_{n+1} = \left(aX_n + b\right) \bmod m \qquad (1)$$

In this research Pseudo-Random Number Generators (PRNG) number generation concepts where a, b and m are large integers, and X_{n+1} is the next X as a series of pseudo-random numbers is used. In other words, value of n must be even. The seed should be filled out with zero to the left in order to create an even valued n-digit. Similar rendered algorithm was used by [12].

2. EMPIRICAL REVIEW

This section considers empirical reviews of similar research efforts. Several researchers have carried out studies on workload allocation and some of the models have been underscored by selected institutions use to conduct their allocation. Surveys based on the impact of several models on the students taking the courses are also investigated in this research. Review of some of the previously proposed frameworks which uses web development platforms such as .Net were also reviewed and adapted in the methodology of this work. The work of [13], emphasised majority of higher institution are guided with policies on sharing workload and its practices however limited to the head of academic departments or other assigned staff. Most times allocation of workloads is seen as an operational level lower than other duties given to the centrality of staff within the university. Allocation of workload is strategic in process and if gone wrong, it causes imbalance in the system [2], [13]. In their work they further differentiated between discretionary inputs from Head of Departments (HODs) and the model they developed towards fine tuning allocations for personnels. This proposed algorithm used a case study that will eliminate the strong imbalance between technical and social problems associated with workload allocation.

Regarding student learning assessment, the works in [14], [15], [16] examined how the significant challenges that work-integrated learning practitioners are facing with respect to academic workload including student perceptions. An Australian university was investigated to reflect the implications of work-integrated learning assessment for teaching in classes and a survey was carried out over a period of 24 months using the responses of 30 staff members during interview. The survey analysis data indicated that student learning evaluation is the biggest single contributor to academic workload in work-integrated teaching classes. The result of their research shows that there were a range of reconciliation acts that need to be discussed at individual, departmental and institutional level if the workload involved in assessing work-integrated learning were to be sustainably managed.

In [17], investigation on how changes have gone in the academic sector over the past twenty years due to various pressures were conducted but no record of proposal was provided. The research questions in [17] paper was tailored towards software developers' acuity on the primary advantages and constraints of designs that are responsive in nature in line with course allocation systems. They queried the advantages and constraints of responsive design of practitioners who have been involved in software development activities for several years. They further inquired; if the advantages and constraints of responsive design are different for professionals with more experience in developing websites or those in different sectors cold contribute greatly to course allocation platforms. The report later turned out that a good user experience is required from software developers end to achieve an efficient course allocation system.

It was emphasized by [17-19] that most workload allocation models are not improving the efficiency and productivity impact in universities. The ssurvey performed by [17] obtained data from Australian universities regarding the real models available for research. After using 'system and software' to further analyse the study. In terms of platform choice for an effective workload sharing solution, authors in [20] researched on .Net framework; it was observed that the .Net framework and the test-first strategy provides synergy for the development of high-quality database applications in the fast-changing vibrant globe. The research emphasized that using Microsoft Visual Studio .Net framework as one of most popular tools to develop enterprise applications could achieve better efficiency in course allocation. This is due to its technique for converting data models between relational databases and object-oriented programs. An investigation on the factors associated with stress among academic staff in Nigeria universities was carried out in [21]. The study was conducted to determine the difference in stress rates between distinct educational employees and academic workload along with student-related problems, research and career development, interpersonal relationships among others. Administrative-related problems were found to be substantially associated with stress based on the research. It was stated that academic rank was one of the stress and socio-dynamic variables with higher academic ranks are more likely to experience less job pressure, as well as work-related stress than those with lower academic ranks. Authors came up with recommendations that academic staff should be given more reasonable workload to manage in order to reduce stress. They should not be given more than a certain number of workloads so that the whole semester would be handled effectively [22], [23].

A study carried out in Australia by [24] examined how academic workloads are being shared among individuals according to gender in Australia. The problem addressed in Australian universities were related to more faculty members were male than female, and in both academic and non-academic positions they continued to outnumber women in senior ranks. The senior staff profiles of two universities were selected for the research. It was observed that different workloads were not always evident between genders. Similarly, [25] carried out research on Workload Allocation Models (WAMs) and how it categorizes academic work into separate operations, along with some grounds for comparison. The basis was found to be often numerical: the model suggested a standard unit of work – for instance presenting a one-hour lecture, and other activities were graded on that basis. For example, preparing (as opposed to presenting) a one-hour lecture might count as two units within the context of the authors findings. It was observed by the author that it was a very sensible way of ensuring fairness in the distribution of work amongst diverse academics – particularly in departments or schools with a variety of academic disciplines, or with a broad gap between teaching-focused and research-focused staff. Other authors in [18], [26], [27], [28] recently contributed in different capacities reflecting challenges encountered in workload allocation.

