**AbOut**

**MTM Program Product**

**Software Requirements Specification**

*Version 4.0*

*3/25/2015*

*Applying MTM SRS Version 4.0*

*Nick Broden, Aaron Hoff, Celia Schahczenski, Ryan Stapley, Chris Tenda, Brian D. Knopp*

SRS Version History

|  |  |  |  |
| --- | --- | --- | --- |
| SRS Version | Date | Authors | Comment |
| 1.0 | 110213 | Nick Broden, Aaron Hoff, Celia Schahczenski, Ryan Stapley, Chris Tenda | Combined group project SRS |
| 1.1 | 110331 | Aaron Hoff, Chris Tenda | Added Use-cases and functional requirements |
| 1.2 | 110414 | Aaron Hoff, Chris Tenda | Added Requirements models and specifications |
| 1.3 | 110421 | Aaron Hoff, Chris Tenda | Updated SRS and requirements models, Added User Interfaces |
| 1.4 | 110428 | Aaron Hoff, Chris Tenda | Updated SRS with corrections and report samples |
| 2.0 | 120229 | Celia Schahczenski | Updated based on feedback from Maintenance SE class, Spring 2012 |
| 2.1 | 120324 | Celia Schahczenski | Updated based on user interfaces and student feedback from DB, spring 2012. |
| 2.2 | 120417 | Celia Schahczenski | Finished earlier work and updated reports |
| 3.0 | 130312 | Brian D. Knopp | MTM SRS Std. v3.4 Compliance; clarified sections, minor reformatting; incorporate defects feedback. |
| 3.1 | 130413 | Brian D. Knopp | Incorporate inspection feedback. |
| 3.2 | 130512 | Frank Ackerman | Restore requirement SM3 to client’s language in v2.2 |
| 3.3 | 121123 | Celia Schahczenski | Updated based on inspection by students in ESOF 326, Spring 2013. |
| 3.4 | 140213 | Celia Schahczenski | Updated based on work done in ESOF 326, spring 2014. |
| 4.0 | 150325 | Mitch Deplazes, Josh Lee, Celia Schahczenski, Luke Schuler, Mack Sutherland | ESOF 328, spring 2015, updated to accommodate EE |

**Montana Tech Software Engineering Students:**

These Montana Tech Method software engineering standards encapsulate Dr. Ackerman’s decades of experience in the software industry, the IEEE software engineering standards, and many suggestions from various texts. They have gone through many revisions and additions over the last several years. They are part of your software engineering studies so that (1) you may have the experience of developing software to a standard (which you may find you need to do if you take a job that requires high reliability software), and so that (2) you will have the experience of developing high quality software. You are also invited to participate in the continuing evolution of these standards by studying them critically and making suggestions for their improvement and correction.

