Review of Computer Engineering Research
2019 Vol.6, No.1, pp.24-34
ISSN(e): 2410-9142
ISSN(p): 2412-4281
DOI: 10.18488/journal.76.2019.61.24.34
© 2019 Conscientia Beam. All Rights Reserved.

DESIGN OF AUTOMATED DEPARTMENTAL LECTURE TIMETABLE SYSTEM

Etuh, Emmanuel[1+]
Ajodoh, Onoja Friday[2]
Yusuf, Serah Omolola[3]

[1]*Department of Mathematics and Computer Science, Arthur Jarvis University, Cross River State, Nigeria*
Email: emma.etuh@gmail.com
[2]*St. Thomas Aquinas School, Lokoja, Kogi State, Nigeria.*
Email: ajodohonoja@gmail.com
[3]*Department of Physics, Arthur Jarvis University, Cross River State, Nigeria*
Email: serahomolola@yahoo.com

(+ Corresponding author)

Article History
Received: 12 December 2018
Revised: 18 January 2019
Accepted: 27 February 2019
Published: 8 May 2019

Keywords
Automated
Courses
Credit Unit
Lecture timetable
Room capacity
Schedule
Timeslot
Venues.

ABSTRACT

A lecture timetable is a tabular list showing the times which a particular lecture is scheduled to hold and the venue in each day of the week. Its content includes the course code for each course, the coded lecture venue, and the time for each lecture. A lecture timetable is designed every semester. In this era of technological advancement, virtually every aspect of human enterprise has been automated (the use of machines especially computers instead of human to accomplish a task). Now students can pay their school fee, register their courses and even check their results on the internet. This is the drive that gave birth to this research work: the design of an automated lecture timetable for the department of mathematical sciences, Kogi State University Anyigba. The project presents the automated design of lecture timetable. The principle of operation is simple and employs the use of computer technology. A program is written in Visual Basic which produces the output in an error free tabular form displaying the courses/venues for each particular lecture schedule.

Contribution/Originality: The paper's primary contribution is the eradication of double scheduling, time clashing, and poor resources' utilization in timetabling. The implementation employs the use of ICT to reduce the cost and time taken to design a timetable, eliminates the drawback of human error. Every semester, the application only needs an update of courses, and venues.

1. INTRODUCTION

A timetable is a table showing a set of placement of a set of meetings at a particular time and place while a meeting is a combination of resources like rooms, people and items of equipment, some of which may be specified by a problem or allocated as part of the solution. Timetables shows when particular events are to take place but do not necessarily imply an allocation of resources [1].

There are several fundamental differences on the timetabling problems depending on the subject and area of discus. For instance, university timetables are usually divided into two main categories; they are: 1) Lecture timetable and 2) Examination timetable [2]. An examination timetable is a set of meetings at a particular time and place in which resources are fixed for a particular period for the purpose of conducting an examination which is a standard test of progress set up for students [3]. On the other hand, a lecture timetable has to do with a set of meetings at a particular time and place in which resources are fixed together for a particular period for the sole purpose of delivering lecture to students. The major differences between the two are:

1) Exams must be scheduled usually at the end of a semester, but lectures must be scheduled before student

enrolments are known.

2) As space is often a constraint, exams often share rooms or are split across rooms, but only one lecture may be held in a room at a time.
3) Examination involves students and their examiner who might not necessarily be their lecturer, but lectures are primarily delivered by the lecturer.
4) The focus and major consideration for lecture timetable is the lecturer while the major consideration for examination timetable is the students.

The overall goal of lecture timetabling problem and the basic challenge is to design a schedule of lectures over a limited period so as to avoid conflicts and to satisfy a number of constraints. An error or conflict occurs whenever a timetable requires any resource to be in two places at the same time.

The main intention is to automate the timetabling process using computing resources, by creating a database of courses and combining it with a timetable generating algorithm to phase out conflicts. Depending on the size of a department and diversity of the courses offered, the time required to produce a timetable schedule can range from a day's work to an intensive month ordeal when using manual processes.

People affected by the timetabling processes are:

1) The administration, which sets the minimum standards that the timetable must conform to.
2) The department, which may request that courses anticipating large enrolment be placed early enough and in large classes.
3) Students who plays a large role into having breaks between courses and prefer well-spaced out courses.

