ABUAD Journal of Engineering Research and Development (AJERD)
ISSN: 2645-2685

Volume 3, Issue 1, 23-33

The Development of a Web-based Application of Examination Seating Arrangement for Student

Sunday ADETONA[1], Edison HASSAN[1], Raifu SALAWU[2], Samuel OMOLOLA[3]

[1]Department of Electrical & Electronics Engineering, University of Lagos, Lagos, Nigeria
sadetona@unilag.edu.ng/edisonhassan@gmail.com

[2]Department of Electrical & Electronics Engineering, The Bells University of Tech., Ota, Nigeria
risalawu@bellsuniversity.edu.ng

[3]Department of Electrical & Electronics Engineering, The Federal Polytechnic, Ilaro, Nigeria
saomolola@gmail.com

Corresponding Author: sadetona@unilag.edu.ng

Date Submitted: 11/11/2019
Date Accepted: 12/03/2020
Date Published: 30/06/2020

Abstract: *The paper presents a web-based application (App) that can be used to solve the problem of "giraffing" in an examination environment by ensuring that students offering the same course are not seated together. The development of the App is related to graph colouring and k-partition problem; which was solved in this contribution with Harmony Search Algorithm (HSA). The implementation of the algorithm into the process of coding the seating generation modules was done with the use of object-oriented programming available in newer versions of PHP. The coding and design of the seating arrangement module to take multiple courses and assign them in all the classes selected was sorted out by multiple increments. The pairing of each student to individual courses was solved by the system of object-oriented programming in JavaScript. When tested, the proposed examination seating arrangement App achieved all its specified functions which include: Implementation of administrative login modules; addition of class and course data through a graphical user interface (GUI) into the database; addition of student data into the database; automatic arrangement of students writing examinations such that students writing the same examinations are not positioned together. After the arrangement has been generated, the administrator is presented with the option of printing which makes the system all the more convenient.*

Keywords: *Giraffing, PHP (Personal Hypertext Pre-processor), object-oriented programming, database, and module.*

1. INTRODUCTION

Examination malpractice is regarded as any process of exploiting wildcat methods to get unjustified advantage in an examination or test with the aim of achieving success. It may include cheating before the examination, that is, students having access to the question papers prior to the commencement of the examination, cheating during the examination which involves copying from textbooks, impersonation, and giraffing, an act of stretching one's neck in order to see and copy the work of a student seated nearby [1-3]. Malpractice may also occur after the examination has taken place, this usually involves students persuading their teachers with personal gifts and favours in order to get better grades.

The syntheses of examination wrongdoing among examinees are dreadful and annihilating; this is because it ruins the basis and framework of educational architecture. It makes formal assessments misleading, educational targets unachievable and forecasts a dim time yet to come for a given state. The coming of electronic cheating has also contributed to the problems of carrying out believable and unquestionable examinations; consequently, a holistic approach; which will involve all stakeholders, and the use of technological, and non-technological solutions, must be come up with to deal with this behemoth.

In order to prevent education sector from ruins due to examination misconducts, various suggestions have been proposed by education and technological researchers. For instance, [4] has identified three main approaches that can be used in curbing examination malpractice. These include: Ethical, Engineering and Enforcement approaches. One of the ways of applying ethical approach proposed by [4], according to [5] is by inculcating honesty and discipline into students. In line with [4] engineering approach, a multi-card operating system that can be used in tackling examination malpractice is proposed by [6]. The proposed system houses the biometric templates of the candidates for a given examination. During the examination, candidates present their cards and the verification of personality and legitimacy to write a particular examination is confirmed. This procedure, despite being quite efficient is only able to solve the problem of student impersonation.

To curb examination misconduct in Universities and high schools, [7], [8] and [9] proposed the use of e-examinations platforms; which is a method of writing an examination with the use of Information and Communication Technologies. The platform offers Multiple Choice Questions (MCQs) and Short Essay Questions (SEQs) examinations, automatic grading and quick certification. With the platform, it becomes quite impossible for questions to be leaked prior to the commencement of the examination. The proposals of [7], [8] and [9] only solve the problem of maintaining questions' integrity.

A Faraday cage that can be used to curb examination malpractice with the use of technological devices is proposed by [10]. The proposed cage creates a metal to shield the examination hall, and as such blocks electromagnetic waves from mobile phones or other electronic devices from propagating into the examination hall. The system proposed by [10] was improved upon by [11]. The system proposed by [11] has ability and capability to detect all mobile devices using the radio frequency spectrum with a wavelength of 30 cm at 872 to 2170 MHz. The detector can be brought into the examination hall and use to uncover the utilization of a mobile technology by a student by a beeping or blinking indicator. As the device gets closer to the source the blinking becomes more intense hence making it easy to apprehend a culprit. The [10] and [11] proposed systems that can only solve the problem of cheating with the utilization of cellular phones, and other communication devices in the halls where examinations are taking place.

In order to reduce complications in the allotment of examination rooms and issuing of examination dockets to examinees during examinations, [12] therefore proposed an examination hall and seating arrangement application using PHP. The docket is used in establishing the eligibility criteria of an examinee of a given department. The App proposed by [12] can only provide a way to apportion examination room for each examinee less any skirmish.

It is evident from the foregoing that curbing examination malpractice which has to do with giraffing has received little or no attention in open and technical literature. And it is worth noting that this constitutes a major problem in conduct of examination in universities and colleges; and as such, solving this problem of cheating during the examination forms motivation for this work. One of the ways of solving the problem is by making sure that no two students offering the same examination or course are positioned close to each other [3]; and this is achievable by scattering students using seating arrangement. Currently, the seating arrangement for examinations is done manually in most of citadels of learning in developing economies. The Faculty Examinations Officer (FEO) has to collect all students' information which includes names, departments, matriculation numbers, and list of registered subjects for the examinations [3]. The FEO then collates the total number of registered students, selects the classes and divides the students among the classes selected. After this, he needs to prepare a students' list for each examination. He also needs to prepare the seating arrangement list for each class based upon the count. All this work needs to be done for each instance of an examination, which is very tedious and error prone; hence the need for an automated seating arrangement. In this work therefore, a web-based examination seating arrangement application (App) is developed and proposed. The proposed App created a web-based software program that arranges students sitting for examinations in an examination hall such that no two students offering the same course are in close proximity to each other. This will help eliminate the problem identified above.