TABLE CONTENTS

1. Introduction 7

2.1 Software Purpose and Scope 7

2.2 Document Purpose and Contents 7

2.3 Definitions, Acronyms, and References 8

1.3.1 Definitions 8

1.3.2 Acronyms and Abbreviations 10

1.3.3 Data Dictionary 10

2.4 References 10

2. General Factors 11

2.5 Product Perspective 11

2.6 Product Functions 11

Function Overview 11

Administrative Functions 11

Faculty Functions 12

Reporting Functions 12

2.7 Environmental Conditions 12

2.8 User Characteristic 13

2.9 Dependencies 13

2.10 Assumptions 13

3. Analysis Use Cases 14

3.1 Actors 14

3.1 Use Cases 15

4. Explanatory User Interfaces 16

5. Specific Requirements 17

5.1 Functional Requirements 17

5.1.1 System (SM) 17

SM1: AbOut login 17

SM2: AbOut logout 17

SM3: Automatic logout 17

5.1.2 Administrator (AM) 17

AM1: Add user information 17

AM2: Edit user information 18

AM3: Delete user information 18

AM4: View user information 18

AM5: Add semester 18

AM6: View all semesters 18

AM7: Default semester 19

AM8: Change default semester 19

AM9: Administrative view 19

AM10: Add student outcome 19

AM11: Edit student outcome 19

AM12: Delete student outcome 19

AM13: View outcomes 20

AM14: Add Performance Criterion 20

AM15: Edit Performance Criterion 20

AM16: Delete Performance Criterion 20

AM17: View performance Criteria 20

AM18: Add course 21

AM19: Edit course 21

AM20: Delete course 21

AM21: View courses 21

AM22: Associate or dissociate a performance criterion with a course 21

AM23: Add course offering 22

AM24: Edit course offering 22

AM25: Delete course offering 22

AM26: View all course offerings 22

AM27: Import a list of students into a course offering 23

AM28: Add student to a course offering 23

AM29: Delete a student from a course offering 23

5.1.3 Faculty (FC) 23

FC1: Semester selection 23

FC2: Export a list of students from a course offering 23

FC3: Basic Characteristics of Offerings 24

FC4: Add/remove students from course offering 24

FC5: Add metric to a course offering 24

FC6: Edit metric 24

FC7: Delete metric 25

FC8: Add student score to metric 25

FC9: Edit student score on metric 25

FC10: View extent to which students met student outcome 25

FC11: Faculty view 25

5.1.4 Reports (RP) 26

RP1: Indicate semester for the Outcome Report 26

RP2: Indicate semester(s) for Course PC and PC Semester Reports 26

RP3: Indicate program(s) for Course PC, Matrix, and PC Semester Reports 26

RP4: Indicate course offering for CORE Report 26

RP5: Indicate show raw data 26

RP6: Show raw data on reports 26

RP7: Indicate scoring method to be used. 27

RP8: Calculate data based on students who met the metric goal. 27

RP9: Calculate data based on average student scores and course weights. 27

RP10: Generate .pdf report file 27

RP11: Generate .csv report file 27

RP12: Generate CORE Report 28

RP13: Generate Course PC Report 28

RP14: Matrix Report 28

RP15: Generate Outcome Report 28

RP16: Generate PC Semester Report 29

RP17: Report view 29

5.2 Non-Functional Requirements 29

5.2.1 Design Constraints (DC) 29

5.2.2 Human Factors (HF) 31

5.2.3 External Interface Requirements (XI) 31

Hardware (HW) 31

Software (SW) 31

Communications (CM) 32

5.2.4 Security (SC) 32

5.2.5 Development Environment (DV) 32

5.2.6 Standards (ST) 32

5.2.7 Delivery Environment (DL) 32

Site (SI) 32

Operations (OP) 32

5.2.8 Performance (PR) 32

5.2.9 Deliverable Items, Dates and Conditions (DD) 32

5.2.10 Cost (CT) 33

5.2.11 Quality (QL) 33

Reliability (RL) 33

Availability (AL) 33

Maintainability (ML) 33

Usability (UL) 33

Enhanceability/Extendibility (EN) 33

Portability (PT) 33

5.2.12 V&V Activities (VV) 33

5.2.13 Database (DB) 33

5.2.14 Adaptability (AD) 33

5.3 Requirements Models 33

6. Illustrative Use Cases (IUC) 33

7. Future Enhancements (FE) 34

# 1. Introduction

## Software Purpose and Scope

The goal of this web application is to simplify and standardize how faculty members in the Computer Science (CS) and Electrical Engineering (EE) Departments at Montana Tech assess their courses in relation to ABET student outcomes. Specifically, help CS and EE faculty members determine the extent to which students in their courses have met student outcomes by streamlining the repetitive tasks which the faculty members were doing by hand. This application should make continual course assessment easier. The customers for this web application are the faculty and staff of the CS and EE department. The web application is called AbOut, for Accreditation Board of Engineering and Technology (ABET) Outcomes.

## Document Purpose and Contents

This Software Requirements Specification (SRS) describes the web application, by detailing its functionality and characteristics. This is likely to be useful to the customers who want this software, the eventual users of the software, those who develop the software, and those who test it. The customers of this web application are the same as the users of it. These people can use this SRS to learn about the web application and to clarify questions about it. Developers use this document to learn what they need to design and implement. Tester can use it to develop test cases for AbOut.

Customers sometimes find sample interfaces easier to understand than documents such as this SRS. Sample user interfaces demonstrate one way that the software could appear. This document goes further to tell precisely what functionality is needed.

This document does not attempt to tell how this software should be implemented except in those cases where the customers want the application to be developed in a particular way. Deciding exactly what a system should do, before deciding how it will do it, reduces development time considerably.

This SRS was first developed by the software engineering (SE) students in the course Requirements and Specifications (ESOF 328) in the spring semester of 2011. In the spring of 2015 it was updated to accommodate the Electrical Engineering program.

## Definitions, Acronyms, and References

The following tables of definitions, acronyms and references may be useful for reading this document.

