



Bayero Journal of Pure and Applied Sciences, 10(1): 320 - 325

Received: June, 2016

Accepted: March, 2017

ISSN 2006 – 6996

DESIGN AND IMPLEMENTATION OF AN ANDROID AND WEB-BASED UNIVERSITY TIMETABLE CUSTOMIZATION SYSTEM

Muhammad, S.H.¹, Galadanci, B.S.¹, Mustapha, A.¹ and Yahaya, A.S²

Department of Software Engineering, Bayero University, Kano¹ Kano, Nigeria.

Center For Information Technology, Bayero University, Kano²

shmuhammad.csc@buk.edu.ng; +2348039647291

ABSTRACT

Timetabling is an essential component of scheduling in all sectors of education. University timetabling entails the scheduling of times and venues for lectures and examinations. These activities are logically structured in a time-wise manner and presented on papers and placed on notice boards in order to avoid conflicts of event. However, timetable schedule forgetfulness, misinterpretation, miscopying, cost of rescheduling and slow time to reach for the rescheduled timetables are the challenges posed by the use of paper and notice board-based timetables; hence, the need for automated timetables. In this paper, we discuss the implementation of Android and web-based timetable customization systems for the Faculty of Computer Science and Information Technology, Bayero University, Kano-Nigeria. The web-based system provides an administrative module that facilitates scheduling and rescheduling of lecture, examination and invigilation timetables. In addition, the Android timetable customization system permits the user to view, customize, set reminders and provide rescheduled timetable updates.

Keywords: Android, Customization, Timetable and Web based

INTRODUCTION

A timetable is a manuscript that serves as a complete guideline for time-wise allocation of resources to subjects, so as to minimize or eradicate the occurrence of conflict (Ozcan, 2003) and satisfy a set of desirable objectives (Kumar, et al., 2013). A timetable specifies which object meets at which resource and at what time. In an educational setting, the objects are students, lectures and courses with infrastructures and time as the resources.

Timetables play an important role in every educational setting (Gonzalez-rubio, 2006). However, apart from its significance towards ensuring that conflicts of schedules do not occur, the complexity attached to understanding scheduled timetables is fast becoming a major problem. Furthermore, students and lecturers have reported several cases of scheduled timetable misinterpretation, miscopying and forgetfulness (students arriving at a lecture or an exam after the scheduled time) (Carter and Laporte, 2004; Qu, et al., 2009). In addition, timetable administrators have reported high costs of scheduling and rescheduling, slow time to reach and cost of damage (Burke & Newall, 2004). This has necessitated the need for individual-centric timetables, and consequently the need for process (timetabling) and product (timetable) re-engineering, hence automation of timetable processes (Werra., n.d.).

Several platforms including web and Android technology have been adopted for automating timetabling and

digitizing timetables. The web timetabling approach simply involves integrating web development technologies for the implementation of web applications that permit timetable scheduling (Chan, & Zhao, 2014). This web application provides interfaces for timetabling through web browsers only (Chan, et al., 2014).

However, there is a growing concern over the single usage of this technology for timetabling rather than hybrid (Attewell, 2005; Aakjeer, 2015). To harness this, using the most common academic institution timetable (for lectures, examinations and invigilation), this paper designs and implements web and Android applications for scheduling, customizing and enforcing timetable schedules. A web application is implemented for scheduling timetables, while a mobile application is used for accessing; customizing and enforcing schedules (setting reminders) based on user category and preferences:- hence hybrid approach. The timetable schedules of the Faculty of Computer Sciences and Information Technology, Bayero University, Kano-Nigeria is used as a study test case.

MATERIALS AND METHODS

Software Engineering system development principles were adopted for the implementation of the timetable customization system. Waterfall and Iterative models (Aggarwal & Yogesh, 2007) were used as the process models for the web and Android application respectively.

The web application was implemented on Programmers Notepad Text Editor using markup, programming, scripting and query languages. Markup language (HTML) and styling sheet (CSS) were used to design and format the Graphical User Interfaces. Programming (PHP) and scripting languages (Javascript and Javascript Jquery) were used to handle the logic and behavior of the layouts. Query language (MySQLi) was used to query the database of the application. For the Android application, ADT configured Eclipse (Kepler Standard Edition) was used for its implementation. This involved the use of XML for Markup, Java for programming and android inbuilt Shared Preference storage for user preference storage. Furthermore, Android Volley Network Library was used to establish the bi-directional communication between the web server and the Android application using JavaScript Object Notation (JSON) as the interpreting language. These applications were tested against ambiguities, Application Not Responding (ANR) issues and user expectation mismatch.

