

University of Nevada  
Reno

**ACAT:  
ABET Course Assessment Tool**

A thesis submitted in partial fulfillment of the  
requirements for the degree of Master of Science  
in Computer Science

by

Eugene O. Essa

Dr. Frederick C. Harris, Jr., Thesis Co-Advisor

Dr. Sergiu Dascalu, Thesis Co-Advisor

May, 2010

We recommend that the thesis  
prepared under our supervision by

**EUGENE O. ESSA**

entitled

**ACAT: ABET Course Assessment Tool**

be accepted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE**

Dr. Frederick C. Harris, Jr., Co-Advisor

Dr. Sergiu Dascalu, Co-Advisor

Dr. Mircea Nicolescu, Committee Member

Dr. Sherif Elfass, Graduate School Representative

Marsha H. Read, Ph. D., Associate Dean, Graduate School

May, 2010

## **Abstract**

This thesis describes the ABET Course Assessment Tool (ACAT). This tool was designed and developed to assist faculty in producing course assessment reports for ABET accreditation. These reports demonstrate that students in the course are achieving the required outcomes. Documentation of this requirement is particularly burdensome. There is no standard method of generating these reports, and each institution handles it differently. This might involve manual collection of the data which is very time consuming. ACAT is a web based tool that allows users to input the data required for a course assessment and then produce a standardized report as a PDF document. The design and implementation of this process is covered, followed by a detailed look at the tool's operation. To validate the design and user interface, a usability study was conducted with Computer Science and Engineering faculty members. Results show that ACAT is a viable tool that streamlines the course assessment process and is an improvement over the existing manual process.

## Acknowledgements

I would like to thank my wife Kristie for her never ending support and encouragement. I couldn't have done it without you. I would also like to thank my parents Dr. Eva Essa and Dr. Ahmed Essa for their wonderful genes and always pushing me to learn and better myself. Many thanks go out to Andy Dittrich with whom I worked on the initial design and prototype of the system. Finally, I would like to thank my advisors Dr. Dascalu and Dr. Harris for all their help, support, and patience.

# Contents

<b>Abstract</b>	<b>i</b>
<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>4</b>
2.1 Assessment in Higher Education . . . . .	4
2.2 ABET Accreditation Process . . . . .	5
<b>3 Existing Tools in Support of Accreditation</b>	<b>8</b>
<b>4 ACAT Software Model</b>	<b>10</b>
4.1 Known Business Rules . . . . .	10
4.2 Requirements Specification . . . . .	12
4.3 Use Cases . . . . .	13
4.4 Design . . . . .	17
4.5 Implementation . . . . .	24
<b>5 ACAT Prototype Operation</b>	<b>27</b>
5.1 General Pages . . . . .	27
5.2 User Pages . . . . .	29
5.3 Administrator Pages . . . . .	34
<b>6 Usability Study</b>	<b>38</b>
6.1 Usability . . . . .	38
6.2 Usability Studies . . . . .	39
6.3 ACAT Usability Studies . . . . .	40
6.4 Results . . . . .	41
<b>7 Comparison With Similar Work</b>	<b>46</b>
<b>8 Conclusions and Future Work</b>	<b>48</b>

8.1	Conclusions . . . . .	48
8.2	Future Work . . . . .	49
	<b>Bibliography</b>	<b>52</b>
	<b>Appendices</b>	<b>54</b>
A	<b>Computer System Usability Study</b>	<b>55</b>
B	<b>Exempt Research Certification</b>	<b>60</b>
C	<b>Sample ACAT Assessment Report</b>	<b>62</b>

# List of Figures

4.1	Use Case Diagram . . . . .	18
4.2	Major Subsystems in ACAT . . . . .	19
4.3	ACAT System Activity Diagram . . . . .	20
4.4	Logon Activity Diagram . . . . .	21
4.5	Administrative Functions Activity Diagram . . . . .	22
4.6	Create New Assessment Activity Diagram . . . . .	22
4.7	Edit Assessment Activity Diagram . . . . .	23
4.8	Create Report Activity Diagram . . . . .	24
4.9	ACAT Database Design . . . . .	25
5.1	General Pages Web Map . . . . .	28
5.2	ACAT Home Page . . . . .	28
5.3	ACAT Login Page . . . . .	29
5.4	User Pages Web Map . . . . .	31
5.5	Assessment Summary Page . . . . .	32
5.6	Select Course Outcomes Page . . . . .	33
5.7	Administrator Pages Web Map . . . . .	36
5.8	Create New User Page . . . . .	36
5.9	Browser Side Validation Error Message . . . . .	37
5.10	Server Side Validation Error Message . . . . .	37
6.1	CSUQ Results . . . . .	43

# List of Tables

4.1	Example Student Scores Spreadsheet . . . . .	11
4.2	General Functional Requirements . . . . .	14
4.3	Assessment Report Functional Requirements . . . . .	14
4.4	Course Outcome Functional Requirements . . . . .	15
4.5	Instruments of Direct Assessment Functional Requirements . . . . .	15
4.6	Report Functional Requirements . . . . .	16
4.7	Non-Functional Requirements . . . . .	17
6.1	Usability Study Results . . . . .	42
6.2	CSUQ Categories and Corresponding Questions . . . . .	42
7.1	Comparison of ACAT to Related Tools . . . . .	47



# Chapter 1

## Introduction

Achieving an accredited status is a very desirable goal for an educational program. This indicates that the program meets certain standards set by the accrediting institution, which typically consists of members of the academic community, professionals, practitioners, and governing boards [11]. This accredited status makes the school a better choice for students, because it indicates that the students will gain the knowledge and skills necessary to be a productive member of their chosen profession [1].

Many professions have an associated accrediting institution. Engineering programs are typically accredited by ABET. ABET defines a set of criteria that an educational program must meet in order to be accredited. The process to achieve or maintain an accredited status is complicated and involves generating reports documenting that the program achieves these criteria. This is a difficult and time-consuming process, and is often perceived as an additional burden on the staff involved in preparing this documentation [14].

One especially burdensome part of the accreditation process is ABET criterion 3. This criterion defines a set of 11 course outcomes, (a) through (k), each of which identifies a specific ability or skill that a student must achieve upon graduation from an accredited program. Educational programs may also define their own course outcomes in addition to the ABET-defined course outcomes. ABET requires that an educational program have a system of assessment in place in order to periodically document the students achievement of these outcomes [1].

In order to perform this assessment, the faculty must first determine which course

outcomes are assessed for each course. Once the course outcomes are defined, then the degree to which the students achieve these outcomes can be directly measured based on student scores on assignments, quizzes, exams, and other instruments of direct assessment. In some cases, the student's scores on these assignments are used to determine if the students are achieving the outcome. In other cases, an assessment team reviews students work and gives a rating that indicates the students achievement of the course outcome.

There are also indirect assessment methods to measure students achievement of course outcomes. This may include student course evaluation surveys, alumni surveys, or job placement statistics [32]. A typical course assessment will contain a combination of direct and indirect methods [21]. Once all of the data is collected, it can be normalized and averaged for each course outcome, and the final result will indicate whether the student achieved that course outcome.

This process is very time consuming and tedious. It requires storing a large amount of data, including descriptions of the instruments of direct assessment, each student's score on each instrument of direct assessment, and other supplementary data such as survey results and documentation of course changes. Preparing the reports is also difficult, and requires analyzing the data, preparing tables and graphs, and including these in a consistently formatted document. The differences in teaching styles and organization of course data may also lead to inconsistent results from different faculty members within the same organization. There is no standard method used throughout academia for this assessment process.

This thesis discusses the ABET Course Assessment Tool, or ACAT. This tool was first suggested by Dr. Sergiu Dascalu in coordination with Dr. Mircea Nicolescu. The goal of this tool is to make the assessment process as simple and efficient as possible for the faculty involved, while requiring minimum changes to the existing teaching methods. It is primarily focused on collecting and organizing the results of the instruments of direct assessment and indirect assessment, analyzing this data, and generating standardized assessment reports that can be used as part of the accredi-

tation process. ACAT is a database driven, web based application that provides an intuitive method to enter assessment data for each course. This includes the course outcomes which will be assessed, the instruments of direct assessment that measure the achievement of the course outcomes, the students scores on those instruments of direct assessment, changes made to the course and the effect of those changes, student course evaluation survey results, as well as general course information. The tool uses this data to automatically generate standard assessment reports for each course. All of the data for each course is preserved in a database, and can be retrieved and modified at any time. ACAT also allows an educational program to define custom course outcomes, which may allow it to be used for more than just ABET criterion 3 outcomes.

This thesis in its remaining parts is organized as follows: Chapter 2 provides a background on accreditation in higher education and ABET accreditation in particular. A review of existing software tools in support of accreditation is provided in Chapter 3. Chapter 4 provides specific details on the specification, design, and implementation of the ACAT software, followed by the operation of the ACAT prototype in Chapter 5. A comprehensive usability study is detailed on Chapter 6, future work is provided in Chapter 7, and finally the conclusions of this thesis are described in Chapter 8.

# Chapter 2

## Background

This chapter examines accreditation in higher education, specifically ABET accreditation of engineering programs and computer science. The requirements for accreditation are presented with an overview of the accreditation process, focusing on the assessment process.

### 2.1 Assessment in Higher Education

Accreditation is an important part of the higher education system today. As a peer-review process, accreditation ensures a standard of quality is achieved and maintained. There are two types of accreditation, institutional which evaluates the overall quality of an institution, and specialized which certifies program of study.

The importance of attaining accreditation is multi-faceted. First, it gives an assurance to college students and their families that the college they choose has a quality program. This, in turn gives the college graduate assurance that they leave well prepared. Accreditation is also used by registration, certification, and licensure boards to screen applicants.

Accreditation in engineering started in the 1932 when the Engineers' Council for Professional Development (ECPD) was established. The original focus of this organization was on guidance, training, education, and recognition [1]. As the focus of ECPD shifted more toward accreditation, the name was changed to the Accreditation Board for Engineering and Technology (ABET).

With the emergence of the field of computer science in the 1970's and early 1980's the Institute of Electrical and Electronics Engineers Society (IEEE) and the Association for Computing Machinery (ACM) established the Computer Science Accreditation Board (CSAB). As the field of computer science grew, CSAB had a significant impact on the quality of these programs. In 2001 CSAB merged with ABET and, in 2005 ABET formally changed its name to ABET, Inc. [9].

In 1997, ABET changed its policies toward accreditation by adopting Engineering Criteria 2000 (EC2000). This shifted the focus of the accreditation process on what is learned rather than what is taught. This is done by focusing on identifying and documenting processes, and utilizing a range of assessment mechanisms to evaluate the program [23].

## 2.2 ABET Accreditation Process

The ABET accreditation process for engineering programs is both complicated and time consuming. ABET provides the criteria for engineering programs to achieve accreditation. There are eight general criteria which are applicable to all programs, and a ninth which is program specific [9].

The first criteria applies to students. The program must evaluate, advise, and monitor students, as well as have procedures in place for transfer students and credits. Additionally, the program must demonstrate that there are procedures in place to ensure students meet all of the programs requirements.

The second criteria focuses on the program's educational objectives. This includes detailed published objectives which are in line with the institution, an evaluation process for the objectives, and a curriculum and processes that ensure these objectives are attained.

The third criteria is a set of program outcomes that the degree program must demonstrate that graduates have, and are as follows:

- (a) an ability to apply knowledge of mathematics, science, and applied sciences

- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to formulate or design a system, process, or program to meet desired needs
- (d) an ability to function on multidisciplinary teams
- (e) an ability to identify and solve applied science problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of solutions in a global and societal context
- (i) a recognition of the need for and an ability to engage in life-long learning
- (j) a knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern scientific and technical tools necessary for professional practice.

The fourth criteria deals with continuous improvement. The program must use a documented process to assess educational objectives and program outcomes in order to evaluate the extent to which they are being met.

Criteria five through eight apply to curriculum, faculty, facilities, and support. Finally, the ninth criteria is program specific which is unique and applies to each given discipline .

The actual accreditation process takes around two years. The major portions of the process are the Self-Study Report and the site visit. The Self-Study Report is prepared by the program under review as a qualitative and quantitative assessment of the strengths and limitations of the program [10].

Programs must clearly demonstrate and document that they meet all of the criteria described above. The entire report, prepared by the program is lengthy and contains numerous sections. However, the documentation for the third criteria, demonstrating achievement of the course outcomes, is especially difficult and time consuming. The software tool, ACAT, which is described in this thesis will help standardize and streamline this portion of the accreditation process.

## Chapter 3

# Existing Tools in Support of Accreditation

This chapter looks at tools and software that have been developed to support the accreditation process. These tools range from database designs to modifications of existing course software to custom programs.

There have been many efforts to streamline and facilitate the assessment process. In DeLyser and Hamstad [12] the faculty was able to reduce the assessment work load significantly by eliminating redundancies in the assessment process. This was done by carefully selecting which instruments of direct assessment were included in the assessment process in order to prevent assessing the same student twice for the same course outcome.