### Definitions

|  |  |
| --- | --- |
| Administer | See “Authorized Administrator” |
| Authorized Administrator | Member in the MTECHS domain who is registered in the AbOut system and has been assigned the administrator role for one or more programs. |
| Authorized Faculty Member | Montana Tech faculty member in the MTECHS domain who is registered in the AbOut system and has been assigned to teach one or more course offerings. Faculty members may be “active” or “inactive”. |
| Authorized Observer | Anyone with an account in the MTECHS system who has been assigned the “observer” role. |
| Authorized User | Montana Tech faculty member or staff in the MTECHS domain who is registered in the AbOut system. In addition, anyone with an account in the MTECHS system who has been assigned the “observer” role. |
| CORE | Course Outcome, Review and Evaluation (CORE) |
| CORE Report | A report which faculty members in the CS Department are required to write for each course offering which they teaching in the CS or SE degree. The AbOut web application generates statistics for this report. (See CORE Statistics) |
| CORE Statistics | Report showing the extent to which student outcomes were met by students in a course offering. |
| Course | Course contributing to measuring student outcomes and offered by the CS department. Courses are identified by a prefix and number, such as *ESOF 328*. The course prefixes are CSCI and ESOF. The course has a name, such as *Software Requirements and Specifications*. |
| Course Offering | A particular section of a course offered in a particular semester. Course offerings are identified by a course, a section, and a semester. |
| Course Outcome | Criteria which students passing the course should meet. These are specific to the course and are different than ‘Student outcomes’ which are specific to either the CS or SE program. AbOut does not measure course outcomes. This term does not appear elsewhere in this document and is included here to avoid confusing this with student outcomes. |
| Course PC Report | Report showing the extent to which students met performance criteria during the semester(s) of interest. This report is divided by performance criteria and courses. |
| Default Semester | The semester used when no semester is given (for instance, creating a course offering or a new course). |
| Faculty Member | See “Authorized Faculty Member” |
| Matrix Report | Report showing the association of courses to performance criteria and the weights of those associations. This is used to get an overview of the extent to which courses are covering performance criteria. |
| Metric | A metric associated with a course offering. The metric is created by the instructor of the course. It consists of a description, maximum number of points, and a list of the student outcomes which it measures. |
| Metric Goal | The overall (percentage) score which a student needs to meet or exceed to be considered to have met the student outcomes. This needs to be stored in such a way so the metric goal can be changed easily. Throughout this document, it is assumed that the metric goal is 70%. |
| Modern Browser | Any web browser which reliably implements the latest (as of 2011) standards in HTML and CSS, with complete support for JavaScript. |
| Outcome Report | Report showing the extent to which students met the selected student outcome during the chosen semester. This report is divided by the performance criteria which measure the outcome. |
| PC Semester Report | Report showing the extent to which students met performance criteria during the semester(s) of interest. This report is divided by performance criteria and semesters. |
| Performance Criterion | A criterion which a program will use to assess a student outcome. |
| Performance Criterion Abbreviation | Abbreviation which will be used to identify an performance criterion. |
| Program | A degree program. This system is to facilitate assessment for the Computer Science, Electrical Engineering and Software Engineering programs. |
| Program Abbreviation | Abbreviation which will be used to identify a program. |
| Raw data | Raw data is the number of students, weight of the course, and percentage. |
| Registered in AbOut | AbOut contains user information for this person. |
| Semester | A Montana Tech semester. Consists of a year and either fall, spring or summer. |
| Semester(s) of Interest | A single semester or a range of semesters for which a report is being generated. |
| Simple Average | An average of each value with no weighting due to characteristics of the values. |
| Student Score | The score that a student earned on a metric. |
| Student Outcome | A criterion which a program will assess for ABET accreditation. |
| Student Outcome Abbreviation | Abbreviation which will be used to identify a student outcome. |
| User | See “Authorized User” |

|  |  |
| --- | --- |
|  |  |

### Acronyms and Abbreviations

|  |  |
| --- | --- |
| ABET | Accreditation Board for Engineering and Technology |
| CAC | Computing Accreditation Commission. |
| CAS | Central Authentication Service |
| CS | Computer Science |
| EAC | Engineering Accreditation Commission |
| IAB | Industry Advisory Board |
| MTECHS | Montana Tech Campus Network Domain |
| NM | Not Measured |
| PC | Performance Criteria |
| SE | Software Engineering |
| SSL | Secure Sockets Layer; secures transmitted data through encryption of data. |
| SRS | Software Requirements Specification |

### Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Composition or Data Type** | **Length** | **Values** |
| Outcome abbreviation | Editable identifier for an outcome | <program name>-<alphanumeric character> | 4 | CS-a, SE-1, EE-k |
| Performance Criterion abbreviation | Editable identifier for a performance criterion | <Outcome abbreviation>.<alphanumeric character> | 6 | CS-a.1, SE-1.9, EE-k.a |
| Program abbreviation | Abbreviation for a degree program | 2 characters | 2 | CS, EE, SE |

## References

ABET, <http://www.abet.org/>

CAS, <https://wiki.jasig.org/display/CAS/Home>

CS Department Student Outcomes, <http://cs.mtech.edu/main/index.php/component/content/article/146>

W3C XHTML validation software, [http://validator.w3.org](http://validator.w3.org/)

W3C CSS validation software, <http://jigsaw.w3.org/css-validator>

# 2. General Factors

## Product Perspective

This web application will be independent of other products except CAS (see Section 2.5, Dependencies).

## Product Functions

This section provides a high-level overview of the functionality of the web application.

#### Function Overview

The AbOut web application will be used to:

* Record student outcomes associated with the computer science, electrical engineering and software engineering programs
* Associate performance criteria with student outcomes
* Associate performance criteria with courses
* Associate faculty and students with course offerings
* Enable faculty to record metrics of the course offerings they teach
* Enable faculty to record the scores which a student earned on an metric
* Generate a variety of reports indicating the extent to which performance criteria were met
* Allow observers to see reports and all of the information leading to the report, with student names redacted

These functions are divided into three overlapping sets: faculty, administrative, and reporting functions.

#### Administrative Functions

Authorized faculty members and administrators are able to do the following:

* Add, edit, delete and view users of the AbOut system
* View the semesters in the system and set a default semester
* Tell the system to generate the next chronological semester
* Add, edit, delete and view student outcomes
* Add, edit, delete and view performance criteria for student outcomes
* Add, edit, delete and view courses, along with the performance criteria associated with them
* Add, edit, delete and view offerings of courses
* Add, edit, delete and view students in a course offering
* Import a list of students into a course.

Note that administrators are not able to add, edit or delete metrics associated with a course offering, and that when administrators view student scores on a metric, student names will be redacted.

If a user is both a faculty member and an administrative, that user will be able to view and edit the information associated with a course offering which they teach.

#### Faculty Functions

Authorized faculty members are able to do the following:

* View the course offerings that they are currently teaching or have taught in the past
* Add metrics to course offerings that they have taught or are teaching
* Add or remove students from course offerings that they taught or are teaching
* Enter student scores on the metrics in the course offering which they taught or are teaching
* Export a list of the students in an offering they have taught or are teaching.

