A Web-Based University Courses Syllabi Generator

Bassam Hussein[1] Samih Abdul-Nabi[2] Zaher Merhi[2] Amin Haj-Ali[2]
1.Department of Industrial Engineering, Lebanese International University, Beirut, Lebanon
2.Department of Computer and Communication Engineering, Lebanese International University, Beirut, Lebanon

Abstract
To improve university students learning experience and in the quest for ABET accreditation, it is crucial to have clear and consistent syllabi encompassing the course outcomes and their relationship to the overall program outcomes for all offered courses. This paper aims to present the automation of syllabi for engineering programs by introducing a web-based software application for course syllabus generation. The application has been developed using the best practices in educational theories and is fully aligned with ABET guidelines for program accreditation. It streamlines the process of writing syllabi and ensures compliance and conformity for all courses offered within a program. In addition, such automation reduces human errors, improves the student learning experience, reduces paper and printing costs and provides an environmental friendly alternative.

Keywords: ABET, engineering education, student learning, syllabi generation, quality improvement.

1. Introduction

Managing courses in a large scale, campus distributed, university with central management is a challenging task. Typically, in this model, several campuses belonging to the same university are located in relatively far locations from the central management (Deans/Chairpersons). The School of Engineering (SoE) at the Lebanese International University (LIU) offers different majors at the undergraduate and graduate levels in 9 different campuses located all across Lebanon [1]. To ensure consistency in material taught in classes and that each student, across all campuses, graduates with the same academic level, a common syllabi approach is deemed necessary. Another important consideration is that syllabi design should adhere to ABET guidelines. ABET is recognized as the worldwide leader in assuring quality and stimulating innovation in applied science, computing, engineering, and engineering technology education. It accredits college and university programs in the disciplines of applied science, computing, engineering, and engineering technology at the associate, bachelor, and master degree levels [2]. Over the past year, the SoE revised the syllabi of all courses offered at the school to ensure that the learning outcomes are clearly stated according to best educational practices in this regard. Understanding that the success of such process depends on the comprehensive implementation of a learning outcomes approach in higher education [3], the SoE put together a plan to align with ABET guidelines in an attempt to seek accreditation in the future.

Clearly, an online solution is best fit for this model since it removes the hassles of managing syllabi through email or via hard copies which is tedious, costly and environmentally harmful. However, designing a web-based syllabi generator that equips course coordinators with the tools necessary to build a well-structured ABET compliant syllabus is challenging. The design should be user-friendly for non-technical users and should provide adaptability for different types of courses. Furthermore, this application should provide a mapping between the program students outcomes (PSOs) and the course outcomes (COs). Probably one of the most challenging part in achieving an ABET compliant engineering program is ensuring that the program has a pyramid like structure when it comes to program outcomes; a structure that reflects the extent of compliance of the outcomes with well-known knowledge domains. Manually constructing the structure and conducting necessary computations would be an exhaustive task that requires time and effort. With the web-based generator the director of the program has an efficient way of creating the structure of the target program in an automated way. Moreover, the application will aid in identifying irregularities in the program structure.

The underlying strategy is to follow a process where the outcomes are simple and clear, achievable, and assessable. In order to achieve that, the well-known Bloom's Taxonomy of educational objectives was adopted. The list of direct unambiguous active verbs presented in Bloom's Taxonomy was strictly adhered to. In addition, the engineering programs were re-designed to comply with Bloom's six domains of knowledge arranged successively in a hierarchy as depicted in Figure 1:

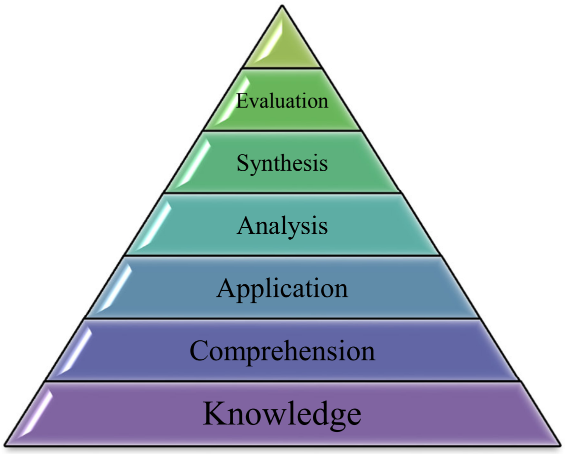


Figure 1. Knowledge domains hierarchy.