2. METHODOLOGY

2.1 Problem Formulation

The proposed App arranges students sitting for examinations in an examination room in such a way that no two students offering the same course are in close proximity to each other. It is a graph colouring and the k-partitioning problem; where the vertices of a chart are coloured in such a way that no two neighbouring vertices divvy the same colour. A basic graph colouring example is shown Figure 1.

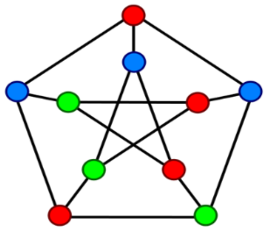


Figure 1: A basic graph colouring

The problem is likened to dividing n number of students among k number of seats available in an examination venue for different types of papers to be taken over a given period in a day; so as to ensure that the problem of cheating that has to do with giraffing is minimized. To achieve this goal, the proposed App will have to meet the following objectives:

- Students offering different courses should be seated on the same seat or adjacent to one another;
- The number of students on each seat should be same.

According to [13], the problem of building the seating architecture can then be fixed using a binary matrix $\omega_{n\times n}$. The elements of this matrix are

$$\omega_{ij} = \begin{cases} 1, & if\ students\ i\ and\ j\ are\ to\ be\ seated\ apart \\ 0, & otherwise \end{cases} \quad (1)$$

With this input matrix, the problem becomes partitioning n students into k subset of students $S = \{S_1, \ldots, S_k\}$; consequently, the objective (fitness) function becomes

$$f_1(S) = min \sum_{i=1 \forall i,j \epsilon S_t : i<j}^{k} \sum \omega_{ij} \quad (2)$$

The task of establishing the beingness of solution to this problem according to [14] is analogous to the $NP - complete\ graph\ k - colourability\ problem$. In the problem, the graph G is defined as

$$G = (V, E) \quad (3)$$

In equation (3), the V = vertex set $= \{v_1, \ldots, v_n\}$ and $E =$ the edge set $= \left\{\{v_i, v_j\} : \omega_{ij} = 1 \wedge v_i, v_j \epsilon V\right\}$; in other words, each student represents a vertex, $v \epsilon V$, and two vertices v_i and v_j are regarded to be contiguous, if and only if, $\omega_{ij} = 1$. It can therefore be stated that, colours correspond to seats, and the task is colouring G using k colours.

The objective function defined in equation (2), provides solution to the first objective of realizing the goal of the study. The solution to the second objective, that is, making sure that the number of students on each seat is the same can be achieved using this objective function [14]:

$$f_2(S) = \sum_{i=1}^{k} \left(min \left(\left| \tau_i - \left\lfloor \frac{n}{k} \right\rfloor \right|, \left| \tau_i - \left\lceil \frac{n}{k} \right\rceil \right| \right) \right) \quad (4)$$

In equation (4), τ_i represents the number of students assigned to each seat i; and is defines as

$$\tau_i = \sum_{\forall v \in S_i} S_v \quad (5)$$

2.2 Implementation of the application

2.2.1 Algorithm

There are many algorithms that can be used to solve the problem at hand like the Tabu Colony, Partial Colony, Greedy, Degree of Saturation (Dsatur), Hybrid Evolutionary, Ant Colony, Hill-climbing, Backtracking Dsatur, and Harmony Search algorithms [13-14]. In this contribution, the Harmony Search Algorithm (HSA) is used because of its simplicity and search efficiency [15] [16].

The HSA is inspired by music phenomenon; where musical performances seek to find pleasing harmony as determined by an aesthetic standard. In music extemporisation, each player sounds any pitch within the possible range, together making one harmony vector. If all the pitches make a good harmony, that experience is stored in each player's memory, and the possibility to make a good harmony is increased next time [15]. In order to use the HSA, the following parameters must be set: Harmony Memory Size (HMS) which ranges from 10 to 100, Harmony Memory Consideration Rate (HMCR) which ranges from 0.7 to 0.99, Pitch Adjustment Rate (PAR) which ranges from 0.1 to 0.5, the difference between two adjacent values in case of discrete variable δ, and Fret Width (in case of continuous variable) FW [15] [16]. Figure 2 presents the steps of the HSA [15] [17] are as follows:

Step 1: Initialize the HSA parameters
Step 2: Initialize the harmony memory (HM). This can be actualized by
Generating randomly vectors x_1^1 ... x_n^{hms} as many as HMS,
Storing vectors generated in HM in matrix form,
Evaluating the fitness function $f(x^k)$ corresponding to each solution vector:

$$\text{HM} = \begin{bmatrix} x_1^1 & \ldots & x_n^1 & |f(x^1) \\ \vdots & \ddots & \vdots & |\vdots \\ x_1^{hms} & \ldots & x_n^{hms} & |f(x^{hms}) \end{bmatrix} \quad (6)$$

Where, x_i^k = seat of the i^{th} students in the k^{th} seating arrangement; and $f(x^k)$ = fitness function,
Step 3: Improvise a new harmony from the HM with the probability$HMCR$ $(0 \leq HMCR \leq 1)$,
Pick the stored value from HM with probability of PAR $(0 \leq PAR \leq 1)$,

Modify the solution chosen with the bandwidth (BW) value using:

$$x_i^{k+1} = x_i^k + rand \times BW \tag{7}$$

Step 4: Is x_i^{k+1} better than the x_i^{worst} in HM? Replace x_i^{worst} with x_i^{k+1}
Step 5: Repeat step 2 to 4 until termination criterion ($when\ i \geq NI$) is satisfied.