Blandford and Hwang [6] suggest using sampling to reduce the overall workload. This can be done by using only a subset of the instruments of direct assessment, by using a subset of the students, or a combination of both. Yamayee, *et al.* [32] placed an emphasis on creating instruments of direct assessment that were focused on a particular course outcome and easy to evaluate. These improvements are helpful in reducing the amount of work overall, and were shown to be effective in streamlining the assessment process, but they do not eliminate the need to collect and analyze the assessment data, and to prepare assessment reports.

Some programs have adopted courseware to facilitate the assessment process. Booth [7] describes a database design that could be used to organize the data required



for an assessment report. This database mapped assignments to course outcomes, collected data for each assignment to measure achievement of the course outcomes. It also collected artifacts of student work, and documented changes to the course. This database is an effective method for organizing the information required for an assessment report, and allowing easy access to that information when preparing reports.

Booth, Preston, and Qu [8] developed a prototype system for mapping department outcomes to course objectives to ABET outcomes. By improving the previously developed database, described above, and applying an existing tool called WebSubmit [20] a new prototype was created to submit assignments and map them to course outcomes.

Abunawass, Lloyd, and Rudolph [2] describe how the University of West Georgia switched from WebCT to an open source course management software called Moodle, and were able to adapt this software to store student portfolios. The Computer Science Program Assessment (COMASS) project uses student portfolios as the basis for the assessment process. This was a major improvement over their existing assessment process, and helped to manage and store the vast amount of data required to document student's achievement of course outcomes. However, this requires a dramatic change in the organization of all course data, which may not be feasible at all institutions. Both of these cases demonstrate that a software solution can be effective in streamlining and automating the assessment process.

One final piece of software is a web based tool that maps subjects to Australian Generic Graduate Attributes [30]. In order to replace a manual process that focuses on an outcome based curriculum, this tool was developed to automate and streamline the process. Generic Graduate Attributes which are similar to ABET outcomes are mapped to course learning outcomes which in turn are mapped to unit (subject) learning outcomes. This approach not only maps courses to outcomes, but also allows users to view the attainment of outcomes for programs of study. This tool creates numerous reports via the graphical interface for curriculum designers, professors, and consultants.

# Chapter 4

## ACAT Software Model

This chapter describes the software design of ACAT. This tool was designed for the specific purpose of creating course assessments reports, which are a major part of the ABET accreditation process. The UML software model [3, 22] of ACAT is fully described. Known business rules for creating an ABET Assessment are examined first, followed by a detailed requirements specification. Next the system design is presented including database design and a web site map. Finally, the implementation of the design will be detailed.

### 4.1 Known Business Rules

ACAT will replace an existing manual method for creating assessment reports at the University of Nevada, Reno. In the future, it may replace similar methods at other educational institutions. In order to be effective, the tool must fulfill the same objectives as the previous methods. In order to better understand these objectives, an interview was conducted with Dr. Sergiu Dascalu and Dr. Mircea Nicolescu at the University of Nevada, Reno. The following guidelines, limitations, and feature requests are a result of that interview.

The input data for the tool includes exam questions, homework assignments, students scores and grades, and student self-assessment survey results. This data is highly sensitive, and must be restricted to only approved individuals. This requires that access to this tool and the reports generated using this tool be restricted as well.

This can be accomplished by requiring a user name and password to log in, and using encrypted network connections such as SSL when necessary. This also requires an administrator for this software who can be trusted to enforce policies and procedures to keep this sensitive data private.

One input to the tool is students scores on the instruments of direct assessment. Each educational program has different methods of tracking this information. This method will also vary from professor to professor within the institution. It would be ideal if the input data format could be standardized and stored in a file which could be uploaded into the tool. However, it is not feasible to force professors to change the way they keep track of scores just to accommodate this tool. It is also not feasible to require manual entry of each score for each student. This would not make the tool user friendly. It was suggested that the tool use a copy-and-paste method to input this data. A typical method for tracking student scores is to use an Excel spreadsheet. This spreadsheet contains a row for each instrument of direct assessment, and a column for each student in the class, or vice-versa. Each cell contains a students score. An example of this is shown in Table 4.1.

Not every student will be included in the final report. Students that do not finish the course should not be included in the final report. It may also be desirable to perform a random sampling of students for large classes. Removing some students from the report could be done manually before entering the data, but it would be more user friendly to add this feature to the tool. This allows the user to enter all of

Table 4.1: Example Student Scores Spreadsheet

	Assign 1	Assign 2	Midterm Exam				Assg 3	Class Partic,
			Q1	Q2	Q3	Q4		
Possible Points	15	20	5	5	10	10	15	10
Alice	15	19	3	5	10	9	12	9
Bob	12	18	4	5	7	9	14	9
Carl	10	20	4	4	8	8	13	10
Dean	13	14	3	5	9	9	10	7

the scores into the tool, and then selectively remove those students who should not be included.

The course outcomes that are measured will change. The accrediting institution may change the requirements from time to time, forcing the educational organizations to change their reports. The educational organization may add their own custom course outcomes in order to measure outcomes not covered by the accrediting institution. In order to accommodate these changes, the tool should allow the user to change the course outcomes. The previous course outcomes should be preserved to allow reviewing reports from the past. This can be accomplished by defining a set of course outcomes, and selecting which set of course outcomes should be used for a particular assessment report. For example, an educational program could use one set of course outcomes for classes in the Fall semester of 2008, and another set of course outcomes in the Spring semester of 2009. This would require two sets of course outcomes, and a user would choose which of these two sets of course outcomes would be used for their assessment report.

The course outcomes provided by the accrediting institution have a default description. These default descriptions are a general definition of the course outcome and are meant to cover a wide variety of courses. A user may need to customize this description for a particular course. For example, ABET's description for course outcome (g) is an ability to communicate effectively. This may be custom tailored for a software engineering course by restating it as an ability to communicate the scope, specification, and status of a software project effectively through scope and vision documents, software requirements specification documents, and project status reports. To accommodate this, the tool should allow the user to create a customized description for each of the course outcomes for a specific course.

## **4.2 Requirements Specification**

This section provides details on the requirements for ACAT. Both functional and non-functional requirements are presented. Functional requirements are defined as services

the system should perform. These are categorized by specific functionality of ACAT. Non-functional requirements are constraints which are outside of the functionality that are placed on the system [22].

The requirements for ACAT are listed with the following naming conventions. These relate to the applicable sections of the system.

- GRXX These are general requirements that define how the user interacts with the system.
- ARXX These requirements define how a user will create and modify assessment reports and the general information used to create them.
- CORXX These requirements define how the course outcomes can be modified by users and administrators in ACAT. This also defines how a user can select and customize the course outcomes for each assessment report.
- IRXX These requirements define how the user specifies the instruments of direct assessment for each course outcome. These also define how the scores are entered for the instruments of direct assessment.
- RRXX These requirements define how the user creates reports and how the contents of the report should be formatted.
- NFRXX These are the non-functional requirements for the system.

Tables 4.2 through 4.7 list the requirements.

### 4.3 Use Cases

This section describes the use cases for ACAT. A use case depicts the interaction a user has with the system. The system is shown in abstract form as a bounded box in the diagram. Users are represented by external actors outside of the system [22]. There are two actors in the ACAT use case diagram, a user and an administrator

Table 4.2: General Functional Requirements

GR1	ACAT shall require the user or administrator to log in to use the system.
GR2	ACAT shall not allow a user to view or modify another users assessment.
GR3	ACAT shall allow a user to change their password.
GR4	ACAT shall allow a user to change their personal information.
GR5	ACAT shall allow an administrator to add a user.
GR6	ACAT shall allow an administrator to remove a user.
GR7	ACAT shall allow an administrator to reset a users password.
GR8	ACAT shall not allow an administrator to view a users password.
GR9	ACAT shall allow an administrator to change a users personal information.
GR10	ACAT shall allow a user or administrator to log out of the system.

Table 4.3: Assessment Report Functional Requirements

AR1	ACAT shall allow a user to create a new assessment report.
AR2	ACAT shall allow a user to modify an existing assessment report.
AR3	ACAT shall create only one assessment report for each unique combination of user, course, and semester.
AR4	ACAT shall allow a user to indicate whether they have taught the course previously.
AR5	ACAT shall allow a user to enter course prerequisites.
AR6	ACAT shall allow a user to indicate whether or not the students were prepared by the course prerequisites.
AR7	ACAT shall allow a user to enter past changes that were made since the last time the course was offered and to enter the effect of those changes.
AR8	ACAT shall allow a user to enter future changes that will be made next time the course is offered, and to enter the purpose of those changes.
AR9	ACAT shall support one or more past changes and one or more future changes for each assessment.
AR10	ACAT shall allow a user to enter comments from the students for each assessment.
AR11	ACAT shall support one or more comments for each assessment.
AR12	ACAT shall allow a user to specify the number of students for each discipline that are enrolled in the course being assessed.

Table 4.4: Course Outcome Functional Requirements

COR1	ACAT shall allow an administrator to add course outcomes.
COR2	ACAT shall allow an administrator to remove course outcomes.
COR3	ACAT shall allow the administrator to define a set of course outcomes to be used in assessment reports.
COR4	ACAT shall allow a user to select which set of course outcomes should be used for an assessment.
COR5	ACAT shall allow a user to select which course outcomes apply to the course being assessed.
COR6	ACAT shall allow a user to enter a description for each course outcome that is tailored for the particular course being assessed.
COR7	ACAT shall allow a user to enter student self assessment results for each course outcome.

Table 4.5: Instruments of Direct Assessment Functional Requirements

IR1	ACAT shall allow a user to define one or more instruments of direct assessment for each course outcome selected for an assessment.
IR2	ACAT shall allow a user to enter the possible points for each instrument of direct assessment.
IR3	ACAT shall allow a user to enter a description for each instrument of direct assessment.
IR4	ACAT shall allow a user to enter the individual scores for each student for each instrument of direct assessment.
IR5	ACAT shall allow a user to copy and paste scores from a spreadsheet into ACAT.
IR6	ACAT shall allow a user to edit previously entered scores.
IR7	ACAT shall allow a user to remove a student and all of the students scores from the assessment report if that student did not complete the course.

Table 4.6: Report Functional Requirements

RR1	ACAT shall allow a user to generate an assessment report once all necessary data has been entered.
RR2	The report shall contain the instructors name, course name, and semester.
RR3	The report shall contain the course prerequisites and indicate if the students were prepared for the course by the prerequisites.
RR4	The report shall contain a section showing the past changes and their effects and the future changes and their purpose.
RR5	The report shall contain the students comments.
RR6	The report shall contain a summary table.
RR7	The summary table shall contain a row for each course outcome. Each row shall contain the student self assessment score, description, list of instruments of direct assessment for the course outcome, and average instrument of direct assessment score for that course outcome.
RR8	The report shall contain a score table.
RR9	The score table shall contain a column for each instrument of direct assessment, and each column shall contain the students scores for that instrument of direct assessment.
RR10	The columns in the score table shall be grouped by course outcome.
RR11	The report shall display all scores on a scale of 1.0 to 5.0, inclusive.
RR12	The score table shall display an average score for each student for each course outcome. This average shall be calculated as the average of the instruments of direct assessment for that course outcome.
RR13	The score table shall display the average score for all students for each course outcome.
RR14	The report shall contain a distribution of scoring table.
RR15	The distribution of scoring table shall display the number of scores for each course outcome that fall in the following ranges: [1.0-1.5], (1.5-2.5], (2.5-3.5], (3.5-4.5], and (4.5-5.0].
RR16	The report shall contain a distribution of scoring graph, which is a bar chart that graphically represents the distribution of scoring table.
RR17	The report shall contain a separate summary table, score table, distribution of scoring table, and distribution of scoring graph for each discipline.
RR18	ACAT shall allow a user to save the report to a file.
RR19	ACAT shall allow a user to print the report.



Table 4.7: Non-Functional Requirements

NFR1	ACAT shall support encrypted network connections.
NFR2	ACAT shall be hosted on an Apache web server.
NFR3	ACAT shall use a MySQL database to store information.
NFR4	ACAT shall be written in PHP.
NFR5	ACAT shall save all data entered by users in a database.
NFR6	ACAT shall allow multiple users to access the system simultaneously.
NFR7	ACAT shall not require that the user install special software to use the system.
NFR8	ACAT shall support commonly used web browsers.

as shown in Figure 4.1. It should be noted that the administrator is also a user. The users have 14 use cases and the administrators have 4. This represents the functionality of ACAT.

## 4.4 Design

This section describes the detailed design of the ACAT system. An overview is provided by explaining the major subsystems in ACAT. Next the functionality of the system is described through activity diagrams. These diagrams model the system behavior [13]. Finally, the database design is shown.

ACAT is a web based system. HTML is used to display pages to the user. This HTML is dynamically generated using server side scripting. The entire system can be viewed in layers, as shown in Figure 4.2. At the top level is the *presentation layer*. This contains the GUI subsystem, which uses HTML to displays data to the end user.