RESULTS

Modeling of the Proposed System

Object oriented analysis and design methods are becoming the most rated and widely used methods for computer system design (Churcher, 2007). This study uses the object oriented design approach which focuses on the data that are to be manipulated by the program and in the program (Aggarwal&Yogesh, 2007). The Unified Modelling Language (UML) has become the standard language in modeling object oriented systems. It is extensively used for modeling software systems and is increasingly utilized for designing non-software systems and organizations (Churcher, 2007). The flow of data for the web and Android systems are described using use case diagram system modeling approach as shown in Fig. 1 and 2 respectively.

The use case diagram in Fig. 1 shows the various possible interactions between the Admin/Lecturer and the web system. The interactions include login, scheduling of the timetable (lectures and examinations), uploading the timetable (lectures, examinations and invigilation), broadcasting latest information, managing accounts and viewing the "about the system".

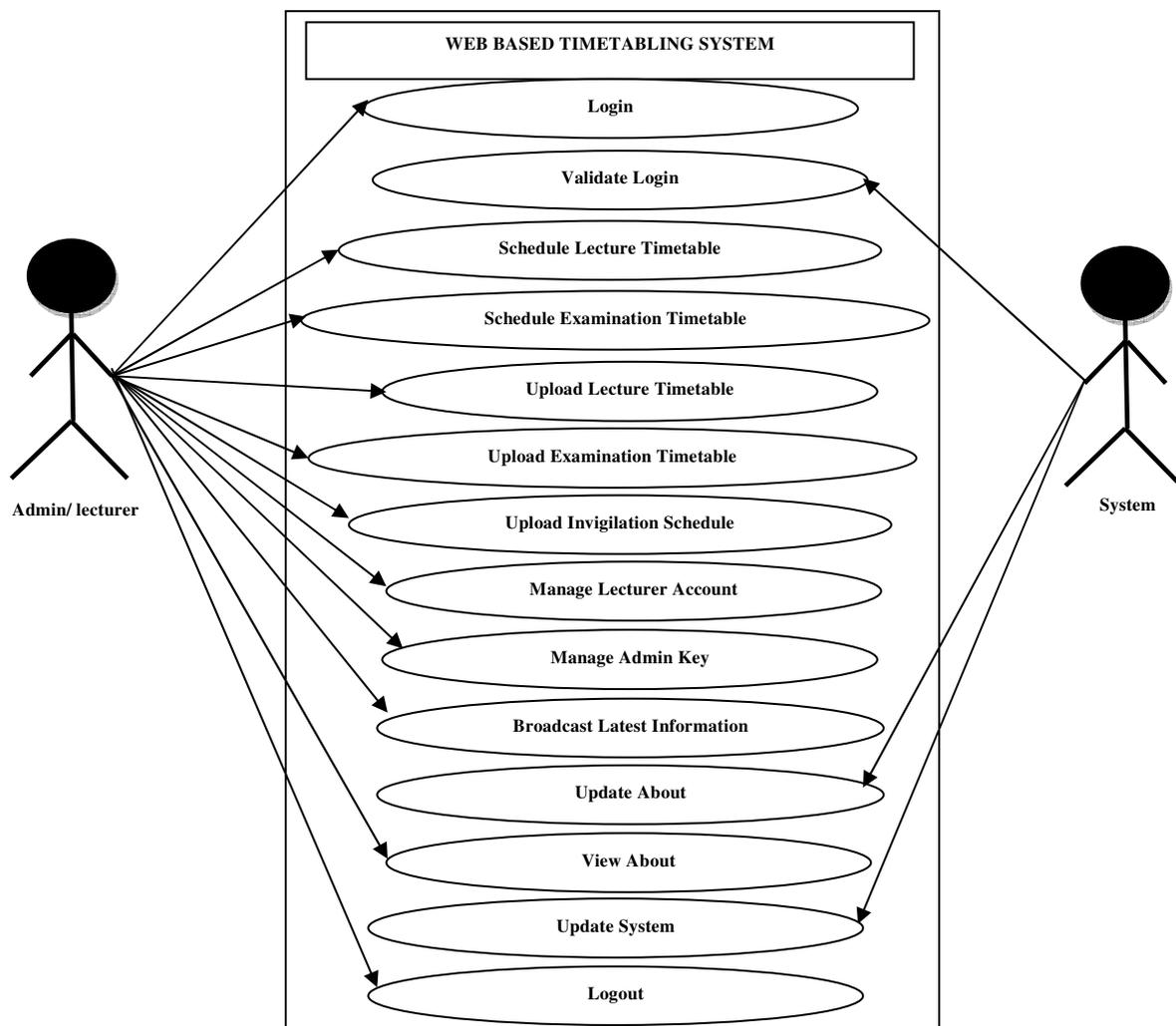


Fig 1: Web Based Timetabling System Use Case Diagram

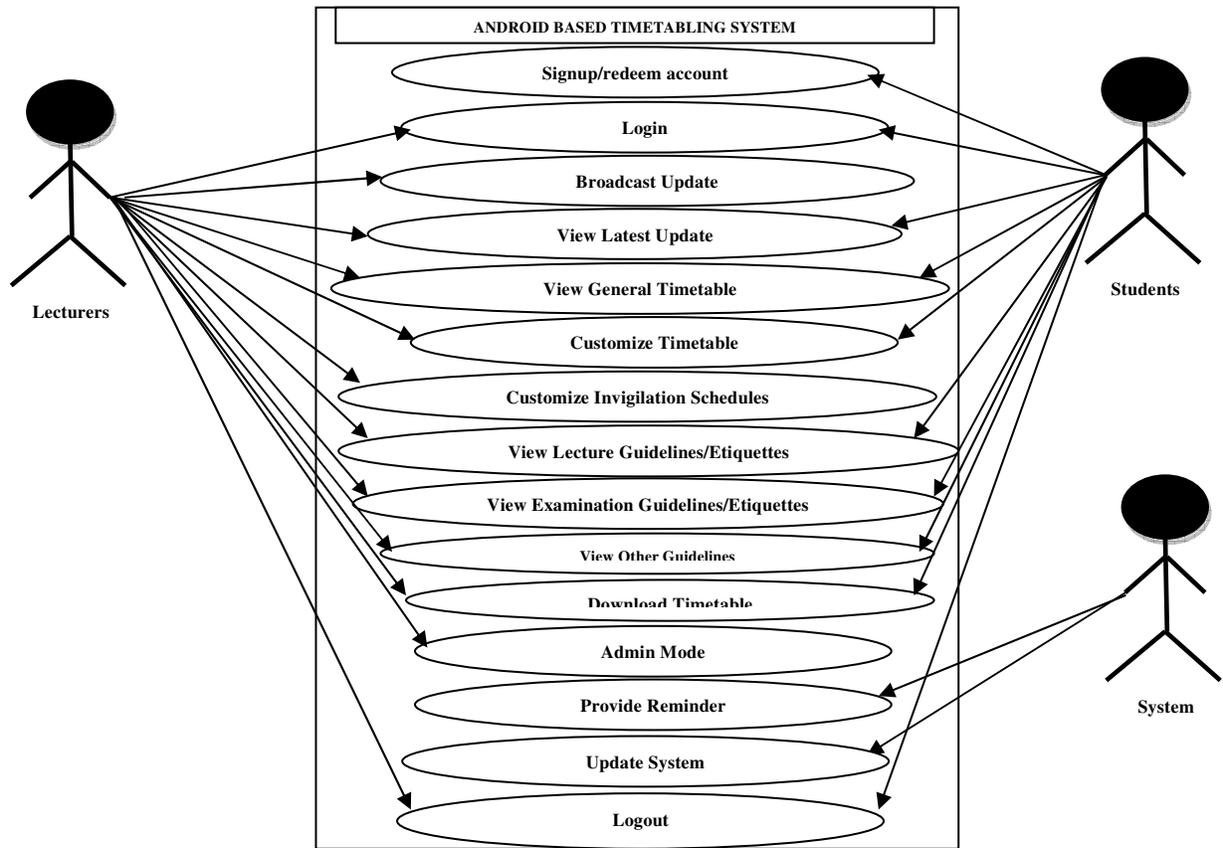


Fig 2: Android Based Timetabling System Use Case Diagram

The use case diagram in Fig. 2 shows lecturers and students as the major actors of the Android application. Basically, the two actors can login, view latest updates, view and customize schedules, download timetables, view guidelines and logout. Specifically, a lecturer can also broadcast update and access admin mode.