Bloom's Taxonomy is not simply a classification; it is an effort to arrange the various thinking processes in a hierarchy. In this hierarchy, each level depends on the student's ability to perform at the level or levels that are below it [4]. The focus here is on the cognitive domain which is quite suitable for disciplines like engineering. Furthermore, Bloom's research was centered on attitudes, feelings and values. This aligns well with ABET's guidelines and thus paves the way to successful accreditation once fully implemented. It must be noted that Bloom's Taxonomy also considers the affective domain. This domain is concerned with issues relating to the emotional component of learning and ranges from basic willingness to receive information to the integration of beliefs, ideas and attitudes [5]. This dimension has not been considered in building the tool as it is more suitable for other non-engineering domains.

This paper highlights the successful implementation a web-based online syllabi generator. The syllabi generator is currently deployed and in use at the SoE at LIU and has proven its ability to scale across many programs over multiple campuses.

2. Issues at stake

Many universities are still developing their syllabi in a traditional fashion where course coordinators use a template to put together each individual syllabus for each respective course. The coordinator would then distribute the syllabus to all instructors teaching the course and in their turn they distribute the syllabus to the students taking the course. This may be done through email or paper hard copies. There are major issues with this approach.

First, the development of the syllabus remains highly arbitrary due to the fact that each coordinator may approach various syllabus items differently. For example, when developing the course outcomes, the coordinator may adopt some vocabulary that is not fully aligned with best educational practices such as Blooms Taxonomy. This would lead to huge discrepancies between different courses syllabi within the same engineering program. Eventually, the program students outcomes (PSOs) would not fully map to individual course outcomes (COs) thus leading to non-conformance with ABET requirements. The COs are learning outcomes composed of statements of what a learner is expected to know, understand and/or be able to demonstrate after completion of a process of learning [6].

Second, since the syllabus template is usually in MS Word™ or other popular word processors format, it poses an issue whenever a new version is created due to a change or enhancement where it must be re-distributed to all course coordinators. "Version wars" become inevitable causing dysfunctional conflict between faculty members which may lead to unsatisfactory experience for instructors. In addition, data inconsistency may be another issue as manual entry and updates of data is an error prone process leading to potential exposure to additional risks.

Third, distributing the syllabus as a hard copy, 5 pages on average, to thousands of students is not environmental friendly and financially costly. While sending it by email becomes a tedious process with students registering or dropping courses within the first week of the course in addition to the fact that some students do not check their emails frequently and promptly or they may miss it if the email is marked as spam or send to the junk folder.

3. Proposed solution

At the top level, the system is based on a multi-tier web application architecture. The main components of that architectures being the web clients (with special mobile features), the workflow orchestrator, and the data vault (database and file system). The architecture is depicted in Figure 2 below.

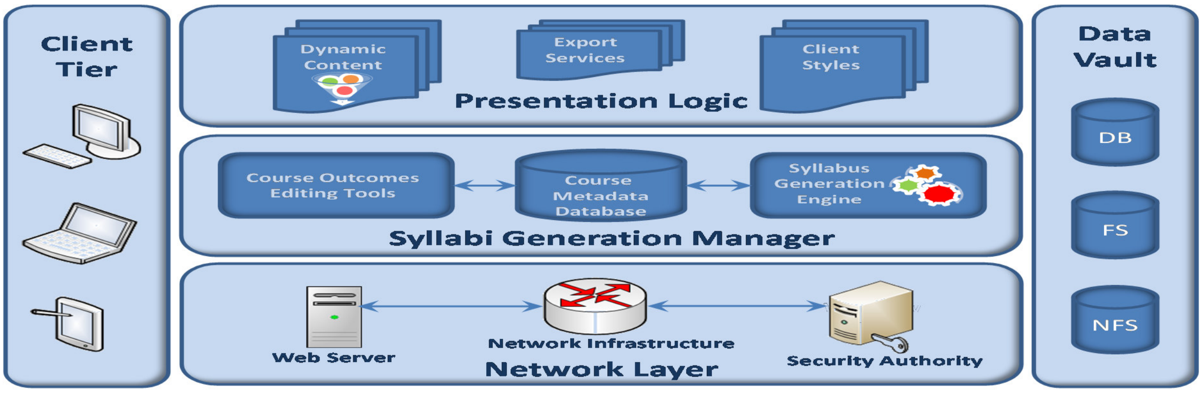


Figure 2: Architectural view.

