Faculty of Arts and Languages Information System for PNU

Renato Jose Molano, Catherine Arguelles, Brendan Lim, Charles Ng, Keith Tan
De La Salle University - College of Computer Studies
Information Technology Department
rene.molano@delasalle.ph [1] *Affiliation (8 pt. Century)*

Abstract: Philippine Normal University's (PNU) Faculty of Arts and Languages (FAL) seeks to improve its lengthy and tedious faculty loading process and take the opportunity to employ faculty profiles to monitor faculty members, their information, and their activities. The proponents analyzed the business processes and areas of improvement of FAL and set out to build a comprehensive information system that encapsulates these areas of improvement, with assistive faculty loading features and faculty profiles. These modules are the core of this innovation project that will serve as the pilot for specialized information systems in administrative functions for the Philippine Normal University. The project aims to assist FAL in assigning faculty members to classes and facilitate creating, updating, and storing faculty information. The Agile Scrum Methodology was used throughout the system development wherein the proponents developed the system at the same time taking into account the constant feedbacks and revisions coming from the users of FAL. The system had undergone four stages of testing: Unit testing, integration testing, system testing and user acceptance testing. The proponents accounted for the testing results to further improve the system. The project will further improve the processes of FAL in both faculty profile and faculty loading. Implementing a centralized database will address the issues of FAL with regards to storing and retrieving faculty profiles, while implementing a score-based algorithm will allow automatic and assistive assignment of faculty members which, making the faculty loading process faster and more efficient. Thus, the lengthy and tedious process of faculty loading has been automated and simplified.

Key Words: Faculty Profile; Faculty Loading; Assignment Algorithm

1. INTRODUCTION

1.1 Background of the Study

The Faculty of Arts and Languages (FAL) is one of the colleges of the Philippine Normal University (PNU), a State University located in Manila. They focus on the Literature education, English education, Music and Arts education, and Filipino education courses. Some processes in FAL are still being done manually. Consequently, the group found opportunities in these manual processes that would improve their data management and processes. The system aims to resolve FAL's issues with the faculty loading and faculty profile.

1.2 Purpose and Significance of the Study

Given this, the project is an innovation request. The setup of the project is unprecedented; PNU has not implemented specialized information systems for faculty loading and faculty profiles before. The success of the system will contribute to FAL's higher level of accreditation and better data management. The system will also allow easier data management for FAL by facilitating tracking and management of faculty subject assignments, as well as facilitating the creating, updating, and storing of faculty subject assignments.

Moreover, there exists a social impact in the project. If the system is effectively implemented in FAL, the system will be implemented throughout the entire university. PNU has also stated that a successful information system will serve as a model for systems in other universities.

2. REVIEW OF RELATED LITERATURE

2.1 Master Data Management

Master data management, according to Loshin (2010), is a process wherein organization incorporates management methods, and tools to implement infrastructures that would support integration and capture of data. Moreover, organization links all of its necessary file into one file which is called "master file". This master file is then used to show common data among the files or common point reference. According to Lewis (2009), phases of master data management are discovery (finding data sources), analysis (identifying discrepancies), design (designing metadata), implementation (implementation of metadata) and establish data governance. Moreover, it can be applied in faculty profile. Based from the multiple forms for faculty profile of FAL, master data management can be applied:

- To find faculty profile related forms from different colleges (discovery)
- To check what were the common fields evident and what were the unique ones which can be used in making a standardized form (analysis and design)
- For FAL to have the standardized form for faculty profile (implementation)
- To manage the usability and security of these data coming from the faculty profile (data governance)

2.2 Personnel Management

Personnel management, according to Grimsley (n.d.) and Juneja (n.d.), involves the management of administration concerning with the work of the employees and their relationship within the organization. Moreover, the objective of personnel management is to assign personnel / employees to certain activities and to ensure that the employees have a good relationship with each other. Faculty loading can be related to personnel management since its process involves the associate dean assigning faculty members to certain subjects. In addition with faculty loading, the associate dean also ensures that the faculty members also have a good relationship with each other with regards to the schedule of teaching. This is why some aspects of personnel management can be applied to faculty loading.

