

A Tool for ABET Accreditation

Abstract:

The accreditation process from ABET is anticipated with trepidation by academic institutions. The review process, however, is well defined and effective steps can be undertaken to succeed in accreditation. Much of the uncertainty can be minimized. Based on our experience and success we have developed a software package to automate the process. We will demonstrate a prototype at the conference.

Background:

The ABET review process, once understood, can lead to self-initiated steps to track, document, analyze, report, and develop strategies for improvement. Our tool will help streamline this for Criteria 3, perhaps the most demanding and important criteria of all the ABET criteria. Generally the Criteria 3 get satisfied across a large number of courses. ABET in their Criteria 3¹ indicate that “The program must have documented student outcomes that prepare graduates to attain the program educational objectives.” The criteria are typically met across a number of junior and senior level courses. We use S/U/- (satisfactory/unsatisfactory/Not Available) notation to identify whether a student met one or more of these criteria in a given course. An S roughly maps to grades of C or better, but is not always a given. Thus, each faculty member should manually record these letter grades for each of his/her course students in a given semester. Our engineering curriculum typically offers 12 courses at junior and senior levels every semester, and has typical enrollments of 18 per course. A typical student spends 6 or more semesters in our program to satisfy the graduation requirements. We graduate 20 to 25 students in the computer engineering program every year. This represents a large database of information. Further, it is a requisite to show not only that all the students who graduated met these criteria, but also that other students still in the program are making adequate progress. Still further, there is a need to show continuous improvement, that is, that the students are acquiring these skills earlier in their career and more consistently, perhaps in more diverse settings.

The present manual approach requires the ABET coordinator to collect all of this info from professors who taught courses during the past six years which should adequately cover all the graduating seniors in a given year. Our App will automate the data collection, analysis, and presentation steps. Further, it is our goal to provide timely feedback so as to ensure that additional courses are scheduled so that any graduating student would have automatically met the ABET Criteria 3. An additional goal is to analyze the results to identify major problems before they arise. As an example, a cohort of juniors may not be making progress towards meeting the ABET Criteria 3. To understand the implication of such a situation, one also needs to gain access to the individual student grades and academic history, presumably from the university’s student transcripts database (typically in the Registrar’s Office), so appropriate measures can be undertaken. The ultimate goal would be to ensure that we are meeting the ABET Criteria 3 in its entirety, thus ensuring that we are graduating our seniors with the best chance of success in the real world.

Approach:

We used the object oriented analysis and design approach in combination with techniques for auto code and GUI (graphical user interface) generation. The former involves the following,

chronologically and iteratively: (1) Identify the use cases; (2) Develop the user requirements; and (3) Design UML class diagrams. The latter uses EMF (Eclipse Modeling Framework) which takes the UML class diagram as the input and generates GUI-based code automatically². This auto code generation covers the communication and interfacing aspects, the areas that typically cause software breakdowns. This ensures that the code is robust, maintainable, and scalable. The behavioral part of the code is local in its impact and can now be easily and rapidly added.

For the tool to be useful, it needs to consider variability in the information technology (IT) infrastructure at different universities and account for the same. This poses certain engineering and user requirements that the software tool (or App) should meet. It should be flexible enough that any university could use it; it should also identify all the appropriate stakeholders and provide user friendly interfaces for all of them. To meet these criteria a web based approach was used for tool development. Using a web based solution the application leverages the Cloud to automatically provide the most recently updated version for all the stakeholders even if they are accessing concurrently. Since the Registrar's offices may use different database formats, we decided to allow their student and course databases to be imported in the meta-format of CSV (comma separated variables). The App thus accepts a .csv file that contains all the data for students, professors, and courses. Our App uploads this data and populates a MySQL database that is later used by all professors and coordinators involved in the accreditation process. For our demonstration, we standardized on a database that contains five tables; three data tables, and two relational tables. A diagram of the tables and their relationships is illustrated in Figure 1.1. With these tables, we can relate students to courses, professors to courses, and students to professors. The front end JavaEE Servlets were created for each of the different screens that a stakeholder will see. Servlets allow the application to dynamically build a page for each stakeholder (typically the professors, the ABET coordinator, and the graduation coordinator), and can update itself as new courses or students are added.