The application allows multiple client types and platforms through browser-based and platform-based (Apple IOS™, Android™, Windows™ Mobile) applications. The browser-based access is made default for effective deployment. The application follows a layered architecture. The rationale behind selecting this type of architecture hinges mainly on a maintenance quality attribute. This quality attribute gives flexibility in replacement of platform/technology at any level in the hierarchy, and thus freeing the dependencies on a specific platform/technology.

At the base of the architecture lies the network layer. This module abstracts the detailed implementation of the web servicing and security authentication from the layer above. All components required for the proper integration of web server and security authority are lumped into the network infrastructure module. This layer provides services to the Syllabi Generation Manger (SGM) layer without exposing the intricacies of the implementation. Such services include: Authentication results, structured query results, etc…

The SGM layer is the main part of the system. Due to its pivotal role, this layer represents the core of the design. The major components of this layer are: The course metadata database, the syllabus generation engine, and the course outcome editing tools. This layer is designed with flexibility and scalability as major requirements. SGM employs data-driven mechanism to eliminate or otherwise minimize any hardcoding of the syllabi generation mechanism. This ensures more flexibility and scalability in the design. The course outcome editing tools allow the creation, editing, and deletion of course metadata. The metadata is saved in the Course Metadata Database in XML (eXtensible Markup Language) format.

The SGM layer hides all its detailed implementation from the presentation layer. The services provided to the presentation logic layer includes all processes needed to render process steps and/or data exports to clients. The last part of the architecture is the data vault. This is simply a conundrum of data services (database, local file system, network file system, etc…). These services are responsible for serializing any information to/from any process.

The architecture described above shows an important attribute; namely, layer-to-layer portability. By this attribute, the architecture is not implementation dependent on any technology/product. This value-added dimension allows institutions employing the ABET syllabus generator application tool to revert to any economical implementation that suits the reliability requirements. In general, portability is central to secure flexible solutions [7]. The basic servers required to implement the syllabus generator application will include:

- Linux: Acting as main OS and file server.
- Apache HTTPD: Web servicing component.
- Apache Tomcat: Application server.
- MySQL (or MariaDB): Database server.

3.1. Tool Description

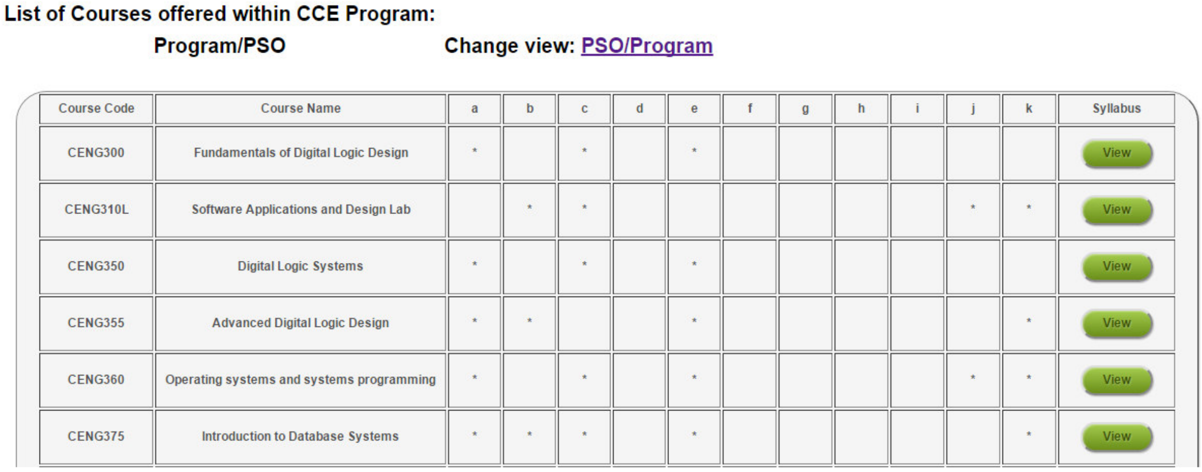
The application allows course coordinators to build their respective course syllabi online and the student to access a website to view syllabus and PSOs/COs mapping. The main purpose of the web application is to allow coordinators to build their syllabi online according to well-established standards and best practices. This application allows building a web-based system with the following features:

- Build the syllabi online by the administrator and the coordinator.
- Possibility to browse courses and view desired syllabus by any visitor online.
- Login as administrator or faculty member. A faculty member can view and print all engineering syllabi or filter syllabi by department.