#### Reporting Functions

Authorized faculty members and administrators are able to do the following.

* Generate CORE statistics showing the extent to which students enrolled in a course offering met the performance criteria associated with that offering
* Generate a Course PC Report showing the extent to which students met performance criteria per course during the semester(s) of interest.
* Generate a Matrix Report showing the weights associated with courses assessing performance criteria for a specified program. Note that this report generates the information for the current semester.
* Generate an Outcome Report showing the extent to which students met performance criteria for a semester.
* Generate a Course PC Semester Report showing the extent to which students met performance criteria per semester over the semester(s) of interest.

## Environmental Conditions

AbOut will be a web application which is accessed from the CS Department website. The CS Department will need a web and database server in order to serve AbOut.

Users will need a Montana Tech account in the MTECHS domain to access this system. They will also need to be registered within the AbOut system, i.e. AbOut needs to contain the user id for this person.

## User Characteristic

The primary users of this system are the faculty and staff of the CS Department. Someone who has a username in the MTECHS, is registered in AbOut and has been assigned the role of “observer” for one or more programs will also be a user of this system. An understanding of the assessment process, a familiarity with web browsers, and proficiency completing forms on a computer, is assumed.

Users can be assigned one or more of the roles: administrator, faculty, and observer.

While other departments at Montana Tech may find this application useful, at this time it is only being developed for the CS and EE Department.

## Dependencies

AbOut will utilize the Central Authentication Service (CAS). CAS is a single sign-on protocol for the web. CAS allows web applications to authenticate users without gaining access to a user’s security credentials.

CAS is offered by Montana Tech’s Campus Technology Services and is used to authenticate Montana Tech users for most campus applications. Without CAS, users of AbOut would need to create and remember another username / password combination to login to AbOut.

AbOut will not depend on any other systems.

## Assumptions

This application will be available on any hardware connected to the Internet and supporting a modern browser. However, developers can assume that the screen display will be that of a typical desktop or laptop, 1024x768 pixels.

Inconsistent browser support for web standards causes problems when applications are rendered on older browsers. Catering to these older browsers increases development costs. Developers of AbOut can assume that modern browsers will be used to access the application (see Section 5, SW1). Developers can also assume that JavaScript and cookies will be enabled.

# 3. Analysis Use Cases

## Actors

|  |  |  |
| --- | --- | --- |
| ***Primary Actor*** | ***Description*** | ***Use Cases*** |
| Administrator | A permission that allows users to do administrative functions for assessments. This permission is meant for administrative staff.  Examples:  Secretaries and Department heads | Add/edit/delete/view student courses  Add/edit/delete/view student outcomes  **Add performance criteria to student outcome**  Edit/delete/view performance criteria  Add/edit/delete/view users  Add/edit/delete/view course offerings  Generate next semester  Import lists of students  Set default semester  View semesters in system |
| Observer | A permission that allows the user to view data, with student names redacted. These users are not allowed to modify data. | View student outcomes  View courses  View course offerings  View metrics associated with a course offering  Generate C.O.R.E. Report  Generate Course PC Report  Generate Matrix Report  **Generate Outcome Report**  View data of a class, with student names redacted. |
| Faculty member active | A permission that allows the user limited add, edit, view, and delete permissions to offerings they are associated with. The user is also allowed to generate C.O.R.E statics. | Add/remove students from course offering  Add/edit/delete/view metric within course offering  **Add score(s) to metric**  Export list of students  View course offering  Generate C.O.R.E. statistics |
| Faculty member inactive | A permission for ex faculty members. Meant to help preserve data for historical reference. | None |

## Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: | Add Student Score(s) to Metric | | |
| Created By: | ESOF 328 students | Last Updated By: | Celia Schahczenski |
| Date Created: | 2/17/2015 | Date Last Updated: | 3/4/2015 |

|  |  |
| --- | --- |
| Actors: | Faculty member |
| Description: | A faculty member enters a student score, or multiple students’ scores, to a metric. |
| Trigger: | Faculty member has created a metric and wants to enter student scores for the metric |
| Preconditions: | 1. Faculty member is logged into AbOut 2. Faculty member is within a course offering which (s)he is teaching and has created a metric for the offering by entering a description of the metric, its maximum number of points and at least one outcome which it measures 3. There is at least one student in the course offering |
| Post conditions: | 1. The score(s) entered are stored in AbOut |
| Normal Flow: | * 1. **Enter a single score.**  1. Faculty member inputs the number of points a student earned on the metric. 2. Faculty indicates “save” 3. The percentage(s) for the score(s) have been updated 4. A message informing the user that the scores were saved is displayed |
| Alternative Flows: | **1.1 Faculty member enters multiple scores (branch before step 1)**   1. Faculty member has a list of newline separated values in the clipboard 2. Faculty member inputs these scores (return to step 2)   **1.2 User indicates “reset” to undo recent edits (branch after step 1)**   1. User indicates “reset” 2. Entered information is set back to previous values (return to step 1) |
| Exceptions: | **1.0.E.1 Illegal score is entered (after step 1)**  1a. The score is a negative number or non-numeric  1b. The score is higher than the maximum number of points  2a. The system indicates that the score is illegal and does not allow score to be saved (return to step 1)  2b. The score is set to the maximum number of points (return to step 2)  **1.0.E.2 User navigates away from page (branch after step 2)**  1. The user navigates away from page after entering one or more scores and before indicating “save”  2. System warns that entered information will be lost and allows the user to stay or go  3a. User stays (return to step 2)  2b. User goes (use case is terminated)  **1.1.E.1 Number of scores do not match the number of students in the offering (after step 2)**  1. The system indicates that the number of scores do not match the number of students and no scores are recorded (return to step 1) |
| Includes: | None |
| Priority: | High |
| Frequency of Use: | Number of metrics\*number of students\*number of offerings each semester. |
| Business Rules: | Metric scores cannot exceed the maximum number of points |
| Special Requirements: | None |
| Assumptions: | * For entering multiple grades at once, the order of student names in AbOut matches the order of the names in the grade source * Multiple grades will be entered via the clipboard |
| Notes and Issues: | * This use case corresponds to requirement FC7 |