Following a comprehensive review of related works, it was observed that most universities still use manual allocation models for academic workload sharing with limited automation and a mix of political undertone [16]. The drawbacks of most of the models is that they give some academic staff workloads that do differ with their individual field of research and other academic specialities and it have brought about some differences with the lecturers along with student challenges in the learning environment. Although, this work focusses more on implementing a more robust stable and efficient algorithmic platforms that will allow flexible customization for decisions on which workload gets assigned to academic staffs based on some certain level of priorities.

3. METHODOLOGY

Sharing of academic workload using a combination of mapping technique accompanied by an algorithm is considered in this work. This algorithm helps determine the mapping pattern and randomization technique. The algorithm implements web applications through the means of client server communication [20], [29]. This methodology addresses first understands the courses that are being allocated to the lecturers in line with their required field of study. The content of the courses is also useful in determining the area of specialization that a lecturer or a group of lecturers should have in order to conduct the course for the semester/session. In some cases where there are no lecturers or other academic staff with the specialties available for the courses, the algorithm checks for closer and set of similar specialization to handle allocation. This method focusses on the development of a client/server architecture web application. In this architecture, the client is the computer used for sharing the workload among the academic staffs and the server is the web server that contains the databases of the workloads and academic staff. The client initiates the request to share workloads and the web server processes the request by sending the appropriate response which is the course lecturer as a result. The application is designed to accommodate course list in a central database sorted based on workload sizes indicating the number of units and other parameters. The integrated database server contains lecturers' information regarding their qualification and area(s) of specialization, list of courses offered in selected departments and their required specialties for the courses to be offered. The workload sharing methodical architecture supports real-time as presented in figure 2.

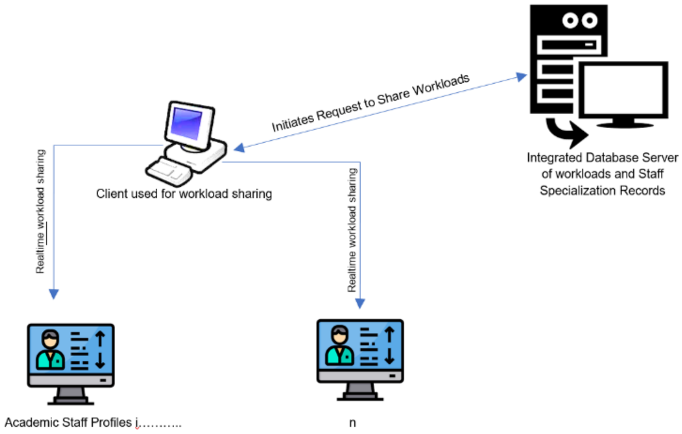


Figure 2: workload sharing methodical architecture

The workload sharing model describes the implementation with the use of Universal Model Language (UML)[30]. The UML provides a visualized system's architecture. The structure diagram emphasizes components that must be present in the system being modelled. The structure diagram of this system comprises of class diagram, component diagram and object diagram. It further establishes the type and object descriptions with their relationships. In the application, two major classes (course and academic staff) serve as the base. The course class consist of attributes that describe the course and the fields of study/research required for it to be lectured by an academic staff. The class designed for academic staff consist of meta-data such as name, qualification, designation and field of research each of which is an attribute that is used to describe a lecturer in a standard system. A typical class and object diagram is presented in figures 3.

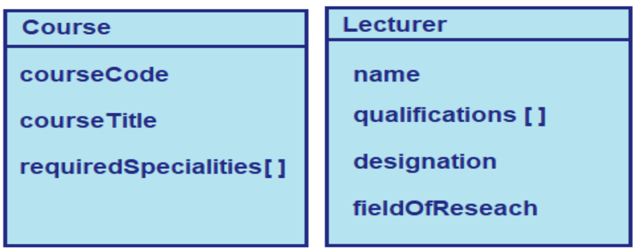


Figure 3: Course and Lecture class diagram

The corresponding object diagram for course and lecturer which is referred to as a graphic description of the instances in a system and the relationship that exists between them is presented in figure 4. A case study of a typical physics department was used as an instance. Some object diagrams that show a course (Classical Mechanics) offered in the department and a dummy academic staff member of the department are associated.