Below this is the *business logic layer*. This contains the general pages subsystem, the user pages subsystem, and the administrative pages subsystem. The general pages subsystem implements functionality for logging in, logging out, displaying the home page, and other basic functionality. The user pages subsystem implements all functionality for entering assessment data and generating reports. The administrative pages subsystem implements all functionality for adding and modifying users, entering

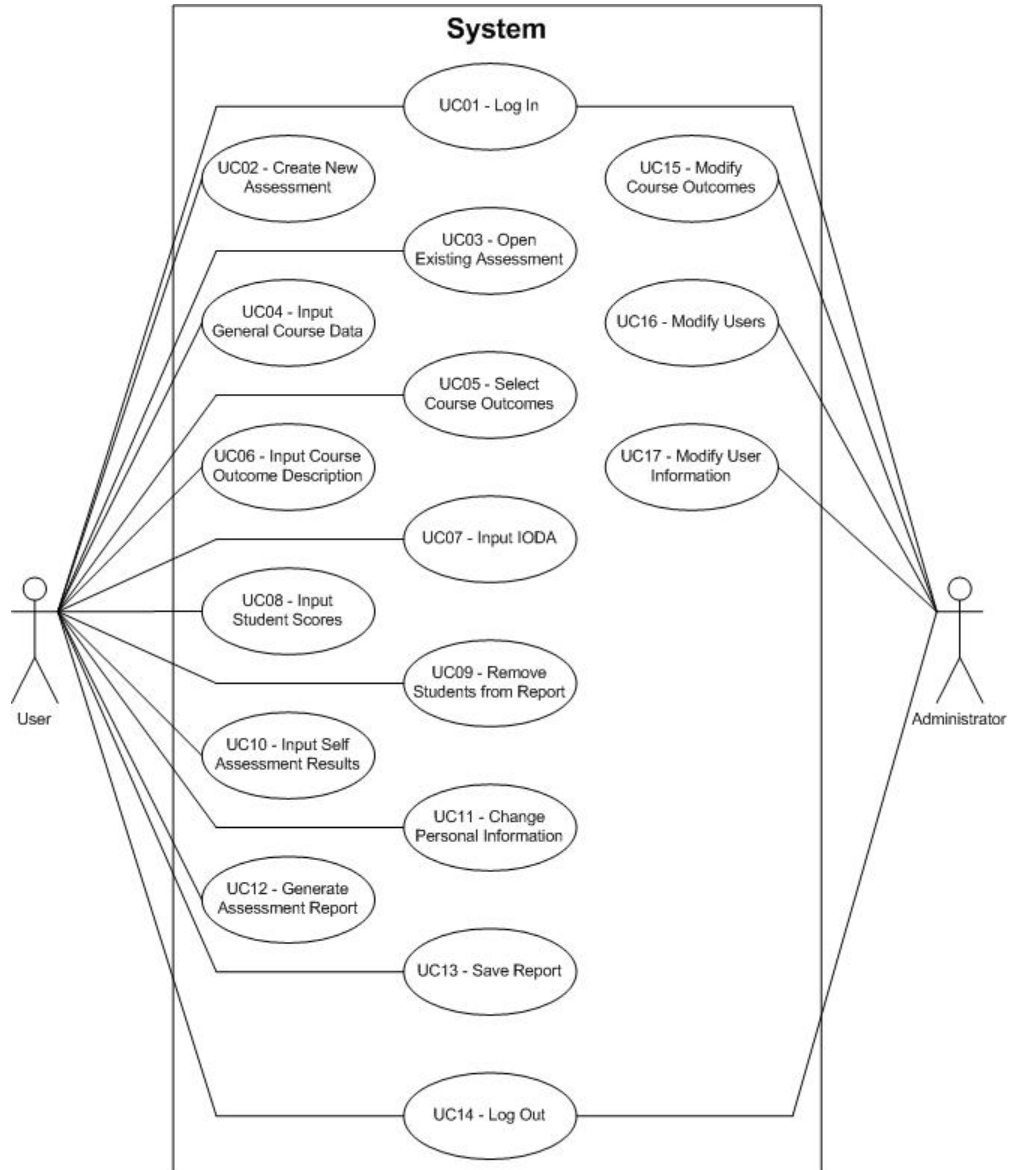


Figure 4.1: Use Case Diagram

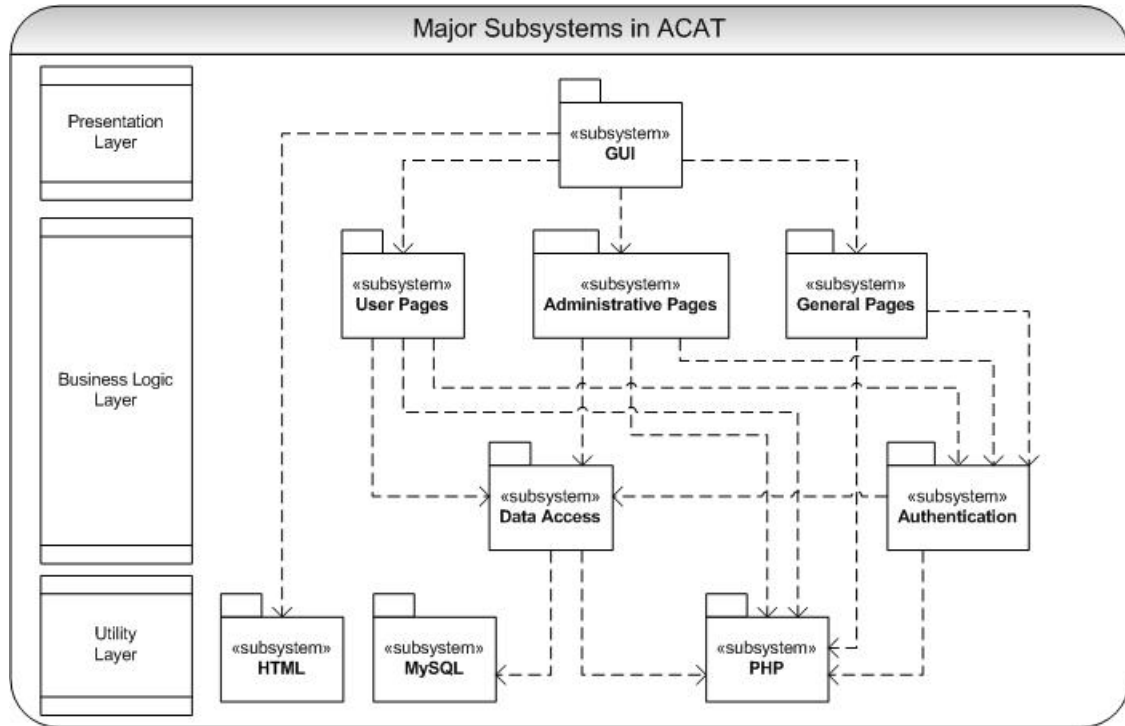


Figure 4.2: Major Subsystems in ACAT

course outcomes, and other administrative functions. Also in the *business logic layer* are the data access subsystem and the authentication subsystem. The data access subsystem implements an interface to the database to allow creating, modifying, and deleting data. The authentication subsystem controls access to different pages within the website based on the users credentials.

At the bottom is the *utility layer*, which includes the HTML subsystem, the PHP subsystem, and the MySQL subsystem. These represent the underlying structure of the system and will be covered in the following section.

The ACAT system is designed to be an intuitive and easy to use data entry and report generation system. The main flow of activity for the entire system is shown in Figure 4.3. The users begin by logging into the system. As in the case of most web sites, the activities can be varied and repeated numerous times until the user logs out. Though not depicted in the diagram, an inactivity timeout may also be used to log the user out after a period of being idle. This is done for security reasons.

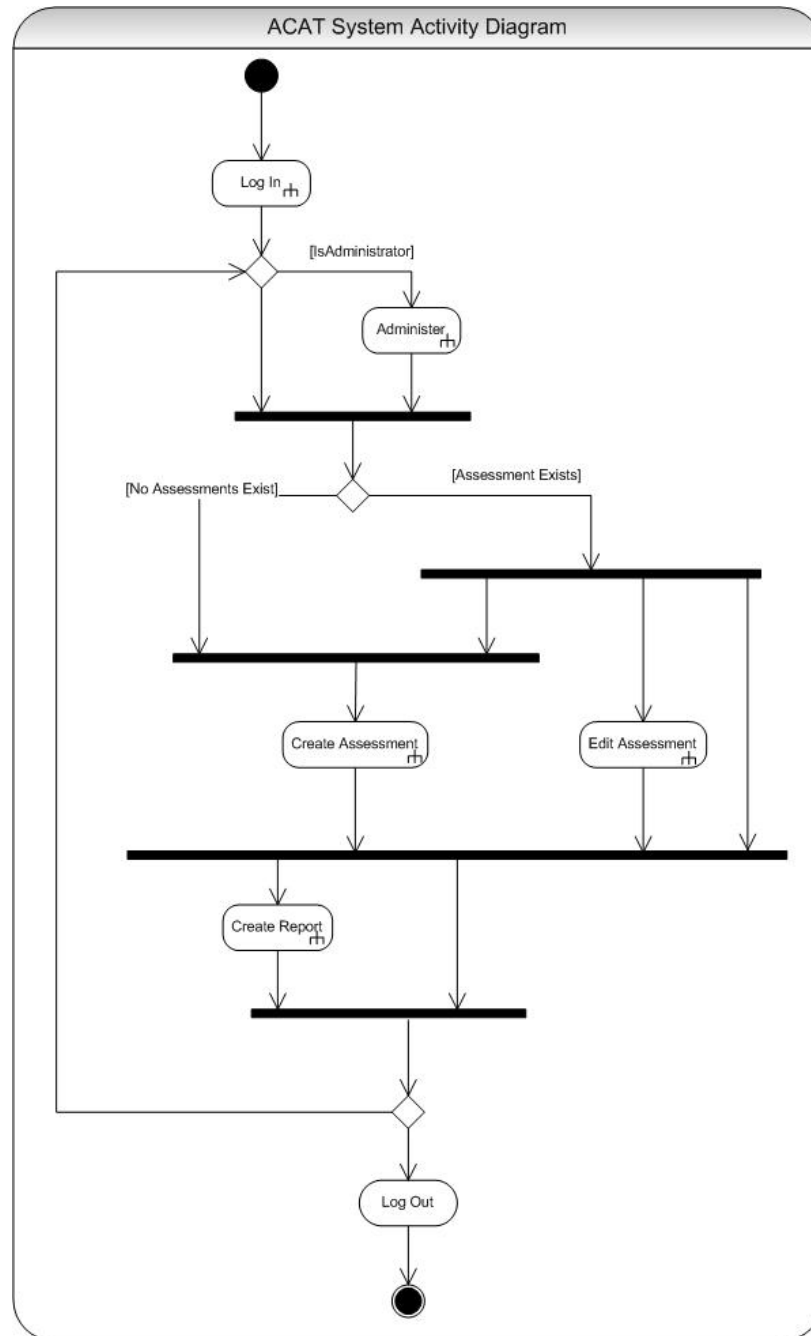


Figure 4.3: ACAT System Activity Diagram

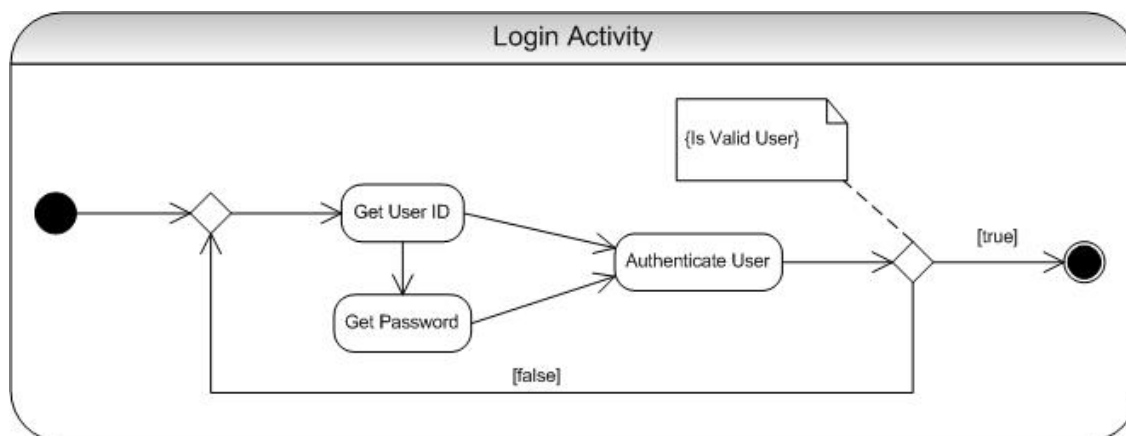


Figure 4.4: Logon Activity Diagram

Logging in to the system is accomplished through a web form. The login activity is shown in Figure 4.4. The user ID and password are entered into the form and the form is submitted. The ID and password are authenticated against the database and the user is either authenticated or returned to the login page with an error message. The authentication also determines if a user is an administrator and grants access to the administrative functions.

The activity diagram for administrative functions is shown in Figure 4.5. These functions allow for routine maintenance of users and the standard course outcomes. User maintenance such as adding new users and editing existing users can be easily accomplished. This functionality is useful when users need a password reset. The functionality for editing course outcomes gives the system added flexibility if these are changed by ABET or if additional outcomes need to be added to the system.