3.2. The PSO's View

The program students outcomes (PSO's) view shows the contribution of each course to the ABET a-k PSO's.

Figure 3 illustrates the view of the various courses that are offered in one of the engineering programs; the Computer and Communications Engineering program in this case. Furthermore, each course has specific learning outcomes that are aligned with education best practices and they map to the overall program outcomes as required by ABET. These are the famous a-k program student outcomes (PSO's). Such a view will indicate if there are any gaps within the offered program so that appropriate action is taken accordingly to address the issues.



List of Courses offered within CCE Program:

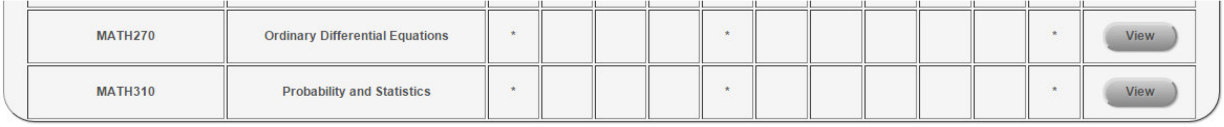
Program/PSO **Change view: PSO/Program**

Course Code	Course Name	a	b	c	d	e	f	g	h	i	j	k	Syllabus
CENG300	Fundamentals of Digital Logic Design	*		*		*							View
CENG310L	Software Applications and Design Lab		*	*							*	*	View
CENG350	Digital Logic Systems	*		*		*							View
CENG355	Advanced Digital Logic Design	*	*			*						*	View
CENG360	Operating systems and systems programming	*		*		*					*	*	View
CENG375	Introduction to Database Systems	*	*	*		*						*	View

Figure 3. Snapshot of Computer and Communications Engineering courses mapping.

3.3. The Pyramid View

At the end of the list, the application puts the measurable learning outcomes into perspective and shows the outcome pyramid model. Figure 4 is an example taken from the Industrial Engineering program at the Lebanese International University. It demonstrates the use of Bloom's verbs representing the various knowledge domains areas and the number of the verbs occurrences which are then represented as a building block of the different pyramid's layers.



MATH270	Ordinary Differential Equations	*				*						*	View
MATH310	Probability and Statistics	*				*						*	View

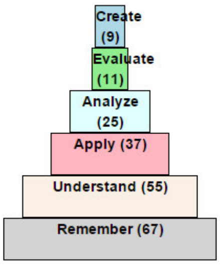
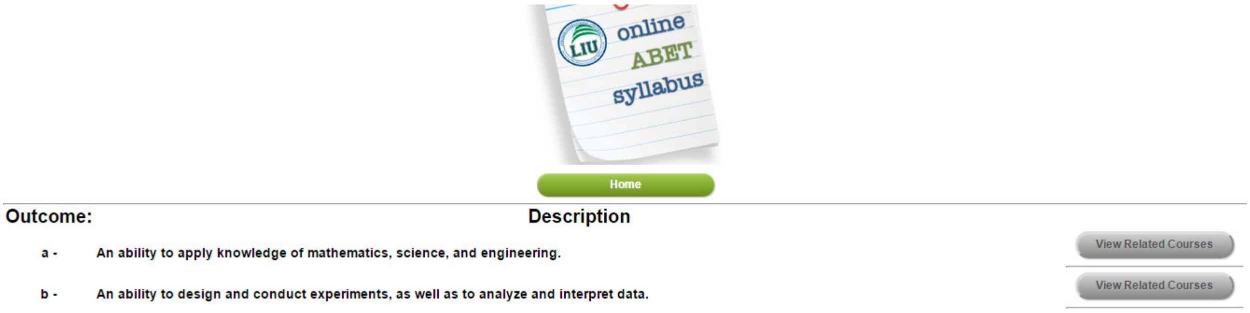


Figure 4. The pyramid view.

3.4. Other Features

The second view lists the ABET a-k skills and offers the possibility to view the program related courses from outcome as depicted in figure 5.



