International Journal for Quality Research 14(4) 1191–1206
ISSN 1800-6450

Duaa AlSaeed[1]

Article info:
Received 20.06.2020
Accepted 14.09.2020

UDC – 005.336.3; 37.018
DOI – 10.24874/IJQR14.04-13

TOWARD ACHIEVING QUALITY IN FACULTY-LOAD ALLOCATION: A DEVELOPED FACULTY-LOAD-MANAGEMENT SYSTEM

Abstract: *In most academic institutes, there is an increasing demand on faculty members to perform not only more teaching and research but also community service and administrative works. Simultaneously, they also face growing pressure to improve the quality of their work. Therefore, maintaining a balance among these different tasks is a major issue in academic institutions, as it is imperative for academic administrators to ensure that the faculty members work productively and are also satisfied. Information systems that will help manage workload allocation and accurately measure academic input and output must be maintained. These systems will directly affect academic performance management. Accordingly, we propose and develop a faculty-load--management system (FLMS) for our university. The proposed system manages workload allocation and time-conflict detection in addition to information management and reporting. The proposed FLMS system is based on the client/server model. Two databases were developed, one for faculty and the other for courses, sections, and administrative tasks. On the client side, there will be a browser, which is responsible for user interaction, and it connects to a web portal (server), which performs data processing. FLMS was developed and tested for our academic department. It was evaluated by the faculty members and administrators, and the results showed that the system met their requirements and helped in maintaining both quality and faculty satisfaction. However, based on user feedback, some improvement to the FLMS is planned as future work.*

Keywords: *Faculty-load allocation, Quality, Scheduling, Database, Usability, Academics*

1. Introduction

For a nation, education establishes the foundation for the overall development. Some of the key members who actively contribute to this activity are faculty members. With the changing times, the landscape of education has changed as well. Because of the globalization of education, one can find multiple streams in different domains, thereby signaling a strategic change from the traditional educational framework to a considerably modern framework. Even the knowledge delivery mechanisms have changed, as they are being studded with new approaches and strategies. However, these

[1] Corresponding author: Duaa AlSaeed
Email: dalsaeed@ksu.edu.sa

changes have impacted the knowledge givers more than the knowledge seekers. Consequently, today's faculty members or teaching staff survive in an environment of continuous learning (Naveed et al., 2020; Muhammad et al., 2020; Bauer & Weller, 2019; Jibola Kadir et al, 2020). A modern-day faculty member is considered human capital who is skilled, knowledgeable, and innovative and one who can meet the requirements of the changing landscape of modern education (Lee, 2004). The result of these changes is the ever-increasing workload on the faculty in varied forms, which means that workload is not limited to teaching activities but has been extended to non-teaching activities as well.

Faculty workload has been a pertinent issue among the faculty of various educational institutions, specifically in the institutions that impart higher education. While there is an increasing demand on faculty members to perform more teaching, research, and community service, there is also increasing pressure to improve the quality of such works (Izabel & Hani, 1997); therefore, maintaining a balance among the aforementioned different tasks is a major issue in academic institutions. It has been identified as one of the top three challenges faced by professors in their institutions, apart from the challenges associated with the career stage and navigating through the institutional culture. However, the situation becomes worse with the lack of attention toward this issue, generally in traditional academic settings (Baker et al., 2012). Moreover, the issue prevalently seems to exist with the administrators who fail to allocate workload in an efficient and fair manner. In addition, there seems to be lack of a mechanism for measuring the workload allocated to the faculty. Faculty job satisfaction is equally important in terms of organizational performance, as is customer satisfaction (Comm & Mathaisel, 2003; Memon et al., 2018), thereby emphasizing the importance of achieving quality and balance in the process of workload allocation, thereby achieving faculty job satisfaction.

In an educational institution, the commitment of the faculty is critical to developing the institution (Naveed et al., 2019). Although the term "faculty" reflects a particular role of the traditional teaching staff, it has a considerably wider scope. This is because, faculty are engaged not only in teaching or imparting education but also performing research activities, thereby improving the quality of the overall education. In addition, they are engaged in different forms of administrative tasks, which are required to organize different activities in a university. Therefore, the tasks allocated to faculty members are combinations of different tasks, and there is no way to differentiate in the timeline of the different types of tasks, neither is there a mechanism using which each proportion of the task can be appropriately measured. For example, a faculty member might like to spend 50% of his/her time teaching, 30% on research, and the remaining 25% on administrative tasks, but he/she may actually be spending 25% of the time on teaching, 20% on research, and the rest on administrative tasks. In total, faculty workload is critical to determining both the quality of education given in educations institutions and level of self-satisfaction derived by the faculty.