To create a new assessment, the users input the basic required information. This is a simple activity, shown in Figure 4.6. This activity collects the course title, semester and year it was taught, whether the course was taught previously, and any prerequisites. Additionally, whether students were prepared and useful comments are entered. Once this activity is complete, the rest of the data for the assessment is entered through the edit assessment activity.

The main functionality of the ACAT system is to add assessment data and edit

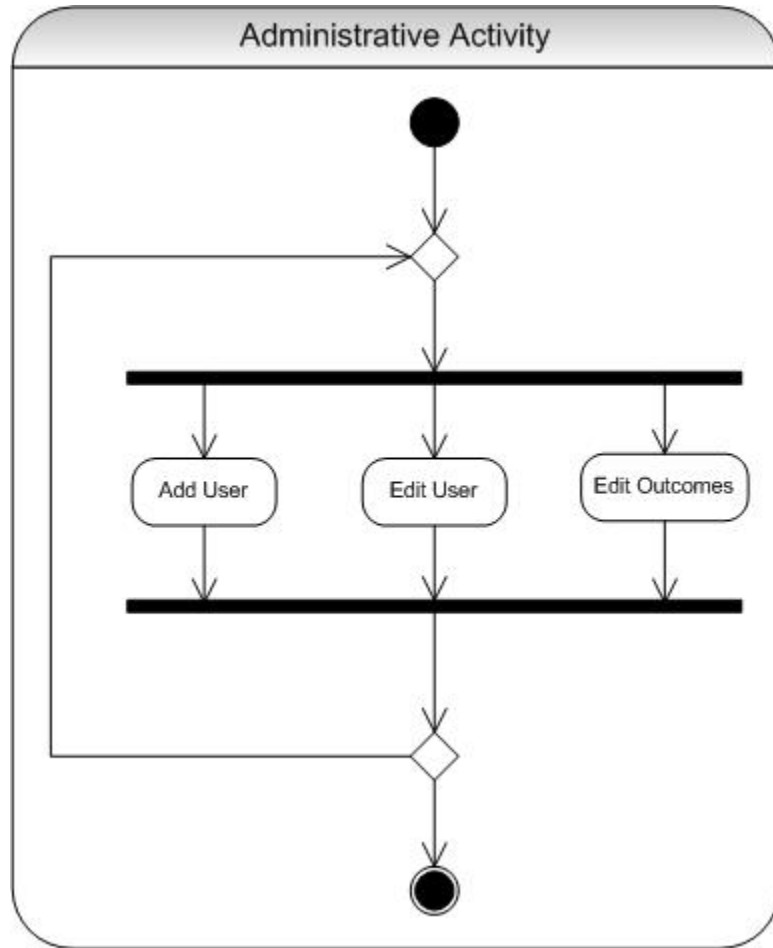


Figure 4.5: Administrative Functions Activity Diagram

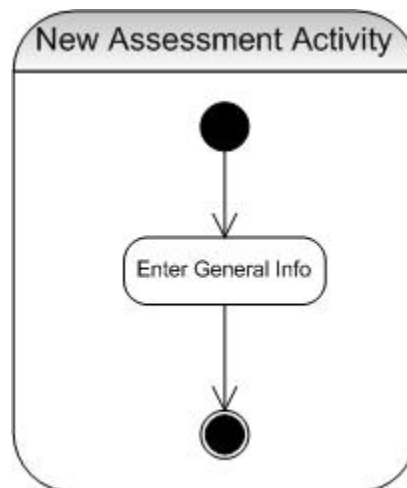


Figure 4.6: Create New Assessment Activity Diagram

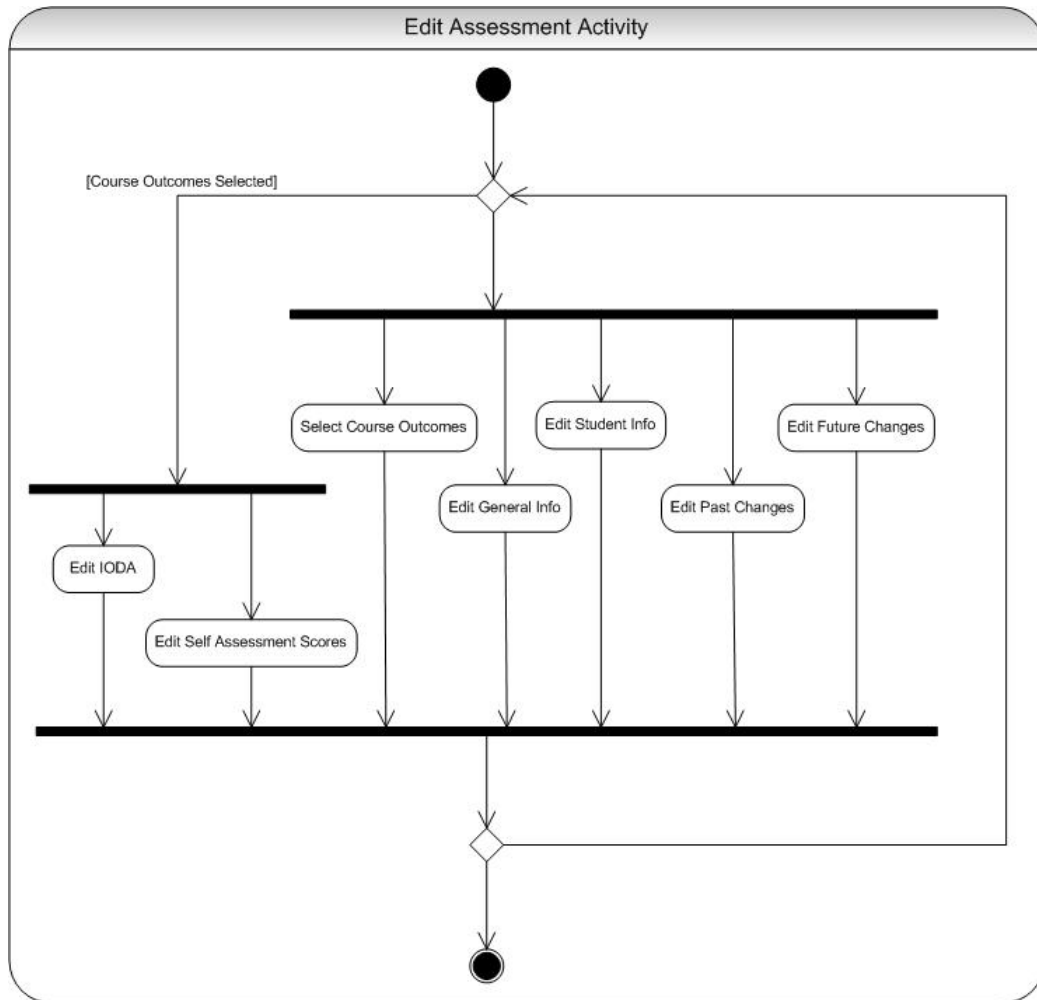


Figure 4.7: Edit Assessment Activity Diagram

this data as necessary. This represents the bulk of the data entry activity and is shown in Figure 4.7. To ensure that all of the assessment data can be easily entered, this activity may be repeated any number of times. Once an assessment has been created per the above activity, past and future changes, outcomes, and student information can be entered and/or edited. The general information can also be edited. Once the outcomes have been selected, additional actions to enter or edit instruments of direct assessment as well as self assessment scores. This is necessary because the outcomes need to be identified before this information is entered.

The final activity is to generate the actual assessment report as shown in Fig-

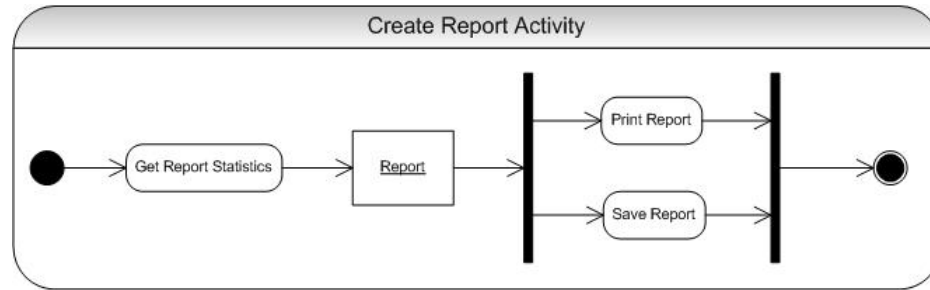


Figure 4.8: Create Report Activity Diagram

ure 4.8. The report is generated from the data entered by the user and by calculating statistics from this data. Once the report is created, the user is able to save a copy or print it for submission. Reports can also be generated at any stage of an assessment and may be created multiple times.

All of the data for the system is stored in a relational database. The database design is shown in Figure 4.9.

## 4.5 Implementation

This section looks at the technologies used in the implementation of the ACAT design. It covers the languages and tools used and follows with the coding philosophy used in creating the software code.

Technologies used in the creation of ACAT:

- **PHP** - There are many choices for programming a dynamic, database driven website. Determining the proper fit is a combination of numerous factors including developer experience, hosting platform, budget, functionality, and personal preference. PHP is a popular language because it is open source and it integrates well with other open source tools such as Linux servers and MySQL databases. It is a powerful language with object oriented support and programming syntax similar to other languages [19]. The choice to use PHP as the language for ACAT was based upon it being open source and it is already in use on the Computer Science and Engineering servers. Additionally, PHP had



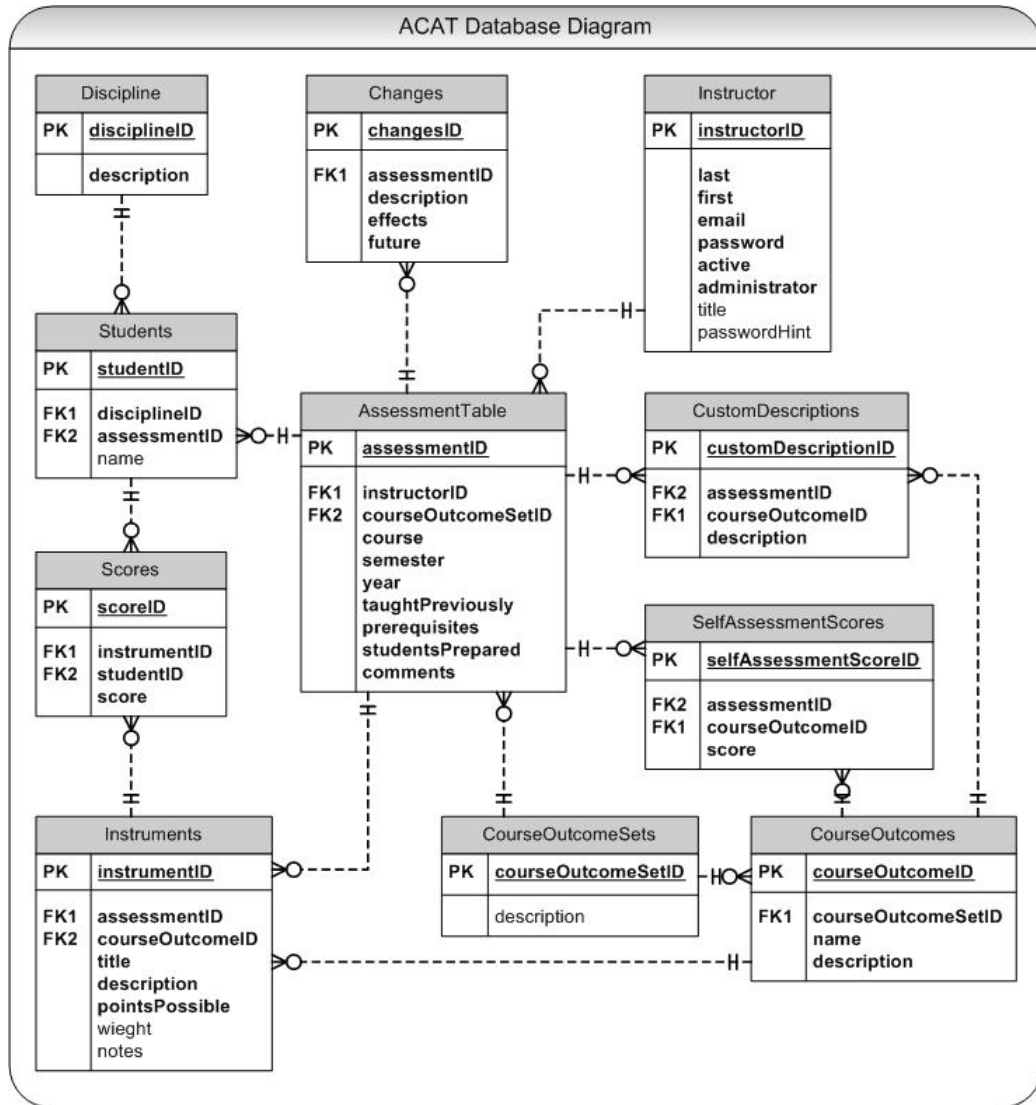


Figure 4.9: ACAT Database Design

all of the functionality required to create ACAT.

