

Revitalizing Internship Registration Portal with an Implementation of Application Lifecycle Management

Bertha Alan Manuel

Maranatha Christian University

bertha.alan.m@gmail.com

Oscar Karnalim

Maranatha Christian University

oscar.karnalim@it.maranatha.edu

Abstract—Internship is an undergraduate activity for gaining working experiences. Hence, it is expected that some universities convert it as an undergraduate course. In our university, registering internship topic is automated using a web-based portal; it takes tremendous effort if done manually. Nevertheless, according to our informal survey, existing portal suffers a lot of limitation, resulting user dissatisfaction. This paper presents a further development of such portal by following phases described in Application Lifecycle Management. According to three evaluation scenarios (black-box, small-group user, and large-group user testing), proposed portal works well and fits user need.

Keywords—internship registration portal, application lifecycle management, computer-aided administrative data management, software development, software maintenance.

I. INTRODUCTION

Internship is an undergraduate activity for gaining working experiences. It is frequently implemented as either working under a direct supervision with real company or working as a research assistant on university. In our faculty, each student is encouraged to experience such activity; it becomes a mandatory undergraduate course.

Despite its benefits, converting internship as a course generates several issues. One of

them is a tremendous effort required to manage all internship data, especially at registration phase. Hence, a web-based portal has been developed at our faculty. It helps the coordinator (i.e. a person who is in charge to manage all internship data) to manage all registration data submitted by the students. This portal has been used since 2012.

Nevertheless, informal observation shows that some users are dissatisfied with existing portal; it has numerous drawbacks (e.g. inconvenient layout and functionality). This paper serves a further development of given portal to handle such drawbacks. As our guideline, we will use Application Lifecycle Management (ALM)[1], a widely-used lifecycle for software development and maintenance.

II. RELATED WORKS

Teaching undergraduates is not a trivial task; it could generate specific issues that are exclusively found on that domain. Those issues, on most occasions, become emerging topics on research. Some emerging topics resulted from such phenomenon are alumni tracer [2], student outcome[3], [4], plagiarism detection [5], [6], computer-aided self-learning[7], [8], and computer-aided administrative data management [9]–[12].

Among these topics, computer-aided administrative data management takes the most attention; all administrative processes can be easily automated. Several successful

examples of such automation are the works proposed in [9], [10], [11], and [12]. Firstly, a work proposed in [9] automates lecturing session allocation which should be done once per semester. Secondly, a work proposed in [10] automates student and lecturer attendance through RFID system. Thirdly, a work proposed in [11] automates saving graduation data through a web-based portal. Finally, a work proposed in [12] automates laboratory helpdesk calling via a web-based portal, connected through local area network.

According to the fact that internship is a beneficial activity to hone student's soft skill, it has been converted as a course on our faculty. On that course, students will experience internship activity for six months. In addition, they will also learn how to document their work with final project template; this mechanism does not only provide an experience to create large document (which is frequently needed at work) but also a preparation for their final project activity.

Before enrolling the internship course, each student should register internship topic one semester before the actual enrollment. At that time, the feasibility of proposed topic is evaluated. Further revision will be given if the topic is not feasible for internship. Since such registration activity is rather complex due to numerous entities involved, an internship registration portal was developed. The portal has been used for internship registration since 2012 in our university.

Generally speaking, internship registration at that portal works as follows:

- a) Students register their topic through the portal on predefined batches. To date, there are two predefined batches; one of them is held at mid-term weeks while the another is held at final-term weeks. Both batches are

used to register internship topics for the successive semester.

- b) Internship coordinator checks the topics and their supplementary files. If some files are missing or do not meet coordinator's criteria, the topic will be on hold and the student should complete it for further processing.
- c) The topics will be distributed to three research groups regarding to their scope. The chief of each research group will check whether the topic fits his/her research group. If it is out of the scope, it will be returned to internship coordinator and the coordinator will assign the topic to other groups.
- d) The chair of each research group will assign a reviewer for each proposed topic; on most occasions, the reviewer will become the supervisor about proposed topic.
- e) Each assigned reviewer will check the feasibility of given topic and provide some revisions if necessary. Further, he/she can also ask the student to meet him/her directly for further enquiries about proposed topic.

Nevertheless, as years go by, some users are dissatisfied toward given portal. Hence, a further development of internship portal is necessary. As our guideline, we will use Application Lifecycle Management (ALM) [1], a widely-used lifecycle for software development and maintenance.

ALM is a widely-used lifecycle that merges several application development lifecycles. It covers from requirement specification to software maintenance phase. Generally speaking, there are four perspectives for implementing ALM: application portfolio management, software development lifecycle, service management/operation, and unified view.