We propose and develop a more structured approach, as well as a solution, for faculty-workload management. Currently, there exists no mechanism or framework using which the workload distribution amongst faculty members could be measured. Consequently, the workload distribution amongst faculty members might be unequal and, simultaneously, the allocation process inefficient. Therefore, in a particular department, one faculty member might be overloaded with tasks while another might have a manageable number of tasks. Moreover, the distribution of different forms of tasks cannot be tracked. This is a gap that exists in the university, and the university administrators, including the Head of Department (HoD), lack the appropriate

mechanism or tool to fill the gap. The following are the main goals of this study:

- Developing a platform for facilitating efficient task allocation among the faculty.
- Striking an appropriate balance between administrative and teaching tasks.

We aim to develop an environment for faculty members in which they can balance their tasks without being overloaded. The following are some of the other key objectives of the solution:

- Developing a faculty-load-balancing tool for the academy.
- Equipping the HoDs with real-time reporting on faculty load, for enabling efficient task allocation.
- Developing a comprehensive database with all the details pertaining to faculty members, departments, courses, and the tasks allocated to the faculty members.
- Facilitating database management for the administrator while performing efficient task allocation.

In the next few sections, we will provide a background on faculty workload, along with the different forms of workload that most faculty members are subjected to. Furthermore, we discuss the research conducted on sample faculty members to develop the practical and detailed understanding of faculty-workload assignment and the impact thereof. Subsequently we propose a detailed solution that can be used to address the research problem. Finally, we draw key conclusions and indicate future recommendations.

2. Background

Academicians have oftentimes asserted that the workload of faculty members frequently differs in terms of individual basis, subjects they teach, and the university or academy where they teach. Therefore, ingtion might not be uniform. Practically, the time spent on the workload of the teaching staff or faculty members usually exceeds that spent on teaching activities or other forms thereof. Moreover, sometimes, they spend more time on non-teaching tasks or administrative tasks, rather than on teaching tasks, which comprise their primary role. These is the main problem of faculty members, directly depriving them of their academic freedom and presenting them with workplace challenges. However, to develop an appropriate solution to this problem, we must understand both the type of workload assigned to faculty members and the manner in which the workload is measured.

2.1 Estimation and Evaluation of Workload

One of the important aspects of workload management is the effective measurement of the workload allocated to faculty members. Without effective measures or evaluation methodologies or tools, it is difficult to determine the amount of workload assigned to faculty members, thereby rendering the administrators with negligible or zero visibility regarding the workload. Accordingly, there is high probability that either the workload allocation amongst faculty members is unequal or it is unfair allocation. Analyzing the existing literature and practices, there exists no framework/mechanism to track the time spent on workload or to determine the proportion of the workload allocated. In addition, there is no pre-determined routine in the allocated tasks; i.e., a faculty member might complete certain tasks in a week or a few. Similarly, some tasks might be completed only at a certain time of the year. Therefore, there could not be a specific pattern in workload completion. In addition, there is no fixed workload-measurement unit in the existing practices. Some studies consider weeks as measurement unit, while others consider a month or quarter.

A different mechanism is used to measure the faculty workload. For example, few institutes use the credit system, which is based on the points assigned for each task and each student. The points are also referred to as credit points. The sum of the total credit points determines the workload assigned to each of the faculty members. However, the major defect of the credit system is that it does not measure the administrative workload, and the measurement merely takes place to teaching workload. However, most institutions follow the self-reporting mechanism, in which faculty members report the tasks they have undertaken. Mostly, they make entries of the task in a dedicated register. Periodically, the administrative officials check the register and determine the workload assigned to faculty members. However, the self-reporting mechanism has multiple defects. First, it does not offer real-time visibility on the allocated workload. Second, the workload could be measured only once the tasks have been registered by faculty members. Third, faculty members could introduce a human error while reporting the tasks..

Therefore, on the basis of the above-mentioned discussion, traditionally, both the credits earned by faculty members or the self-reported contact hours by the faculty members have been the mechanisms to measure the work hours. Although contact hours show the work hours of the faculty members, they do not have considerable information on the complexities involved in the tasks, or more details regarding the tasks. However, credits reflect both the nature and complexities of the task, but they do not correctly measure the work hours. Therefore, the current mechanisms clearly lack a platform that will record the contact hours and credits, and generate instant reports on faculty workload. Notably, instant reporting enables clear visibility of the current faculty workload.

Some of the top lecturer management software available are : LearnSpeed (Learn speed, 2020) and SkillsLogic (Skills logic, 2020) TimeCruncher (Tutor cruncher, 2020) TutorPanel (Tutor panel, 2020), all these are examples of business modelled educational information management systems. These systems focus more on the business part of billing, scheduling and recruitment of faculty members without providing features for monitoring or controlling the tutors' workload.