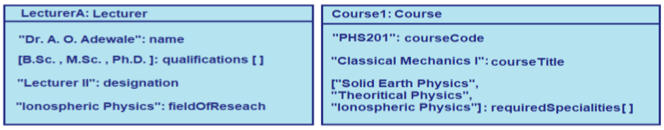


Figure 4: Lecturer and course object diagram

The information in the object diagrams is a dummy from a typical university programme template [31- 32]. The component diagram which represents the model for insight of the metadata structure of the system is illustrated in figure 5. The components in this design are the databases that stores the information about the courses and lecturers, the allocator otherwise referred to as the mapping algorithm -maps a course to a lecturer and the databases holds the information about the courses that have been mapped and the lecturers already assigned.

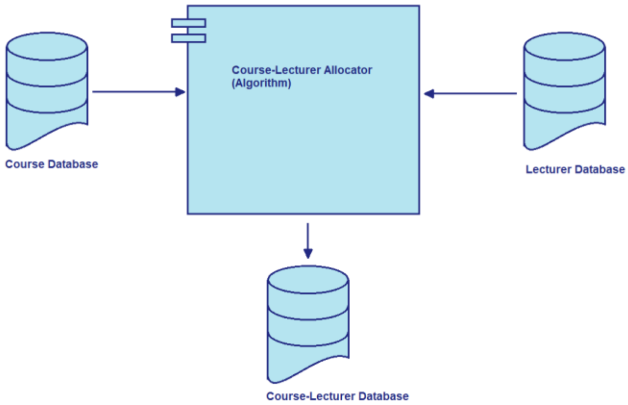


Figure 5: Workload allocation component diagram

The behaviour representation depicts the elements that are dependent of time and show the dynamic concept using the activity diagram presented in figure 6. It explains the workflow of the academic workload sharing application.

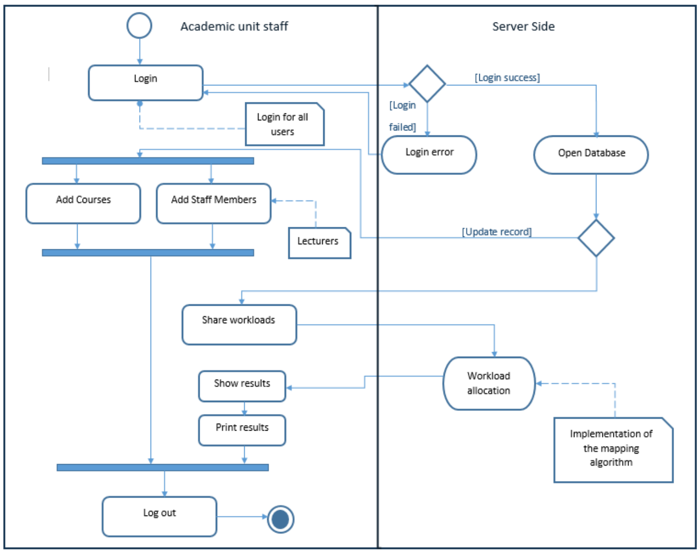


Figure 6: Activity diagram of the web application

Tables 1 and 2 shows the description of the databases for both course and lecturer objects while table 3 describes the corresponding metadata of the tables.

Table 1: Description of the course database

ID	course_code	course_title	required_specialities
1			
2			
…			
n			

Table 2: Description of the lecturer database

ID	name	qualifications	designation	field_of_research
1				
2				
…				
n				

Table 3: Database attributes (column descriptions)

Column name	Data Type	Description
ID	Integer	This is what uniquely identifies an object (course or lecturer). It is an integer.
course_code	String	course code of a course.
course_title	String	full title of a course.
required_specialities	String	required area of specializations needed by lecturers to have in order to be qualified to handle the course.
name	String	name of the lecturer
qualifications	String	academic qualifications of a lecturer.
designation	String	the position of a lecturer.
field_of_research	String	field of research of a lecturer.

3.1 Mapping Algorithm

The algorithm for course-lecturer allocator component is sub-divided into main, map-function and the programming implementation.

Main algorithm:

Step 1: Store course objects in a list, C
Step 2: Store lecturer objects in another list, L
Step 3: Pass the C to a map function that returns the final list, C-L that stores the objects that describes the courses with the lecturer that's most qualified for the job.
Step 4: Store the C-L list to the course-lecturer database.