1.1. Problem Definition

Timetable scheduling has a lot of constraint satisfiablility that are very hard to meet [4]. To develop a timetable that will be free of course, venue, and lecturer schedule clashes is extremely hard to achieve [5, 6]. The existing system has main problem, which is the clashing of courses, venues, and a lecturer may be assigned to two classes at the same time. This creates a big problem to the institution because it affects both the lecturers and students that unluckily have more than one course at the same time. The information is manually collected from departments and is updated manually per semester in case new courses are introduced or dropped which increases or reduces the courses offered. The scheduling of lecture timetable for a university is a complicated and often expensive affair [6]. The timetable problem is essentially concerned with scheduling a number of courses in limited number of timeslots and available venues in order to satisfy a set of specified constraints. Most universities still schedule their timetable manually and this takes longer process and time, error prone, and unable to cater for inevitable last minute changes that may arise.

Manual data processing is done using physical human effort to process the data that yield the information. The raw data includes prospective courses, credit units, available venues and courses. Problems got from manual operation and procedures include co-ordination, collection and computation of data supplied by the lecturers and various departments. Flaws are generated from co-ordination, placing of courses into rows and columns in a table and sometimes more than one course is likely to be scheduled at the same time, and may be fixed in different venues for students who obviously cannot be in more than one place at a time. Fatigue and lack of concentration causes this from the handler which makes the manual method inefficient. It can be therefore classified into two constraints that vary between institutions.

2. HARD CONSTRAINTS

A timetable which breaks a hard constraint is not a feasible solution and must be repaired or rejected by the timetable algorithm [4]. That is, no person can be permitted to attend more than one lecture at a time; no lecturer must be assigned two courses at the same time.

2.1. Soft Constraints

They are usually impossible to avoid breaking most times whichever method is applied, timetables are usually rated by a penalty function which calculates the extent to which a timetable has violated its soft constraints [1, 4] like when there are no available venues.

2.2. Procedure for Solving Timetable Problems

The procedures are:

1) General courses should be scheduled first before other courses.
2) Faculty courses are scheduled next where all the interrelated courses for all the departments at faculty level are scheduled.
3) Departmental requirements are analyzed, synthesized and synergized before courses are allocated for each department or group of departments that share the same course.
4) Courses that are a level below or above the other in the same department should not be scheduled at the same time or day to give room to students that may have failed a course and want to redo the course.

For one to know that an optimal solution is feasible in performing an automated timetable, these questions must be answered:

1) Are the lecturers and students satisfied?
2) Are there clashing of courses?
3) Are the courses properly spaced out?
4) Are the venues for general courses enough?

2.3. Computer and Areas of Application

Computer is an electronic device that accepts data, processes it and produces a reliable output at a very fast speed using suitable program [7]. Computers are everywhere and are applied to virtually every area of human activities, they are used in government, businesses and industries, academic field, laboratory work, research and other professions finds it important and useful. In an academic environment, computers help lecturers and students to perform better in various activities ranging from sourcing for research materials to publishing. Computer aided library helps both lecturers and students save time and assists in efficient use of the library. The idea of applying computers to difficult and time consuming problems including school timetable systems is a big advantage in academic environment.

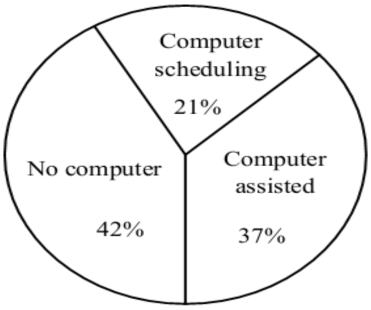


Figure-1. Computer usage in British university examination timetabling 1995 [4].

The research has revealed that of all the universities that use some sort of computer aid, about 21% of British universities have a facility to perform automated timetable scheduling, although these may still require a certain amount of manual input and previous knowledge of the particular timetable problem. 37% uses the computer to some extend while about 42% are still using manual operation to carry out their activities [4]. Each department is to produce their own draft timetable and use a central administration facility to merge them all into a master timetable.

2.4. System Analysis

System analysis is the process that involves careful examination of an already existing system either manual or automated for the evolution of a new system which will be more efficient and reliable [8]. In the design of a lecture timetable, analysis of the existing system is very important as it helps the analyst to:

1) Know the various procedures and steps followed in order to achieve the set goal o obtain the desired result (in this case a lecture timetable).
2) Know the problem with the existing system; turning it into constraints which are used in the design the new system.
3) Discover the area that needed improvement and looking for ways to improving upon it in order to increase efficiency in its performance.
4) Design the new system taking the constraints analyzed into consideration to meet the requirement.

From the analysis, input specifications which include the data that are collected and used to build the timetable system are obtained from the mathematical science department. These input specifications are the available venues, courses, and time slot for each schedule.