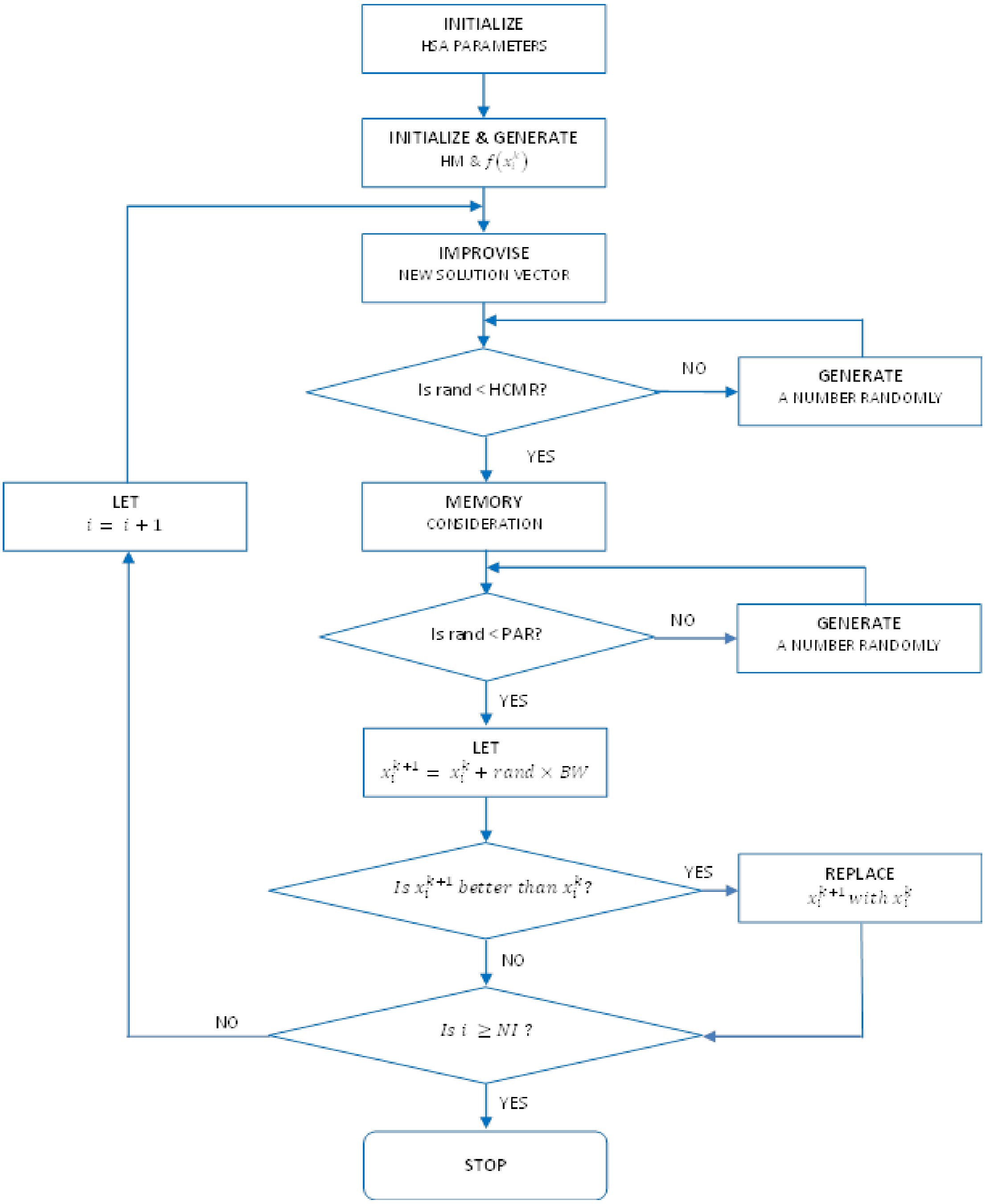


Figure 2: The flowchart for HSA

2.2.2 Program

The implementation of the algorithm into the process of coding the seating generation modules was done with the use of object-oriented programming available in newer versions of PHP. The PHP is deployable in this contribution because it can be used on many web servers and operating systems; and can be used with many relational database management systems (RDBMS).

Figure 3 presents a sequence diagram for the App, which shows how the four essential objects that are involved in operations of the App are sequentially arranged, and their timely interactions. It is evident from the figure that the App is linked to the database, Administrator (Admin), and printer. The function of the database is to hold the details of students' information, various halls for examinations, and courses to be examined in the examinations. The needed students' information includes; names, matriculation numbers, and list of registered courses for the examinations. The Admin has shown in the figure, creates or updates the data in the database. He instructs the App to generate the seating arrangements; and also gives details of what the App is to be done after it generates the seating arrangements, either to only view or print the generated seating arrangements. In a situation where there is need to print the seating arrangement, the App sends the information to the printer for the execution.

