International Journal of Scientific & Engineering Research, Volume 9, Issue 8, August-2018 809
ISSN 2229-5518

An Automated Scheduling System for University Lectures and Examinations

Iwara ARIKPO[1], Abigail OKOKON[2]

Abstract— Scheduling has been a real-life combinatorial problem over the years. University timetabling, which is part of time scheduling, represents a difficult optimization problem, which most universities are grappling with; hence, developing software systems that automate timetabling has also been a challenging task. The aim of this study was to develop an automated system that can simplify the lecture and examination timetabling in the University. The system was designed with an object-oriented analysis and design approach using the unified modelling language. The resulting web application was implemented using the Java Enterprise Edition version 6, with MySQL Server as the backend database system. The resultant software schedules lectures and examinations interactively and in batches for the eleven departments of the Faculty of Science at the University of Calabar, Nigeria. Lectures are scheduled randomly in timeslots, satisfying the constraints provided in the system. The output of the resulting timetable is by far better than the manual approach, eliminating venue and time clashes in the process.

Index Terms— scheduling, automated, timetabling, lecture, examination, algorithm, software

1 Introduction

SCHEDULING is a real-life combinatorial problem that is concerned with programming a certain number of events within a specific timeframe. [1], report that, the general area of scheduling has been the subject of intense research for several years now. Researchers have studied literally thousands of scheduling problems. Generally, scheduling problems involve tasks that must schedule on machines, subject to certain constraints to optimize some objective function [2]. [3], defined scheduling as “establishing the timing for performing a task”. Scheduling problems arise in many areas of human endeavour, such as transportation, workforce (e.g. Nurse scheduling), education, sports, and so on. Scheduling and timetabling are synonymous; however, the two can be viewed as two separate activities, with the term scheduling used as a generic term to cover specific types of problems in this domain. Timetabling can therefore, be considered as a special case of generic scheduling [1]. In Universities, scheduling problems often, arise in the course of generating lecture and examination timetables, and are referred to as timetabling. The term timetabling describes the allocation, subject to constraints, of given resources to objects being placed in space-time, in such a way as to satisfy or nearly satisfy a desirable set of possible objectives [4]. Solving a university timetabling problem manually often requires a lot of time and resources in terms of manhours and losses due to clashes and cancellations. The focus of this study is the development of an automated timetabling system for scheduling lectures and examinations.

2 Literature Review

Many researchers have shown great interest in the area of scheduling and timetabling. A wide variety of studies, from the field of operations research and artificial intelligence, have addressed the spectrum of university timetabling problems [5].

Evolutionary techniques have been applied to solve the timetable scheduling problem. Methodologies like Genetic Algorithms (GAs) and Evolutionary Algorithms (EAs) have been used with mixed success. Researchers have employed many different approaches over the years in an attempt to generate ‘optimal’ timetabling solutions subject to a list of constraints.

[6], have developed a mathematical programming model for faculty lecture assignments, where each variable represented a full teacher schedule, with the problem formulated as a set of partitioning problems with side constraints. [7] conducted a research in Germany on the school timetabling problem. This study, it will be noted, described software implementations and the underlying approaches utilized by the German institutions, most of which were based on direct heuristics. [8] stated the various timetabling problems in a formal way and provided different formulations for them. [9] surveyed the approaches to the examination timetabling problem with a focus on approaches that are based on the reduction to graph colouring problems. [10] surveyed the application of genetic algorithms to timetabling. [11] addressed what the study termed “Quantitative Modelling and Technology Driven

Departmental Lecture Scheduling". [12] studied "Decision Support Systems for University Lecture Scheduling". His work was centred on the development of a tool to provide human-assisted timetabling. [13] presented a decision support system using Analytic Hierarchy Process (AHP). [14] applied genetic algorithms (GA) and sequential local search to generate a lecture timetable. This study selected GA as a well-suited tool for university course timetabling problem because it is used to search large nonlinear solution spaces, where there is a lack of expert knowledge, or the encoding is difficult. [15] applied Modified GA and Cooperative GA to solve the university timetable scheduling problem. Many researchers have proposed different approaches such as, simulated annealing ([16]; [17]), tabu search [18] and evolutionary algorithms [19]; each solving a particular timetabling problem.

3 Materials and Methods

3.1 System design

The system design approach for this lectures and examinations time scheduling is based on the object-oriented analysis and design (OOAD) methodology, using the unified modelling language (UML). The system design components are presented in Figures 1 – 3.