The last view is the most general one since it merges the first three perspectives. The detail of unified view can be seen on Fig. 2.1. It starts with portfolio rationalization, a subset of application portfolio management, followed by software development lifecycle

and operations (operations are a subset of service management/operation). These phases will generate business value as their result. If change request occurs or new release is required, the whole processes can be restarted from portfolio rationalization.

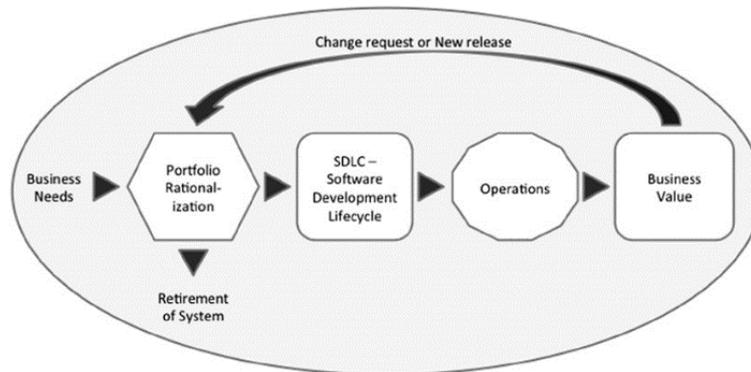


Fig. 2.1 Unified View[1]

III. ALM IMPLEMENTATION

Since the portal has been developed before, ALM implementation will be started from change request (described as an arrow from business value to portfolio rationalization on Fig. 2.1) to business value.

Portfolio rationalization is responsible to analyze what features are required on proposed system. Since the portal has been developed before, in our case, this phase is more focused on additional features instead of the whole features. Required features are collected through a survey involving 45 internship students and 10 lecturers (who act as either internship coordinator, the chair of research group, and reviewer). Each respondent is asked to provide feedbacks about existing internship registration portal. According to our survey, 22 additional features are required, starting from providing more-friendly user interface to simultaneous review mechanism (at that time, existing portal only allowed the reviewer to comment once and the student is only able to read the comment without providing any feedbacks). The detail of these features can be seen on Table 3.1.

Table 3.1 Additional Features

Feature Description
More-friendly user interface
Frequently Asked Questions
E-mail notification for every topic status change
Rich text area to fill given topic
A monitoring mechanism for internship coordinator and the chair of each research group
Advanced searching and sorting
Modifiable number of batch
More-convenient user interface
Information about important dates
Information about supervision schedule
Draft storing mechanism
Direct email mechanism to reviewer
Larger file upload limit
A mechanism to see topics given by lecturer
A mechanism to see all topics from lecturer role
More-detailed profile data
Modifiable topic similarity algorithm
Workload-balancing monitoring for the chair of each research group
Internship track record for students
Download feature for all roles
Supervision log
Simultaneous review between student and reviewer

Software development lifecycle is responsible to implements required features. It consists of six sub-phases: analysis, management decision, initial requirements, initial architecture, development, and delivery.

Analysis sub-phase is responsible to enlist required features. In our case, this sub-phase is merged with portfolio rationalization.

Management decision sub-phase is responsible to determine whether required features are possible to be implemented regarding analysis sub-phase. In our case, all required features are possible to be implemented, resulting the execution of the 3rd sub-phase, initial requirements.

Initial requirements sub-phase is responsible to further analyzing required features from semi-technical perspective. In our case, this sub-phase is conducted by merging several required features that have similar characteristic or target same data.

Initial architecture sub-phase is responsible to formalize required features to software engineering design. Since existing portal was developed in procedural manner, this sub-phase is conducted by generating data flow diagram, entity relationship diagram, and the prototype of new user interface.

Development sub-phase is responsible to convert software engineering design to application. It is conducted by writing the code regarding given design. In our case, the portal is developed as a web-based application using PHP and MySQL. Two sample views of resulted application can be seen on Fig. 3.1 and Fig. 3.2. On the one hand, Fig 3.1 is a figure about home view for internship coordinator. It links to numerous features, starting from assigning batch

schedule to monitoring all registered topics. On the other hand, Fig. 3.2 is a figure about topic registration by student. It is used to initiate internship registration process.

Delivery sub-phase is responsible to evaluate whether implemented features work well. In our case, this sub-phase is implemented by performing black-box, small-group user, and large-group user testing.

Black-box testing is conducted directly by the 1st author. She tries all features and checks whether each result is in-sync with feature's goal. According to this testing, all features work correctly in controlled environment.

Small-group user testing is conducted by inviting 7 participants. They will act as predefined roles on internship registration portal; 2 of them act as students, 2 of them act as reviewers, 1 of them acts as the chair of research group, 1 of them acts as internship coordinator, and 1 of them acts as administrator. All participants are gathered in one room and they are asked to simulate the whole internship registration process using newly-developed portal. If they experience some errors, they should write it down on a blank page, which will be collected later by the 1st author. According to this testing, some minor errors are defined. These errors are then solved before continuing to the next phase: service management / operations.