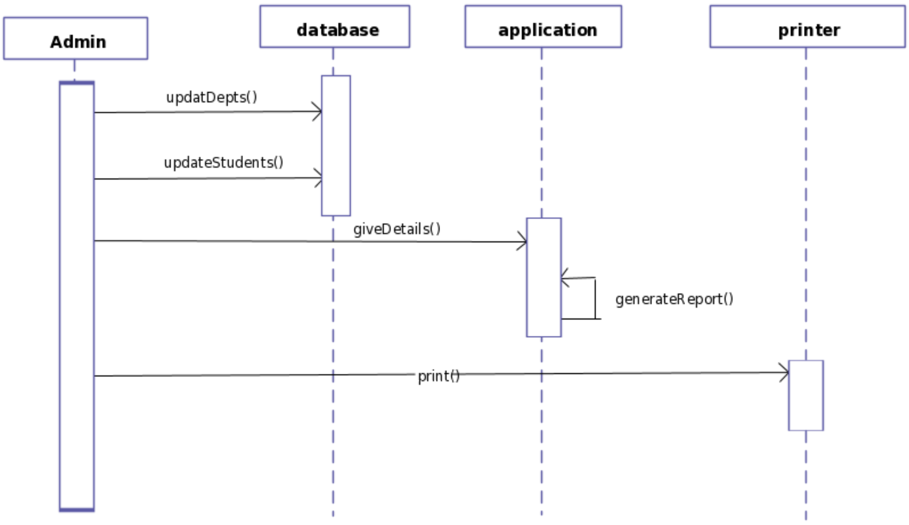


Figure 3: Sequence diagram of the proposed App

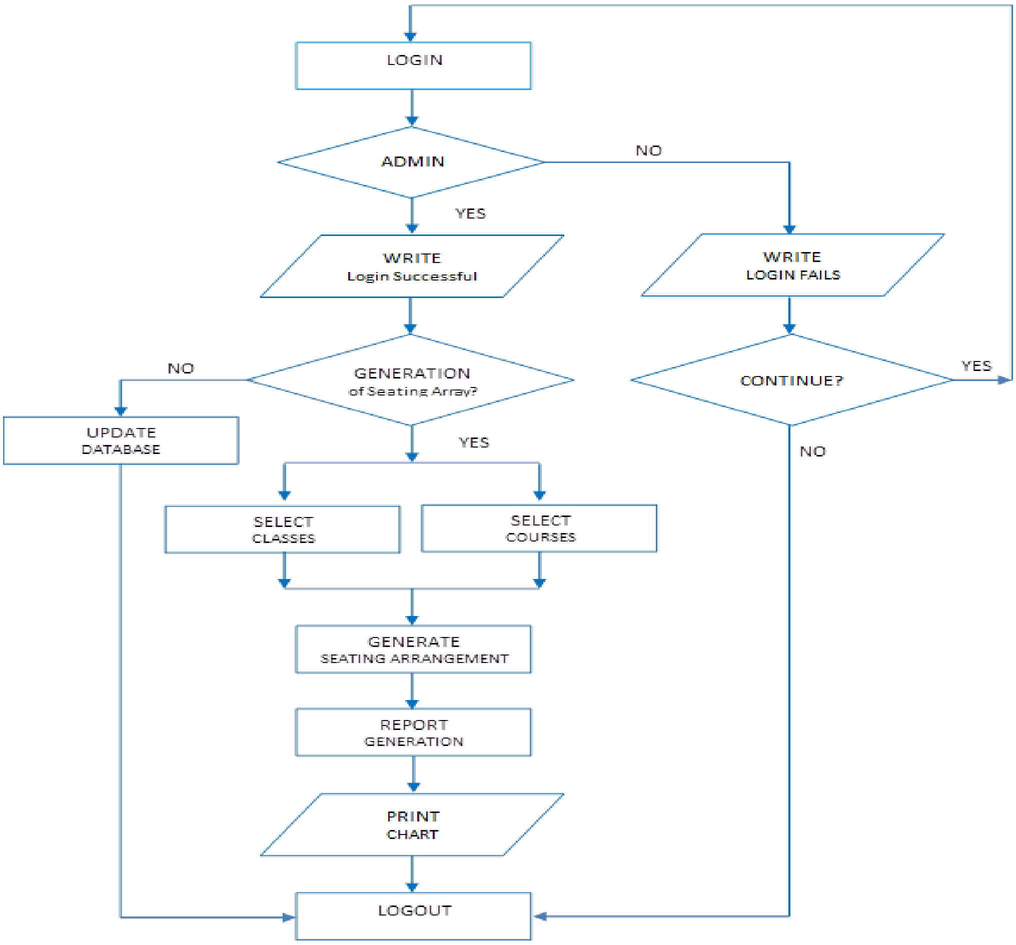


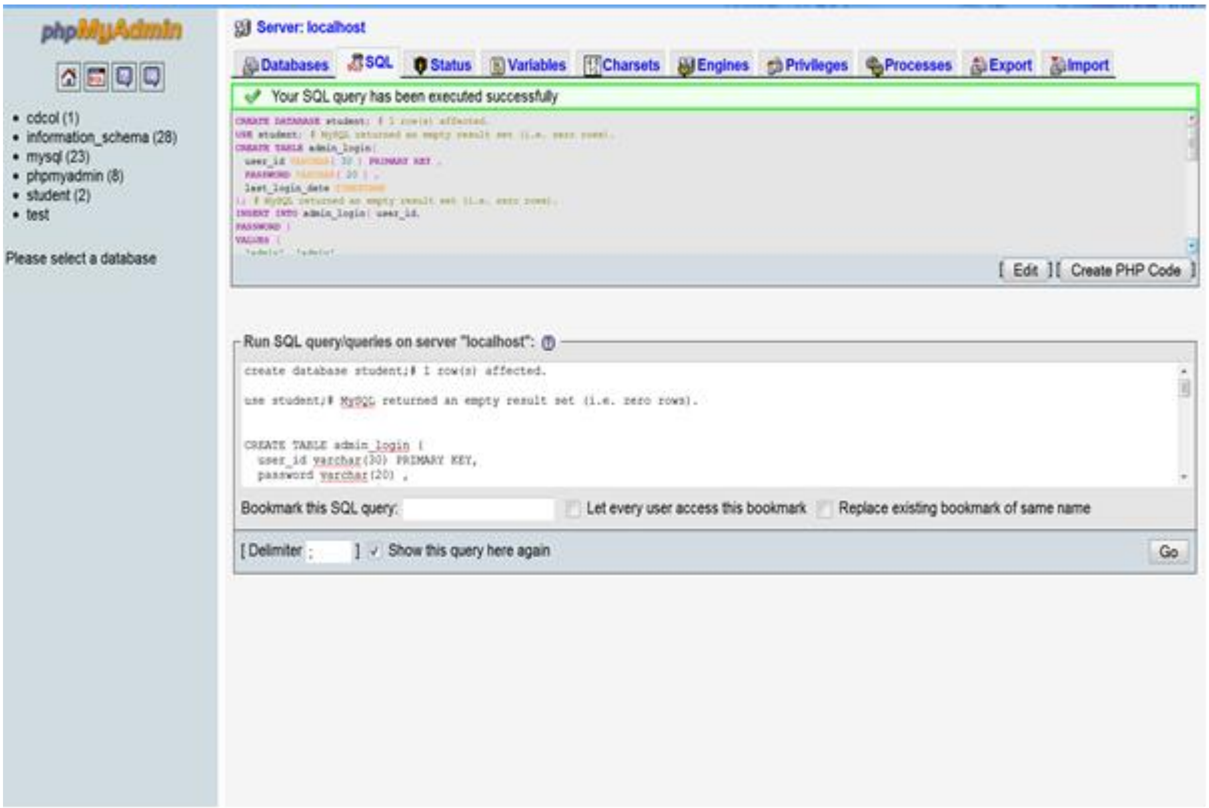
Figure 4: Data flow diagram for the seating arrangement system

Figure 4 depicts the data flow diagram that was used for the coding of the proposed App. The Admin is in charge of the proposed App. The App expects the Admin to first and foremost logs on to the system. If the login fails, the App will ask the Admin to log in again and again. In a situation where the login is successful, the App will ask the Admin whether he wants to generate examination seating arrangement or just to update the existing data on the database. If he is interested in updating the contents of the database, the proposed App is expected to let the Admin do that and thereafter logout of the App. If the Admin is interested in generating seating arrangements for a given examination at a given period of the time, the App is expected to let the Admin selects various examination courses that would be done at that period; and rooms that are available for the examination at that period. Based on this information supplied by the Admin, the App is expected to execute HAS to carry out its function. After the generation of the seating arrangement, the App is expected to present to the Admin the option of printing.