We can conclude, that it is considerably important for academic administrators to ensure that faculty members work productively, and, therefore, we must maintain information systems that would produce correct information, such as academic workload, and accurately measure academic input and output, which is an important issue that will directly affect academic-performance management (Parsons & Slabbert, 2001).

Faculty load can be categorized into, at least, the following four broad categories: (1) teaching, (2) research, (3) administration, and (4) community interaction (Bitzer, 2007). Each of these types will be explained in the following subsection.

Teaching Workload

The primary responsibility of faculty members is to perform teaching tasks, as per their specialties or streams. A faculty member is primarily hired in the institute on the basis of the teaching assignments given to him/her upon hiring. The teaching workload comprises contact-based teaching (classroom teaching / online teaching), supervision of the students (preparation of course materials, follow-ups with the students), attending scheduled faculty meetings (inter-department faculty meet, faculty orientation meeting). Faculty members spend approximately 40 percent of their time in teaching. Because teaching is their primary responsibility, it is fairly liable that faculty members spend most of their time on teaching, as previously established via different studies. However, some studies show that oftentimes teaching workloads are extended to weekends as well.

Consequently, faculty members have to spend their weekends on completing their teaching workloads. In addition, studies show that working during weekends affects the performance of the faculty staff in the long run. The administrators might not be aware of this without an appropriate reporting platform, thereby resulting in dissatisfied and underperforming faculty members.

Research and Consultation Workload

Faculty members often spend some time in research and consultation regarding the subject in which they specialize. This helps them gain key insights on the latest developments in that particular subject. Simultaneously, thorough research helps them discover the unknown aspects of the subject. Sometimes, they are also involved in peer reviews, which entail thorough consultation and discussion. Peer review involves reviewing both the research and findings of other faculty members from the same or different institute. The overall workloads related to research and development helps faculty members acquire more knowledge on a given subject and help them excel professionally. Although this benefits the institute, most institutes consider this as the individual achievement of a faculty member or his/her individual effort to gain more skills and knowledge. Therefore, no mechanism exists to track the workload related to research and consultation. In some institutes, the workload related to research and consultation are considered the part of the teaching workload. Accordingly, there is no clear demarcation between the research and consulting workload and the teaching workload. In addition, no framework exists to track the hours spent on research by a particular faculty member.

Most institutes encourage their faculty members to participate in research-related activities, as it helps the faculty members, as well as professors, to gain more knowledge, thereby indirectly helping the institutes to raise the overall quality of education imparted by them. In addition, when a professor publishes papers or research findings or peer reviews a paper, his/her credentials are usually associated with his/her university or institute, thereby enhancing the reputation of the educational institute as well. Therefore, institutes always encourage their teaching staff to be involved in research-related activities.

Administrative Task Workload

Faculty members have to spend approximately 30 percent of their time on administrative tasks. Different types of administrative tasks are assigned to the faculty staff. Administrative tasks are an important contribution made by the faculty members of an institute toward the better functioning of the institute. Some key administrative tasks assigned to faculty members are the following: preparation of different kinds of reports (such as reports at organizational and departmental levels), coordination of different kinds of programs within the institute or university (such as cultural programs, research programs, fests, and sport programs), heading different committees or teams formed for different purposes and occasionally even different types of clerical tasks. It is estimated that a faculty member spends approximately 20 percent of his/her work hours on administrative tasks, which are not even his/her primary job role. However, if the aforementioned percent figure exceeds, the faculty member might suffer from task overload, thereby affecting the primary job role, i.e., teaching. However, the current approaches lack the appropriate framework to limit the workhours spent on the administrative tasks and prevent any kind of task overload created by the assignment of administrative tasks.

Community Service and Development

Oftentimes, universities conduct different programs for the betterment of society or the community as a whole. These programs are

implemented with the help of faculty members and volunteering students. They include weekend education for poor children, slum-cleaning drives, plantation drives, and issue-based awareness drives. Usually, these programs are voluntary, and, accordingly, no mandatory workload is assigned. Therefore, it completely depends on the faculty members whether they want to participate in these programs. Community-development programs are implemented and made successful with the joint efforts of teaching staff, non-teaching staff, and students. Although the programs are voluntary, there is no platform to record the contributions of the teaching staff toward community development. In addition, there are no appropriate records regarding the nature of the program in which the staff participated or coordinated.

3. Methodology

To determine the current problems pertaining to faculty workload, we had to perform a study that would provide better visibility to the underlying problem. The design adopted for this research is exploratory, and it is based on qualitative-research methodology. This design enabled us to explore the research problem to the maximum extent, thereby allowing the participants to provide the maximum details on the underlying problem. For this research, ten faculty members were selected from the university via random sampling. All the selected faculty members were from the same university. Upon receiving their consent for the research, they were inducted into the research and were provided interview slots. The primary instrument for this research was an open-ended structured questionnaire. Each participant was provided the same questionnaire, and his/her response was recorded separately. The responses of the participants were analyzed using thematic analysis, which is the most common data-analysis procedure in qualitative methodology.