- **MySQL** - In contrast to the abundance of web programming languages, the choices for databases are considerably smaller. Available databases range from desktop application to large enterprise systems. Because the database behind ACAT is quite small and the amount of data is relatively minor, a large enterprise system was not warranted. However, a stable and free solution was also desired. MySQL was the logical choice because of these reasons as well as the near seamless integration with PHP [25].
- **JavaScript** - Because PHP is a server side scripting language, it is called only at the request of the user through the web browser. The request is processed and the resulting web page is sent back to the browser. It is sometimes desirable to have some processing available on the browser before the request is submitted. The most common tool for this is JavaScript and it is commonly used to validate form data before sending it to the server. The ACAT software uses JavaScript in this fashion.
- **TCPDF** - The report which is generated by ACAT needs to be presented in a common and usable format. PDF is one of the most widely used formats. Native PHP support for PDF generation is extremely limited so an open source package called TCPDF was chosen. This PHP class supports HTML and creates a PDF in the browser [26]. It integrated nicely into ACAT for report generation.

The philosophy behind the coding was one of code re-usability and extensibility. A uniform page design and theme was achieved by creating single header and footer functions which were used to create each page. This allows for changes in the look and feel to be accomplished in a single PHP file and affect the entire site.

Additional encapsulations of code include having a single file for all database functions and a single security file which is included in every user page to check authorization for viewing the page. PHP files were created based on the function or page which they represent.

# Chapter 5

## ACAT Prototype Operation

This section describes the operation and layout of the ACAT web site. The web site map will be shown with descriptions of the individual pages. The pages are grouped into operational categories.

The main program units for ACAT are web pages. Each web page in ACAT implements some portion of functionality for the system. These web pages may rely on common functionality implemented in the Data Access or Authentication modules, but are otherwise separate. There are three categories of pages: general pages, user pages and administrator pages. The relationships between these pages are best shown in website navigation diagrams, which are shown in Figure 5.1, Figure 5.4, and Figure 5.7.

### 5.1 General Pages

- **Home and about pages** - The home page and about page belong to the general pages subsystem, as shown in Figure 5.1. These pages are HTML only. They have no dynamic content, so PHP is not required. They are accessible to all users, so authentication support is not required. The home page of the ACAT prototype is shown in Figure 5.2.
- **Authentication pages** - The login page and logout confirmation page are part of the general pages subsystem, as shown in Figure 5.1. The login page allows the user to log in as shown in Figure 5.3. The logout confirmation

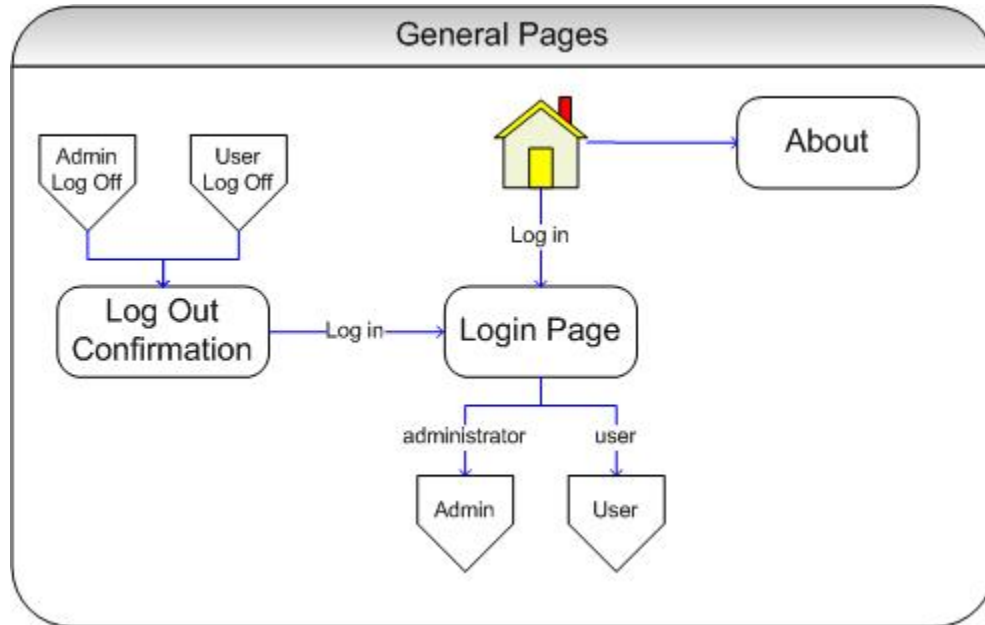


Figure 5.1: General Pages Web Map

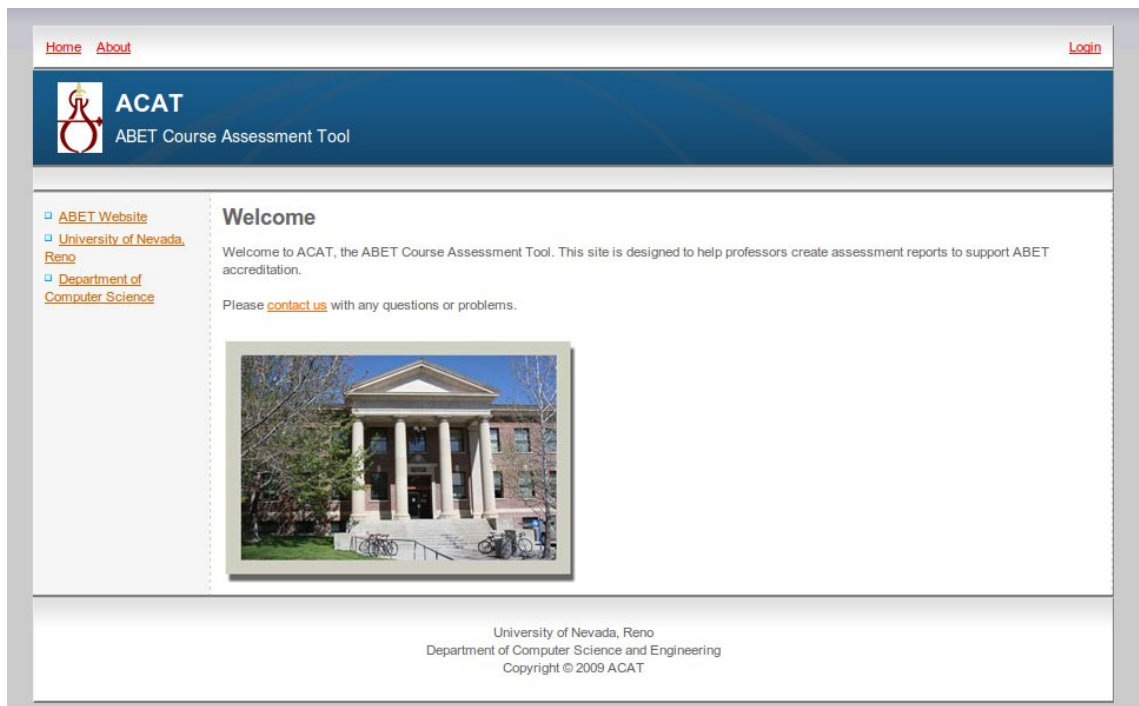


Figure 5.2: ACAT Home Page

Home About Login

**ACAT**  
ABET Course Assessment Tool

Members log in here:

Username:

Password:

University of Nevada, Reno  
Department of Computer Science and Engineering  
Copyright © 2009 ACAT

Figure 5.3: ACAT Login Page

page is displayed to let the user know that they have logged out successfully. These pages are very similar and are implemented in the same module. This module uses functionality from the authentication subsystem. The bulk of the functionality is in the login page, which takes the user name and password as input and passes it to the authentication subsystem. The authentication system indicates whether the data is valid, and starts a new session. When the user logs out, they go to the logout confirmation page, which displays a message confirming that the user is logged out, and offers an option to log back in. This is just a different version of the log in page. Note that although it is not explicitly mentioned, each page contains a link that allows the user to log out.

## 5.2 User Pages

These pages are available to any authenticated user in the ACAT system as illustrated in the website map in Figure 5.4. The user pages represent the main functionality of the system and consist of all of the data input and report creation pages. Additionally,

there is functionality available for users to view and edit their profile information.

- **User Welcome Page** - The user welcome page is displayed to a user immediately after they log in. This is part of the user pages subsystem as shown in Figure 5.4. This page provides links to create a new assessment and to modify the users account information. It also displays a list of assessments that the user has created previously, and a link to modify each assessment. The input data for this page is the current user ID, which is retrieved from the authentication subsystem. This page uses functionality from the data access subsystem to display the list of assessments for the current user.
- **Account Information Page** - The account information page is part of the user pages subsystem. This page displays the users information such as their email address, title, and name, and allows the user to modify this information if necessary. It also allows the user to change their password. The input data for this page is the current user ID, which is retrieved from the authentication subsystem. This page uses functionality from the data access subsystem to retrieve and update the users information in the database. The authentication subsystem is also used when changing the current users password. All passwords are encrypted using the PHP sha1() function, which uses the US Secure Hash Algorithm 1 [31].
- **Assessment Summary Page** - The assessment summary page is displayed when the user selects an existing assessment from the user welcome page, or after the user enters basic information for a new assessment and is shown in Figure 5.5. This page is part of the user pages subsystem, as shown in Figure 5.4. This page displays a summary of all of the data entered for the selected assessment, and gives the user links to pages where the user can enter missing data or update existing data. The input for this page is the current user ID, which is retrieved from the authentication subsystem. It also uses the selected assessment ID as input, which is retrieved from the current session data. This

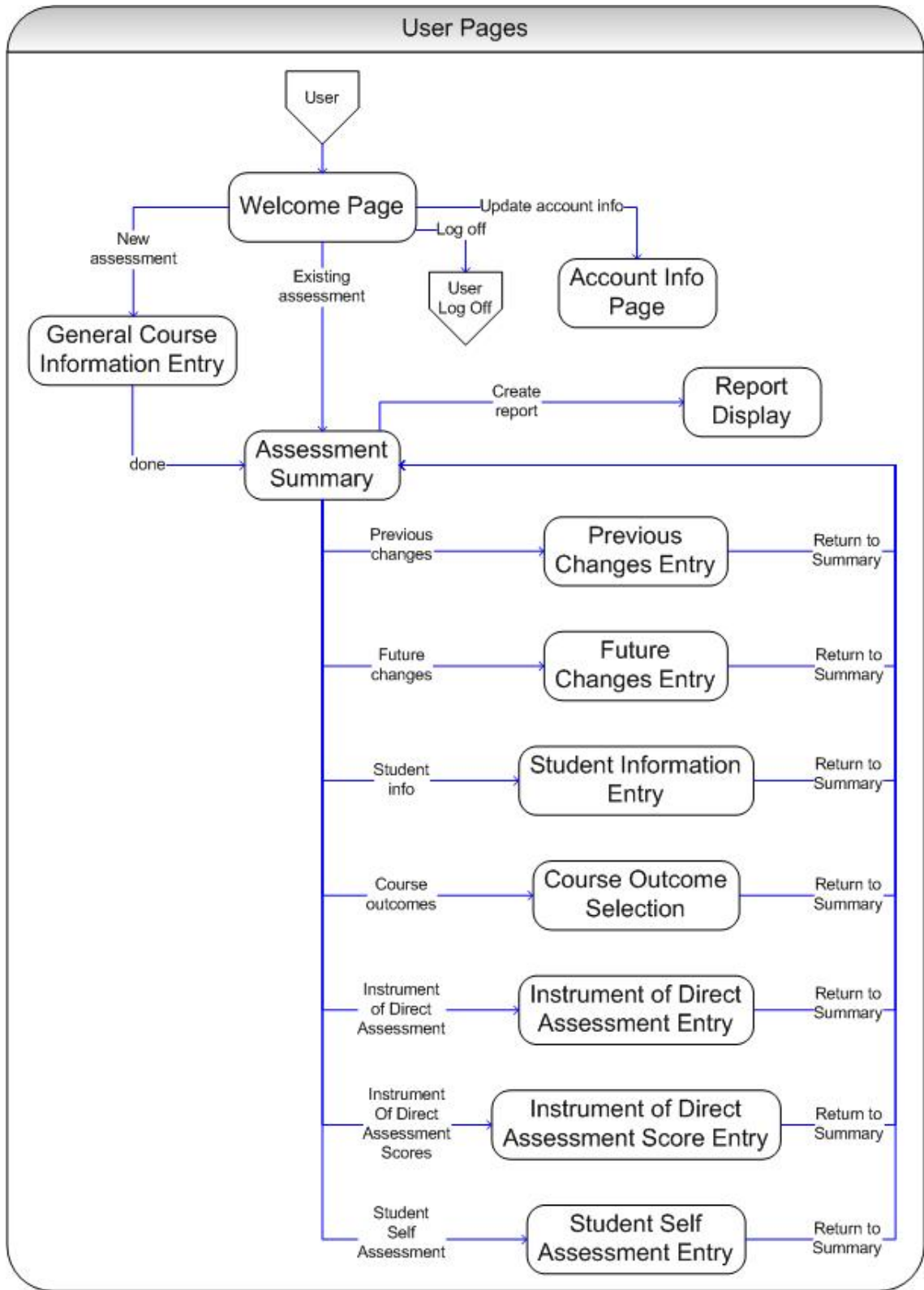


Figure 5.4: User Pages Web Map

[Main](#)

[Edit Profile](#)

[Administrative Functions](#)

## Assessment Summary for CS326 Fall 2006

[Help](#)

[Create Report](#)

General Info	
Course	CS326
Semester	Fall
Year	2006
Have You Taught This Course Before?	✓
Course Prerequisites	CS302
Were the Students Prepared?	✓
Useful Comments	Great class!