# 4. Explanatory Use Cases

This document does not contain any explanatory use cases.

# 5. Specific Requirements

This section provides details of the AbOut system. The functional requirements describe the functionality that AbOut must provide. The non-functional requirements give other characteristics of the AbOut system.

## Functional Requirements

The following requirements describe the functionality that AbOut will provide.

### System (SM)

#### SM1: AbOut login

The AbOut system will use CAS to authenticate users. The user shall input a Montana Tech login name and password into CAS. If the credentials are accepted by CAS and if the user is registered in AbOut, the user is authorized to use AbOut.

*Rationale:* Login exists for security reasons, to protect sensitive data and data change. At this time only faculty and staff of the CS department will use AbOut.

#### SM2: AbOut logout

Users who are logged into the AbOut system can log out at any time that there isn’t an action which must be attended to.

*Rationale:* It should be easy for users of the system to quickly disconnect from AbOut.

#### SM3: Automatic logout

The AbOut system shall automatically logout users who have been inactive for more than 3 minutes.

*Rationale:* Users may be called away when working with AbOut and they should not remain logged in while they are not actually using the software.

### Administrator (AM)

Authorized faculty members are able to perform all administrative functions.

#### AM1: Add user information

Administrators shall be able to add a user who is authorized to use AbOut. The user information consists of the MTECHS username, the person’s name, the role or roles the person can play, and the program or programs the person is associated with.

*Rationale:* New users will need access to AbOut.

#### AM2: Edit user information

Administrators shall be able to edit the name, roles, and programs for users who are authorized to use AbOut. If a faculty member’s name is changed, that change does not permeate to the previous course offerings that the faculty member taught. The new name will be used for any later course offerings.

*Rationale:* Users may change their name and status as faculty members. MTECHS user names shall not be editable since these serve to connect users to course offerings.

#### AM3: Delete user information

Administrators shall be able to delete users who are no longer authorized to use AbOut. If a user is deleted, metrics associated with offerings taught by that user will no longer be accessible. The offering itself, will stay in the system, along with any scores earned by the students. The instructor name associated with the offering will stay the same. An alternative to deleting a faculty member, is to make the faculty member inactive.

*Rationale:* Faculty and staff may leave the department and administrators may want to remove them from the system.

#### AM4: View user information

Administrators shall be able to a view a list of all users of the system sorted alphabetically by last name.

*Rationale:* Administrators may need to view a list of all users registered with the AbOut system.

#### AM5: Add semester

Administrators shall be able to generate the next chronological semester and set it to the default semester if desired. (See “Change default semester” for changing the default semester outside of adding a semester)

*Rationale:* Semesters must be generated in order. It is highly likely that the new semester will made the default semester.

#### AM6: View all semesters

Administrators shall be able to a view a list of all semesters in the system, ordered from the latest (most recent) semester to the earliest.

*Rationale:* Administrators need to view the semesters that have been created in AbOut.

#### AM7: Default semester

At all times, AbOut shall have a default semester.

*Rationale:* Most AbOut operations are dependent on a semester. Pre-populating fields with the default semester may save time.

#### AM8: Change default semester

Administrators shall be able to change the default semester.

*Rationale:* As new semesters are created, the system’s default semester will need to be changed. Changing the default semester is not done automatically by the system since the semester may be created before the date it goes into effect.

#### AM9: Administrative view

When an authorized administrator logs into AbOut, the administrative functions shall be displayed and easily accessible.

*Rationale:* Displaying faculty, administrator, and report functions separately may make the system easier to use. Additionally, this provides a stable default view for administrators.

#### AM10: Add student outcome

Administrators shall be able to add a student outcome to the AbOut system. Student outcome information includes an abbreviation, textual description, the program area to which it refers (CS, SE, or EE), the semester when the outcome came into effect, and the semester when the outcome is no longer in effect.

*Rationale:* New student outcomes may need to be measured and the system will need to know when these new student outcomes came into effect and when it is no longer in effect.

#### AM11: Edit student outcome

The system shall allow administrators to edit student outcomes.

*Rationale:* Student outcomes will change over time. Allowing administrators to easily accommodate those changes will increase system usability and consistency.

#### AM12: Delete student outcome

Administrators shall be able to delete student outcomes from the AbOut system if that student outcome has no performance criteria associated with it.

*Rationale:* Student outcomes may be mistakenly added to the system and administrators need to be able to clean up the system. A student outcome which has no performance criteria associated with it would not have contributed useful information to a report and can be deleted.