3.1 Challenges and issues with faculty-load management

During the analysis of the responses collected from the interview, some common themes that emerged using thematic analysis indicated the underlying problem. One of the common themes reflected in all the ten responses was the "lack of accountability." The respondents mentioned that there was no accountability for the workload allocated to them, as there was no mechanism to track the workload allocated to them. Among the ten respondents, six mentioned that due to the lack of accountability, most of the time workload allocation was not fair, or rather it resulted in "unfair allocation." They further elaborated that sometimes unfair allocation would create excessive workloads, especially in cases wherein one of the faculty members was available for tasks, but still tasks would be allocated to another member who was previously burdened with tasks. Therefore, it is clearly shown that unfair task allocation resulted in work overload, which was one of the two interdependent themes. In addition, six participants responded that on multiple occasions they had more administrative workloads than teaching workloads, thereby directly affecting their primary role of teaching. Consequently, it affected the completion of the curriculum assigned to them; on many occasions, the completion was delayed because of the excessive overload of the administrative tasks. Another key theme that emerged from the responses is that most participants rued that although technology was important in the day-to-day functioning of the university, it was not implemented to perform faculty-workload management.

From the qualitative analysis, it is evident that there are multiple issues and challenges that have negatively affected faculty-workload management. Therefore, the issues and challenges could be summarized as follows:

- Currently, no framework/ platform/ mechanism is available for performing fair workload allocation amongst faculty members.

- There is no method to determine the proportion of the faculty workload, which comprises teaching, research and consulting, administrative, and community service and development workloads.
- There is no real-time visibility of the workload allocated to faculty members.
- There are instances of task overload due to inappropriate and unorganized workload allocation.

4. Proposed FLMS system

On the basis of the issues and challenges arising in faculty-workload management, a solution framework was modeled and then designed. The key solution to the current problems was developing a platform that would act as tool to perform faculty-workload management.

4.1 Framework

The requirements of faculty members were analyzed using the responses collected during the interview. A separate session was conducted with the administrators who assigned tasks, and their requirements were also analyzed; alternately, two members from the IT team were selected to understand their requirements from the implementation perspective. The user requirements determined the functional specifications of the developed platform. The following are the key user requirements:

- Developing an inventory in the form of a database that will store all the key information of faculty members and their workloads.
- Developing a platform to record, store, edit, and manage information.
- Assigning tasks using the platform to keep appropriate records of the tasks assigned.
- Developing real-time visibility on workload assignment by generating real-time reports and providing the option to print the reports.
- Options to customize the platform as and when required.

Upon analyzing and then documenting the user requirements, the key users, along with their characteristics, were determined. The following are the key users:

- DB-admin: He/she registers faculty members in the databases with their skillsets, reports, and tasks assigned (including administrative tasks teaching tasks).
- Faculty-load-management system (FLMS) Administrator: He/she assigns tasks, generates real-time reports on task assignment, and has the authority to manage the database.

From the requirement analysis, the following main requirements were established:

- There will be the following two types of user logins based on the user characteristics: the FLMS-administrator Login and the DB-admin Login.
- All the information of faculty, courses, sections, and tasks will be registered by the DB-admin.
- The DB-admin can print customized reports on faculty workloads
- The DB-admin might add and edit the administrative tasks, as well as print real-time reports.
- The FLMS-administrator login will be for the administrator only.
- The FLMS-administrator will assign and unassign tasks via the FLMS-administrator login.
- Administrators can reset the workload by clearing the workload from the FLMS-administrator login.
- The FLMS-administrator can generate real-time reports on the workloads of the faculty members via the FLMS-administrator login.
- The FLMS-administrator can perform all the tasks of the DB-admin, but the vice versa is not possible.

4.2 Algorithm

The challenge of workload allocation and management can be divided into the following three main issues:

- How to measure and calculate workload?
- How to detect a conflict in a schedule?
- How to allocate tasks among faculty members?

To calculate the workload hours for administrative and teaching tasks, we use the following formula:

$$Total\ workload\ = \\ Administrative\ workload\ + \\ Teaching\ workload$$

where administrative and teaching workloads are defined as follows:

$$Administrative\ workload\ = \\ \Sigma\ assigned\_tasks\ (weight\ hours)$$

$$Teaching\ workload\ = \\ \Sigma\ assigned\_sections\ (contact\ hours)$$

To detect a conflict, we define time slots during the week, for each section; subsequently, when a section is assigned to a faculty member, the system will check whether there is a conflict with a previously assigned section, and if a conflict is detected, the user will be alerted and the assignment rejected.