2.4.1. Input Specification

Data entry is through the input device of a computer system (keyboard and mouse). Each field has a specified width and type definition that help to check against entry of wrong data. The character for each field is defined; hence there is a check on the data that goes into the input file, which is the storage facility of the inputs. This style makes provision for only the specified inputs important for developing the desired timetable. The specifications are:

1) Alphanumeric course code
2) Numeric course unit (course strength/work load)
3) Course title
4) Number of students taking the courses
5) Number of courses in each option
6) Alphanumeric available venues
7) Sizes (capacity) of the lecture halls/class rooms

2.4.2. Available Venues

This input deals with the list of all usable and conducive venues having the right lecture facilities like seats and tables, also venues that have adequate capacity for the lecture. For the purpose of this research simulation, our venues shall be limited to the twin lecture theatre and the two blocks in the natural sciences faculty. The twin lecture theatre rooms are labeled as LT1 and LT2 for both the right and left wings respectively. Also the block class room close to the computer laboratory shall be labeled as block A with each of the class room as A1, A2, and A3 respectively. Similarly, the block class room close to the administrative block of the natural science faculty shall be labeled block B with each of the class room as B1, B2, and B3 respectively. Table 1 shows the venue specification and the room capacity for each lecture room.

Table-1. Available Venues and their Capacity.

Venue	Room Capacity
LT1	473
LT2	490
A1	40
A2	31
A3	72
B1	60
B2	35
B3	90

Source: Faculty of Natural Science, Kogi State University.

2.4.3. Courses

This deals with all the courses available to all students either core or elective, which are listed with the course unit and the total number of students in each level taking the courses which determines the number of students to be arranged in each venue. Due to the complexity of the university course timetabling problem, the complete university timetabling problem is decomposed into a series of sub problems to be solved at the academic department level, where the resources required to provide instructions are controlled. This research work is limited to mathematical science department courses. Table 2 shows the simulated courses used for timetabling. Each record represents the course code, the credit unit that determines the allotted hour per lecture and the total number of students that registered for the course

Table-2. Simulated courses, credit units and number of registered students.

Courses	Credit Unit	Number of Students
CSC104	2	450
MAT102	3	120
MAT304	3	88
GST102	2	480
GST106	2	450
STA11	2	100
CSC304	2	80
MAT204	2	120

Source: Faculty of Natural Science, Kogi State University.

2.4.4. Output Specification

The output of the program is a finished report for processed and organized information. That is, a timetable generated in a tabular form. This is displayed on the monitor (VDU) or sent to the printer if hardcopy is needed for public view/use. The output gives adequate information on the following

1) Various courses spread out on different days.
2) The days of the week for lectures
3) The time/session allocated for each course
4) The venues affixed to different courses/time sessions.

2.5. The Existing System in K.S.U Anyigba

The existing mode of operation at the university is manual form of generating timetables. The staff presently in charge of lecture timetable design in natural science faculty is Mr. Salehdeen M.U of chemistry department. The individual departments in the natural science faculty sends all necessary information either in smaller departmental timetables or bare (raw data) like the total number of students offering the course, total number of students in each level/department, then each departments own timetable are analyzed and integrated into one central timetable after error checks which is time consuming is carried out. Though the staff in charge of timetabling has some stable information like venues available, and number of sitting capacity per venue from which the timetable is generated; but this is tedious, long process, time-consuming and inefficient in most cases due to human factor.

2.6. Problems Associated with the Existing System

In an attempt to analyze the existing system of operation in KSU, interviews and observations were used as fact-finding techniques on timetable production. The problems that the existing system has are:

1) Inefficiency on the part of the generated timetable like course clashing.
2) Vulnerability to error due to human factors: stress, fatigue.
3) Unavoidable data omission; due to collection of too many data.
4) Data redundancy.

2.7. Solution to the Problems Mentioned Above

In accordance with the problems mentioned above, it has become imperative to introduce an automated means of timetable scheduling to meet the requirements needed to solve various problems of course timetabling.

The issue of course clashing will be eliminated when automated system is employed in the design of course timetable. The algorithm processes iteratively taking all the constraints of the courses into consideration. During each step, an unassigned variable (course) is selected and assigned a timeslot and once a slot is assigned to a course at a particular time, no other course is expected to assume that same time slot during the iteration.

The software will help in checking errors and removing the problem of data redundancy, course clashing and any other shortcoming that may arise when manual system is used.