Functional Components of the System

The timetabling system comprises of the web and Android applications. The web based application is made up of an administrative module while the Android application is made up of student and lecturer modules.

A. Administrative Functional Components

The web application is strictly for the administrative management of the system. It consists of the authentication (Fig. 3) and dashboard (Fig. 4) pages.

The authentication page provides the interface for authenticating the user of the system. Upon successful authentication, the dashboard page loads the lecture schedule module by default which consists of schedule timetable, upload timetable, manage accounts, FAQs, contacts and credits tabs. The schedule timetable provides the sub-menu for course-wise scheduling of lectures and examinations. In addition, the upload timetable provides the sub-menu for broadcasting latest information and uploading lecture, examination and invigilation schedules in image and Portable Document Format (PDF). Furthermore, the manage accounts holds the sub-menu for managing lecturer accounts and creating administrative keys.

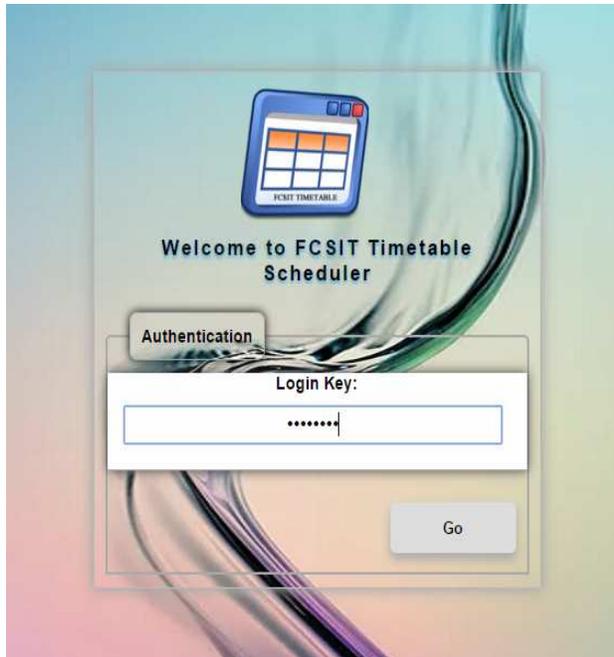


Fig 3: Index Page

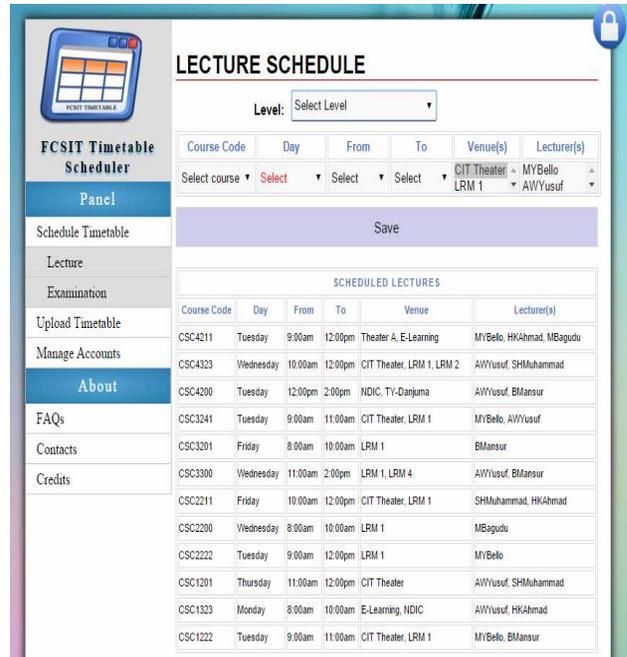


Fig 4: Dashboard Page

Additionally, the about menu has a sub-menu that includes the Frequently Asked Questions (FAQs), contacts and development credits. Also, positioned at the top-right part of the page is a logout icon for terminating administrator session.

B. Lecturer Functional Components

The functional components available to lecturers are provided on the Android application as shown in Fig. 5 and 6. However, lecturer accounts are created by the administrator on the web application. Therefore, using an Android powered or .apk executable mobile phone, the installation and execution of the application gives

the lecturer the privilege to logon to the application's main activity, broadcast and view the latest schedule updates, and view and customize lecture and invigilation timetables. In addition, a lecturer can also be an administrator, and can thus view lecturing, invigilation and other etiquettes.