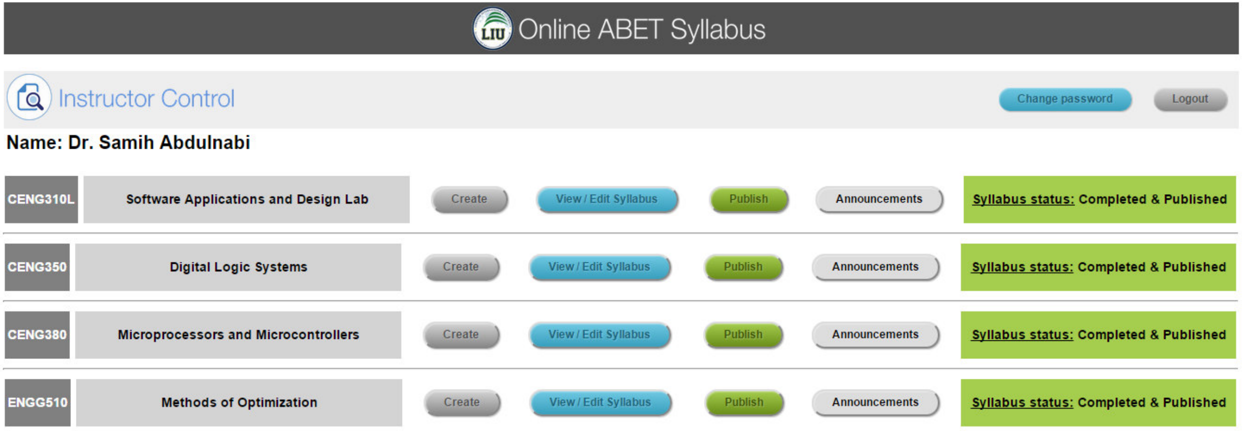
Outcome: **Description**

a - An ability to apply knowledge of mathematics, science, and engineering. View Related Courses

b - An ability to design and conduct experiments, as well as to analyze and interpret data. View Related Courses

Figure 5.View of program courses from outcomes.

Moreover, the application allows each coordinator to edit and publish the syllabi of courses being coordinated. This contributes to the strategy of using the outcomes based approach to align with ABET criteria which provides an opportunity to enhance understanding of teaching and learning. Developed by technical professionals from ABET's member societies, this criteria focus on what students experience and learn. [2]. Figure 6 shows the list of courses coordinator by a faculty member and the available options. The coordinator of a course has the possibility to create a new syllabus or to modify and publish existing one. It must be noted that when creating the syllabus, the tool will force the coordinator to use the active verbs only available in Bloom's Taxonomy thus avoiding any undesired deviation both on the CO's and PSO's levels.

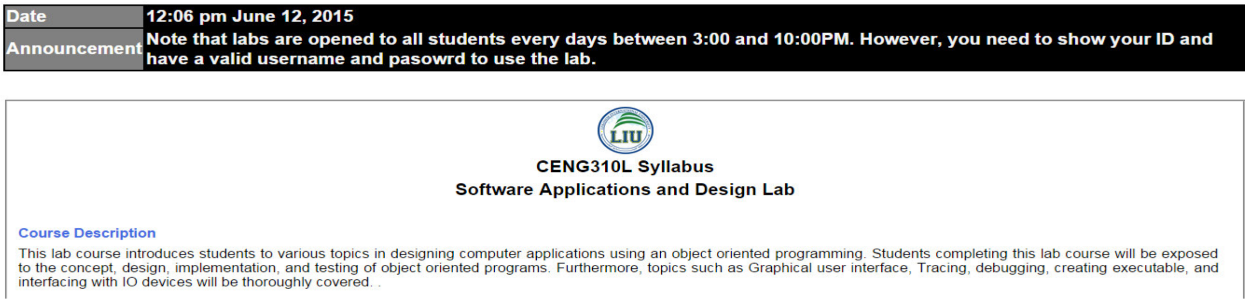


Name: Dr. Samih Abdulnabi

CENG310L	Software Applications and Design Lab	Create	View / Edit Syllabus	Publish	Announcements	Syllabus status: Completed & Published
CENG350	Digital Logic Systems	Create	View / Edit Syllabus	Publish	Announcements	Syllabus status: Completed & Published
CENG380	Microprocessors and Microcontrollers	Create	View / Edit Syllabus	Publish	Announcements	Syllabus status: Completed & Published
ENGG510	Methods of Optimization	Create	View / Edit Syllabus	Publish	Announcements	Syllabus status: Completed & Published

Figure 6. Creating and editing syllabi by the course coordinator.

In addition to writing the syllabus, the coordinator has the possibility to add a general note to the students as an announcement. Figure 7 captures an example of such an announcement.