2.8. Method of Investigation

This has to do with the fact-finding techniques employed in this research work in order to gather the needed information on the existing system that leads to the evolution of the new system. They include observation and interview.

2.8.1. Interview

One of the best ways to obtain critical study facts is to conduct interview. It is defined as an exchange of information between individuals concerning what is required and how these requirements are met. In essence, gathering information from the staff in charge of lecture scheduling Mr. Salehdeen M.U who gave the guidelines for preparing timetable and the consideration for fixing a lecture. Although timetable is prepared at faculty level and information are gathered from every department in the faculty. Due to time and the complexity of timetabling, the project is limited to one department. Information was also gathered from mathematical science department. The interviews generated all the needed information used in designing the new system.

2.8.2. Observation

Observation was used to verify the information acquired in the course of the interview. This method was most useful because it gives a report determining what is being done, how it's being done and the time it takes to execute the job.

2.9. Need for Automated System

Due to advancement in software engineering, computer application have proliferated every sphere of human activities in recent times. Feasibility reports show that the design of an automated system will favour the university and when designed, it will aid in eliminating most problems of inefficiency, unavoidable data omission, data redundancy and course clashes due to human factor. The system will have the following features

1) User-friendly interface.
2) Easy to generate timetable.
3) Better scheduled timetable.
4) Error/clash free timetable.

3. SYSTEM DESIGN

This section is going to analyze the design and implementation of the automated system for lecture timetable, the sets of requirements that are used to build the system and the choice of the programming language. System design is the activity of proceeding from an identified set of requirements for a system to a design that meets those requirements. The system should have an interface where courses, venues, and time slots are entered into the database as input requirements. A programming language that implements the algorithm which is Visual Basic generates an error-free timetable if all the requirements are satisfied.

3.1. Database

Every system that responds intelligently is driven by a will structured data that interact with the user using a program [9]. A database is responsible for the display of intelligence by a computer system. Microsoft access was used to design the database used to create the timetable. This database contains two tables: one for the courses and the other for venues that are loaded whenever load button is clicked during the runtime of the application. The timetable will be created and displayed on a notepad when build menu is clicked.

3.2. Procedure Chart

The design of any system requires a layout or schematic diagram of the system because it shows the main components of the system, their roles and inter-relationships.

During the run of the program, the timetable system interface is displayed showing the File, Edit, Build, and windows menus. Each of their tasks in the timetable generation is highlighted below.

1) **File Menu:** This menu helps the user to exit the timetable system window that is when the timetable has been built, click on file menu and select exit to automatically terminate the program.
2) **Edit Menu:** This menu enable the user to effect changes to either the courses or the venues and at the same time load the venues and courses respectively from the database. Click on edit menu and select load venues from the buttons at the bottom of the venues window, equally select courses from edit menu and click on load courses, this exercise load both the venues and courses in the database into the system and the system is ready to build a timetable with the loaded information.
3) **Build Menu:** This command menu automatically builds the timetable based on the loaded courses and venues. The lecture timetable is built and can be retrieved from the application folder.
4) **Window Menu:** As the name implies is the menu that holds the windows being opened during the run of the application. It could be the course window or the venue window.

4. PROGRAM FLOW CHART

A low-level graphical representation of the structure of the program with emphasis on the control flow and primitive actions performed by the program is shown Figure 2.