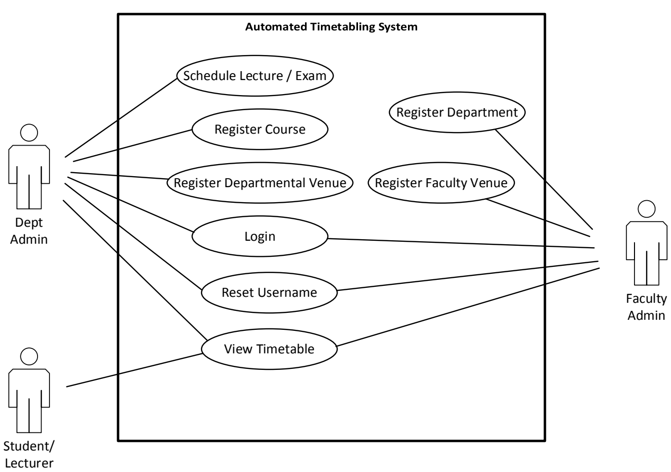

Figure 1: Use case diagram for the automated timetabling system

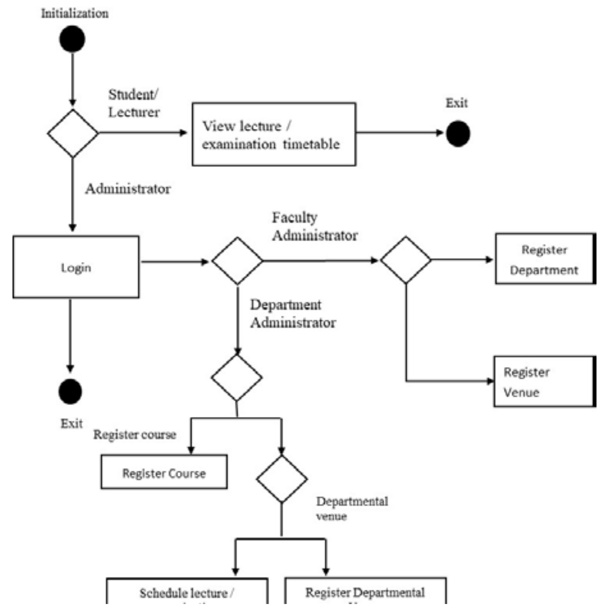

Figure 2: Activity diagram for the automated timetabling system

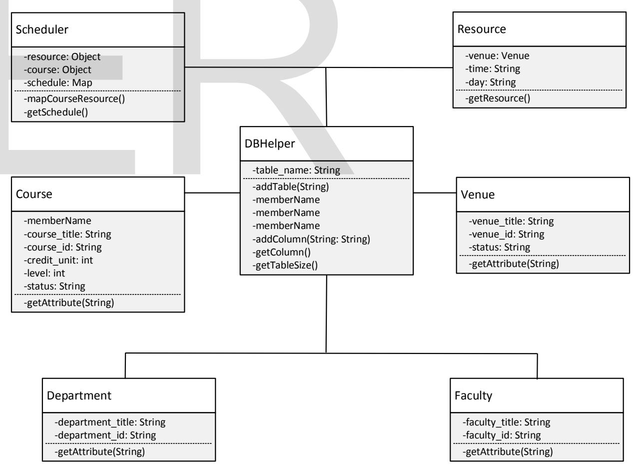

Figure 3: Class diagram for the automated timetabling system

The processing logic of the system, programmed as Servlets, was implemented using the Java Web Technology running the Java Enterprise Edition version 6 (JEE6); while the user interface and presentation logic was developed with Java Server Pages (JSP) and HTML. At the backend, the sever side was deployed with Apache Tomcat, while the database system was implemented on MySQL Server.

International Journal of Scientific & Engineering Research, Volume 9, Issue 8, August-2018 811
ISSN 2229-5518

4. Results

The automated time scheduling system for University, handles the creation of timetables, which undertakes the scheduling of time for lectures and examinations, as well as the venues of these activities for the entire Faculty, including joint lectures involving multiple departments. As should be anticipated, the system validates users, granting higher privileges to the system administrators. Besides, the system provides database management mechanisms to facilitate persistent data storage as regards courses, examinations, lecture venues, departments, and so on.