[Edit General Info](#)

Past Changes			
Change Description	Effects of Change		
Remove 2nd midterm	Remove 2nd midterm to give students more time to work on projects	<a href="#">Edit</a>	<a href="#">Delete</a>

[Add Past Changes](#)

Future Changes			
Change Description	Purpose of Change		
make class better	teach better so students learn more	<a href="#">Edit</a>	<a href="#">Delete</a>
Make class shorter	Students fall asleep	<a href="#">Edit</a>	<a href="#">Delete</a>

[Add Future Changes](#)

Figure 5.5: Assessment Summary Page

page uses functionality from the data access subsystem to retrieve information for the current assessment.

- **Assessment Data Entry Pages** - A separate data entry page exists for each element of data that must be entered for an assessment. This includes a page to enter or update the following:
  - Basic information such as the course name and semester
  - Past changes and proposed future changes to the course and their effects
  - Student information including the number of students for each discipline
  - Course outcomes applicable to the course (Figure 5.6)



Home About Logged in as test. Logout

**ACAT**  
ABET Course Assessment Tool

Main  
Edit Profile  
Administrative Functions

Help

### Select Course Outcomes for CS326 Fall 2006

Select	Outcome	Default Description
<input type="checkbox"/>	a	an ability to apply knowledge of mathematics, science, and engineering
<input type="checkbox"/>	b	an ability to design and conduct experiments, as well as to analyze and interpret data
<input type="checkbox"/>	c	an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
<input type="checkbox"/>	d	an ability to function on multidisciplinary teams
<input type="checkbox"/>	e	an ability to identify, formulate, and solve engineering problems
<input type="checkbox"/>	f	an understanding of professional and ethical responsibility
<input type="checkbox"/>	g	an ability to communicate effectively
<input type="checkbox"/>	h	the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
<input type="checkbox"/>	i	a recognition of the need for, and an ability to engage in life-long learning
<input type="checkbox"/>	j	a knowledge of contemporary issues
<input type="checkbox"/>	k	an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Update Cancel

Figure 5.6: Select Course Outcomes Page

- Instruments of direct assessment to measure each course outcome
- Scores for each instrument of direct assessment
- Student self assessment scores

Each of these pages has the same basic functionality. The input to each page is the current assessment ID, which is retrieved from the current session data. Each page extracts the data for the current assessment using functionality from the data access subsystem. It then displays this data in an appropriate format, with links or editable forms that allow the data to be entered or modified. In some cases, a method is provided to allow the data to be deleted. Each page has an option to save the data. Selecting this option first validates the data to

ensure that it is valid input, and that no required data is missing. If the data is valid, then it writes the changes to the database, again using functionality from the data access subsystem. When the user is finished with a data entry page, they are returned to the assessment summary page, which is updated with the changes.

- **Report Page** - The report page is accessed through a link on the assessment summary page. In order to generate a report, an assessment must first exist and be open. The link will automatically create a report and display it on the page. This unit presents the most computation and processing of all the components. To generate the report, the system accesses all of the assessment data the user has input. Statistical calculations are made and the numbers are forwarded to a report generation system. This system formats the report per the report design template. Tables and graphs are automatically generated and the report is presented in Adobe PDF. The capabilities of this format allows the user to easily print or save the report. Because of the nature of the PDF plug-in, users must use the back button on their browser to return to the report summary.

### 5.3 Administrator Pages

Users who are designated administrators have access to special administrative pages. A link to these pages is automatically made available to users on the login page. A summary of the administration pages is shown in Figure 5.7. The functions are listed on the administration home page as links. There is also the facility to log off or return to the welcome page.

- **User Modification Page** - One of the tasks of an administrator is user upkeep. New users need to be added and occasionally, the administrator needs to add or change user information, for example resetting a forgotten password. The administration home page has links to forms that can create new users and edit existing users. The create user page has all of the required information for

creating an account. An example of this page is shown in Figure 5.8. Error checking is incorporated to ensure all of the required fields are filled in. A submit button allows the form to be processed and the user information will be stored in the database. The page to edit users can be accessed from the administrative home page. A list of users is presented and the administrator can choose one to edit. The user information field becomes editable and changes can be made. Those changes may then be saved or discarded. There is also a facility to delete a user from the system. Because of data integrity, users are not be deleted from the database; they are only marked as inactive.

- **Course Outcome Modification Page** - In some cases, ABET may choose to change the definition of a course outcome or possibly add new outcomes. Additionally, schools may choose to word an outcome definition differently than ABET's definition and they may also want to add custom ones. An ACAT administrator can accomplish these tasks through the Course Outcome Modification page. These values represent the default outcomes and definitions that can be used in an assessment. These definitions can also be customized by users for individual assessments.

Error detection and validation is performed at two levels. The first is at the browser level using Java Script functions. This allows data to be validated before being sent to the server for processing. Error messages are shown in red next to the input section for easy identification as shown in Figure 5.9. The second level of validation occurs at the server level before processing the data. Error messages are given at the top of the form as shown in Figure 5.10 and require interpretation to find the error. Having a two layer validation system improves functionality and the robustness of the system.

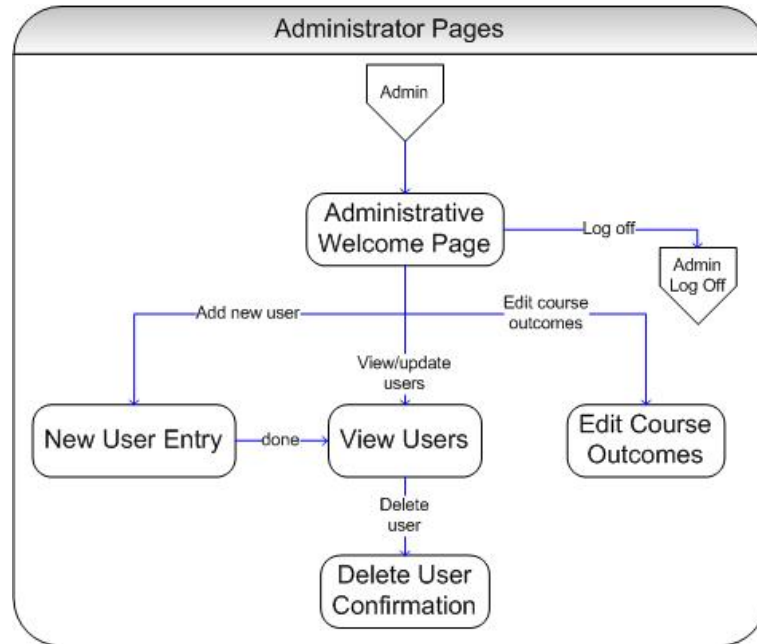


Figure 5.7: Administrator Pages Web Map

The screenshot shows the "Add a New User" page within the ACAT (ABET Course Assessment Tool) interface. The page header includes navigation links for "Home", "About", and "Contact", and a "Logout" link. The ACAT logo and "University of Nevada, Reno" branding are visible. A sidebar on the left contains links for "Home", "View/Update Users", "Add User", and "Edit Outcomes". The main content area features a form with the following fields:

Title	<input type="text"/>
First Name	<input type="text"/>
Last Name	<input type="text"/>
Email	<input type="text"/>
Password	<input type="password"/>
Password Hint	<input type="text"/>
Administrator	<input type="checkbox"/>
Active	<input type="checkbox"/>
<input type="button" value="Insert"/>	

The footer of the page contains the text: "University of Nevada, Reno", "Department of Computer Science and Engineering", and "Copyright © ACAT".

Figure 5.8: Create New User Page

ABET Code	<input type="text" value="a"/>
Value *	<input type="text"/> Value is Required
<a href="#">Insert</a> <a href="#">Cancel</a>	

Figure 5.9: Browser Side Validation Error Message

**Could not log you in.**

Members log in here:

Username:

Password:

Figure 5.10: Server Side Validation Error Message

# Chapter 6

## Usability Study

This chapter describes the usability study which was conducted to evaluate the ACAT software. It covers a background of usability and usability studies, the preparation for the study, the execution of the study, and finally the results.

### 6.1 Usability

As computers and the Internet have revolutionized the way we store and use information, the interfaces to access the data have also grown in complexity. One key to effective user interface and system design is usability. The International Organization for Standards (ISO) defines usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use [29]. With the proliferation of websites as the standard for information dissemination, the design and implementation of web interfaces with a high degree of usability is crucial. It is often the sole discretion of the designer or programmer to produce quality and usable web pages. As can easily be seen on the Internet, website quality ranges from outstanding to disastrous.

Usability can often be defined by two factors, ease of learning and ease of use. These factors are a measurement of the time and the number of actions required to perform a task [4]. Krug believes that the purpose of any website should be explicitly evident to the user. A user should be able to figure out what a web page is about and how to use it without thinking about it [16].

Determining usability is a little less clear. This depends more on the purpose of the site, information, sales, research, downloading software, etc. The underlying goal is to provide people with information in an efficient manner to allow them to make decisions [24].

It is also important to differentiate usability and user experience. While usability focuses on ability to carry out a task, user experience takes a broader view. The entire interaction, thoughts, feelings, and perceptions all go into user experience [27]. Maximizing the entire user experience is the goal of usability.

## 6.2 Usability Studies

The true measure of system comes from the actual intended end user. The aim is that the design of a system be tested by this target group in as near realistic conditions as possible [5]. Careful planning of the study is crucial to ensure accurate measurement of product. The goal of the study must first be determined followed by the plan for the actual study. Factors in planning a study include the method, participants, sample size, metrics, and analysis. Budget is also an important aspect of the study.

The first step in a usability study is identifying the goal for the study. Reasons may include testing new functionality of a system or comparing two or more products. The study may be formative, an iterative approach to testing the design of a product. It may also be summative, evaluating how well a product meets its design goals or how well it performs against the competition. User goals may also be taken into consideration. Performance and satisfaction are two common user goals.

The method of the study must also be determined. This can range from personal interviews to focus groups to group studies to online surveys. The method of collecting data needs to be selected as well as the time frame for the study [27]. If a questionnaire is utilized, the format and questions are critical to the success of the study. Standardized surveys may be used or custom questions and evaluations may be developed.

Participants are the most important part of a usability study. Research must be

accomplished to identify the characteristics and demographics of the intended users. Age, skill ratings, experience, and education are all examples of this. Additionally, the number of participants must be determined. This ensures an adequate cross section of users as well as a valid statistical quantity. Finally, recruitment methods must be identified [5].

Once the study is complete, the data is analyzed and interpreted. Statistical analysis and summations of the data are performed and presented in a manner that is consistent with the goals of the study. The results can be then be presented in a final usability study report.

### **6.3 ACAT Usability Studies**

ACAT was designed with a specific purpose of creating course assessments for a narrow user base of university instructors. A usability study was deemed the best method for determining the effectiveness and use of the software. The goals of the study are to determine whether the product meets its design goals and whether it is a valuable tool for the target audience. Additionally areas for improvement are an important aspect of the study.

It was determined that self-reported data from the users experience after using the software would determine the goals. There are numerous surveys available to evaluate system usability. The Computer System Usability Questionnaire (CSUQ), shown in Appendix A, was chosen for this study. This survey was developed by IBM in the early 1990s. It consists of 19 questions to assess the users' perceived satisfaction with the computer system. The factors evaluated in this survey are system usefulness, information quality, interface quality, and overall satisfaction [17].

Any research that involves human subjects at the University of Nevada, Reno is subject to review and approval by the Office of Human Resource Protection. To protect the rights of individuals involved in research activities the University maintains Institutional Review Boards (IRBs) to review research protocols involving human subject [28]. The process consists of online training for all principal investigators and



research personnel in the history of human resource protection and the review process. The usability study for ACAT required an application and IRB review. In order to expedite the process, which can take up to several months, the study was requested to be exempt research. The primary requirement for exempt research is minimal risk to those involved in the study. An additional requirement for the exemption was that the questionnaire could not be electronic or online. The usability study for ACAT received the exempt research approval with the stipulation that the survey be hard copy. The Certificate of Approval of Exempt Research, E09/10-034, is included in Appendix B.

ACAT is a tool designed for university faculty, therefore the participants chosen for this study are teaching faculty from the Computer Science and Engineering Department at the University of Nevada, Reno. These participants are characterized as educated and experienced computer users who can be very critical of software design and operation. All 16 faculty member were invited to participate in the study as well as several graduate students.