Map function algorithm:

Step 1: Create a third list. This list would hold the objects that have both courses and lecturers assigned to them.
Step 2: Iterate through the list of lecturers and examine the ones that have the required specialties for the first course.
Step 3: If the number of courses the lecturer is already given is ≤ m, (where m is the maximum number of courses a lecturer can take in a semester)
Store the course and the lecturer object as an element in the course-lecturer list. else,

Check for the next lecturer object in the lecturer list until the last lecturer object is scanned.

Step 4: Check for the courses that have not yet been added to the third list. This is due to the fact that the lecturers available for them have [9]. Use the method of generating pseudorandom numbers that's guided by the recurrence equation

$$X_{n+1} = (aX_n + b) \bmod m$$

Where $a = t^3$; $b = t^2$; $m = t$

n is an integer (0, 1, 2, 3, …); X_0 = number of lecturer objects t = time user logged in (seconds)

X_{n+1} = the next random number generated in the recurrence. The lecturer object that would be assigned to the course object that is unassigned is the lecturer object with ID = $X_{n+1} \bmod X_0$

Step 5: Return the list created in Step 1 (Course – Lecturer list) so that it would be available in the main function.

A detailed representation of the mapped function process is shown in figure 7.

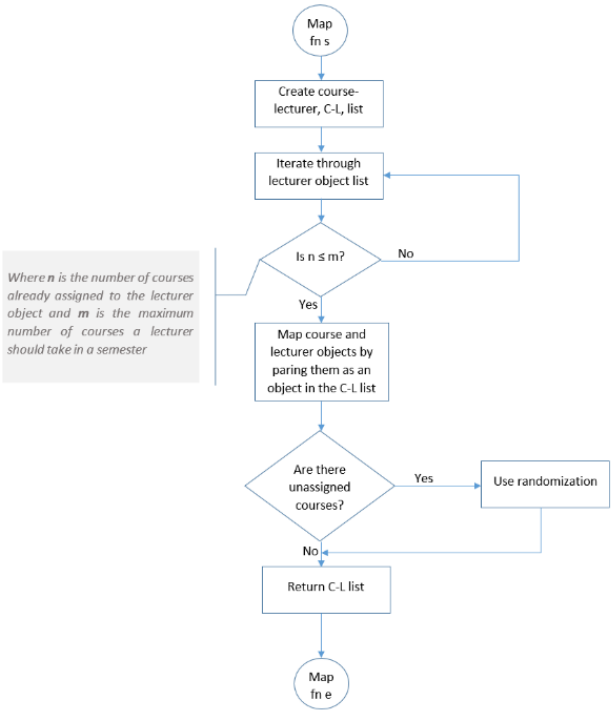


Figure 7: Flowchart showing the map function process

4. IMPLEMENTATION AND RESULTS

Having discussed the appropriate methodology in the previous section. A detailed simulation of the proposed architecture was carried out using a console application written in C# to simulate the course sharing function based on the mapping algorithm. Except of the simulator code is show in figure 8. The simulator consists of three classes for course, lecturers and 'course lecturer'. The course class and lecturer class were created based on the class diagrams designed in the methodology section. The 'course lecturer' class holds the details of a course and the assigned lecturer. The member variables of this course-lecturer class are objects of the course class and lecturer class. The program also consists of a function that maps lecturer objects to all the elements in an array of course objects, a function that assigns a single lecturer object to a single course object. In addition, is the main function that runs program, create objects, fill with data on courses and lecturers, then calls the mapping function and prints the result on the console output shown in figure 9. Subsequently, more classes were created for general courses and the objects that would map several lecturers to the respective 'general course' because there are some courses (which may be general in most cases) that would require more than one lecturer to handle due to the fact that students from other departments register for the course. C# was chosen for this simulator so that the code could just be transferred to the main ASP.NET project without much modification since ASP.NET technology makes use of C# for webpage functionality, the syntax for the console application and the course sharing part of the web application which ran on the server using the same logic.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace FYP1
{
    class Program
    {
        //Some info determined from the database
        public const int max_number_of_courses = 10;
        public static int number_of_courses = 8;         //Number of course objects
        public static int number_of_u_courses = 1;
        public static int number_of_lecturers = 10;      //Number of lecturer objects

        public static int number_of_general_courses = 1;


        class Course
        {
            public int ID;
            public string course_code;
            public string course_title;
            public string[] required_specialities;

            public int number_of_specialities;

            public bool assigned;