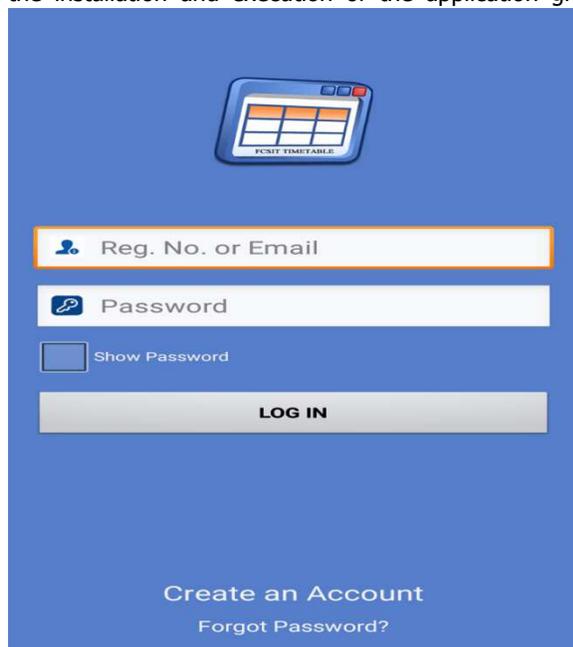


Fig 5: Authentication Activity

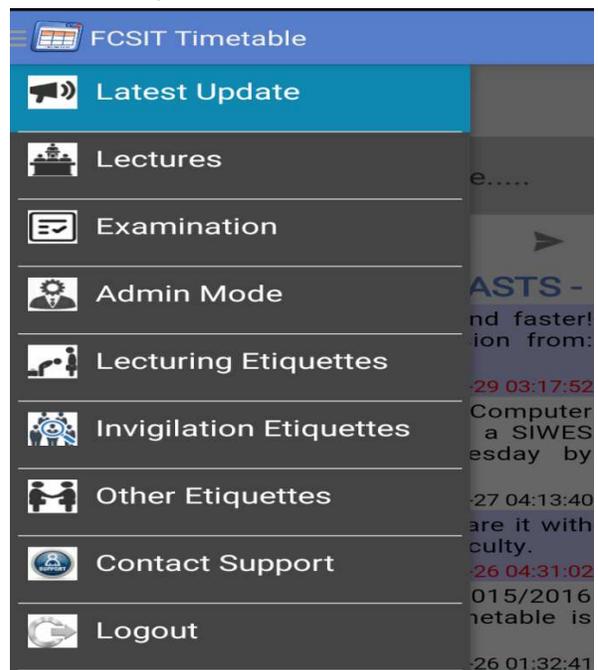


Fig 6: Lecturer Functional Components

The "General Timetable" navigation option shown in Fig. 6 triggers the timetable customization component shown in Fig. 7. This is the major component of the system. It consists of four Buttons, one Spinner and two hidden Text Views. The Spinner holds the various course codes in descending order of course level. The

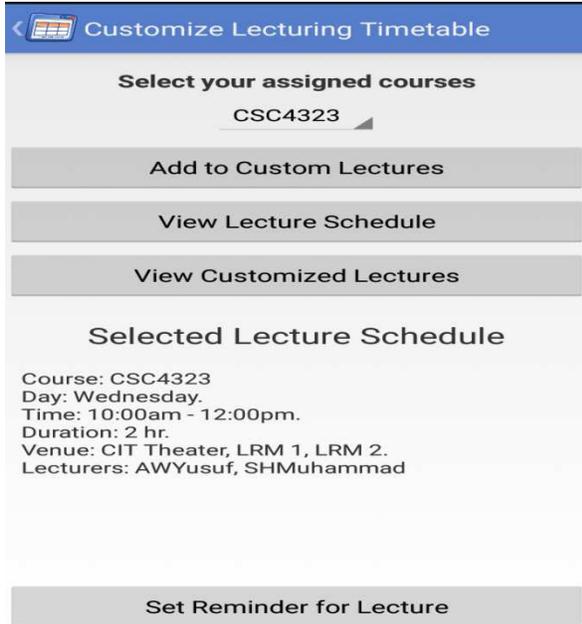


Fig 7: Timetable Customization

The "view customized lectures" button redirects the lecturer to a fragment that holds the list of all the customized schedules (Fig. 8). The fragment presents a structured table headed with course code, day of lecture, duration of lecture, venue of lecture and also the lecturer(s) facilitating the course.