Date	12:06 pm June 12, 2015
Announcement	Note that labs are opened to all students every days between 3:00 and 10:00PM. However, you need to show your ID and have a valid username and pasowrd to use the lab.

CENG310L Syllabus
Software Applications and Design Lab

Course Description

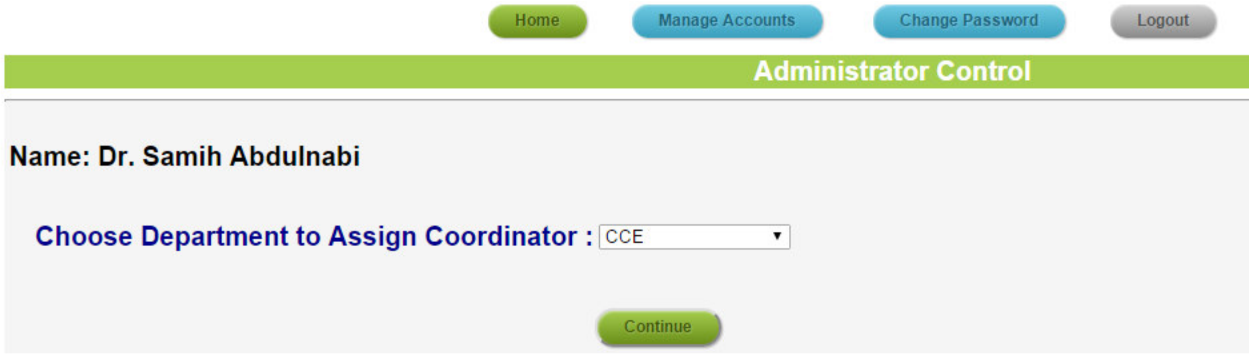
This lab course introduces students to various topics in designing computer applications using an object oriented programming. Students completing this lab course will be exposed to the concept, design, implementation, and testing of object oriented programs. Furthermore, topics such as Graphical user interface, Tracing, debugging, creating executable, and interfacing with IO devices will be thoroughly covered. .

Figure 7. Snapshot of an announcement.

When logged in as an administrator, the user has the possibility to perform one of the following functions:

1. Assign coordination.
2. Import courses (List of courses and their coordinators).
3. Import programs (List of program courses).
4. Manage accounts (Add/remove/update users).
5. Change or reset passwords or users.

Figure 8 illustrates the assign coordination function where the department is specified and the coordinator is selected and assigned to respective course(s).



Name: Dr. Samih Abdulnabi

Choose Department to Assign Coordinator : CCE

Figure 8. Assign coordination function.

4. Critical assessment

Despite its value-added benefits, the adoption of the implemented generator posed some challenges and risks that require careful assessment. Some issues were related to common difficulties associated with the deployment of any software artifact while others are directly related to the process which the application is automating.

The web-based ABET syllabi generator offers the means necessary to create, modify and distribute syllabi in an efficient way. However one of the major points still to be made lies in the complete integration of the generator with the already existing University Management System (UMS). Having multiple distinct standalone applications to learn and to use distracts ordinary users from the intended goal of the application.

While the application generates the syllabi in Adobe™ pdf format which is a popular format, some users require them in Microsoft Excel™, or Microsoft Word™ which is a limitation of the software application. Furthermore, in any engineering program, there are some courses that are offered by other schools within the university such as arts, sciences, education and humanities. Such courses are repeated across different engineering programs and insofar the application requires manual entry of such courses syllabi for each individual program. Automating this process will have big advantages in terms of limiting manual entries where they would be entered once and all programs would have access to them. Moreover, it will limit the amount of overhead in administrative work done by the faculty members.

The solution to the identified issues is not beyond realization. In fact, a simple, yet effective solution is currently being implemented. The new version will be integrated into the web-based UMS system to offer an embedded experience to both faculty and students.

5. Conclusion

University learning improvement is an on-going process that needs to be done in a continuous manner in order to secure sustainability. This is crucial to institute a process that will facilitate continuous improvement [8]. ICT (Information and Computer Technology) plays an important role in supporting such a process. This paper described the automation of course syllabus in the multi-campus Lebanese International University. The solution was an online software application which may be used at any university developing course syllabi according to best practices and in alignment with the ABET approach. Although the application is simple, it provides a good basis to develop the courses syllabi for any program across multi-campuses. Enhancement and evolution of the online syllabus application is possible and there are already plans to integrate it with the existing university management system which will provide an embedded system where all needed applications may be accessed in a consolidated manner.