Large-group user testing is conducted by inviting 30 students to register their internship topic through the portal. After the registration process, they will be asked whether the features of given portal is feasible to be used in practice. According to this testing, 28 students responded positively while the other two responded in reverse.

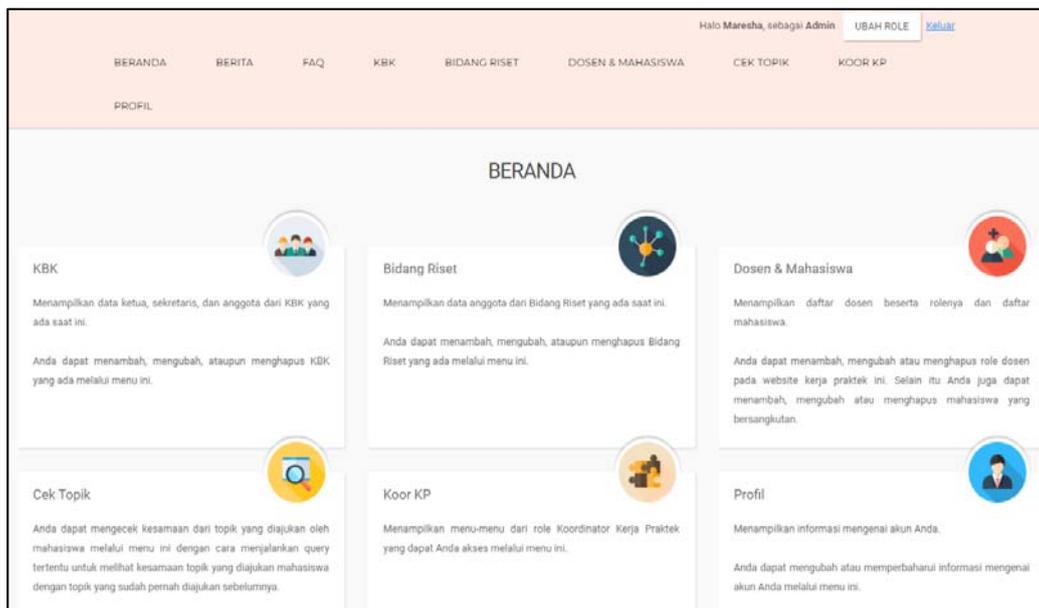


Fig. 3.1 A Home View for Internship Coordinator

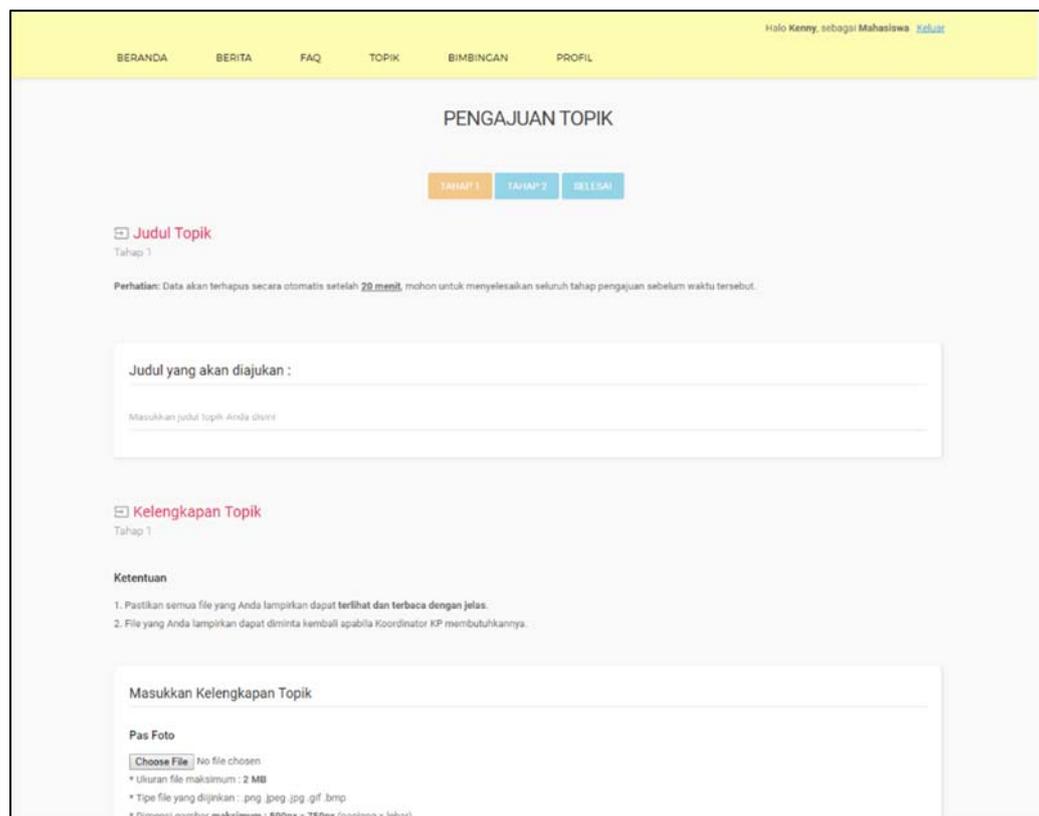


Fig. 3.2 Topic Registration Form View

Service management / operations phase is responsible to maintain developed system (i.e. portal in our case). This phase is planned to be conducted for 6 months, covering two batches of internship registration. Afterwards, the business value of newly-

developed portal will be measured. However, at the time of writing this article, these two phases have not been completed; service management / operations phase has just been started while business value can only be measured after service management /

operations phase is completed. Hence, there is no further information about those phases that can be given on this article.

IV. CONCLUSION AND FUTURE WORK

This paper presents a further development of internship registration portal in our faculty. As our guideline, an implementation of Application Lifecycle Management (ALM) has been developed. We would argue that such implementation fits our necessity; its phases describe the development and maintenance process perfectly. According to our evaluation, proposed portal works well and fits user need.

For further work, we plan to develop web services for given portal so that any applications can interact with our portal regardless of their containing platform. In addition, we also plan to develop a mobile application for our portal; according to our informal survey, the number of mobile users in our students are quite high.

REFERENCES

- [1] J. Rossberg and M. Olausson, "Introduction to Application Lifecycle Management," in *Pro Application Lifecycle Management with Visual Studio 2012*, Berkeley, CA: Apress, 2012, pp. 21–36.
- [2] H. Toba, E. A. Wijaya, M. C. Wijanto, and O. Karnalim, "Enhanced Unsupervised Person Name Disambiguation to Support Alumni Tracer Study," *Global Journal of Engineering Education*, vol. 19, no. 1, 2017.