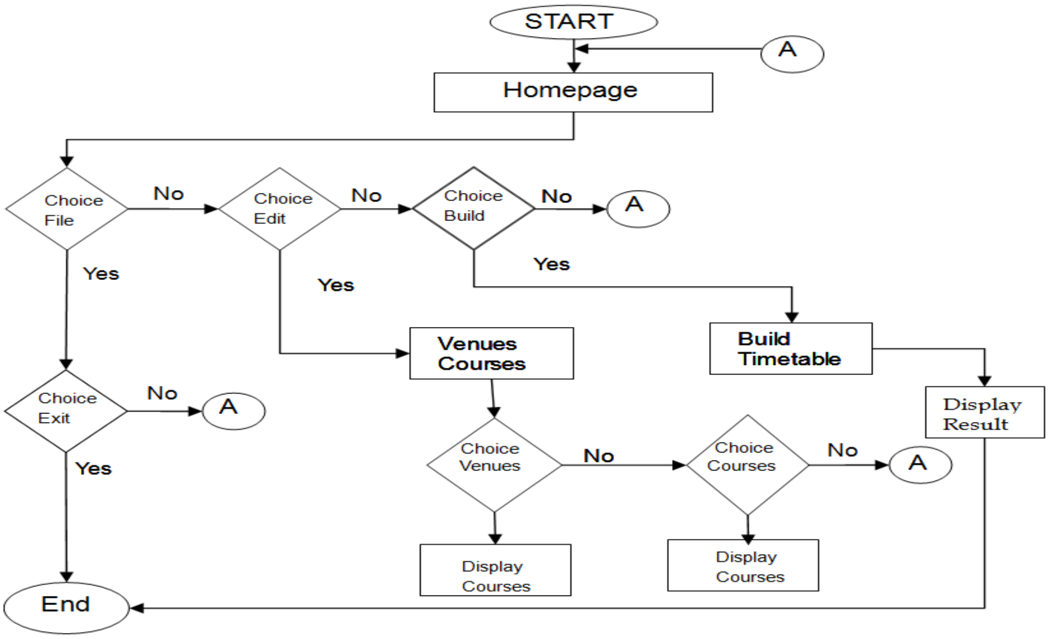


Figure-2. Lecture Timetable Algorithm Flow Chart.

4.1. Choice of Programming Language

In implementation, Visual Basic programming language was used in writing the source code required for the desired system in this research work. The language was chosen based on several reasons some of which are highlighted below:

1) Visual Basic enforces object oriented programming and as a result provides great flexibility, modularity and reusability.
2) It is simple in that it replaces the multiple inheritances in language like C++ with a simple language construct called an interface and eliminates pointers, it also has a clear syntax which makes visual basic programming easy to write and read.
3) It is used in designing a graphical user interface (GUI). In a GUI programming, many tasks goes on concurrently, this is achieved in Visual Basic because it has the capacity to perform several task simultaneously.

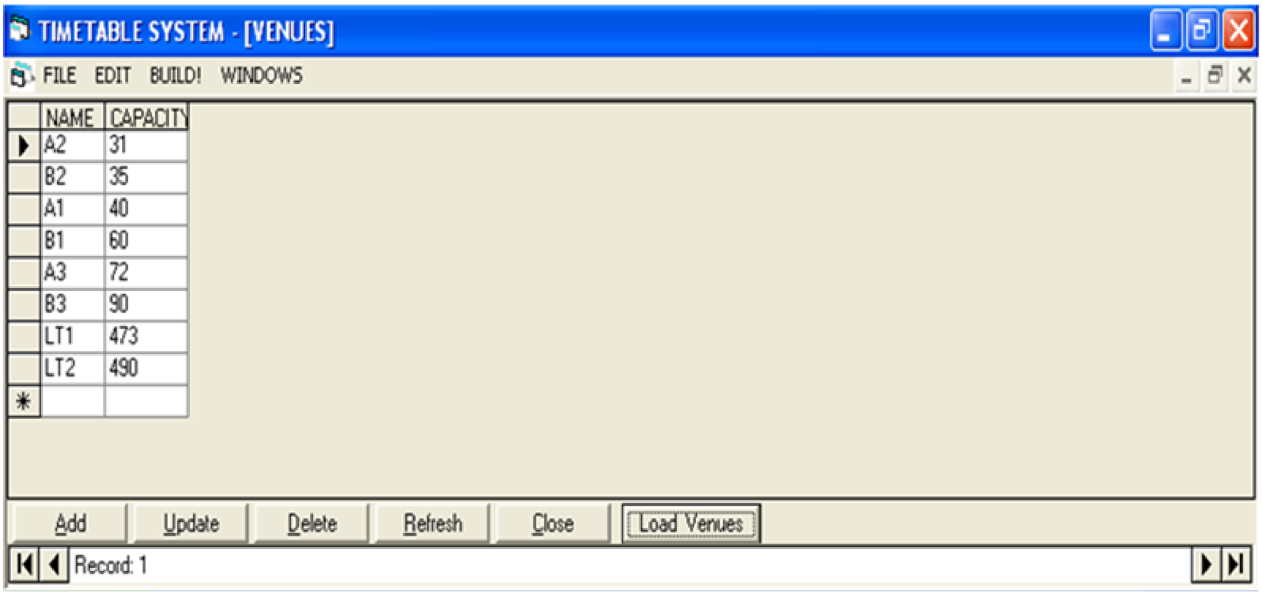
4.2. Hardware Requirements

To build course timetable system, a personal computer with the following basic configuration and components are desired: (1) 32MB of RAM, 250 HD, 14 inch monitor, CD-ROM, U.P.S, Printer, A4 paper.