#### AM13: View outcomes

Administrators shall be able to a view all student outcomes in the system including past student outcomes.

*Rationale:* Administrators must be able to view all outcomes that have been created in AbOut.

#### AM14: Add Performance Criterion

Administrators shall be able to add a performance criterion to an outcome. Performance criterion information includes an abbreviation, textual description, the semester when the performance criterion came into effect, and the semester when the performance criterion is no longer in effect

*Rationale:* New performance criteria will need to be measured, and the system will need to know when each criterion came into effect and when it is no longer in effect.

#### AM15: Edit Performance Criterion

The system shall allow administrators to edit performance criteria.

*Rationale:* Performance criteria will change over time. Allowing administrators to accommodate those changes will increase system usability and consistency.

#### AM16: Delete Performance Criterion

Administrators shall be able to delete performance criteria from the AbOut system if that criterion has no metrics measuring it.

*Rationale:* Performance criteria may be mistakenly added to the system and administrators need to be able to clean up the system. A performance criterion which has no courses associated with it would not have contributed useful information to a report and can be deleted.

#### AM17: View performance Criteria

Administrators shall be able to view all performance criteria in the system, including past performance criteria.

*Rationale:* Administrators must be able to view all performance criteria that have been created in AbOut.

#### AM18: Add course

Administrators shall be able to add courses to the AbOut system. Course information includes an abbreviation, course name, the semester when the course became part of the curriculum, and the semester when the course is no longer part of the curriculum. Administrators can also associate performance criteria with a course (see “Associate or disassociate a performance criterion with a course”).

*Rationale:* In the future, courses may be added to a department’s curriculum and AbOut needs to be able to accommodate the new courses.

#### AM19: Edit course

The system shall allow administrators to edit a course.

*Rationale:* Courses will change over time. Allowing administrators to easily accommodate those changes will increase system usability and consistency.

#### AM20: Delete course

Administrators shall be able to delete a course from the AbOut system if there are no course offerings associated with that course.

*Rationale:* Courses may have been mistakenly added to the system and administrators need to be able to clean up the system.

#### AM21: View courses

Administrators shall be able to a view all courses in the system including past courses.

*Rationale:* Administrators may need to view a list of all courses that have been created in AbOut.

#### AM22: Associate or dissociate a performance criterion with a course

Administrators shall be able to associate or dissociate a performance criterion with a course. The system shall present a list of all active performance criteria associated with the programs of the course to the administrator.

*Rationale:* The performance criteria associated with a course will change through time.

#### AM23: Add course offering

Administrators shall be able to add a course offering to the AbOut system. The administrator shall select a semester, or use the default semester. (S)he shall also select a course, a section number, and the faculty member who will teach this course offering. The name of the faculty member shall be copied into this course offering. The performance criteria that this course offering will measure shall be automatically connected to this course offering. Note that the performance criteria associated with this course offering cannot be changed here. In order to change which performance criteria are associated with this offering, an administrator must delete this offering, make the new associations between the course and performance criteria, and recreate the offering.

*Rationale:* Every semester new offerings of courses will be created. The name of the faculty member will be copied into this offering so that if the faculty member later changes his/her name, the change will not be reflected in this offering. If changes are made to the text of the performance criteria, these changes will be reflected in later reports.

#### AM24: Edit course offering

Administrators shall be able to edit the section number and user (current faculty member who will be teaching the course) associated with a course offering. Students can also be associated with this course offering (see AM27 and AM28).

*Rationale:* Course offerings may need to change. The semester, course and outcomes associated with the offering cannot be changed. The semester and course are intrinsic to the offering. The outcomes are a function of the course. The system will be more consistent if the outcomes associated with the course offering cannot be changed.

#### AM25: Delete course offering

Administrators shall be able to delete a course offering from the AbOut system if there are no metrics associated with this offering.

*Rationale:* Course offerings may have been mistakenly added to the system and administrators need to be able to clean up the system.

#### AM26: View all course offerings

Administrators shall be able to a view all courses offerings in the system, ordered by the latest offerings, alphabetically by the course abbreviation, to the earliest course offerings.

*Rationale:* Administrators may need to see a list of all courses offerings.

#### AM27: Import a list of students into a course offering

Administrators shall be able to import a list of students into a course offering by providing the file name of a comma-separated list of students.

*Rationale:* Entering student information one by one could be time consuming and the information is likely to be available in a comma-separated list.

#### AM28: Add student to a course offering

Administrators shall be able to add students to the course offering. If the course offering already contains metrics and student scores associated with the metric, the new student scores will default to zero and the user shall be informed that the added students are getting a zero for this metric.

*Rationale:* Students may join the course at a later date or the administrator or the faculty member may decide to add students individually to the course offering.

#### AM29: Delete a student from a course offering

Administrators shall be able to remove a student from the course offering, causing all metric scores for that student to be removed. If the course offering had one or more metric, with a score for the student being dropped, the user shall be informed that the deleted student’s score is being lost.

*Rationale:* Students may drop the course, students may be taking the course offering who are not CS or SE majors so do not need to be measured, and failing student scores may need to be removed from the metrics for that course offering.

### Faculty (FC)

#### FC1: Semester selection

Users shall be able to select a semester different from the default semester for which operations will be performed. This will not modify the default semester.

*Rationale:* Periodically users will need to perform operations on semesters other than the default semester.