References

[1] Lebanese International University Web site. Available as of June 21st, 2015 at: www.liu.edu.lb.

[2] ABET Web Site. Retrieved on June 21st, 2015 from: www.abet.org.

[3] Rauhvargers, A., Deane, C. and Pauwelset, W. *Bologna Process Stocktaking Report*, Leuven, Bologna Follow-up Group to the Ministerial Conference in Leuven/Louvain-la-Neuve, 2009. Retrieved on June 21st, 2015 from: http://www.ond.vlaanderen.be/hogeronderwijs/bologna/conference/documents/Stocktaking_report_2009_FINAL.pdf.

[4] Bloom, B. S., Masia, B. B. and Krathwohl, D. R. *Taxonomy of Educational Objectives Volume II: The Affective Domain*. New York, New York: McKay Publishing, 1964.

[5] Kennedy, D., Hyland, A. and Ryan, N. *Writing and using Learning Outcomes*, Bologna Handbook, Implementing Bologna in your Institution. Berlin, Germany: Raabe Verlag Publishing, 2006.

[6] Donnelly R. and Fitzmaurice M., *Designing Modules for Learning*, in: O'Neill G. et al., Emerging Issues in the Practice of University Learning and Teaching, Dublin, Ireland: AISHE Publishing, 2005.

[7] Pereira, R., Azambuja, M., Breitman, K. and Endler, M. *An Architecture for Distributed High Performance Video Processing in the Cloud*. Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing, pp. 482-489, July 2010.

[8] Hussein, B., Abdul-Nabi, S., Harb, A., Hage-Diab, A. & Haj-Ali, A., *Curricula Management and ABET Alignment at the Lebanese International University School of Engineering*, Journal of Education and Practice, 5, 22, pp. 163-1688, July 31st, 2014. ISSN (Online) 2222-288X.

About the authors

Bassam Hussein holds a B. Eng. in Computer Engineering, an MBA in Management and a PhD in Engineering Management. Dr. Hussein has worked for several international telecommunication corporations and educational institutions. He led multi-million dollar software development projects and rolled-out many products that are still widely in use by hundreds of millions of users across the globe. Dr. Hussein supervised many academic projects, participated in many conferences, symposiums and workshops and taught courses in the fields of software engineering, operating systems, engineering ethics, project management, human factors engineering, systems engineering and organizational behavior. He is currently the Chair of the Department of Industrial Engineering at the Lebanese International University (LIU).

Samih Abdul-Nabi received his B.E. degree in Electrical Engineering in 1990 from the Lebanese University, Beirut, Lebanon and his M.S. degree in Computer Sciences in 1993 from the University of Montreal, Canada. Since 1993, Mr Abdul-Nabi worked in many related engineering fields in both Canada and the U.S. His last industrial assignment was at Diebold Inc. as Senior Systems Engineer providing technical support to one of the largest banks in Canada. He received his Ph.D Degree in electronics and communications from the University of INSA, Rennes in 2015. Dr Abdul-Nabi is currently the chairperson of the computer and communications department at the Lebanese International University.

Zaher Merhi received his M.Sc. Degree in Computer Engineering in 2007, and his Ph.D. in Computer Engineering in 2010 from University of Louisiana at Lafayette, LA, USA. Since 2010 Dr. Merhi joined the Lebanese International University as an Assistant Professor in Computer Engineering. In 2011 he was assigned as the Assistant Dean for the school of Engineering- Tripoli Campus and in 2012 as associate chair to the computer and communications department.

Amin Haj-Ali is the Dean of the School of Engineering at the Lebanese international University. He received his B.E. degree in Computer and Communications Engineering in 1993 from the American University of Beirut, Beirut, Lebanon, his M.S. degree in Electronics and Computer Control Systems in 1994, and his Ph.D. in Electrical Engineering (Systems and Soft Computing) and Computer Science (Minor) in 2002 from Wayne State University, Detroit, USA. Dr. Haj-Ali earned a Masters in IT Project Management from George Washington University, Washington DC, USA. Since 1996, Dr. Haj-Ali worked in many related engineering fields in both Lebanese and international corporations. Before Joining LIU Dr. Haj-Ali was the chair of the Department of Software Engineering at ALHOSN University Abu Dhabi - UAE. Dr. Haj-Ali has a number of refereed journal and conference papers