The PHP, HTML, MYSQL, and Apache [18-20] were employed to develop Server-Web, Front-End, Back-End and Web-Server softwares respectively of the App. In order to prevent redundancy and make the App more responsive, JavaScript was employed; and it is responsible for Pop-Ups that are used for data intake. In this App also, XAMPP was used for the installation and testing of PHP, MySQL, and Apache.

2.3 Creation of Database and Insertion of Data

In the App, the use and creation of a database is inevitable. A database which contains all the lists of students, courses registered by each student for a particular semester and numbers of classes as well as the capacity of each class were created. The database design of the system was created in the PhpMyAdmin module available in the XAMPP installation. A table containing administrative login information was also created in order to prevent unauthorized access to the system. A screenshot of a successful creation of the entire database needed for the web application using PhpMyAdmin is shown in Figure 5. The creation of these databases will enable the Admin to add new students, delete students and rectify any discrepancy that might exist in the system.



Your SQL query has been executed successfully

```
CREATE DATABASE student; # 1 row(s) affected.
USE student; # MySQL returned an empty result set (i.e. zero rows).
CREATE TABLE admin_login(
 user_id VARCHAR( 30 ) PRIMARY KEY ,
 PASSWORD VARCHAR( 20 ) ,
 last_login_date DATETIME
); # MySQL returned an empty result set (i.e. zero rows).
INSERT INTO admin_login( user_id,
PASSWORD )
VALUES (
 'admin', 'admin'
```

Run SQL query/queries on server "localhost":

```
create database student;# 1 row(s) affected.

use student;# MySQL returned an empty result set (i.e. zero rows).


CREATE TABLE admin_login (
  user_id varchar(30) PRIMARY KEY,
  password varchar(20) ,
```

Bookmark this SQL query: □ Let every user access this bookmark □ Replace existing bookmark of same name

[Delimiter ;] ☑ Show this query here again

Figure 5. Creation of database in MySQL using phpMyAdmin

2.4 Project Modules

To develop the proposed App, four modules were developed. They are:

Login Module: This module is used by the administrator to log on to the App; using his or her unparalleled user ID and PIN. For security purposes, it is only the administrator that is empowered to access the module.

Database Module: This module is used by the administrator to update the student, course, class and departmental information. It is imperative because inaccuracy in class capacity, departmental size and student information may lead to misallocation of students.

Allocation Module: This module is responsible for the allocation of students; based on the parameters selected by the administrator. After the allocation of students is completed, this module passes the generated chart to the display module.

Display/Report Module: This module collects data from the allocation module and generates a report of the seating arrangement while displaying the seating chart

2.5 System Requirement and Specification

The App presented in this contribution is cross platform and web based; hence the hardware requirements are very basic as any internet enabled device can be used to access the system. The following hardware specifications support the App: Operating System: Windows 8.1 Pro, Processor: Intel Core i3-4005U, and RAM: 8GB

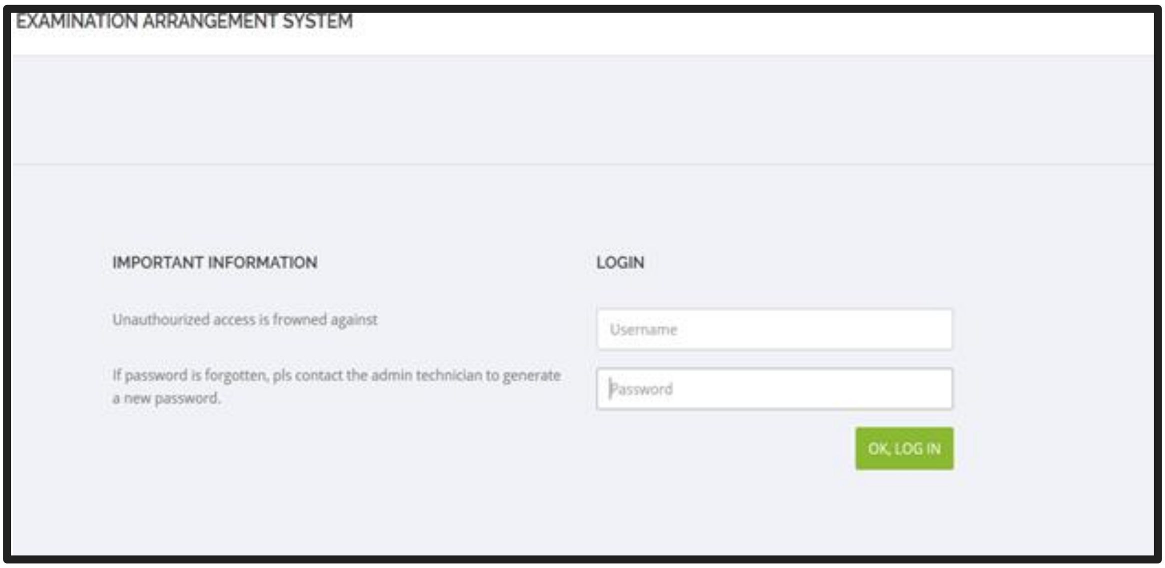
2.6 Installation and Launching of the App

- Install XAMPP or WAMP
    - Start XAMPP
    - Start Apache on XAMPP window
    - Start MySQL on XAMPP window
    - Minimize XAMPP
    - Extract the project files from Zip folder that contains the Giraffe Software App
- Move all the extracted files to the htdocs folder
    - Open your browser
    - Navigate to localhost/phpmyadmin
- Create a new database
    - Click new
    - Give the database a name e.g. exam_seating
    - Choose utf-8 unicodcci as the encoding format/collation
    - Click create
    - The database has been successfully created
- Launching of the App
    - On your browser, navigate to localhost/database name
    - Login to the App with your username and password

3. RESULTS AND DISCUSSION

3.1 Flow Analysis

In launching the proposed App, the Admin is first presented with a login platform which can only be accessed with the correct credentials. If there are numerous administrators, unique credentials are generated for each of them. The login page is shown in Figure 6.



EXAMINATION ARRANGEMENT SYSTEM

IMPORTANT INFORMATION

Unauthourized access is frowned against

If password is forgotten, pls contact the admin technician to generate a new password.

Figure 6. Administrative login page

After successful login, the administrator is presented with three data input options and the generated options. The data input options include:

Add Student Details: This option allows the system administrator to add any number of students into the database based on the courses that each student is offering. This module is shown in Figure 7.

Add Registered Classes: This option enables the administrator to add halls/rooms that have been designated for the examinations to take place. During this process, the administrator has to enter the rows and columns in a particular class and the system generates the number of seats that are available. This option is shown in Figure 8.