5. IMPLEMENTATION

5.1. Venues Preview

This preview displays the database of the various venues for lecture. During the run of the program, the preview of the venues can be achieved by clicking on the edit menu and selecting venues, this option displays the names of the venues and their corresponding capacities in Venue Window. The user can add more venues by clicking on the 'Add' button, delete a venue by selecting a particular venue record and clicking the 'Delete' button. After each editing, click on 'Update' to save the last changes made to the venues database. 'Load Venues' is a command used to load the venues used in building the lecture timetable. Close button is used to close the venues window after it has been loaded into the system. Figure 3 is the screen preview of the venue interface where the timetable developer enters all the venues that are ready to be used for lecture under the semester of design.



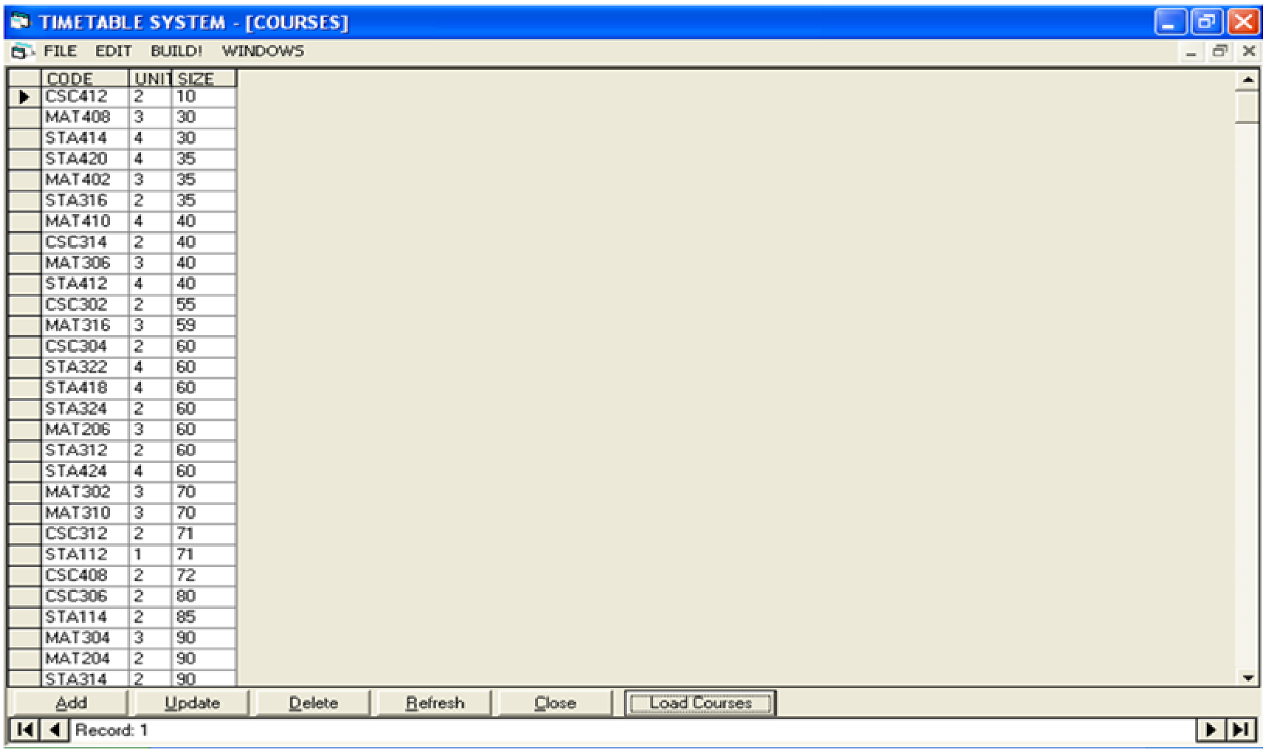
NAME	CAPACITY
A2	31
B2	35
A1	40
B1	60
A3	72
B3	90
LT1	473
LT2	490

Figure-3. Screen - Venue Window.

Source: Implementation Screenshot.

5.2. Course Preview

The course preview interface displays the database of the various courses that can be offered by the students in the department with each record showing the course code in alphanumeric character length of six, the credit unit, and the total number of students offering the course, these are the criteria that determine the placement priority in the generated timetable. Editing the courses takes the same procedure as in the venues described. Figure 4 is the screen preview.