To assign tasks, the proposed solution is based on the pre-emptive algorithm, which functions in a Round-Robin (RR) manner. Pre-emptive task allocation ensures that the task is allocated to the faculty member for a fixed time only, and that automatically the resource (faculty member) will be marked as free after the assigned time is over. The algorithm is detailed as follows:

- There will be two queues, namely, the task queue and resource queue.
- Tasks will be registered and queued in the task queue.
- All free resources will be queued in the resource queue.
- As soon as a resource is free, it will be added to the resource queue.
- The task will be assigned to the first available resource in the resource queue, in an RR manner without any priority.
- The assigned task will be scheduled for a fixed time to the resource in the resource queue; after the fixed time elapses, the task will be automatically marked as free.
- If the resource is unable to finish the task within the assigned time slot, he/she must raise an exception, following which a new task will be created. The newly created task will be allocated as a separate task to the next available resource in the resource queue.

The algorithm ensures that none of the resources are overloaded with tasks, and that all the resources receive an equal share of tasks.

4.3 Requirements

Based on the solution framework developed in the last section, a higher-level design of the solution was created by adding more functionalities to both the modules, i.e., “Administrator” and “DB-admin,” As per the solution framework, the proposed solution will the following two sub-modules, each with a different set of user characteristics and functionalities: the FLMS-administrator and the DB-admin. The solution has been developed on the basis of the assigned characteristics of the given user types. Each submodule has an assigned set of specific tasks, which are aligned to its functionalities. The following are the functional requirements of the system developed:

- The FLMS-administrator should be able to assign and unassign tasks.
- The FLMS-administrator should be able to generate real-time reports on the workload allocated.

- The FLMS-administrator should be able to reset the workloads for all the faculty members.
- Faculty members should be able to register themselves in the database.
- The system should allow editing the details of the faculty members, whose details previously exist in the database.
- The system should allow deleting the record of a given faculty member, for example, deletion of faculty members who have left the university.
- The system should allow printing report for the individual workload.

The following are the non-functional requirements of the system:

- Usability: The system developed will be user friendly, i.e., with a user-friendly interface and navigation links.
- Security: The system developed will be completely protected via authorization mechanisms. As of now, only a single level of password-based authentication has been incorporated, thereby ensuring only authorized users can access the tool.
- Maintainability: The current solution can be maintained with the least possible downtime. Moreover, the maintenance can be scheduled during non-business hours, thereby not negatively affecting the business.
- Reliability: The solution developed is completely reliable, as confirmed via different cases.
- Scalability: The proposed system is scalable, as new modules and features can be easily added without making any major changes to the system.
- Performance: The system has been designed to offer maximum uptime, minimal disruption, and fast response rate.

5. Implementation and Testing

The system was implemented using required technologies. The following are the technological/software-related requirements for the system:

- Platform: The platform chosen for the application is ASP.NET. It is an open-source platform, which allows customizing the application as per user requirements. In addition, it allows the integration of the application to the cloud infrastructure, thereby adding considerably more scalability to the application. One of the key reasons for selecting this platform was its better performance than those of other platforms.
- Web Server: The Apache Web Server has been used as the webserver for hosting the application. It is an open-source framework. The key reasons for selecting it as the webserver are its low cost (because it is free), satisfactory performance, and enhanced security. Furthermore, it is flexible to use because of its module-based structure.
- Database: MySQL has been used as the backend database for this system. It is free and can process considerable number of concurrent requests, meaning that even in the case of heavy traffic, it can perform in an optimized manner. In addition, it is one of the most secure databases used for web applications.
- Scripting Language: PHP has been used as the scripting language for the following two reasons: first, it is free, thereby reducing the total cost of ownership, and second, it is perfectly compatible with MySQL. Because it is a server-side scripting language, by default it is protected by the security framework of the server.

- Testing: Testing is significantly important before deploying a particular solution in a production environment. The solution was thoroughly tested using the tool "TestRail." Testing was performed in terms of performance (load testing), usability, and security. TestRail provides an interactive dashboard, which provides a snapshot of the test results in a single screenshot.

5.1 System architecture

The FLMS system is based on the client/server model. The server holds a database for the information of system users and courses, sections, administrative tasks, faculty members, and system users. On the client side, there will be a browser, which is responsible for user interaction. In addition, the browser connects to a web portal (Server), which performs data processing. The client/server architecture is adopted because of the need for centralized data processing. The architecture of the proposed FLMS system is depicted in Figure 1.