Add Courses: This option allows the administrator to add individual courses that examinations are to be taken into the system. This module can take as many courses as are available. The add courses module is shown in Figure 9.

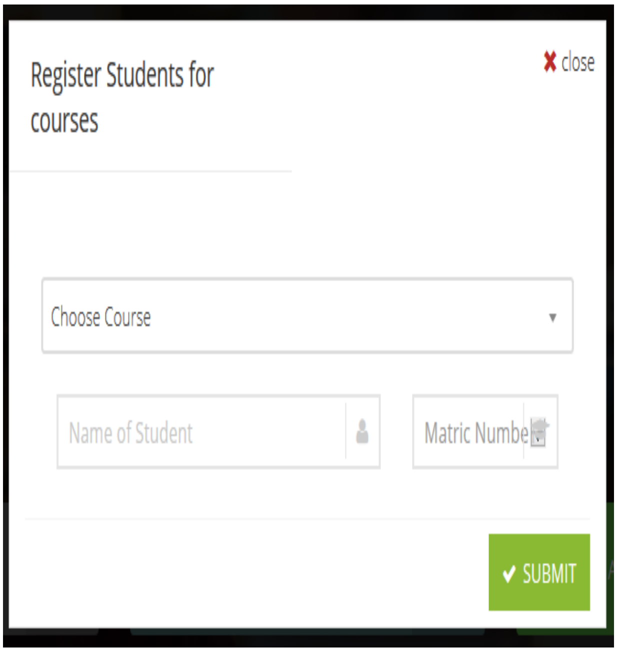


Figure 7. Add students module

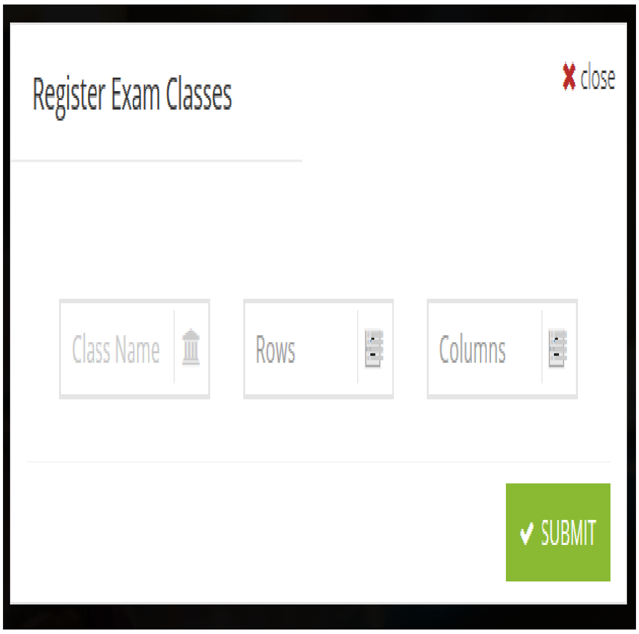


Figure 8. Add exam classes module

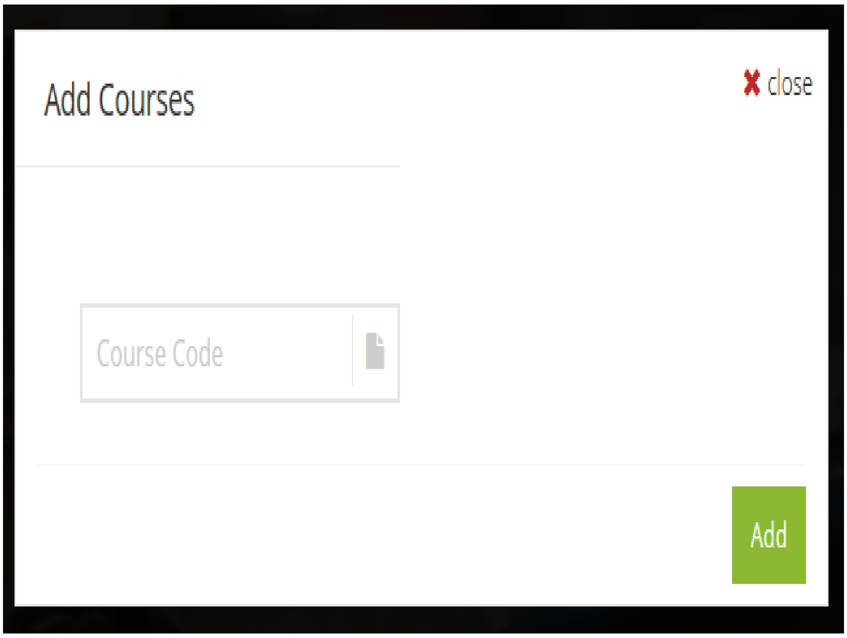
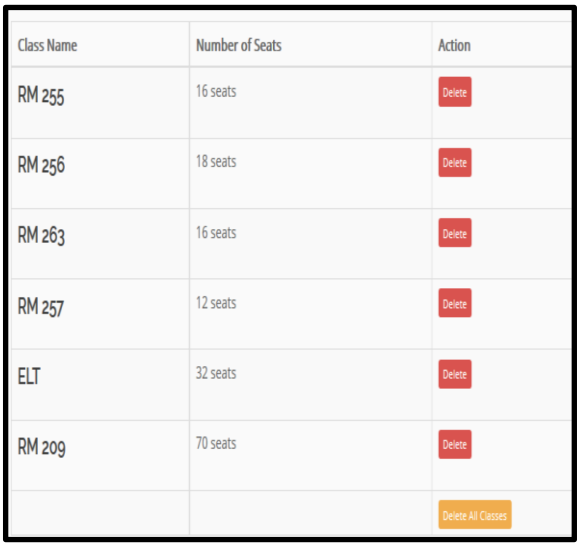


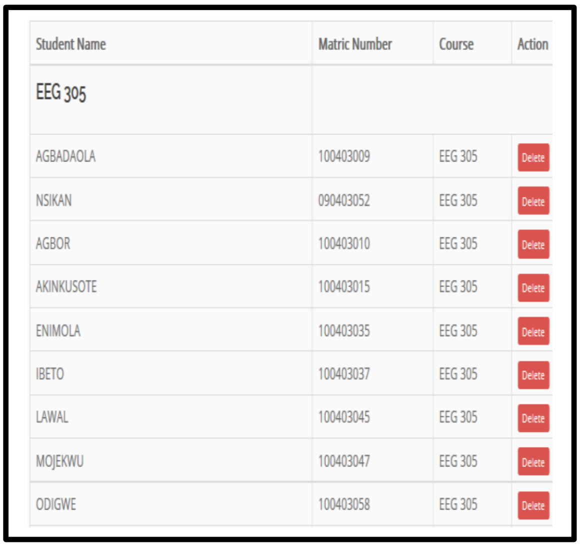
Figure 9. Add courses module

The information received from all the data input options is stored in a MySQL database. The seating arrangement application has a graphical interface that can be used to access the database. Here, student information, classes and courses can be viewed, modified or deleted when necessary. The graphical databases implemented in the application are shown in Figures 10, 11 and 12.