#### FC2: Export a list of students from a course offering

Faculty members shall be able to export a list of students from a course offering which they teach by providing a path to which a comma-separated list of students will be written.

*Rationale:* Exporting the list of students in an offering may be useful to the faculty member in creating their grading sheet.

#### FC3: Basic Characteristics of Offerings

Once a faculty member is working with a course offering, the course, semester, section and students outcomes of the offering cannot be changed.

*Rationale:* The course, semester and section are inherent in the offering and changing one of these essentially creates a new offering. The student outcomes cannot be changed because outcomes student outcomes are associated with the course. If student outcomes are to be changed, they must be changed at the course level. The current offering would need to be deleted, the outcomes of the course changed and the offering re-created.

#### FC4: Add/remove students from course offering

Faculty shall be able to add and remove students associated with a course offering which they teach

*Rationale:* Students may add the course, drop the course, or not be in the program for which metrics are being collected.

#### FC5: Add metric to a course offering

The faculty member teaching a course offering shall be able to add a metric to the offering. Metric information includes a short textual description of the metric item, the program that the metric will be associated with, the maximum number of points that a student can achieve on the item, and one or more student performance criteria to be measured by the metric. The available performance criteria to be measured will be limited to only criteria that are associated with the program that the metric is to be associated with.

*Rationale:* Metrics are needed to determine how well students perform on student outcomes.

#### FC6: Edit metric

The faculty member teaching a course offering shall be able to edit the description, maximum number of points, and list of student outcomes which this metric will measure. The student outcomes being measured must be associated with the program that the metric is associated with.

*Rationale:* Faculty members may reconsider a metric description and the student outcomes it measures, and should be able to modify these. They may have mistakenly entered the wrong number of points for the outcome and should be able to fix their mistake.

#### FC7: Delete metric

The faculty member teaching a course offering shall be able to delete a metric from that course offering. If a metric is deleted, all student scores associated with that metric will be deleted.

*Rationale:* Metrics may be mistakenly added to the system and the faculty member teaching the offering must be able to clean up the system.

#### FC8: Add student score to metric

The faculty member teaching a course offering shall be able to add a student score to a metric. Student scores must be in the range of 0 to the maximum number of points for the metric. If a score is not entered for a particular student, the score will default to 0.

*Rationale:* Student scores are needed since they are what enable the system to report the extent to which student outcomes are met.

#### FC9: Edit student score on metric

The faculty member teaching a course offering shall be able to edit a student score on a metric provided the new score is within the range of 0 to the maximum number of points for the metric. If the score is set to anything else (blank for instance) it will default to 0.

*Rationale:* Student scores may need to be changed due to data entry mistakes.

#### FC10: View extent to which students met student outcome

The faculty member teaching a course offering shall be able to view the extent to which students in the offering met the student outcomes associated with that course, as percentages. If scores have not yet been entered for any metric(s) which measures a student outcome, the percentage will be 0%.

*Rationale:* As student scores are entered for metrics, faculty members will want to know the extent to which students met the student outcomes.

#### FC11: Faculty view

When an authorized faculty member logs into AbOut, the offerings that the faculty member is teaching or has taught, shall be displayed and the faculty functions easily accessible.

*Rationale:* Displaying faculty, administrator and report functions separately may make the system easier to use. Additionally, this enables faculty members to easily access the courses they teach or have taught.

### Reports (RP)

#### RP1: Indicate semester for the Outcome Report

Users shall be able to indicate the semester of interest for the Outcome Report.

*Rationale:* The Outcome Report displays information relevant to a single semester.

#### RP2: Indicate semester(s) for Course PC and PC Semester Reports

Users shall be able to indicate the semester(s) of interest for which a report is to be generated. They may indicate a single semester or provide a start and end semester for the report.

*Rationale:* All of these reports are relevant to a semester or a range of semesters.

#### RP3: Indicate program(s) for Course PC, Matrix, and PC Semester Reports

Users shall be able to indicate the program(s) for which the report is to be generated and whose student’s scores will be used in calculating data.

*Rationale:* Reports for individual programs are needed when reporting to ABET. However, combined reports might be desired when the CS Departments reports to their advisory board.

#### RP4: Indicate course offering for CORE Report

Users shall be able to indicate the course offering for which CORE statistics are to be generated. To select the course offering, users indicate the course, semester and section of the offering.

*Rationale:* In order to generate CORE statistics for an offering, the offering must be identified.

#### RP5: Indicate show raw data

Users shall be able to indicate that raw data is to be shown on any report except the Matrix Report. Raw data is the number of students, weight of the course, and percentage.

*Rationale:* Users may want the raw data shown, or they may want a less crowded report.

#### RP6: Show raw data on reports

For all reports, excluding the Matrix Report, the system shall be able to display the raw data (number of students in the course and the weight of the course for its performance criterion), in addition to the corresponding percentage representation.

*Rationale:* Users may want to view the raw data for analytical purposes.

#### RP7: Indicate scoring method to be used.

Users shall be able to indicate the method required for the report to calculate its data.

*Rationale:* Different departments use different formulas for assessing performance, requiring reports to be able to use these different formulas for calculating data.

#### RP8: Calculate data based on students who met the metric goal.

The system shall implement a method of calculations which provide scores that represent the percentage of students who earned the metric goal or higher.