            public Course()
            {
                required_specialities = new string[10];
                assigned = false;
            }
        }
```

Figure 8: Simulator code except

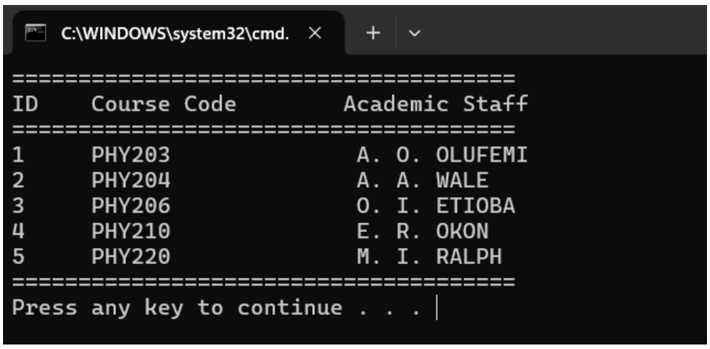


Figure 9: Console output of the Mapping Simulation

After successful simulation a web application is then developed for proper implementation using HTML, CSS and JavaScript. A combination of other tools includes C#, ASP.NET and MySQL for the back-end implementation. These tools assisted with extensions used for Updated views, Cross-platform support, Embedded database library, Stored procedures, Triggers, Cursors, Partitioned tables with pruning of petitions in optimizer and Native storage engines. Other requirements in required during testing is the visual studio integrated development environment (IDE) for building the application. Google Chrome web browser was used to testing and evaluation due to its extremely fast nature; It loads and displays web pages very quickly; it has a very efficient built-in tool to help web developers and it supports the use of plugins to add more functionalities. MySQL Server - an open-source database management system was then used to manage the databases that hold the information about the courses and lecturer.

The hardware that is essential to run this application is a Personal Computer - PC with the following minimum specifications: Processor – Intel Core i3 or later/AMD, Clock Rate - 2.0GHz or more, RAM – 4GB or more, HDD – At least, 250GB, Cache Memory – 5MB or more, Monitor – 1280 x 720 (HD) screen resolution or higher as well as a windows Operating System – Windows 7-11.

4.1 Server and Database Setup

The created database consists of tables each of which is built for courses, lecturers and various other relationship components. The character length was set to 50 for the strings columns except qualifications which was set to 200 because the information about that field could be long especially when an individual has more than five qualifications. Null type had to be permitted for the qualifications column because not all the academic staffs was obtainable from the database. For the course table, 10 separate fields were created for the required specialties in order for them to be assigned to the elements in the array of specialities. The database tables and their resulted properties are shown in figures 10 and 11.

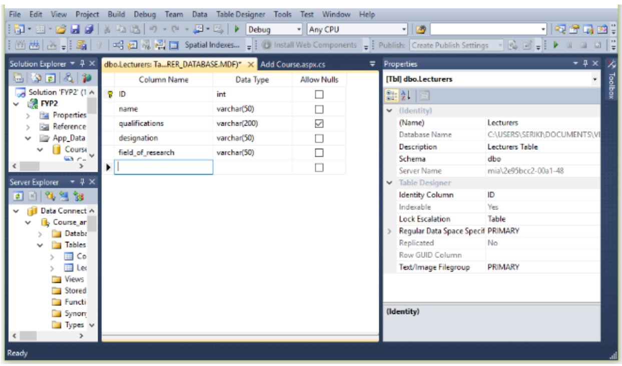


Figure 10: Database Interface - Lecturer table.

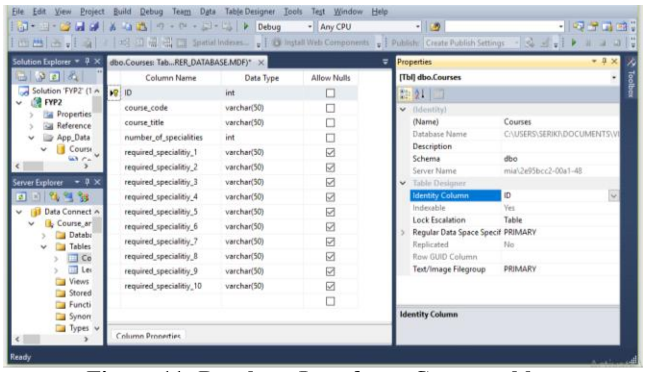


Figure 11: Database Interface - Course table.

During the experimental setup, Microsoft SQL server was installed for the database to be connected in real-time during course allocation. It is important to note that prior to the installation, previous versions of .NET framework (3.0 and 3.5) were executed. The login interfaces and the basic configuration required for the implemented web application using ASP.NET and styled with CSS are shown in Figures 12 and 13.

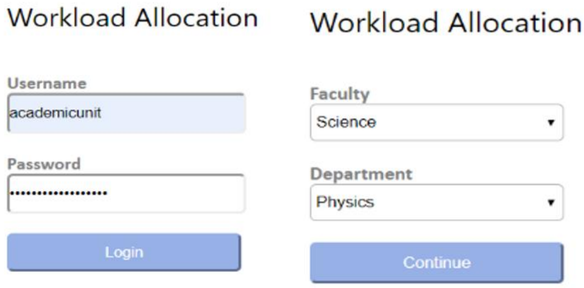
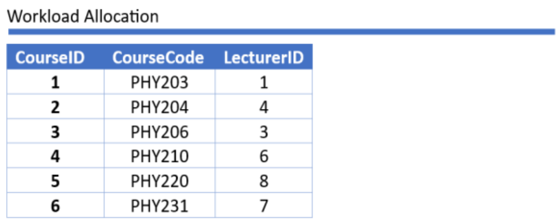


Figure 12: Login and basic configuration page (Select Faculty and Department)



Workload Allocation

CourseID	CourseCode	LecturerID
1	PHY203	1
2	PHY204	4
3	PHY206	3
4	PHY210	6
5	PHY220	8
6	PHY231	7

Figure 13: Workload/Course allocation by Lecturer IDs

The mapping function is invoked whenever the 'Share Workload' button is activated. This is where the workload allocation takes places. A typical department of physics is used as a case study for the purpose of testing and evaluation. Table 4 and 5 consists of dummy data of 200 level courses offered in the department and a dummy sample academic staff record.

Table 4: Data of 200 level physics courses offered in first semester [31]

Course code	Course title	Units	required specialization Code