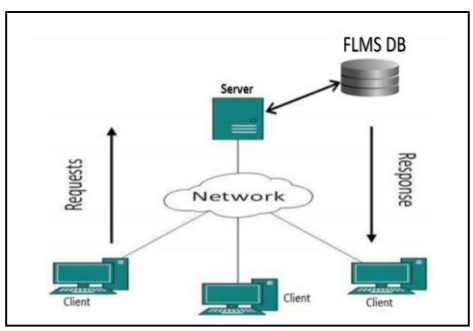

Figure 1. FLMS system architecture

5.2 Main components

The proposed FLMS system has two main components, namely, FLMS database management and workload management, each of which is related to a certain task. The main screen of the FLMS system, with both the main components is depicted in Figure 2.

- **FLMS database management:** This part of the system builds and manages the FLMS database, which contains the information of all faculty members, courses, sections, and administrative tasks. This part of the FLMS systems can only be accessed by the DB-admin and the administrator. The main screen of this component is depicted in Figure 3, which contains the following four basic parts of the FLMS database: faculty, courses, sections , and administrative tasks. For each part of the database, the user can perform basic database-management operations such as add, edit, view, and delete.
- **Workload management:** This forms the core of the system, and it performs workload assignment\editing. This part of the FLMS systems can only be accessed by the FLMS-administrator of the system. The following are the basic load-management operations:
  - **Assign task:** Using this operation, a task is assigned to a faculty member, where the task can be a teaching load (sections of courses) or an administrative task such as related to committees or a student advisory task. While assigning a task, the admin will be able to select the available (unassigned) course, sections, or admin tasks from the lists saved in the FLMS database. In the case of assigning teaching load (course sections), the system will detect any time-conflicts with previously assigned tasks, and if time-conflicts are detected, it will not allow the assignment operation. After each assignment operation, the system will update the total workload for the faculty member.

- **Unassign task:** Using this module, the user can unassign certain tasks from the list of the tasks assigned to a faculty member, and the system will accordingly update the total workload for that faculty member.
- **View\Print all work:** This module enables the user to view/print all the workload assignments for all faculty members or the list of them.
- **Clear all work assignments:** The user will be able to clear all assignments, and this is usually done at the beginning of a semester to clear all the assignments to begin a new workload-allocation process.

In Figure 4, the main modules in the screen of the workload-management component are depicted.

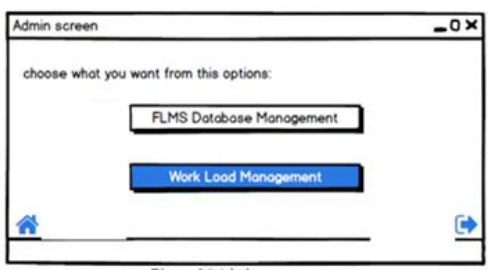

Figure 2. FLMS main screen—view of the administrator

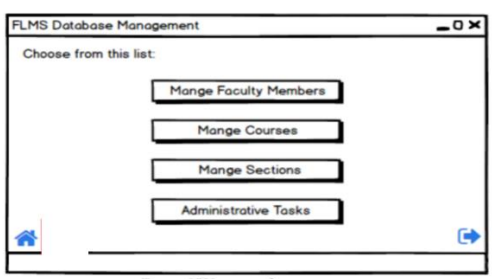

Figure 3. Main modules of the FLMS database management component

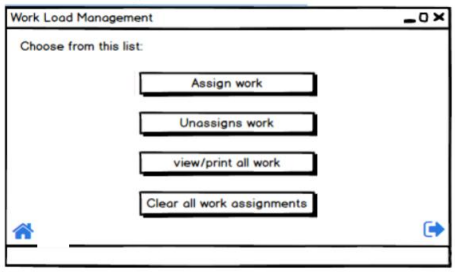

Figure 4. Main modules of the FLMS Faculty-load management component

5.3 Testing

Before deploying the solution, testing was performed in different phases. The testing was necessary to ensure that the developed solution remained stable post implementation and met the solution requirements. An appropriate and effective testing will ensure the elimination of post-implementation risks. The following testing modules were applied to the solution developed:

- **API Testing (Unit Testing):** A solution comprises multiple APIs, each of which is referred to as the unit of software. In API testing, all the software APIs were tested as per the API specifications. Primarily, we checked whether the APIs performed as per the specifications or their assigned functionalities. This test was automated via the SoapUI testing, where the behavior of each API was tested using different inputs and condition checks.
- **Integration and Regression Testing:** In integration testing, new modules were added to the previously tested modules, to integrate a module with the previously tested modules or APIs. Via integration testing we checked how the previously tested modules accepted the newly integrated module, and vice versa. The aforementioned step is extremely important, as the modules are not

completely independent of one another and are interdependent on one other. After integration testing, regression testing was performed. In regression testing, we checked whether the solution met the functional requirement, after introducing small changes in the code. The changes were incorporated to address the solution glitches that occurred during the integration testing, thereby ensuring that the end solution completely met the functional requirements. The regression testing was implemented via iteration testing. In this kind of testing, the insight was taken from previous tests, and the tests were iteratively performed.