The study was conducted over a one week period. The participants were recruited with a flier describing the study and in person visits. Instructions were provided describing the procedures. Users were asked to create an assessment using ACAT with any valid data. They were then requested to create a report with the software and fill out and return the provided Computer System Usability Questionnaire. The time for the study was estimated to be 30 to 45 minutes. Because of the time constraints and varying schedules of the faculty members, the study was conducted at the location and time of the participants choice.

## 6.4 Results

Of the 16 faculty members who were requested to participate in the study, 9 responded. This is an expected sample size as schedules and work loads can be difficult to work around. The results for overall satisfaction, system usefulness, information quality, and interface quality are shown in Table 6.1 and the corresponding questions

Table 6.1: Usability Study Results

	<b>Average (Out of 7)</b>	<b>Standard Deviation</b>	<b>Percentage</b>
Overall Satisfaction	5.21	1.24	74.5
System Usefulness	5.58	1.24	79.7
Information Quality	4.81	1.48	68.7
Interface Quality	5.15	1.41	73.6

that comprise these averages are shown in Table 6.2 [17]. Note that the Likert scale for the survey is scored from 1 being to highest ranking to 7 as the lowest. For this section, the scores are reversed for better readability.

The highest category, system usefulness, at 79.7% indicates that the users involved in the test believe that ACAT will be a useful tool. However, as indicated by the 68.7% approval of information quality, there is room for improvement. The scores for general categories, as well as the averages for all of the questions are shown in Figure 6.1.

The questions that scored the lowest were 9 through 11. These related to error messages, error handling, and online help. This corresponds to several comments regarding poor help and instructions. Conversely, the highest individual score was for “it was easy to learn the system.” The discrepancy is due to the fact that half of the respondents did not encounter any errors and marked those items as not applicable. A few of the users who did encounter errors gave lower scores.

Equally important in this study were the comments included with the CSUQ

Table 6.2: CSUQ Categories and Corresponding Questions

<b>Category</b>	<b>Items</b>
Overall Satisfaction	Questions 1 - 19
System Usefulness	Questions 1 - 8
Information Quality	Questions 9 - 15
Interface Quality	Questions 16 - 18

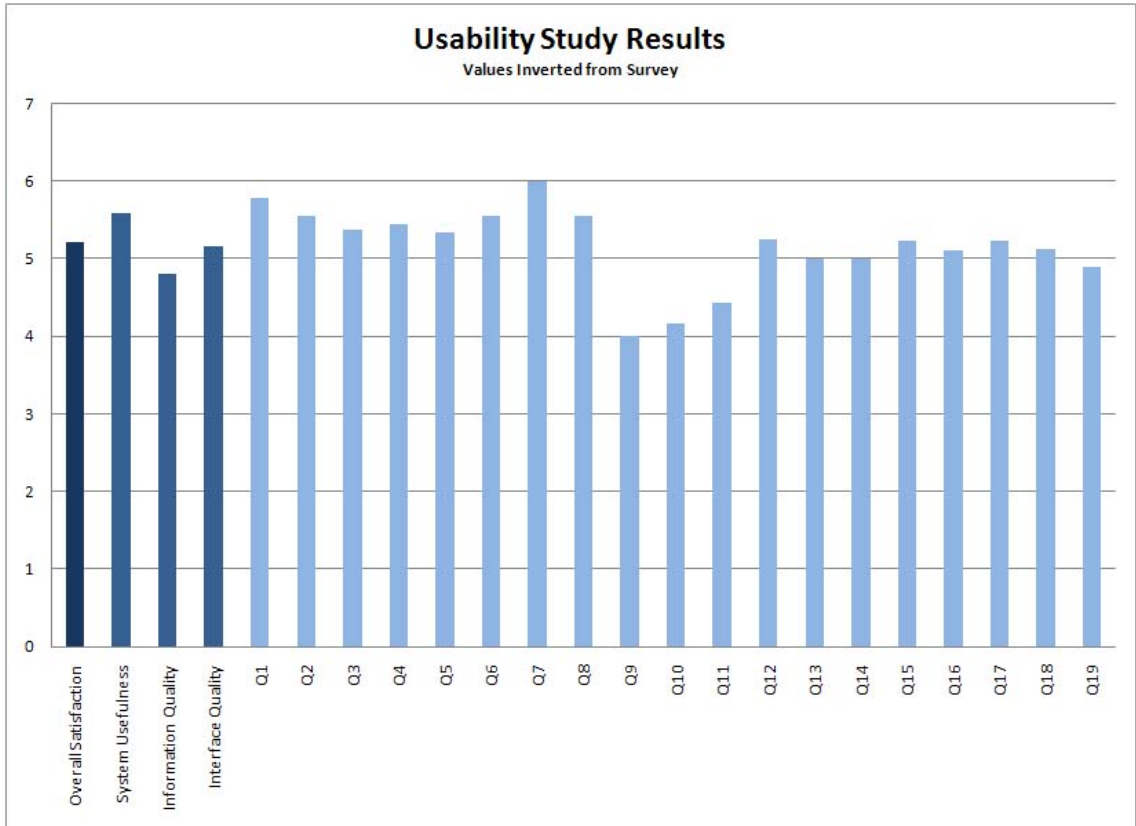


Figure 6.1: CSUQ Results

surveys. These highlights the most significant areas for improvement. Many of these suggestions are listed below and represent ways to increase the usability of the tool. Additionally, the comments relate to common usability issues in Human Computer Interaction.

- The most frequent comments were regarding help and understanding how to use the system. More guidance and support is needed. Basic help pages were included, but a fully integrated help system is required and should be part of the design and implementation process.
- Wording for actions such as submitting a form must be precise. “Update” is understood for submitting a change on existing data, but can be confusing for entering new data. “Insert” is also not standard web phraseology. “Save” is the commonly accepted word.
- One user encountered an error occurred while producing the PDF report due to an incompatible browser. While the software was tested on several different browsers and operating systems, it is vital to also test on older browsers.
- The graphs that are produced in the report are not correct. They were missing data.
- Navigation needs to be in place from the home page or the about page for an authenticated user. If these pages are visited while logged in, there is no link back to the user home.
- Since the design of ACAT, some requirements have changed, such as using numbers instead of letters for outcomes.

The system was released to the users for this usability study directly from the development stage. Many of the errors and problems noted in the surveys could have been discovered prior to the usability study if a review by the accreditation and assessment chair and one or two of the committee members was performed.

While the overall usefulness of the ACAT system was measured at 74.5%, there is certainly room for improvement. One major lesson learned from this study is that tasks that seem obvious or intuitive to the designer may not be that way to a user. This is highlighted in that Computer Science faculty, who are considered expert computer users, still come across the same issues that affect novice users. In addition to the problems noted, there were numerous positive remarks on the system's usefulness and ease of use.

# Chapter 7

## Comparison With Similar Work

Assessment tracking and outcome based curriculum management are high visibility areas of research today. Tools are being developed to aid in the tracking of course and program outcomes and to help map the actual courses to these outcomes. Much of the research has focused on the collection of assessment data from self assessments and instruments of direct assessment.

Initial research focused more on the process and less on tools designed to aid in assessments. This was often in response to a recent ABET visit [6, 32]. Databases and interfaces have also been developed to store the large amounts of data used in assessment reporting data by storing and mapping student data [7, 8, 20]. While these are all excellent tools in assisting with assessment, they do not address the actual reporting requirements of accreditation.

Two software tools were created with the intent of collecting, analyzing and reporting data [2, 30]. The first, COMPASS from the University of West Georgia, uses an existing, open-source classroom management system to assist in this process. Functionality to map course objectives to assignments was added. This allowed additional information regarding learning objectives and student performance in a database for further review and analysis. While this is a major improvement over the current process, there is no reporting mechanism in place. The data that can be used for a course assessment report is available, but must be retrieved and formatted for the official report.

The second tool is custom software and integrated user interface, database, and

reporting. The system is based on an Outcomes Database and was developed at the Curtin University of Technology in Perth, Australia. It differs from COMPASS and ACAT in that it focuses on the Australian requirements for technical programs. This tool maps assessment items through three layers of outcomes - course, degree program, and generic attributes. Numerous reports were developed for use of curriculum developers, instructors, and students. These reports show the relationships and mappings between courses and outcomes as well as degree programs and generic attributes.

The goal of ACAT is to streamline the course assessment process and standardize reporting. This is accomplished through a custom and integrated interface, database, and report engine. It encompasses data required for a complete assessment report, including student outcomes of direct assessment, self assessment values, and outcomes. A comparison of the features incorporated in ACAT, COMPASS and the Outcomes Database is shown in Table 7.1.

Table 7.1: Comparison of ACAT to Related Tools

	<b>ACAT</b>	<b>COMPASS</b>	<b>Outcomes DB</b>
Custom User Interface	✓		✓
Custom Database	✓		✓
Map Outcomes to Student Instruments	✓	✓	✓
Custom Outcome Wording	✓	✓	
Use Self Assessment Data	✓		
Course Report	✓		✓

# Chapter 8

## Conclusions and Future Work

### 8.1 Conclusions

Accreditation for a technical program of study is highly desirable. It shows that the program meets certain requirements and that the graduates possess a higher level of skill and competence. The requirements for accreditation from an organization such as ABET are stringent. Reporting and documentation specifications for certification and re-certification are very time consuming. Course assessments are a major portion of this reporting and most every faculty member is involved.

A need exists to automate and standardize the process for generated ABET course assessments. A detailed analysis and requirements development was performed and a comprehensive design was created for developing a software tool, ACAT, aimed at facilitating the assessment process. The emphasis on the design centered on ease of use and functionality.

Based on this design, a prototype of ACAT was developed. It provides the facility for a user to enter all of the data for a course assessment, including general information, course outcomes to be assessed, instruments of direct assessment for each outcome, student scores for each instrument, and student self evaluation survey results. The system deployed to department web server for evaluation by faculty.

A usability study was accomplished measuring the satisfaction with the functionality of ACAT. Faculty members tested the system and provided feedback and assessment based on the Computer System Usability Questionnaire. Results revealed



that the software is an acceptable method to create course assessments and suggestions were made to improve the software. This validates the design and usability goals of the tool.

ACAT also has the potential to support the assessment process in other accredited universities. ACAT collects data that is common in assessment reports for, and could be tailored to a specific programs needs with slight modification. For example, the University of Michigan at Dearborn prepares assessment reports that list the course outcomes being assessed, what the students were asked to show their achievement of that outcome, and the average score [18]. ACAT collects this data currently, and could be adapted to prepare reports in the desired format for this program as well as many other similar programs.

Overall, the research contained in this thesis shows that the continued development and support ACAT is a worthwhile investment in time. It can be a valuable resource for ABET course assessments and can be extended further for periodic evaluation and assessment of programs of study. Future additions can propel this software into a highly productive and dependable tool.

## 8.2 Future Work

Many hours of work have gone into designing ACAT and creating a user friendly and useful interface. While the prototype can currently be used for an ABET accreditation visit, additional work can make it even better. The modifications to improve this tool range from short range cosmetic and usability fixes to integration with external systems. This chapter covers these changes for a better ACAT.

The first priority is to address the shortcomings and suggestions identified during the usability study. This includes improved instructions for entering data as well as more descriptive error messages. While there is a basic help system for ACAT, a more comprehensive version would improve the usability of the tool. Next error messages to confirm deleting data were recommended. Finally, the ability to delete an outcome for an assessment was recommended. This is a more complex task to ensure cascaded

deleting of data in the database to ensure data integrity.

In an effort to further reduce the time required to create an ABET assessment, there are pieces of data that could be automatically loaded into the ACAT database from external system.

- A list of courses taught by the instructor and displayed in a drop down list by year and semester to create a new assessment.
- Course prerequisites as listed in the course catalog for the year the course was taught.
- Self assessment data for each course.

The assessment report is created by PHP code and HTML tables and then converted to a PDF document. While this created a standard report, it is complicated to make changes or additions. Further investigation into dedicated reporting software will improve the report modification process and facilitate adding additional reports as described next.

As part of the course assessment report, copies of examinations are often included. The ability to store scanned copies of these examinations and add them to the generated report is desirable. Adding this functionality to ACAT will also keep all of the documentation for courses in a single location.

Additional reports in support of ABET accreditation can be developed. A course outcomes matrix could be developed fairly quickly mapping each course to the outcomes for a given semester. Additionally, an outcome assessment report in support of program outcome 1 can also be created. This is a more complicated report to develop, but the data required for it is already being collected in ACAT. All of the course data for the required time period would be required for these two additional reports.

Finally, there is a significant amount of data that is collected and stored in the ACAT database representing student performance and satisfaction with a given

program of study. Future reports and analysis of the data can be developed to track trends and help evaluate curriculum.