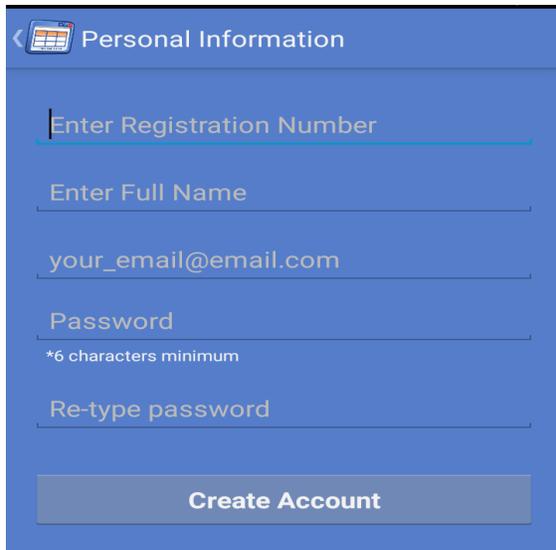


Fig 9: Account Creation

Buttons are arranged in order of precedence; hence, a course must first be added to custom before it can be viewed. Similarly, a reminder can only be set for viewed courses. In a case where some courses are added to custom, the courses can be viewed by tapping the "view customized" button.

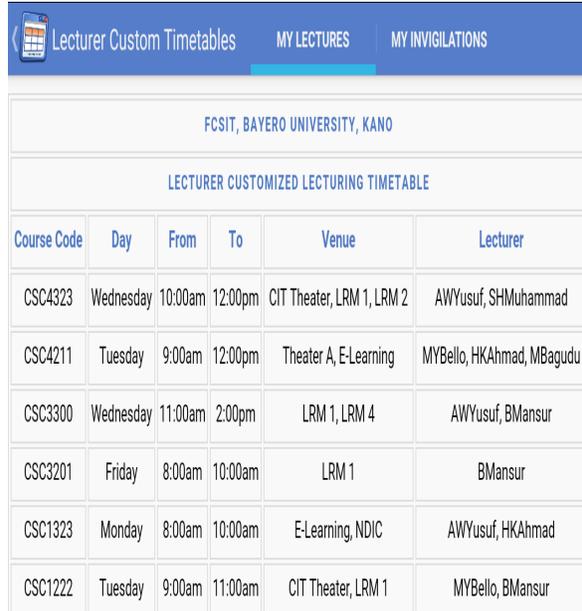


Fig 8: Customized Schedules

C. Students Functional Components

As a student, the functional components are provided on the Android application. Unlike lecturers, students are required to create accounts (Fig. 9) on the Android application.

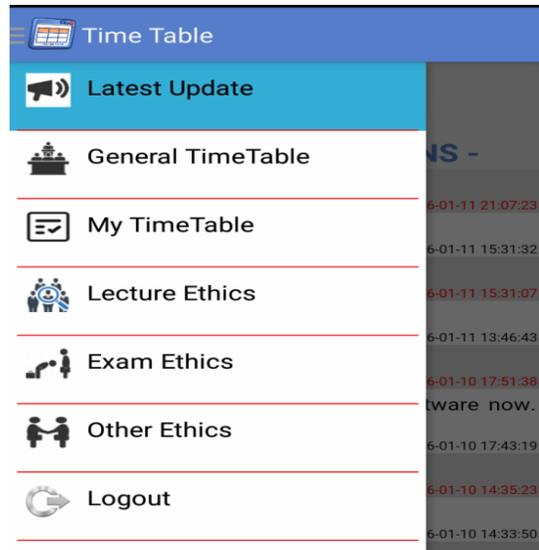


Fig. 10: Student Functional Components

However, students share the same authentication activity (Fig. 5) with the lecturer. Upon successful authentication, the functional navigation option available to students as shown in Fig. 10 includes latest update, general timetable (lectures and examinations), my timetable (lectures and examinations), ethics (lectures, examinations and others). Additionally, the schedule customization procedure for students is similar to that described previously in Fig 7. and 8.

DISCUSSION AND CONCLUSION

The complexity attached to understanding scheduled timetables, timetable misinterpretation, miscopying and forgetfulness of schedules are fast becoming a major problem in every university. This paper discussed the design and implementation of an Android and web-based timetable customization system. The system oversees the scheduling, rescheduling, customization and on-the-go reminder facility for timetable schedules.