Lectures are scheduled randomly in timeslots, satisfying the constraints provided in the system. The output of the timetable when compared with the manual equivalent is better; resolving venue and lecture time clashes whilst adhering to the constraints defined in the systems model. Figures 4 – 7 are screenshots from the automated timetable scheduling application.

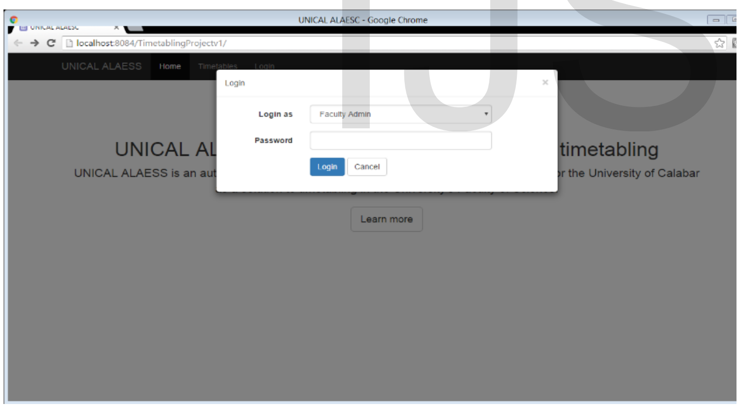

Figure 4: Session login for users

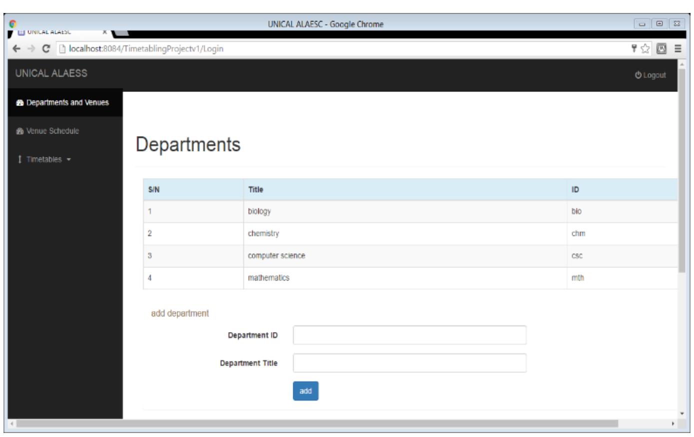

Figure 5: Administrator Panel

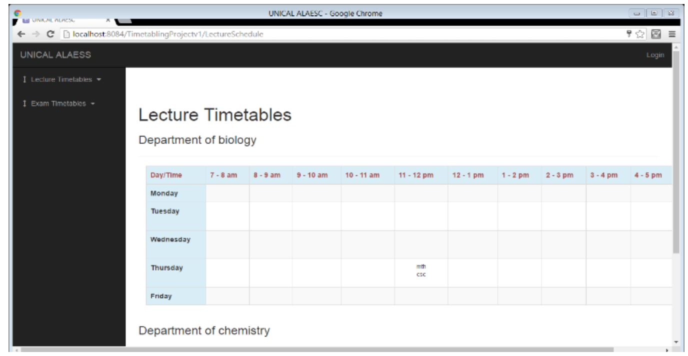

Figure 6: Partial lecture timetable

```
import java.io.IOException; import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession; import tts.DBHelper;
@WebServlet(name = "Login", urlPatterns = {"/Login"})
public class Login extends HttpServlet {
  protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
      throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    DBHelper db = new DBHelper();
    String username = request.getParameter("admin");
    String password = request.getParameter("password");
    if (db.accessUser(username).equals(password) && username.equals("faculty")) {
      HttpSession session = request.getSession(true);
      session.setAttribute("department", username);
      request.getRequestDispatcher("/admin.jsp").forward(request, response);
    }else if (db.accessUser(username).equals(password) &&
!username.equals("faculty")) {
      HttpSession session = request.getSession(true);
      session.setAttribute("department", username);
      request.getRequestDispatcher("/dept_admin.jsp").forward(request,
```

Figure 7: Code Listing for a segment of the Session Management Servlet

5. Conclusion

The tasks of lecture and examination scheduling are a critical component of any University Administration. The automation of the timetabling problem has consequently attracted the attention of several researchers. This study, in contribution to efforts by other researchers, has developed a timetabling system for lectures and examinations, which was tested within the Faculty of Science, at the University of Calabar, Nigeria. The authors believe that, with minor modifications, the software can be a useful tool for generating timetables for other faculties in Universities.