PHS201	Classical Mechanics I	2	8
PHS203	Electricity & Magnetism	2	6
PHS204	Modern Physics	2	4
PHS206	Electronics 1 A	2	5
PHS210	Properties of Matter	2	5
PHS220	Practical	1	8

Table 5: Sample Dummy Academic Staff (Lecturers)

ID	Name	Qualification	Designation	Field of research [Code]
1	XYZ, M. A.	B.Sc. (Physics), ABU, M.Sc. (Physics), Switzerland, Ph.D. (Physics), USA	Professor	Solar Physics [8]

Single lecturer will not be mapped to courses that requires multiple specialization and team of lecturers; this is also similar approach for courses offered from another department. The algorithm is flexible and can be configured on the server for special cases of unmapped courses. Data from course information and lecturer's tables were passed into the database for workload sharing model implementation and the results obtained after mapping are presented in Table 5. The level of accuracy was measured according to how qualified the academic staff is to take the course from the manual benchmarks. Since the courses has list of required specialities, the ones that have perfect matches the first required field of study gets 100% accuracy for the course. If the lecturer's field of research matches the second required specialty, then the accuracy reduces by 5% which makes it 95%. If the lecturer's field of research is the third, the accuracy reduces by 10% and this could be re-configured based on user intention. If there are no lecturer available for the course due to lack of matching field of research – specialities, the accuracy drops by 50%. Accuracy of general courses are always 100% because they do not require someone from a specific field of study to take them. The courses are classified into generic or non-generic courses in the G/NG column of table 5. G represents the general courses while NG represents the non-general courses.

Table 5: Excerpt of mapped results from the application indicating level of Accuracy

Course Code	Course ID	G/NG	Lecturer ID	Accuracy
FSC105	1	G	5	100%
PHS101	2	G	9	100%
PHS102	3	G	11	100%
PHS201	3	NG	6	100%
PHS203	3	NG	8	100%
Average Accuracy				90%

A further breakdown of courses taken by individual lecturers are also achieved by the varieties of reports obtainable from the applications. The corresponding chat showing the overall accuracy of the application performance is shown in figure 14.

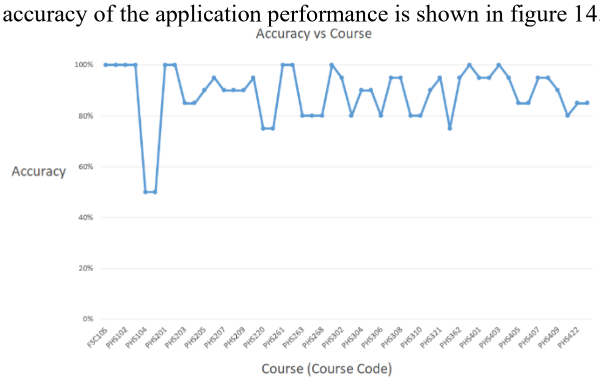


Figure 14: Chart showing all the mapped courses and their respective percentage accuracy

4.2 Discussion

Universities globally are putting in efforts to promote transparency in workload allocation as much as possible. This work has provided a more accurate means by which workloads can be shared at the departmental level with more flexible features and varieties of customized reporting tools. The result of over 90% accuracy obtained from the mapping algorithm is quite commendable. The mapped courses with lower accuracy were due to the fact that the most qualified lecturers were not available as the program could not find a match, it had to result to randomization. To this end, the application could also be used to identify scare resources in the department for possible employment opportunities to fill the gap. In other words, there is an indication that qualified lecturers contributes to higher accuracy. Over 5,000 records were used to test the system and the accuracy was maintained with varying parameters. This signifies that the application has the capacity to store data of several other departments without overloading the server. Therefore, the application can be used for the entire university with departmental clusters and administrative profiles. The result of this work can also be used to identify redundant resources in case of lectures with idle specialities. Overall, a complete data migration and integration from the existing systems based on required records (Course code, course title and required specialities) and academic staffs (lecturer name, qualifications, designation and field of research) into the centralized database as indicated in the architecture rather than individual data entry. This workload sharing model can be used by any higher institution of learning as it is configurable for universities.