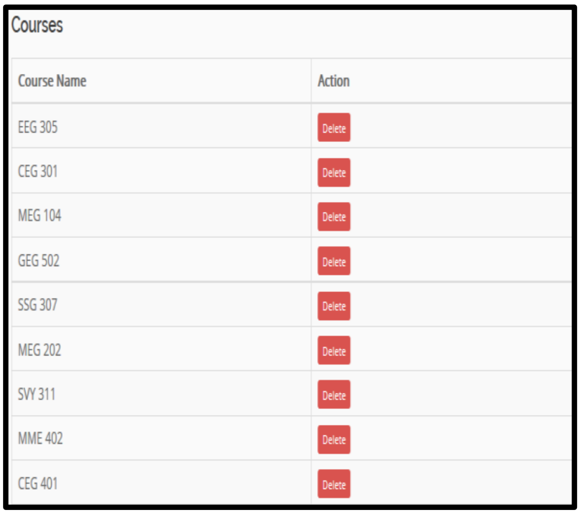
Class Name	Number of Seats	Action
RM 255	16 seats	Delete
RM 256	18 seats	Delete
RM 263	16 seats	Delete
RM 257	12 seats	Delete
ELT	32 seats	Delete
RM 209	70 seats	Delete
		Delete All Classes

Figure 10. Class database



Student Name	Matric Number	Course	Action
EEG 305			
AGBADAOLA	100403009	EEG 305	Delete
NSIKAN	090403052	EEG 305	Delete
AGBOR	100403010	EEG 305	Delete
AKINKUSOTE	100403015	EEG 305	Delete
ENIMOLA	100403035	EEG 305	Delete
IBETO	100403037	EEG 305	Delete
LAWAL	100403045	EEG 305	Delete
MOJEKWU	100403047	EEG 305	Delete
ODIGWE	100403058	EEG 305	Delete

Figure 11. Student database



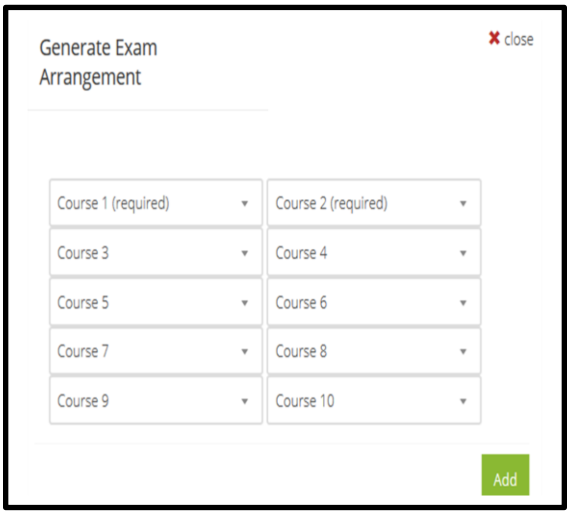
Courses

Course Name	Action
EEG 305	Delete
CEG 301	Delete
MEG 104	Delete
GEG 502	Delete
SSG 307	Delete
MEG 202	Delete
SVY 311	Delete
MME 402	Delete
CEG 401	Delete

Figure 12. Course database

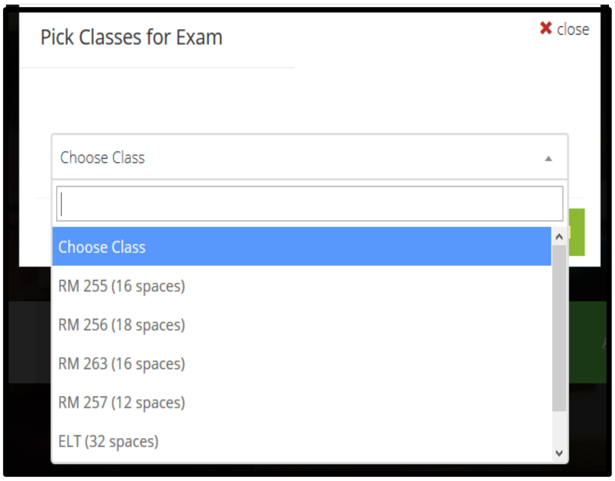
3.2 Discussion

The App presented in this contribution automates the arrangement of students in an examination so as to ensure that no two students writing the same examination are seated close to each other. The seating arrangement module gathers all the information that has been stored by the administrator in the database and uses the graph search algorithm considering the courses being offered and generates the arrangement of students into the various halls/rooms selected. Once the administrator selects the courses to be taken at a particular time and classrooms available for the examination, the system automatically generates the chart. The selection of courses and subsequent generated chart of students are shown in Figures 13, 14 and 15.