- **Performance and Load Testing:** Performance testing was performed using the IBM Rational Performance Tester. The performance of the application was tested in different test cases, considering various scenarios and different inputs. Simultaneously, load testing was performed using various loads, and the application performance was analyzed for different loads. Both performance and load testing also determined the overall effectiveness of the application. The following are some of the key tests performed using the rational tool:
    - The URL-based testing was performed to discover non-functional and broken links. Using this testing, we checked whether all the navigational links within the portal were functional.
- The URL-based testing was performed to discover non-functional and broken links. Using this testing, we checked whether all the navigational links within the portal were functional
    - The database-based testing was performed, wherein multiple tasks were used for storing data in the database. In Figure 5, a screenshot of the database testing is depicted.
- Stress testing: During stress testing, a load mix was developed and applied on the solution to test the endurance thereof. The load mix comprised randomized clicks from various users. A screenshot of the stress testing is depicted in Figure 6.

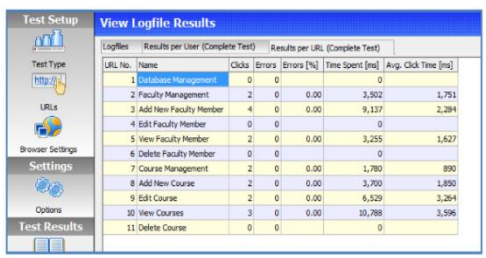

Figure 5. Log screenshot of database testing

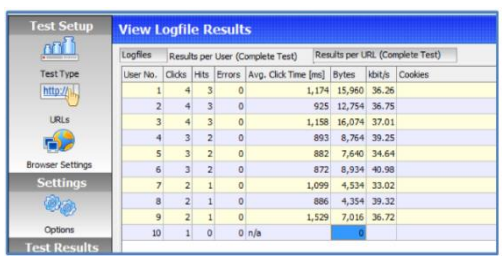

Figure 6. Log screenshot of stress testing

- **User-Acceptance Testing:** Other than tools-based testing, user-acceptance testing was manually performed. The testing involved performing common user tasks by the users, i.e., the FLMS-administrator and DB-admin. The purpose of the testing was to evaluate the FLMS website in terms of the provided functionalities and design conveniences. It also helps gain more accurately predict the usability and effectiveness of the application. During the user-acceptance testing, the end-users

(customers) of the system evaluate how the system confirms to its initial requirements. We chose some faculty members and administrators as DB-admins and FLMS-administrators, to test our system. In Figures 7 and 8, the results of the user-acceptance testing for the FLMS-administrator and DB-admin, respectively are depicted.

Testing Task	#of Errors	Time in Sec.	Result
Login to the System	0	20	No Difficulties
Forget Password	0	30	No Difficulties
Management Faculty Member Add - Edit – View – Print - Delete	0	150	No Difficulties
Management Course Add - Edit – View – Print - Delete	0	150	No Difficulties
Management Tasks Task Add - Edit – View – Print - Delete	0	150	No Difficulties
Management Section Add - Edit – View – Print– Delete New Timeslot – Edit Timeslot – Delete Timeslot	0	250	No Difficulties
Management Workload Assign Teaching Load – Assign Administrative Tasks – Un-Assign Sections – Un-Assign Administrative Tasks – Clear all Workload – View Workload	0	300	No Difficulties

Figure 7. Results of the user-acceptance testing for the FLMS-administrator

Testing Task	#of Errors	Time in Sec.	Result
Login to the System	0	20	No Difficulties
Forget Password	0	30	No Difficulties
Management Faculty Member Add - Edit – View – Print - Delete	0	150	No Difficulties
Management Course Add - Edit – View – Print - Delete	0	150	No Difficulties
Management Administrative Task Add - Edit – View – Print - Delete	0	150	No Difficulties
Management Section Add - Edit – View – Print – Delete - New Timeslot – Edit Timeslot – Delete Timeslot	0	250	No Difficulties

Figure 8. Results of the user-acceptance testing of for the DB-admin

6. Conclusion

The solution developed clearly addresses the challenges that were identified as associated with faculty-workload management. One of the key concerns related to workload management was the lack of visibility. The current solution addresses this concern from the perspective of both the administrator and the faculty DB-admin. Consequently, after

the implementation of the solution, an administrator can always generate real-time reports on workload allocation and assign tasks on the basis of both the availability and workloads assigned to faculty members. Similarly, using real-time reports, administrators can ensure that non-teaching workloads do not exceed the teaching workload, which should be the primary workload of faculty members. Therefore, in conclusion, the current solution will facilitate the fair allocation of workloads, thereby reducing overload and human errors and motivating faculty members to perform better.