5. CONCLUSION

Workload allocation is an important factor in the learning environment and it is a sophisticated process that involves careful means of selection and assignment. The means at which the workload is being shared is guided by an efficient algorithm which ensures that the most qualified academic staff is being allocated to a particular workload. The set, map and randomization mathematical theories proved useful in guiding the mapping algorithm. The application was built to run on web platforms so a web application with a user-friendly graphical user interface was developed with the technologies discussed in methodology section. The entire application was modelled as a system using the Universal Model Language (UML) where the class and diagrams of course and lecturer were designed in order to model the application that implements the mapping algorithm. From the results of the final workload sharing, courses were mapped accurately. It has also been concluded that this work addressed a very critical gap in the institution of higher learning therefore meeting its objectives of minimizing time spent in discussing course and workload allocations in meetings. It also eradicates the bias among teaching staff members on which courses are allocated with a balance workload approach. The seamless integration capability is a strength of the effort put into this work. In future, expanding this system to accommodate non-teaching staff members and take care of laboratory technicians as well as making it suitable for different university could be considered. No doubt that this computerized application for sharing workload among academic staffs in a department has been successfully designed and implemented, tested and evaluated successfully.

6. REFERENCES

[1] Adebisi J.A, B. O. Moses, and others, "The Role of Network Technologies in the Enhancement of the Health, Education, and Energy Sectors," Network and Communication Technologies, vol. 7, no. 1, pp. 1–39, 2022.

[2] G. Bissias, C. Cousins, P. N. Diaz, and Y. Zick, "Deploying Fair and Efficient Course Allocation Mechanisms," arXiv preprint arXiv:2502.10592, 2025.

[3] A. Bezuidenhout, "Implications for academic workload of the changing role of distance educators," Distance education, vol. 36, no. 2, pp. 246–262, 2015.

[4] T.-T. Zhu, F. Yang, and Y.-J. Zhang, "Evaluating the efficiency of online course resource allocation in universities of China," Eval. Program Plann., vol. 107, p. 102481, 2024.

[5] M. A. Munim and M. J. I. Mony, "Design of a Fair and Scalable Course Allocation Framework for University Faculty," Int. J. Comput. Appl., vol. 975, p. 8887, 2025.

[6] M. Triossi, A. Romero-Medina, and others, "Strategic priority-based course allocation," J. Econ. Behav. Organ., vol. 226, 2024.

[7] N. Jamil, R. Rosli, and M. S. Mahmud, "Algebraic trends impact school mathematics education: A bibliometric review," Multidisciplinary Reviews, vol. 8, no. 9, p. 2025261, 2025.

[8] S. Stewart and J. Troup, "Linear Algebra Education: A Scoping Review," ZDM--Mathematics Education, pp. 1–13, 2025.

[9] J. Von Neumann and others, "Various techniques used in connection with random digits," John von Neumann, Collected Works, vol. 5, pp. 768–770, 1963.

[10] N. R. Council et al., The mathematical sciences in 2025. National Academies Press, 2013.

[11] M. Haahr, "Random numbers," 2019. [Online]. Available: https://www.random.org/randomness/