6. Future Research

While this study has addressed the problem of time scheduling for intra faculty lectures and examinations in a University, it is the view of the authors that, this has only provided opportunities for future research. For instance, the application developed in this study, has not addressed time clashes for "carryover" students (those who offer lower level failed courses while in higher levels). Another potential avenue for further research is the scheduling for multi-faculty venues, (i.e., those shared by multiple faculties). Addressing these issues in a future research will enhance the efficiency of this application and those developed by other researchers.

Acknowledgment

The authors wish to thank all the Staff of the Faculty of Science, and the Central Timetable Officers at the University of Calabar, Nigeria, who provided the expert information during the development of the timetabling application.

References

[1] Sandhu, K. S. (2001). Requirements for the degree of Doctor of Philosophy. School of Management. Nathan Campus. Griffith University: *Automating Class Schedule Generation in the Context of a University Timetabling Information System.*

[2] Abdennadher S. & Edward M. (2007). Department of Computer Science, 5th Settlement New Cairo City. *Constraint-Based Examination Timetabling for the German University in Cairo*.

[3] Wight, O. W. (1984). Van Nostrand Reinhold Company, NewYork: *Production and Inventory Management in the Computer Age*.

[4] Wren, A. (1996). Scheduling: *timetabling and rostering – A special relationship,* pp. 46–75.

[5] Abdullah, S. (2006). Heuristic Approaches for University Timetabling Problems. PhD thesis: The School of Computer Science and Information Technology, University of Nottingham.

[6] McClure & Wells, C. (1984). A mathematical programming model for faculty course assignment: Decision Science.

[7] Junginger, W. (1986). Timetabling in Germany: A Survey. *Interfaces*, 16(4), pp. 66-74.

[8] de Werra, D. (1985). An introduction to timetabling. *European Journal of Operational Research,* 19(1), pp. 151–162.

[9] Carter, M. W. (1986). OR Practice – A Survey of Practical Applications of Examination Timetabling Algorithms. *Operations Research,* 34(2), pp. 193-202.

[10] Ross, P., Corne, D. & Fang, H. (1994). Improving evolutionary timetabling with delta evolution and directed mutation. In: Davidor, Y., Manner, R., Schwefel, H. (eds.): *Proceedings of the 3rd International Conference on Parallel Problem Solving from Nature*, Springer, Berlin, pp. 556-565.

[11] Boronico, J. (2000). Quantitative modelling and technology driven departmental course scheduling. *Omega*, 28(2), pp. 327-346.

[12] Pesenti, M. (2002). Decision Support System for University Course Scheduling.

[13] Parthiban, P., Ganesh, K., Narayanan, S. & Dhanalakshmi, R. (2004). In IEEE International Engineering Management Conference: *Preferences Based Decision-making Model (PDM) for Faculty Course Assignment Problem*.

[14] Abdullah, S., Burke E. & McCollum, B. (2007). In CEC: *A Hybrid Evolutionary Approach to the University Course Timetabling Problem*.

[15] Ghaemi, S., Mohammad T. V. & Ali A. (2007). Using a Genetic Algorithm Optimizer Tool to Solve University Timetable Scheduling Problem: Signal Processing and Its Applications. *9th International Symposium*.

[16] da Fonseca, M., Martins, I., Ye, M., Constantino, M. & Cadima, J. (2014). Modeling target volume flows in forest harvest scheduling subject to maximum area restrictions. *Top*, 22(1), pp. 343-362.

[17] Zhang, D., Liu Y., M'Hallah R. & Leung S. C.H. (2010). European Journal of Operational Research: *A simulated annealing with a new neighbourhood structure based algorithm for high school timetabling problems*, 203(1), pp. 550-558.

[18] Bello, G. S., Rangel, M. C. & Boeres, M. C. (2008). 7th International Conference on the Practice and Theory of Automated Timetabling: *An Approach for the Class Teacher Timetabling Problem*.

[19] Raghavjee, R. & Pillay N. (2010). In Second World Congress on Nature & Biologically Inspired Computing: *Using genetic algorithms to solve the South African school timetabling problem*, pp. 286–292.

AUTHOR DETAILS:

Corresponding Author:

Abigail OKOKON[2]: Department of Computer Science, University of Calabar, Nigeria.

Iwara ARIKPO[1]: Department of Computer Science, University of Calabar, Nigeria. Email: iwara.arikpo@unical.edu.ng; iiarikpo@gmail.com