REFERENCES

- Aakjeer, A. (2015). Open Innovation: Popularity, Performance and Usability of Mobile Application Development Technologies. *Cambridge, MA, Harvard Business School Press*, 1–10.
- Abdullah, S., Burke, E., & Mccollum, B. (2005). An investigation of variable neighbourhood search for university course timetabling. *The 2nd Multidisciplinary ...*. Retrieved from [http://www.asap.cs.nott.ac.uk/external/mista/2005/papers/An Investigation of Variable Neighbourhood Search for University Course Timetabling.pdf](http://www.asap.cs.nott.ac.uk/external/mista/2005/papers/An%20Investigation%20of%20Variable%20Neighbourhood%20Search%20for%20University%20Course%20Timetabling.pdf)
- Aggarwal, K.K & Yogesh, S. (2007). *Software Engineering* (3rd ed). New Age International Publishers.
- Asratian, A.S., and D. D. (2002). A university timetable problem: Complexity and existence problems. *European Journal of Operational Research*.
- Attewell, J. (2005). Mobile Technologies and Learning: A Technology Update and m-Learning Project Summary. *The Learning and Skills Development Agency, London, 1*, 1–19.
- Burke, E., Kingston, J., & Werra, D. De. (2004). 5.6: Applications to Timetabling. *Handbook of Graph Theory*. Retrieved from https://books.google.com/books?hl=en&lr=&id=mKkIGIea_BkC&oi=fnd&pg=PA445&dq=Burke,+E.K.,+de+Werra,+D.,+Kingston,+J.+Timetable&ots=VX6ERPv1vu&sig=lvjUJtbj8Ei18TghTTGtePxA7E
- Burke, E., & Newall, J. (2004). Solving examination timetabling problems through adaption of heuristic orderings. *Annals of Operations Research*. Retrieved from <http://link.springer.com/article/10.1023/B:ANOR.0000030684.30824.08>
- Carter G., and Laporte, M. W. (2004). Recent Developments in Practical Course Timetabling. In E. Burke and M. Carter (eds.), *Practice and Theory of Automated Timetabling II*. *Springer-Verlag New York, 1408*, 3–19.
- Carter, M.W., Laporte, G., Lee, S. T. (1996). Examination timetabling: algorithmic strategies and applications. *Journal of the Operational Research Society, (47)*, 373–383.
- Chan, S.W., & Zhao, W. (2014). An Implementation of a Web-based Timetabling System. *School of Information Systems. Curtin University of Technology, 3(1)*, 1–13.
- Churcher, C. (2007). *Beginning Database Design*. (J. Gennick, Ed.). *New York, United States of America: Springer-Verlag New York, Inc.*
- Ferland J.A. and Fleurent, C. (2008). SAPHIR: A Decision Support System for Course Scheduling. *Interfaces, 24(2)*, 105–115.
- Gonzalez-rubio, R. (2006). Multi-Site Timetabling, 416–419.
- Kumar, A., Singh, K., & Sharma, N. (2013). Automated Timetable Generator Using Particle Swarm Optimization. *Ijritcc.Org*, (September), 686–692. Retrieved from [https://www.ijritcc.org/IJRITCC_Vol_1_Issue_9/automated timetable generator using particle swarm optimization.pdf](https://www.ijritcc.org/IJRITCC_Vol_1_Issue_9/automated%20timetable%20generator%20using%20particle%20swarm%20optimization.pdf)
- Michalewics, Z and Fogel, D. B. (2000). *How to Solve It: Modern Heuristics*. *Springer*.
- Mom, J. M., & Enokela, J. A. (2012). Implementation of a Time Table Generator using Visual Basic . NET, *7(5)*, 548–553.
- Ozcan, E. (2003). Towards an XML based standard for Timetabling Problems: TTML. *TTML*.
- Post G, Ahmadi S, Daskalaki S, Kingston JH, Kyngas J, Nurmi C, Ranson D, and R. H. (2008). An XML Format for Benchmarks in High School Timetabling. In *7th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2008)*, Montreal. Montreal.
- Qu, R., Burke, E., Mccollum, B., Merlot, L., & Lee, S. (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*. Retrieved from <http://link.springer.com/article/10.1007/s10951-008-0077-5>
- Silva, J., Burke, E., & Petrovic, S. (2004). An introduction to multiobjective metaheuristics for scheduling and timetabling. *Metaheuristics for Multiobjective ...*. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-17144-4_4
- Werra., D. de. (2004). An introduction to timetabling. *European Journal of Operational Research, 19(2)*, 151–162.