[12] D. E. Knuth, Art of Computer Programming, Volume 4, Fascicle 4, The: Generating All Trees--History of Combinatorial Generation. Addison-Wesley Professional, 2013.

[13] L. C. Barrett and P. S. Barrett, "The management of academic workloads: Full report on findings," 2008.

[14] A. A. Bilgin, A. D. Rowe, and L. Clark, "Academic Workload Implications of Assessing Student Learning in Work-Integrated Learning.," Asia-Pacific Journal of Cooperative Education, vol. 18, no. 2, pp. 167–183, 2017.

[15] L. T. Soriano, J. N. B. Capucao, and T. D. Palaoag, "Decision Support Model for Managing Academic Workload in State Universities and Colleges," International Journal of Information and Education Technology, vol. 8, no. 7, pp. 491–495, 2018.

[16] L. Bachman and C. Bachman, "Student perceptions of academic workload in architectural education," J. Archit. Plann. Res., pp. 271–304, 2006.

[17] R. Watson, R. King, S. Dekeyser, L. Baré, and C. Baldock, "Current practice in academic workload allocation processes in Australia," in Tertiary Education and Management Conference 2015: Refereed Papers (TEMC 2015), 2015.

[18] T. Mgutshini, J. Murray, and A. Mavundla, "Workload Allocation and Postgraduate Research Supervision: From praxis to theory development," in Higher Education Forum, 2025, pp. 99–124.

[19] Y. Wu, "Workload or Education Development in Global Context," in Proceedings of the 2024 3rd International Conference on Educational Science and Social Culture (ESSC 2024), 2025, p. 475.

[20] J. R. Shah, Lee S M R Mehta, "Test Driven Development in the .Net Framework. In," in The Proceedings of the Conference on Information Systems Applied Research 2008, 2008. [Online]. Available: https://iscap.us/proceedings/conisar/2008/4113/CONISAR.2008.Lee.ppt

[21] J. T. Akinmayowa, P. A. Kadiri, and others, "Stress among academic staff in a Nigerian university," Covenant journal of business and social sciences, vol. 6, no. 1, 2016.

[22] S. Dekeyser, R. Watson, and E. Baré, "Comparing academic workload models: How Australian universities resource teaching activities," in Tertiary Education Management Conference: From Rhetoric to Reality. Retrieved from https://www. atem. org. au/documents/item/576, 2016.

[23] M. M. Javed and M. M. Qadri, "PERCEPTION AND THE LEVEL OF AWARENESS ABOUT WORKLOAD AMONG FACULTY: A TOOL OF EXPLOITATION FOR ACADEMIC CAREER AND FACULTY DEVELOPMENT," 2018.

[24] A. Dobele, S. Rundle-Thiele, F. Kopanidis, and M. Steel, "All things being equal: Observing Australian individual academic workloads," Journal of Higher Education Policy and Management, vol. 32, no. 3, pp. 225–237, 2010.

[25] R. Hull, "Workload allocation models and 'collegiality' in academic departments," Journal of organizational change management, vol. 19, no. 1, pp. 38–53, 2006.

[26] D. K. Mayer, D. Havens, and L. Copel, "Reducing PhD Faculty Workload Through a Faculty Fellows Program," Nurse Educ., pp. 10–1097, 2024.

[27] Q. Ma, "Predicting Physician Burnout Deterioration in Medical Students: A Model based on Academic Stress, Workload and Personal Characteristics," in Proceedings of the 2025 International Conference on Big Data and Informatization Education, 2025, pp. 387–394.

[28] A. Quickfall and P. Wood, "The perfect storm for teacher education research in English universities: the tensions of workload, expectations from leadership and research," Educ. Sci. (Basel)., vol. 15, no. 4, p. 434, 2025.

[29] F. Almeida and J. Monteiro, "The Role of Responsive Design in Web Development.," Webology, vol. 14, no. 2, 2017.

[30] G. Booch, J. Rumbaugh, and I. Jacobson, "Unified modeling language user guide,(the 2nd edition)," 2005, Addison-Wesley Professional.

[31] U. of Lagos, "University of Lagos, Faculty of Science Prospectus. 2013 – 2015 Edition," 2015.

[32] U. of Lagos, "University of Lagos, Faculty of Engineering Prospectus. 2013 – 2015 Edition.," 2015.