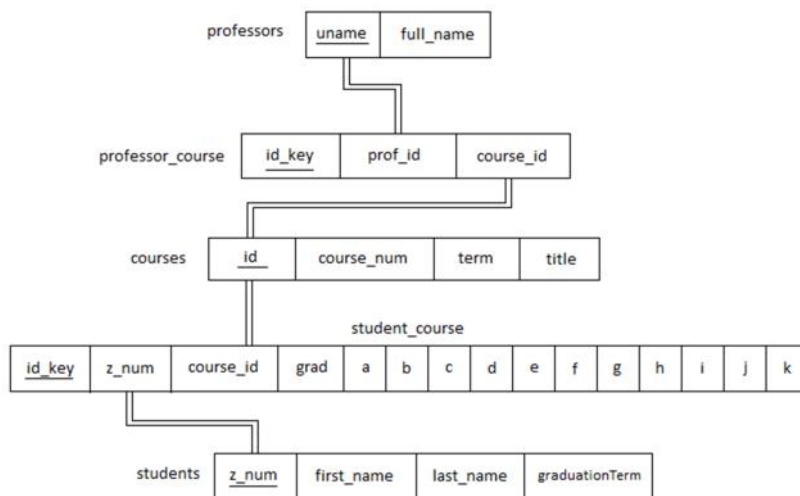


Figure 1.1 – Database model

All of the development of Java code was done with the open source Eclipse integrated development environment (IDE). Additionally, a plug-in called Eclipse Modeling Framework (EMF) was used to provide automatic code generation (EMF 2012). EMF allows the App developer to model the structure of the application and, with one click, to generate code for all the classes. Member functions are still implemented by the developer (or programmer) , but

much of the code is automatically generated, allowing the programmer to focus on more important areas.

Requirements:

Our prototype will have interfaces for the ABET Coordinator, individual faculty members and the Graduation Coordinator. The faculty member will enter their S/U grades on a list provided by the software that is specific to their course and semester of offering. It will automatically send reminders to those faculty members who are behind on their entries. The tool will automatically tabulate the results for all the students. Once the graduation coordinator identifies the graduating students in a given semester, the tool will generate a tabulated view of the criteria 3 grades for all the students, and summarize the results and action items.

Our Prototype Application (App):

Our current prototype combines the roles of the ABET coordinator and the graduation coordinator into one identified as “ABET Accreditation Manager.” We also identified individual faculty members as “professors.” There may be nuances to address later on, to distinguish various types of faculty members.

The App’s Details:

1. When a professor accesses the application they are prompted to login with their user name and password (provided by each university) (see Figure 1.2). When the professor submits this information the credentials are verified and if accepted the application calls the doPost() method of the professors home page. This method calls the database manager which in-turn queries the database to retrieve all the courses associated with that professor’s user name. From this screen the professor can select a course and to evaluate students (see Figure 1.3) for satisfying ABET criteria. Once the professor selects a course the doPost() method on the course selection class is executed. Once again the database manager will query the database and return a list of all the students that took that course with that professor (see Figure 1.4). On this course evaluation page the ABET criteria is listed and the professor can enter Satisfactory , Unsatisfactory , or Not Applicable (S/U/-)) grade for each of the criteria. After completing the entry process, the professor can navigate back home and select another course. When the professor is done evaluating he/she can logout, returning to the login screen.
2. The ABET coordinator accesses his/her home page in a similar fashion. From the login screen the coordinator enters the user name “coordinator” and the provided password (see Figure 1.5). The coordinator’s home page is where information from the Registrar’s Office can be uploaded (as a .csv file), and where the coordinator can generate progress reports for students (see Figure 1.6). The .csv file must be in a specific format for the data to be imported properly. The given format is as follows:
Student Number, Course Number, Course Title, Unique Course ID, Course Term,
Instructor Name, Student Last Name, Student First Name, Grade, Instructor ID, Student Graduation Term

An example file would look like this:

Z15044219,COP 3530003,Data Struct/Algorithm Analysis,11658,200801,Clovis
Tondo,Doe,Jon, A,tondo@fau.edu,201201

Additionally, reports can be printed for each student from the coordinator's home page. The coordinator can select a term and the group of interest (Graduating students or Non-Graduating 'ongoing' students) by selecting one of the radio buttons (see Figure 1.7). If 'ongoing' is selected, the selected term will return a list of students who took a course that term. When 'graduating' is selected a list of students who graduated that term is returned. For either list the coordinator can select a student to get that student's detailed report for by clicking on their name or their student ID (Z-Number). See Figure 1.8. This results in the display of a list of all the courses (to date) a student has taken along with grades for all the criteria (a-k) that the student has received (see Figure 1.9). With this report the coordinator can provide proof to ABET that the student individually indeed has met the specific requirements for the university's program to maintain its accreditation.