2.3 Data Centralization

Data Centralization is when data is gathered, stored, and updated in one place that can be accessed through several points (Jeske, 2015). The benefit of data centralization include quicker searches, less complexity in the IT infrastructure, and reduced redundancy. As for FAL's case, since its faculty profile is based from different faculties and organization, this led them to have difficulties in coming up with reports for its professors. Having the faculty profile centralized in FAL, FAL can easily access its professors' profiles so that necessary reports can be easier to generate.

3. METHODOLOGY

3.1 Agile Scrum Methodology

To make this project possible, Scrum was chosen because one of the attributes is better customer satisfaction due to the active role of the product owner in the methodology (Feliciano-Misla, n.d.). Scrum is also more adaptable and flexible to change than more traditional methods (Cervone, 2011). This is important because the nature of the project involves constant revisions and modifications throughout development. Therefore, the group and the methodology must be able to cope with the changes. The Scrum method is also more efficient and faster than traditional methods. This saved the development team much needed time, given a limited number of months. The team also chose Scrum because of familiarity and usage in previous projects.

3.2 Research Design and Methodology

To follow the Scrum methodology, the team either visited PNU for client interviews or contacted the client through email in order to have the client more involved in the project. These meetings were for the scope and specific requirements for the project. The information and requirements were recorded. These information were then analyzed through weekly consultations with the team's thesis advisor, and through online research.

4. DISCUSSIONS AND RESULTS

4.1 Problem Areas

The motivation of this project stems from problem areas that include delays in faculty loading.

4.1.1 Faculty Loading

Faculty loading is the assignment of faculty members to classes. The group analyzed that FAL has difficulty and delays in generating faculty loads. FAL's associate dean stated that it takes around two (2) weeks to generate the faculty loads because of the manual process of constantly revising and requesting the feedback of faculty members. Considering that the schedule of classes is provided to FAL one month before the start of classes, taking additional constraints, revisions, and approval of faculty members into account can become tedious and time consuming for FAL. In extreme cases, classes start without any faculty members assigned. Students become victims of FAL's inability to resolve the faculty assignment on time.

4.1.2 Faculty Profile

To support the faculty loading process and centralize data collection, the proponents proposed faculty profiles. They contain information about faculty members and their activities within and outside the university. These information include degrees, recognitions, presentations, instructional materials, and extension works.

4.2 Proposed System

With the issues stated above, the group designed a system that addresses them. The system consists of two modules, which are faculty profile and faculty loading. In the faculty profile module, the system allows the storage of profiles in a centralized database. The system is also designed to use the information stored in assigning faculty members during the assignment process. Inside the faculty profiles module, users can create and update profiles of faculty members within FAL's scope, and print their individual profiles.

In order to address the issues in faculty loading, the group designed a score-based algorithm in assigning faculty members, based on the rankings and preferences of the associate dean of subdocuments (supporting documents or activities), professor ranks, number of times a subject has been taught, and number of preparations needed. The system has a feature wherein it automatically assigns faculty members to their classes. Furthermore, the system's algorithm takes into account the information of the faculty member in the previously mentioned module where the scores vary according to their relevance and relationship with the program and the class itself. The system also allows manual loading which assists the user in manually assigning faculty members using the algorithm. The module's features include adding and deleting classes, assigning faculty members, and viewing schedules and the term overview. Furthermore, the module allows users to print term schedules, faculty schedules, and term overviews.

Based on the scoring system below, the system calculates for the faculty member with the highest score, and either automatically assigns them or suggests them in manual assignment.

Table 1. Scoring System Table

Category	**Description**
Professor Rank	Each professor rank has its own corresponding score: - Instructor: 500 pts - Assistant Professor: 400 pts - Associate Professor: 300 pts - Full Time Professor: 200 pts - Part time: 100 pts
Subdocuments	Each subdocument has its own initial points: - Degree: 500 pts - Instructional Material: 400 pts - Presentation: 300 pts - Extension Work: 200 pts - Recognition: 100 pts
Number of Times Taught	- Faculty members who taught the same subject in the past get additional pts depending on the number of times taught but the law of diminishing returns apply. gain=base points*0.8^(x-1) base points = 100 pts
Number of Preps	- Faculty members who teach same subjects but with different time slot get additional 100 points in their score. The limit for the number of preps a faculty can take in the automatic assignment is two (2).
Load Limit	- Faculty members who are below the load limit get additional 100 in their score.
Time Preference	- If the time slot of the class is preferred by the faculty member then the faculty member gets additional 100 pts in their score.