# Bibliography

- [1] ABET. Why should I choose an accredited program? [http://www.abet.org/why\\_choose.shtml](http://www.abet.org/why_choose.shtml) (Accessed October 27, 2008).
- [2] Adel Abunawass, Will Lloyd, and Edwin Rudolph. COMPASS: a CS program assessment project. In *ITiCSE '04: Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*, pages 127–131, New York, NY, USA, 2004. ACM.
- [3] Jim Arlow and Ila Neustadt. *Uml and the Unified Process: Practical Object-Oriented Analysis and Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.
- [4] Albert N. Badre. *Shaping Web Usability: Interaction Design in Context*. Addison-Wesley Longman Publishing Co., Inc., Indianapolis, NJ, 2002.
- [5] David Benyon, Phil Turner, and Susan Turner. *Designing Interactive Systems - People, Activities, Contexts, Technologies*. Addison-Wesley Longman Publishing Co., Inc., Essex, England, 2005.
- [6] Dick K. Blandford and Deborah J. Hwang. Five easy but effective assessment methods. In *SIGCSE '03: Proceedings of the 34th SIGCSE technical symposium on Computer science education*, pages 41–44, New York, NY, USA, 2003. ACM.
- [7] Larry Booth. A database to promote continuous program improvement. In *SIGITE '06: Proceedings of the 7th conference on Information technology education*, pages 83–88, New York, NY, USA, 2006. ACM.
- [8] Larry Booth, Jon Preston, and Junfeng Qu. Continuous program improvement: a project to automate record-keeping for accreditation. In *SIGITE '07: Proceedings of the 8th ACM SIGITE conference on Information technology education*, pages 155–160, New York, NY, USA, 2007. ACM.
- [9] J.T. Cain. Engineering and computer science accreditation. In *Information Communications Technology, 2008. ICICT 2008. ITI 6th International Conference on*, pages 45–50, Dec. 2008.
- [10] Computing Accreditation Commission. Self-study questionnaire 5-28-09. <http://www.abet.org/Linked%20Documents-UPDATE/Program%20Docs/C003%20Self-Study%20Questionnaire%205-28-09.doc> (Accessed February 17, 2010).

- [11] Council for Higher Education Accreditation. Informing the public about accreditation. [http://www.chea.org/public\\_info/index.asp](http://www.chea.org/public_info/index.asp) (Accessed April 9, 2009).
- [12] R. R. DeLyser and M. A. Hamstad. Outcomes based assessment and a successful ABET 2000 accreditation at the university of denver. In *FIE '00: Proceedings of the 30th Annual Frontiers in Education*, pages T1A/1–T1A/6, Washington, DC, USA, 2000. IEEE Computer Society.
- [13] Alan Dennis, Barbara Wixom, and David Tegarden. *Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach (3rd Edition)*. John Wiley and Sons, 2009.
- [14] J. Enderle, J. Gassert, S. Blanchard, P. King, D. Beasley, P.Jr. Hale, and D. Aldridge. The ABCs of preparing for ABET. *Engineering in Medicine and Biology Magazine, IEEE*, 22(4):122–132, July-Aug. 2003.
- [15] Essa Eugene, Andrew Dittrich, Sergiu Dascalu, and Frederick C. Harris Jr. ACAT: A web-based software tool to facilitate course assessment for ABET accreditation. In *Information Technology, 2010. ITNG '10. Seventh International Conference on*, april 2010.
- [16] Steve Krug. *Don't Make Me Think: A Common Sense Approach to the Web (2nd Edition)*. New Riders Publishing, Berkeley, CA, USA, 2006.
- [17] James R. Lewis. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *Int. J. Hum.-Comput. Interact.*, 7(1):57–78, 1995.
- [18] Bruce R. Maxim. Closing the loop: assessment and accreditation. *J. Comput. Small Coll.*, 20(1):7–18, 2004.
- [19] PHP Website. What can PHP do? <http://www.php.net/manual/en/intro-whatcando.php> (Accessed March 22, 2008).
- [20] John Preston and Devid Wilson. Achieving consistency and communication in online education via an online grading system. In *TOHE '04: Teaching Online in Higher Education*, 2004.
- [21] Kathryn E. Sanders and Robert McCartney. Program assessment tools in computer science: a report from the trenches. In *SIGCSE '03: Proceedings of the 34th SIGCSE technical symposium on Computer science education*, pages 31–35, New York, NY, USA, 2003. ACM.
- [22] Ian Sommerville. *Software Engineering: (Update) (8th Edition) (International Computer Science)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.
- [23] N. Soundarajan. Engineering criteria 2000: the impact on engineering education. In *Frontiers in Education Conference, 1999. FIE '99. 29th Annual*, volume 1, pages 11A1/25–11A1/30 vol.1, 1999.

- [24] Jared M. Spool, Tara Scanlon, Carolyn Snyder, Will Schroeder, and Terri DeAngelo. *Web site usability: a designer's guide*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- [25] Steve Suehring, Tim Converse, and Joyce Park. *PHP 6 and MySQL 6 Bible*. Wiley Publishing, 2009.
- [26] TCPDF Website. PHP class for creating PDF documents. <http://tcpdf.org> (Accessed November 17, 2009).
- [27] Thomas Tullis and William Albert. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Morgan Kaufmann Publishers Inc., 2008.
- [28] UNR Office of Human Resource Protection. Policies and procedures for human resource protection. <http://www.unr.edu/ohrp/policy.htm> (Accessed April 24, 2010).
- [29] Usability Net Website. International standards for HCI and usability. [www.usabilitynet.org](http://www.usabilitynet.org) (Accessed June 21, 2009).
- [30] Brian R. von Kinsky, Allan Loh, Mike Robey, Susan J. Gribble, Jim Ivins, and David Cooper. The benefit of information technology in managing outcomes focused curriculum development across related degree programs. In *ACE '06: Proceedings of the 8th Australian conference on Computing education*, pages 235–242, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.
- [31] W3 Schools. Php sha1() function. [http://www.w3schools.com/php/func\\_string\\_sha1.asp](http://www.w3schools.com/php/func_string_sha1.asp) (Accessed April 28, 2010).
- [32] Z.A. Yamayee, R.J. Albright, M.I. Inan, M.S. Kennedy, K.H. Khan, and V.D. Murty. Work in progress - streamlining assessment process in response to a successful ABET visit. In *Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference*, pages S1C –S1C, Oct. 2005.

# Appendix A

## Computer System Usability Study

This questionnaire (which starts on the following page), gives you an opportunity to tell us your reactions to the system you used. Your responses will help us understand what aspects of the system you are particularly concerned about and the aspects that satisfy you.

To as great a degree as possible, think about all the tasks that you have done with the system while you answer these questions.

Please read each statement and indicate how strongly you agree or disagree with the statement by circling a number on the scale. If a statement does not apply to you, circle N/A.

Please write comments to elaborate on your answers.

As you complete the questionnaire, please do not hesitate to ask any questions.

Thank you!

1. Overall, I am satisfied with how easy it is to use this system.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

2. It was simple to use this system.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

3. I can effectively complete the tasks and scenarios using this system.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

4. I was able to complete the tasks and scenarios quickly using this system.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

5. I was able to efficiently complete the tasks and scenarios using this system.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**



6. I felt comfortable using this system.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

7. It was easy to learn to use this system.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

8. I believe I could become productive quickly using this system.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

9. The system gave error messages that clearly told me how to fix problems.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

10. Whenever I made a mistake using the system, I could recover easily and quickly.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

11. The information (such as on-line help, on-screen messages and other documentation) provided with this system was clear.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

12. It was easy to find the information I needed.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

13. The information provided for the system was easy to understand.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

14. The information was effective in helping me complete the tasks and scenarios.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

15. The organization of information on the system screens was clear.

<b>SSTRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
----------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

16. The interface of this system was pleasant.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

17. I liked using the interface of this system.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

18. This system has all the functions and capabilities I expect it to have.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

19. Overall, I am satisfied with this system.

<b>STRONGLY AGREE</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>STRONGLY DISAGREE</b>	<b>N/A</b>
---------------------------	----------	----------	----------	----------	----------	----------	----------	------------------------------	------------

**COMMENTS:**

## Appendix B

# Exempt Research Certification

The certification for exempt research form the University of Nevada, Reno Office of Human Resource Protection is included in this appendix.



### *Certification of Approval of Exempt Research*

Date: April 15, 2010  
To: Eugene Essa  
Department of Computer Science and Engineering  
2390 Roanoke Trail  
Reno, NV 89523  
CC: Frederick Harris PhD  
Department of Computer Science and Engineering / 0171

---

Exemption:	<b>E09/10-034</b>
	<b>ACAT - ABET Course Assessment Tool</b>
Sponsor:	N/A
VA Research:	No
UNR Assurance Number:	<b>FWA00002306</b>
Action Item:	<b>New Protocol: Exempt Research</b>
Level of Review for Action:	<b>Exempt</b>
Exemption Category	<b>2</b>
Action Date:	<b>4/14/10</b>
Approval Date:	<b>4/15/10</b>
Approval Period	<b>12 months</b>
Expiration Date:	<b>4/15/11</b>

---

**This approval is for:**

- Computer system usability questionnaire, as revised
- Recruitment flyer, as revised
- Questionnaire instructions, as revised
- Application for Exempt Research (4/15/10), as revised

**PI responsibilities**

- Proposed changes must be reviewed and approved by the Office of Human Research Protection (OHRP) prior to initiation, except where necessary to eliminate apparent immediate hazards to subjects. Such exceptions must be reported to the OHRP at once.
- Any unanticipated problems which may increase the risks to human subjects or unanticipated adverse events must be reported to the OHRP within 10 days of becoming aware of the issue.
- Exempt applications are not renewable and are not subject to continuing review. If you would like to continue the research or analysis of identifiable data beyond the expiration date, you will need to submit a new Application for Exempt Research to the Office of Human Research Protection for review and approval.

Please reference the exemption number above on all related correspondence with the OHRP. If any additional information is necessary, please contact J. Logan Hamill at 775.327.2368.



\_\_\_\_\_  
Susan Ford Publicover, MA, CIP, OHRP Director

# Appendix C

## Sample ACAT Assessment Report

The following information is used as input to ACAT. The report generated from the system follows.

Course: CS 201  
Semester/Year: Spring  
Prerequisite: CS 123  
Taught course previously: yes  
Were students prepared: yes  
Useful Comments: Excellent class

Past change description: Too much homework  
Past change effect: Reduced homework by 2 assignments

Future change description: More guest speakers  
Future change purpose: Bring in guests from the industry

4 Computer Science Students

Course outcomes: a and c

Instruments of direct assessment:

Outcome a, Midterm problem 4, 5 points, scores 4, 5, 3, 4

Outcome c, Final question 3, 10 points, scores 9, 7, 10, 8

Outcome c, Homework 2 problem 1, 5 points, scores 5, 5, 4, 4

Self assessment scores:

Outcome a: 4.35

Outcome c: 4.78

# CS & E Course Assessment Form

63

**CS 201**

**Instructor:** Dr. Ima Test

**Semester/year:** Spring 2009

The purpose of this form is to document the achievement of ABET Criteria 3 outcomes in the courses that you instruct. Answers to the questions below should cite supporting evidence from your own observations, student performance on assignments and examinations, student self assessment forms, and other feedback.

First time taught by this instructor

Course taught previously

Course prerequisite(s): CS 123

Were the students adequately prepared by prerequisite course(s)? Yes  No

Were changes implemented since the last time this course was taught? Yes  No

If yes, what changes were made since the last time this course was taught?

<i>Changes Made Since Last Time</i>	<i>Effects of Change</i>
Too much homework	Reduced homework by 2 assignments

Are changes called for the next time this course is taught? Yes  No

If yes, what changes should be made the next time this course is taught?

<i>Changes Made Since Last Time</i>	<i>Effects of Change</i>
More guest speakers	Bring in guests from the industry

<i>Most Useful Comments from Students</i>
Excellent class.

## Achievement of Objectives / Demonstration of Outcomes

64

Did the students demonstrate achievement of the learning objectives/expected outcomes specific to this course? In the table provided, cite evidence using student responses on the student self assessment questions and evidence from your direct assessment of student work.

### Computer Science

#### Mapping between ABET Outcomes - Course Outcomes - Instruments of Assessment

ABET Criterion 3 Outcomes	Student Self Assessment	Course Outcomes	Instruments of Direct Assessment by Instructor	Outcome Average
a	4.350	an ability to apply knowledge of mathematics, science, and engineering	Midterm - Question 4	4.00
c	4.780	an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability	Final Exam, Question 3 Homework 2, Problem 1	4.38

#### Student Samples

ABET Criterion 3 Outcomes	a		c		
	Midterm Q 4	Avg.	Final Q 3	HW 2 P 1	Avg.
Student Samples (5 = excellent to 1 = poor)	4.00	<b>4.0</b>	4.50	5.00	<b>4.8</b>
	5.00	<b>5.0</b>	3.50	5.00	<b>4.3</b>
	3.00	<b>3.0</b>	5.00	4.00	<b>4.5</b>
	4.00	<b>4.0</b>	4.00	4.00	<b>4.0</b>
<b>Outcome Averages</b>	<b>4.00</b>		<b>4.38</b>		



# Distribution of Scoring

ABET Criterion 3 Outcomes	Distribution of Scoring				
	Poor [1.0-1.5]	Fair (1.0-2.5]	Good (2.5-3.5]	Very Good (3.5-4.5]	Excellent (4.5-5.0]
<b>a</b>	0	0	1	2	1
<b>c</b>	0	0	0	3	1