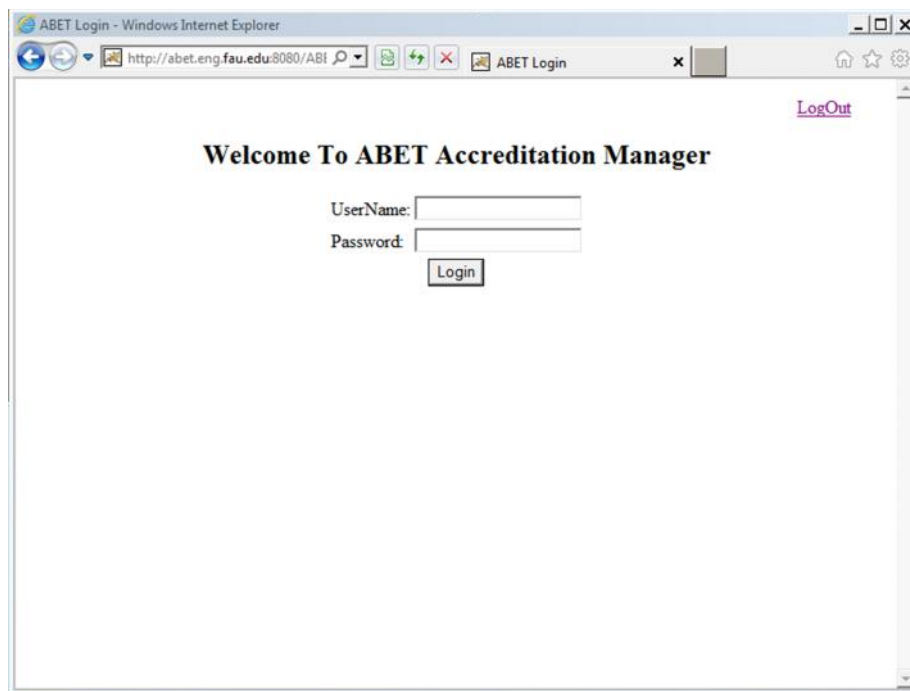


Figure 1.2 – The main login page

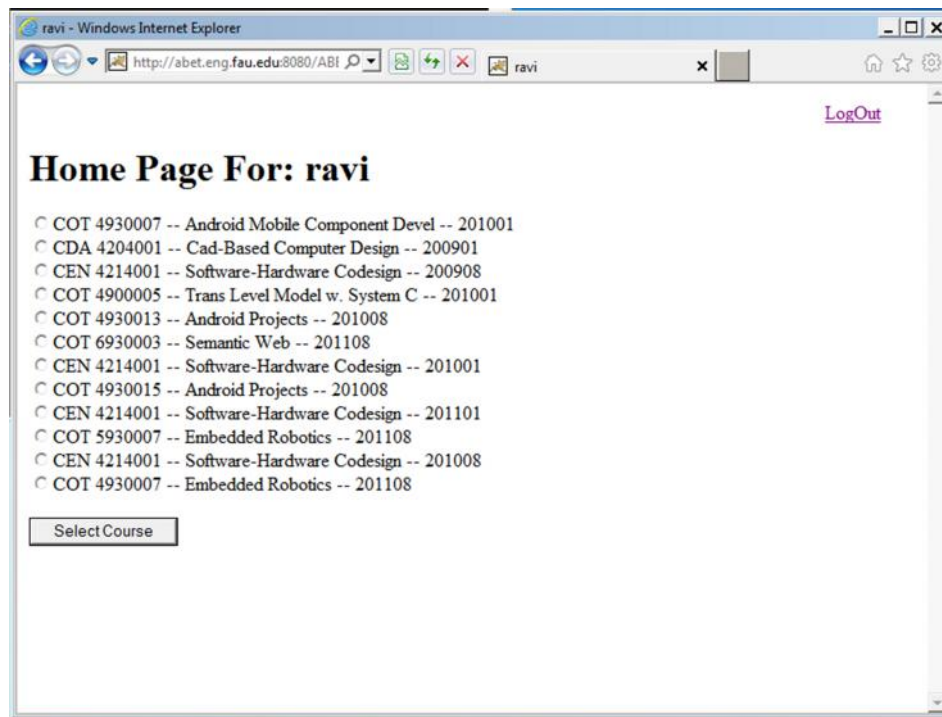


Figure 1.3 – Professor's Home Page



Figure 1.4 – Students' list for professor's evaluation

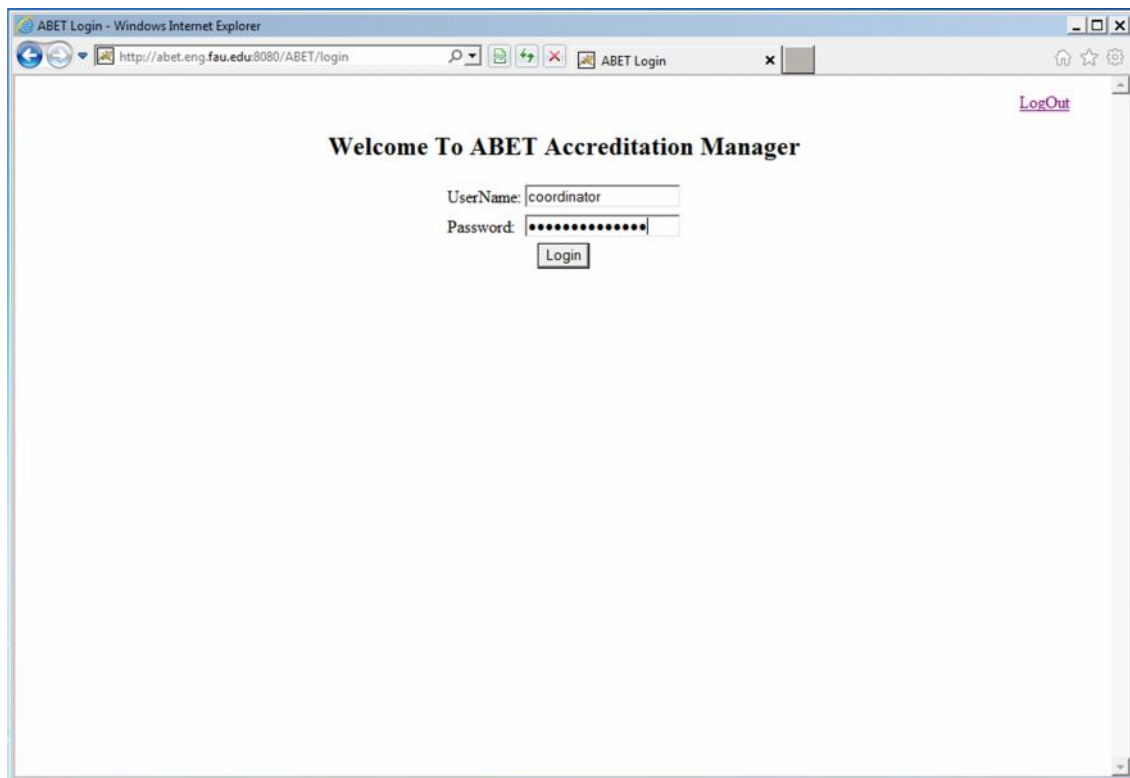


Figure 1.5 – Login for the ABET coordinator

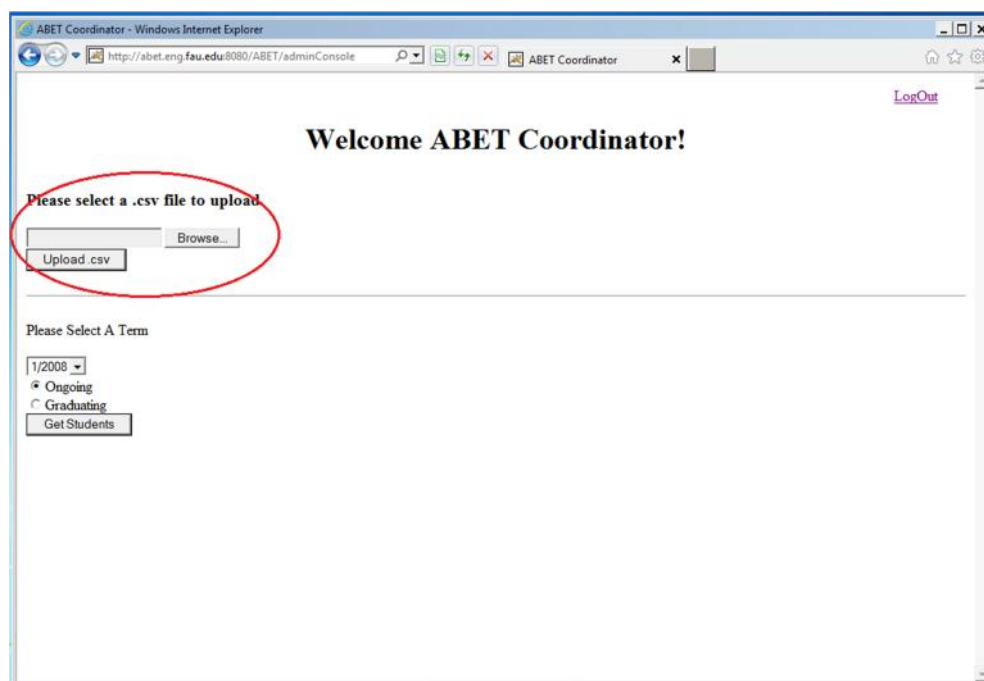


Figure 1.6 – Uploading a .csv file

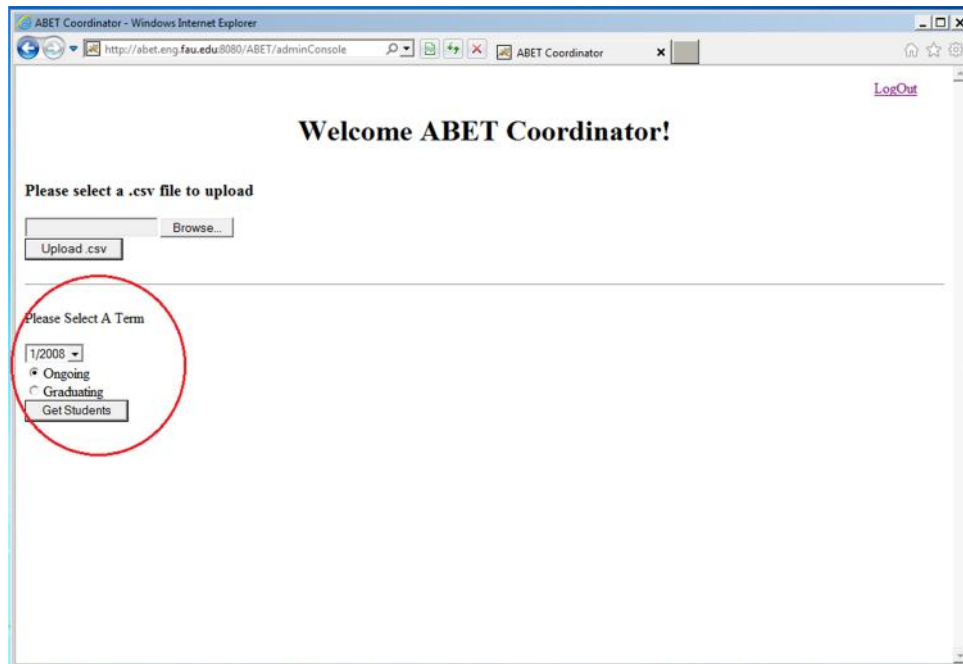


Figure 1.7 – Selecting a Term for a student record

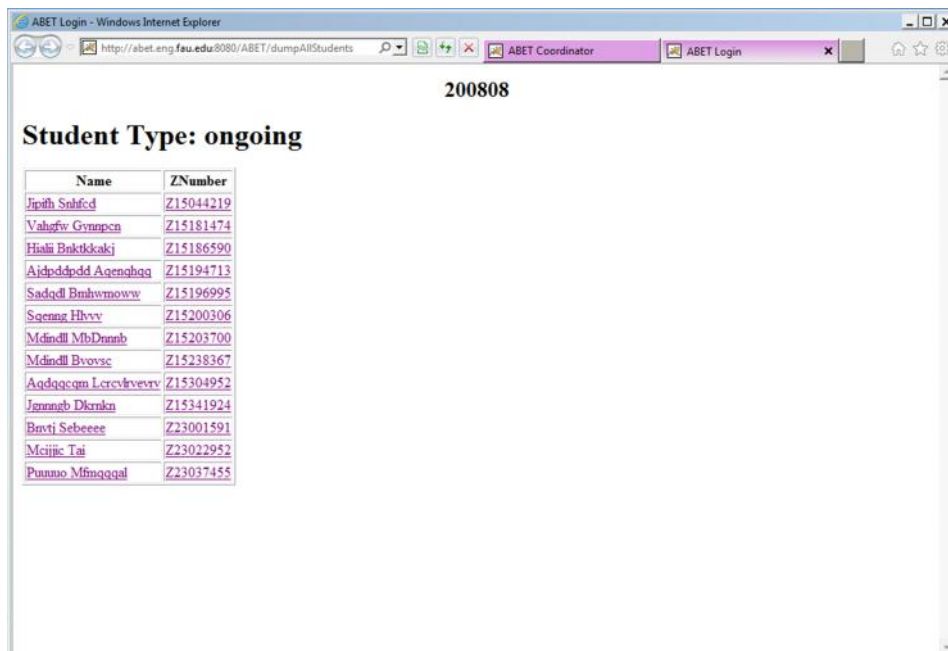


Figure 1.8 – List of students that can have a report generated for them

Mdindll Bvovsc

Course Number	Title	Term	Grade	A	B	C	D	E	F	G	H	I	J	K
COP 4610002	Computer Operating Systems	200801	A	S	S	S	S	S	S	S	S	S	S	S
STA 4821001	Stochastic Models for CS	200801	A	S	S	S	S	S	S	S	S	S	S	S