CODE	UNIT	SIZE
CSC412	2	10
MAT408	3	30
STA414	4	30
STA420	4	35
MAT402	3	35
STA316	2	35
MAT410	4	40
CSC314	2	40
MAT306	3	40
STA412	4	40
CSC302	2	55
MAT316	3	59
CSC304	2	60
STA322	4	60
STA418	4	60
STA324	2	60
MAT206	3	60
STA312	2	60
STA424	4	60
MAT302	3	70
MAT310	3	70
CSC312	2	71
STA112	1	71
CSC408	2	72
CSC306	2	80
STA114	2	85
MAT304	3	90
MAT204	2	90
STA314	2	90

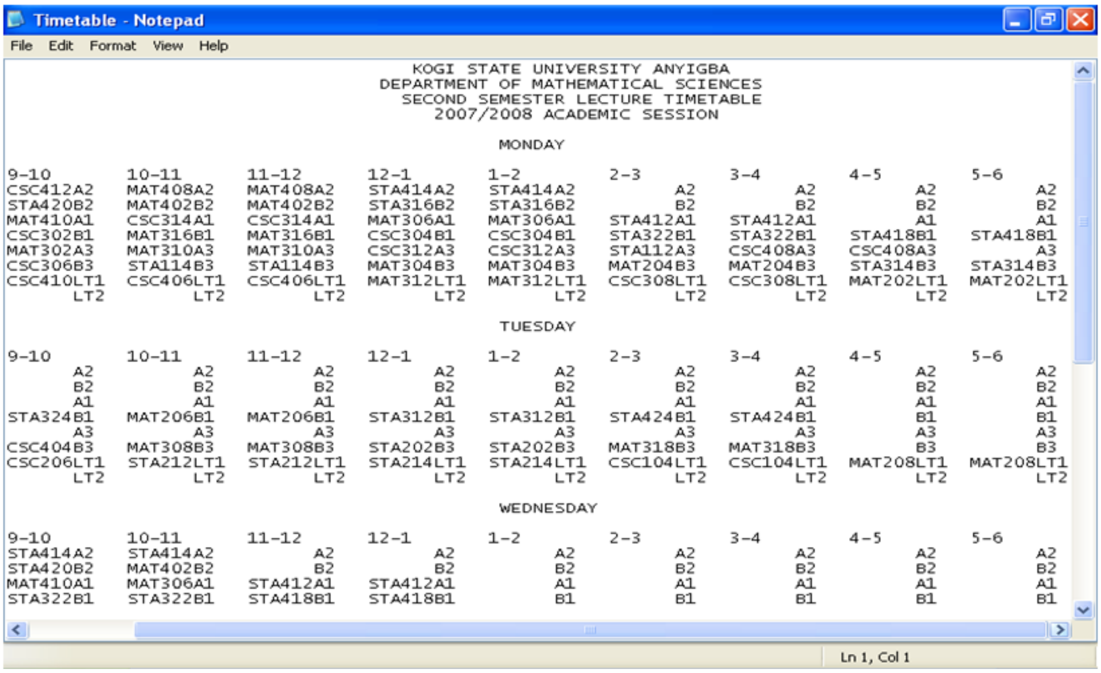
Figure-4. Screen - Courses Window.

Source: Implementation Screenshot.

5.3. Output Preview

The output preview in Figure 5 shows the desired output, the program during runtime generates the timetable automatically using the venues and courses loaded into the timetable system when build button command on the menu bar is clicked. The program is intelligent in the sense that it allocates a venue to a particular course having a class size that best fits a particular venue capacity in the timeslots and days of the week appropriate for lectures in a tabular form. The time allocated to each course is dependent on the unit of the course.