Another thing that the algorithm considers is the Law of Diminishing Returns, which states that in every productive process, adding more of one factor while having others constant, will at some point yield lower incremental per units return (Nordhaus & Samuelson, 2001). Law of Diminishing Returns applies when a faculty member teaches a subject multiple times or adds a subdocument, the incremented score decreases. This is required to reflect the real world characteristics. For example, when a professor teaches a subject multiple times, the experience and knowledge gained will not be as significant as the first time they taught the subject. It also prevents spamming repetitions to unfairly boost their points in the system.

4.3 System Development

To create the system, TypeScript was used as the primary programming language. This language was chosen mainly because it is designed for development of large applications (*Bierman et al, 2014*). Tools were used such as React, MobX, Axios, and Material UI for the client side, while Koa, TypeORM, PostgreSQL and Node.js for the server side. To ensure that data is secure, the passwords are protected, hashed, and salted.

4.4 System Testing and Results

The system had undergone 4 stages of testing: Unit testing, integration testing, system testing and user acceptance testing (UAT). For unit testing, integration testing and system testing, the group made several test cases to check whether each module and functionality works properly.

Table 2. Sample Taken from Actual Test Cases

Test Case ID	**auto_assign1**
Objective	Auto-assign faculty to class schedule
Steps	1. Click classes tab 2. Click auto-assign button 3. Click auto-assign on pop-up
Input Data	None
Expected Result	Shows the schedule with faculty members assigned in each class
Actual Result	Shows the schedule with faculty members assigned in each class
Remarks	Pass

As for UAT, the group went to PNU and the system was tested by FAL's associate dean, FAL's clerk, and three of FAL's faculty members. Based on the results from the user acceptance testing, the overall system had an average rating of 4.77 out of 5. Some users rated 4 out of 5 under the overall system criteria. The user interface gained an average rating of 4.9 out of 5. Most of FAL's users were impressed by the user interface of the system, and commended that it is simple and organized. On the other hand, the functionalities of the system gained an average rating of 4.8 out of 5. Lastly, for the security of the system, it had an average rating of 5 out of 5. This means that when FAL tested the system, it is secure in terms of sensitive information and access controls. However, there were minor bugs found while the client was testing the system.

Table 3. User Acceptance Testing Average Result

Criteria	Average Score (5 being the highest)
Overall System	4.77
User Interface	4.9
Functionalities	4.8
Security	5

5. CONCLUSION

Given that the system developed has addressed the problem areas, there is good reason to believe that the project will improve the processes of FAL. The faculty profile module will store the faculty members' details in a centralized database and assist in class assignment. On the other hand, the faculty loading module will solve the problems in gathering time constraints, assigning classes, and gathering feedback found in the assignment process.

6. ACKNOWLEDGMENTS

This project would not have been possible without our panelists, Prof. Michelle Ching and Prof. Liandro Tabora for their time and insightful comments.

We would also like to thank Mr. Johannes Acuna for aiding us in the project proposal, Dr. Maria Acuna, associate dean of the Faculty of Arts and Languages of The Philippine Normal University, and the faculty members for the continuous cooperation throughout the project.

7. REFERENCES

Bierman, G., Abadi, M., & Torgersen, M.(2014). Typescript. In European Conference on Object-Oriented Programming (pp. 257-281). Springer, Berlin, Heidelberg.

Feliciano-Misla, C. (n.d.). 4 reasons to use scrum in software development. Retrieved from https://invidgroup.com/ 2017/06/06/4-reasons-to-use -scrum-in-software-development/

Grimsley,S. (n.d.). Humanresourcemanagement vspersonnelmanagement. Retrieved from https://study.com/academy/lesson/ human-resource-management -vs-personnel-management.html

Jeske, S. (2015). The benefits of streamlining data centralization [Blog post]. Retrieved from https:// www.datasciencecentral.com/profiles/ blogs/ the-benefits-of-streamlining-data-centralization

Juneja, P. (n.d.). Personnel management. Retrieved from https:// www.managementstudyguide.com/ personnel-management.htm

Lewis, R. (n.d.). 5 Steps of Master Data Management. Retrieved from https:// www.slideshare.net/embarcaderotech net/5-steps-to-master-data-management

Loshin, D. (2010). Master data management. Morgan Kaufmann.

Nordhaus, W. & Samuelson, P. (2001). Microeconomics. McGraw-Hill, New York USA.