CEN 4010002	Prin Software Engineering	200801	A	S	S	S	S	S	S	S	S	S	S	S
COP 3540001	Intro to Database Structures	200808	A	S	S	S	S	S	S	S	S	S	S	S
COT 4930010	Game Programming	200808	A	S	S	S	S	S	S	S	S	S	S	S
CNT 4403001	Intro Data & Ntwrk Security	200808	A	S	S	S	S	S	S	S	S	S	S	S
COT 4935002	Senior Seminar	201008	A	S	S	S	S	S	S	S	S	S	S	S
COT 4930007	Blackberry Java Development	201008	A	S	S	S	S	S	S	S	S	S	S	S

Overall

A	B	C	D	E	F	G	H	I	J	K
S	S	S	S	S	S	S	S	S	S	S

Figure 1.9 – Detailed report on the criteria (a-k) for each course and the overall criteria (a-k) for a specific student.

We have completed the coding and will be able to demonstrate the App at the conference. We were able to create an application that takes .csv files as an input and breaks the info up into a database. With this data a professor can easily evaluate a student from any class they have taught from any semester. The coordinator can print reports that show that students have satisfied all the ABET criteria 3.

Discussion:

The ABET review process, once understood, can lead to self-initiated steps to track, document, analyze, report, and develop strategies for improvement. Our tool will help streamline this for Criteria 3, perhaps the most demanding and important criteria of all the ABET criteria. Generally the Criteria 3 get satisfied across a large number of courses. There are strategies one can implement across the program to strengthen the program and reduce the number of courses that are needed to satisfy Criteria 3. This will reduce the confusion and frustration in managing the review process. This will also improve the program in terms of predictability, continuous improvement, and potential job placement after graduation.

Conclusions:

We have developed a software tool that can help the engineering ABET faculty coordinators to manage the documentation for Criteria 3 in an easy and intuitive manner. The tool will be extended later to provide early warnings and to provide actionable results.

Bibliographic Information:

1. ABET General Criteria 3: Student Outcomes, Retrieved September 17, 2012, from <http://www.abet.org/engineering-criteria-2012-2013/>
2. Steinberg, D., Budinsky, F., Paternostro, M., and Merks, E., *Eclipse Modeling Framework (EMF)*. 2nd edition. Addison-Wesley: Upper Saddle River, NJ.