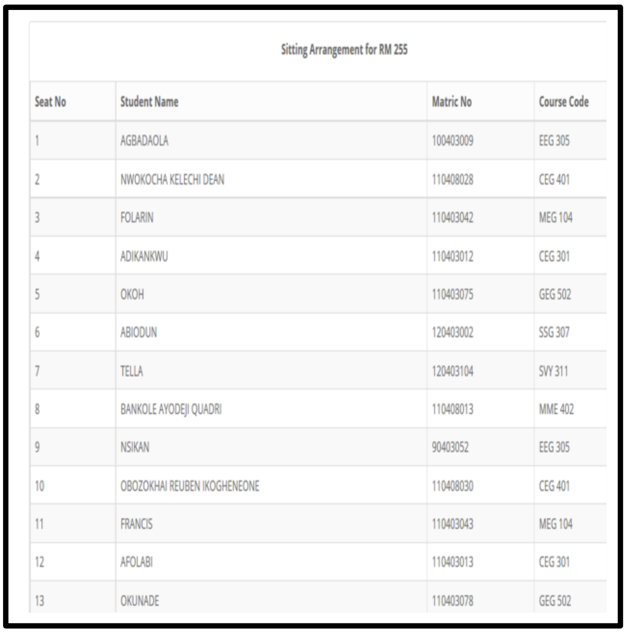
Generate Exam Arrangement

Figure 13. Selection of examination courses



Pick Classes for Exam

Figure 14. Selection of examination classes



Sitting Arrangement for RM 255

Seat No	Student Name	Matric No	Course Code
1	AGBADAOLA	100403009	EEG 305
2	NWOKOCHA KELECHI DEAN	110408028	CEG 401
3	FOLARIN	110403042	MEG 104
4	ADIKANKWU	110403012	CEG 301
5	OKOH	110403075	GEG 502
6	ABIODUN	120403002	SSG 307
7	TELLA	120403104	SVY 311
8	BANKOLE AYODEJI QUADRI	110408013	MME 402
9	NSIKAN	90403052	EEG 305
10	OBOZOKHAI REUBEN IKOGHENEONE	110408030	CEG 401
11	FRANCIS	110403043	MEG 104
12	AFOLABI	110403013	CEG 301
13	OKUNADE	110403078	GEG 502

Figure 15. Generated sitting arrangement

A careful examination of Figure 15 shows the seating chart generated. It can be seen that no two students offering the same examination are positioned next to each other. The arrangement module requires a minimum of two (2) courses and a maximum of ten (10) courses to be offered at any instance of an examination in order to ensure and improve efficiency. At any point in time, the more the number of courses being offered, the more efficient the App becomes in arranging the students; because the sample space of courses must have increased.

4. CONCLUSION

The study shows that eliminating the examination malpractice, which has to do with giraffing has received little or no attention in the literature; and that this constitutes a major problem in the conduct of examinations in universities and colleges. One of the ways of solving the problem is by making sure that no two students offering the same examination or course are positioned next to each other; and this is achieved by scattering students using seating arrangement. Presently, the seating arrangement for examinations is done manually; which is very tedious and error prone; hence the need for automated seating arrangement. In this work therefore, a web-based App that arranges students sitting for examinations in an examination hall such that no two students offering the same examination or course are in close proximity to each other is therefore developed and proposed.

After adequate testing, the proposed examination seating arrangement App achieved all its specified functions which include: Implementation of administrative login modules, addition of class and course data through a GUI into the database, addition of student data into the database, automatic arrangement of student writing examinations such that students writing the same examinations are not positioned together. After the arrangement has been generated, the administrator is presented with the option of printing which makes the system all the more convenient.

REFERENCES

[1] Anyanwu G. (2014). *Success Above Average: Real Life Lessons for Parents and their Children*, Author House, UK.

[2] Oko S.U., & Adie R.I. (2016). Examination malpractice: Causes, Effects and Possible Ways of Curbing the Menace, A study of Cross River University of Technology, *International Journal of Management Studies and Research*, 4(1): 59-65.

[3] Adetona S.O., & Akinade A. (2016). Curtailing Examination Malpractice In Citadel of Learning: Technological Options, *Security and Safety Reviews*, 1(1): 64-68.

[4] Hinman, L.M. (2000). *Approaches to Cheating and Plagiarism*, [Online]. Available: http://www.academicintegrity.org/Values.asp.

[5] Onyechere I. (2004). Cheating: A World-wide problem, In “comes at least 33 ways” retrieved from AllAfrica.com

[6] Odejobi O.A., & Clarke N.L. (2009). Implementing Biometrics to Curb Examination Malpractices in Nigeria,” Advances in Communications, *Computing, Networks and Security* 6: 115-123.

[7] Fluck A. (2010). E-Examinations Development and Acceptance, In N. Reynolds, M. Turcsanyi-Szabo (eds.): IFIP TC 3 International Conference, KCKS 2010, Held as Part of WCC 2010, Brisbane, Australia, September 20-23, Proceedings pp. 126.135, DOI 10.1007/978-3-642-15378-5_12.

[8] Fluck A., Pullen D., & Harper, C. (2009). Case study of a computer based examination system. *Australasian Journal of Educational Technology*, 25(4): 509-523.

[9] Abas O.A, Olajide S.A, & Babafemi O.S. (2017). Development of Web-Based Examination System Using Open Source Programming Model, *Turkish Online Journal of Distance Education*, 18(2): 30-40.

[10] Underwood J. & Sarbo, A. (2004). Academic Offences and E-learning: Individual Propensities in Cheating, British *Journal of Educational Technology*, 34: 467 – 478.

[11] Ajasa A.A., Shoewu O., & Nwamina P.O. (2014). Design and Development of a Mobile Phone Signal Detector, *Pacific Journal of Science and Technology*, 15(2): 167-172.

[12] Gokila R., & Dass A.R (2018). Examination Hall and Seating Arrangement Application using PHP, *International Journal of Engineering Science and Computing*, 8(2): 16059-16065.

[13] Lewis R.M.R (2016), *A guide to graph Colouring: Algorithms and Applications*, Springer International Publishing, Switzerland.

[14] Lewis R., Thompson J., Mumford C., & Gillard J. (2012). A wide-ranging computational comparison of high performance graph colouring algorithms. *Computers and Operations Research*, 39(9): 1933–1950.

[15] Gao, X.Z, Govindasamy V., Xu H., Wang X., & Zenger K., (2015). Harmony Search Method: Theory and Application, Computational Intelligence and Neuroscience, Vol. 2015, Article ID 25849.

[16] Geem Z.W., Kim J.H., & Loganathan G.V. (2001). A New Heuristic Optimization Algorithm: Harmony Search, *Simulation*, 76(2): 60-68.

[17] Omar M., Ebrahim M.A., Abdel-Ghany A.M., & Bendary F., (2016). Reduced Size Harmony Search Algorithm for Optimization, *Journal of Electrical Engineering*, 16(1): 395-402.

[18] MySQL API Programming, [Online]. Available: http://zetcode.com/db/mysqlc.

[19] Carroll F. & Lewis R. (2013). Important considerations for the design and development of web applications for solving a complex combinatorial optimization problems, *World Journal of Computer Application and Technology*, 1(3): 75–82.

[20] Lee W. (2009), *Wireless and Cellular Telecommunications*: 3rd Edition, McGraw- Hill Engineering.