7. Future Work

The current system suffered from a drawback in terms of its accessibility, as it could be accessed only with the intranet of the university, meaning it could not be accessed from an internal network, thereby limiting the accessibility of the platforms. Therefore, one of the key recommendations for future is to integrate the solution to the cloud. Because the current application is based on the ASP.NET platform, integrating it with cloud is a definite possibility. The integration will ensure that users can access it from anywhere, at any time, as per their convenience. Second, in the current solution, the workload is manually assigned by the administrators using the tool, once they print the report and check the availability. Therefore, the second key recommendation is to automate the workload-allocation process performed by the administrators. Using an inbuilt algorithm, the tool will check the workload and assign the workload. Lastly, today, most of the computing happens on mobile; therefore, it is recommended that a mobile version of the application is developed, thereby providing considerably more flexibility to both faculty members and administrators.

Funding: Researchers supporting project number (RSP-2020/281) King Saud University, Riyadh, Saudi Arabia.

Acknowledgment: The author extends her appreciation to the researchers supporting project number (RSP-2020/281) King Saud University, Riyadh, Saudi Arabia.

I would like to present my appreciation and special thanks to my students, namely, Alaa Alharbi, Samah Dammas, Fatimah Aldossari, and Eman Basha, for helping in implementing the FLMS system.

References:

Baker V., L., Lunsford L. G., & Pifer M. J. (2012). *Developing Faculty in Liberal Arts Colleges*. New Brunswick, NJ: Rutgers University Press.

Bauer, U., & Weller, S. (2019). Controlling the Quality of Industrial Engineering and Management. *International Journal of Industrial Engineering and Management* (IJIEM), *10* (1), 1-6.

Education: Structural Analysis of Austrian Degree Programs and

Implications for Higher Education Institutions

Bitzer, E. M. (2007). Attempting a fair and equitable academic workload distribution in a Faculty of Education. *South African Journal of Higher Education*, *21*(1), 23-15)

Comm, C., & Mathaisel, D. (2003), A case study of the implications of faculty workload and compensation for improving academic quality. *International Journal of Educational Management*, *17*(5), 200-210.

Izabel, S., & Hani, S. (1997). Academic workload and quality, *Assessment & Evaluation in Higher Education*, *22*(2), 135-157.

Jibola Kadir, A. N., Tijani, A. A., & Mafara, R. M. (2020). Proceedings on Engineering Sciences, 2(1), 21-30, doi: 10.24874/PES02.01.003

Learn Speed. (2020). *Manage tutor scheduling, Session Invoicing and Payment Processing.* Retrieved March 17, 2020, from LeasrnSpeed: https://www.learnspeed.com/

Lee, M. N. N. (2004). *Restructuring higher education in Malaysia*. Kuala Lumpur: College teachers.

Memon, F. A., Saeed, S. and Shaikh, A. (2018). Systematic Approach of Customer Relationship Management in Much Different Organization. IBT *Journal of Business Studies* (JBS), *14*(2).

Muhammad, A., Shaikh, A., Naveed, Q. N., & Qureshi, M. R. N., 2020. Factors Affecting Academic Integrity in E-Learning of Saudi Arabian Universities. An Investigation Using Delphi and AHP. *IEEE Access*, *8*,16259-16268.

Naveed, Q. N., Qureshi, M. R. N. M., Shaikh, A., Alsayed, A. O., Sanober, S., & Mohiuddin, K. (2019). Evaluating and ranking cloud-based e-learning critical success factors (CSFs) using combinatorial approach. *IEEE Access*, *7*, 157145-157157.

Naveed, Q. N., Qureshi, M. R. N., Tairan, N., Mohammad, A., Shaikh, A., Alsayed, A. O., Shah, A., & Alotaibi, F. M. (2020). Evaluating critical success factors in implementing E-learning system using multi-criteria decision-making. *Plos one*, *15*(5), p.e0231465.

Parsons, P. G., & Slabbert, A. D. (2001). Performance management and academic workload in higher education. *South African Journal of Higher Education*, *15*(3), 74-81.

Skills logic. (2020). *Online Tutor Management System.* Retrieved March 17, 2020, from Skills Logic: http://www.skillslogic.com/services

Tutor Cruncher. (2020). *Scheduling and Timetabling-TutorCruncher.* Retrieved March 17, 2020, from TutorCruncher: http://www.timecruncher.com/features/scheduling/

Tutor Panel. (2020). *Tutor Scheduling- Tutor Panel.* Retrieved March 12, 2020, from Tutor Panel: https://www.tutorpanel.com/scheduling

Duaa AlSaeed
College of Computer and
Information Sciences
King Saud University
Zip code 4545
Riyadh, Saudi Arabia
dalsaeed@ksu.edu.sa