The display of the courses and venues codes is an alphanumeric code with a maximum size of nine widths. The first six represents the course code while the last two or three represents the venue code depending on the code for the courses and venues. The program has been tested and implemented for the second semester program of the Mathematical Science Department, Faculty of Natural Sciences Kogi State University Anyigba.



```
                                           KOGI  STATE  UNIVERSITY  ANYIGBA
                                       DEPARTMENT  OF  MATHEMATICAL  SCIENCES
                                         SECOND  SEMESTER  LECTURE  TIMETABLE
                                            2007/2008  ACADEMIC  SESSION

                                                    MONDAY

9-10        10-11       11-12       12-1        1-2         2-3         3-4         4-5         5-6
CSC412A2    MAT408A2    MAT408A2    STA414A2    STA414A2          A2          A2          A2          A2
STA420B2    MAT402B2    MAT402B2    STA316B2    STA316B2          B2          B2          B2          B2
MAT410A1    CSC314A1    CSC314A1    MAT306A1    MAT306A1    STA412A1    STA412A1          A1          A1
CSC302B1    MAT316B1    MAT316B1    CSC304B1    CSC304B1    STA322B1    STA322B1    STA418B1    STA418B1
MAT302A3    MAT310A3    MAT310A3    CSC312A3    CSC312A3    STA112A3    CSC408A3    CSC408A3          A3
CSC306B3    STA114B3    STA114B3    MAT304B3    MAT304B3    MAT204B3    MAT204B3    STA314B3    STA314B3
CSC410LT1   CSC406LT1   CSC406LT1   MAT312LT1   MAT312LT1   CSC308LT1   CSC308LT1   MAT202LT1   MAT202LT1
      LT2         LT2         LT2         LT2         LT2         LT2         LT2         LT2         LT2

                                                    TUESDAY

9-10        10-11       11-12       12-1        1-2         2-3         3-4         4-5         5-6
      A2          A2          A2          A2          A2          A2          A2          A2          A2
      B2          B2          B2          B2          B2          B2          B2          B2          B2
      A1          A1          A1          A1          A1          A1          A1          A1          A1
STA324B1    MAT206B1    MAT206B1    STA312B1    STA312B1    STA424B1    STA424B1          B1          B1
      A3          A3          A3          A3          A3          A3          A3          A3          A3
CSC404B3    MAT308B3    MAT308B3    STA202B3    STA202B3    MAT318B3    MAT318B3          B3          B3
CSC206LT1   STA212LT1   STA212LT1   STA214LT1   STA214LT1   CSC104LT1   CSC104LT1   MAT208LT1   MAT208LT1
      LT2         LT2         LT2         LT2         LT2         LT2         LT2         LT2         LT2

                                                    WEDNESDAY

9-10        10-11       11-12       12-1        1-2         2-3         3-4         4-5         5-6
STA414A2    STA414A2          A2          A2          A2          A2          A2          A2          A2
STA420B2    MAT402B2          B2          B2          B2          B2          B2          B2          B2
MAT410A1    MAT306A1    STA412A1    STA412A1          A1          A1          A1          A1          A1
STA322B1    STA322B1    STA418B1    STA418B1          B1          B1          B1          B1          B1
```

Figure-5. Screen - Timetable Output.

Source: Implementation Screenshot.

6. CONCLUSION

A simple way of using an automated system to generate a lecture timetable has been designed, implanted, and tested. It is suitable for any institution that uses timetable to map out their academic lecture timetable. The research project has been undertaken to make life easy, averting inputting the same courses or getting worried over course clashes and data omission, inputs are stored in the database without duplication and are used to generate the timetable in a manner preferred by the user with the help of the program.

The program, automated timetable involves providing a simple and cost effective means of generating a timetable that is easy to maintain, update in the sense of adding or deleting courses when the need arises in an institution.

Funding: This study received no specific financial support.
Competing Interests: The authors declare that they have no competing interests.
Contributors/Acknowledgement: All authors contributed equally to the conception and design of the study.

REFERENCES

[1] V. A. Bardadym, *Computer-aided school and university Timetabling*. Ukraine: Springer-Verlag, 1996.

[2] S. D. Bloomfield and M. M. Mcsharry, "Preferential course scheduling," *Interfaces*, vol. 9, pp. 24-31, 1979.Available at: https://doi.org/10.1287/inte.9.4.24.

[3] M. W. Carter, *A general examination scheduling system*. Ontario, Canada: Canadian Management Institute Press, 1994.

[4] E. Burke, K. Jackson, J. H. Kingston, and R. Weare, "Automated university timetabling: The state of the art," *The Computer Journal*, vol. 40, pp. 565-571, 1997.Available at: https://doi.org/10.1093/comjnl/40.9.565.

[5] P. Nortons, *Introduction to computers*, 6th ed. New Delhi: McGraw Hill Companies Inc, 2006.

[6] B. P. Romero, *Lecture scheduling in large engineering school* vol. 12. Spain: Spanish Institute of Management Press, 1982.

[7] M. Tomas, *Constraint-based timetabling Charles University in Prague, mathematics and physics*. Prague: Tomas Muller, 2005.

[8] J. H. Winston, "Bibliography on the practice and theory of automated timetabling, Sydney, Australia." Available: hpp://ftp.esusvd.edu.au//jeff/-timetabling/timetabling.bib.gz/, 1997.

[9] O. C. Akinyokun, *Data structure and algorithm*. Anyigba: Kogi State University, CSC305, 2006.

Views and opinions expressed in this article are the views and opinions of the author(s), Review of Computer Engineering Research shall not be responsible or answerable for any loss, damage or liability etc. caused in relation to/arising out of the use of the content.