- [3] M. Ayub and O. Karnalim, "Predicting outcomes in introductory programming using J48 classification," *World Transactions on Engineering and Technology Education (WTE&TE)*, vol. 15, no. 2, 2017.
- [4] T. M. Christian and M. Ayub, "Exploration of classification using NBTree for predicting students' performance," in *2014 International Conference on Data and Software Engineering (ICODSE)*, 2014, pp. 1–6.
- [5] O. Karnalim, "Detecting source code plagiarism on introductory programming course assignments using a bytecode approach," in *The 10th International Conference on Information & Communication Technology and Systems (ICTS)*, 2016, pp. 63–68.
- [6] O. Karnalim and A. Aldiansyah, "Python Source Code Plagiarism Attacks in Object-Oriented Environment," *Computer Engineering and Applications Journal*, vol. 6, no. 3, Oct. 2017.
- [7] L. Christiawan and O. Karnalim, "AP-ASD1: An Indonesian Desktop-based Educational Tool for Basic Data Structure Course," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 2, no. 1, Apr. 2016.
- [8] E. Elvina and O. Karnalim, "Complexitor: An Educational Tool for Learning Algorithm Time Complexity in Practical Manner," *ComTech: Computer, Mathematics and Engineering Applications*, vol. 8, no. 1, p. 21, Mar. 2017.
- [9] M. C. Johan, R. Tan, O. Karnalim, E. Imandha, and T. Cahyadi, "Analisis dan Perancangan Aplikasi Penyusunan Jadwal Mengajar Sesuai Jadwal Ketersediaan Mengajar Dosen Di Fakultas Teknologi Informasi (Studi Kasus: Jurusan Teknik Informatika)," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 1, no. 2, Aug. 2015.
- [10] R. Tan, D. S. Kartawihardja, and I. Christian, "Penerapan Teknologi RFID untuk Purwarupa Pencatatan Presensi Mahasiswa di

- Laboratorium Komputer,” *Journal of Information Systems Engineering and Business Intelligence*, vol. 3, no. 2, p. 122, Oct. 2017.
- [11] A. D. Widiatoro, “Information System of Graduation in Administration and Academic Bureau Soegijapranata Catholic University,” *SISFORMA*, vol. 2, no. 1, p. 13, Nov. 2015.
- [12] W. F. Senjaya, T. Witono, and N. Alkhala, “Perancangan dan Evaluasi Usability Aplikasi Pengelolaan Laboratorium Komputer,” *Journal of Information Systems Engineering and Business Intelligence*, vol. 3, no. 2, p. 101, Oct. 2017.