*Rationale:* The CS/SE department assesses their performance through analyzing the percentage of students that had scores greater than a certain cutoff line.

#### RP9: Calculate data based on average student scores and course weights.

The system shall implement a method of calculations which provide scores that represent weighted average student scores.

*Rationale:* The EE department assesses their performance by analyzing weighted student averages.

#### RP10: Generate .pdf report file

The system shall be able to generate PDF files for all report types. This file will contain the entirety of the report. The filename for the .pdf file will be the report name, along with the course, program, or outcome that is the focus of the report.

*Rationale:* Users will be using these reports in a variety of circumstances for which a .pdf file is helpful.

#### RP11: Generate .csv report file

The system shall be able to generate a Comma Separated Value (.csv) file for all report types. This file will contain the analyzable subset of the report.

*Rationale:* Users may want to analyze the report data using another application.

#### RP12: Generate CORE Report

Users shall be able to prompt the system to generate statistics for a CORE Report which displays the score of each performance criteria associated with a course offering. The text of each student outcome associated with the course offering shall be displayed in the statistics. If there are no student scores for any metric of an outcome, the percentage shall be 0.

*Rationale:* Faculty members will be able to attach the generated page into their CORE Reports, which will save them time and may make the reports more accurate. Generating statistics for the CORE reports are one of the tedious, repetitive tasks which AbOut was created to alleviate.

#### RP13: Generate Course PC Report

Users shall be able to prompt the system to generate a Course PC Report which displays the scores of performance criteria vs. courses. If there are no student scores for any metric of an outcome, the percentage shall be 0. The report shall also display a summary across all performance criteria, for each course, and a summary, across all courses, for each performance criteria.

*Rationale:* Providing a Course PC Report will allow users, ABET evaluators, and IAB members to look more deeply at what each course contributed to the extent that performance criteria were met during a semester range.

#### RP14: Matrix Report

Users shall be able to generate a Matrix Report which displays, by courses and outcomes, which outcomes each course assesses. If a course assesses a given outcome, the report will denote it with a value representing the course’s weight, while courses that do not will be unmarked. The report shall also display a sum across all courses for each performance criteria, and a sum across all performance criteria for each course.

*Rationale:* Providing a Matrix report will allow AbOut users to quickly view which courses are responsible for assessing individual outcomes when planning to add or remove outcomes from a course.

#### RP15: Generate Outcome Report

Users shall be able to prompt the system to generate an Outcome Report which displays the score of performance criteria associated with the chosen outcome vs. courses for the chosen semester. If there are no student scores for any metric of an outcome, the percentage shall be 0. The report shall also display a summary, across all performance criteria, for each semester, and a summary, across all semesters, for each performance criteria.

The report shall contain a list of any student outcome which was active during a portion of the indicated semester range, but not during all of it.

*Rationale:* Providing an Outcome Report will allow AbOut users, ABET evaluators and IAB members to quickly see the extent to which each performance criterion was met during a semester.

#### RP16: Generate PC Semester Report

Users shall be able to prompt the system to generate a PC Report which displays the scores of semesters versus performance criteria within the range of semesters indicated. If there are no student scores for any metric of an outcome, the percentage shall be 0. The report shall also display a summary across all semesters, for each performance criteria.

The report shall contain a list of any performance criteria which was active during a portion of the indicated semester range, but not during all of it.

*Rationale:* Providing an PC Semester Report will allow AbOut users, ABET evaluators and IAB members to quickly see the extent to which each performance criteria was met during a semester range.

#### RP17: Report view

The interface for generating statistics and reports shall be separated from the administrative and faculty functions.

*Rationale:* Displaying faculty, administrator and report functions separately may make the system easier to use.

## Non-Functional Requirements

While the functional requirements detail the functions which the system can perform, the non-functional requirements describe characteristics of the system. These characteristics typically apply to the entire system.

### Design Constraints (DC)

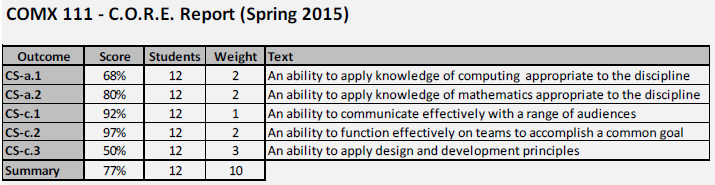
DC1: This application is to be developed using MySQL, PHP and Javascript.

DC2: It must be easy to change the value of the metric goal. A programmer should only need to change the code in one place, and all reports will be generated using the new metric goal.

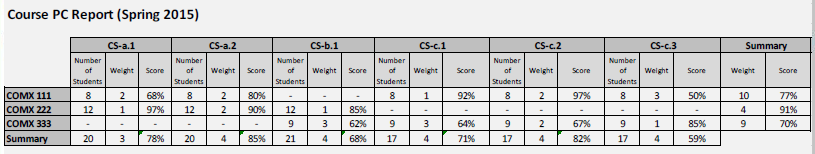
DC3: When a faculty member logs into AbOut, a list of the course offerings the faculty member is teaching or has taught shall be displayed in reverse chronological order by semester. If the faculty member is not assigned to teach any offerings and has never taught any offerings, the system shall state this clearly. Administrative functions shall be accessible from this Faculty View, but with less prominence.

DC4: When an administrator logs into AbOut the Administrative functions shall be prominent.

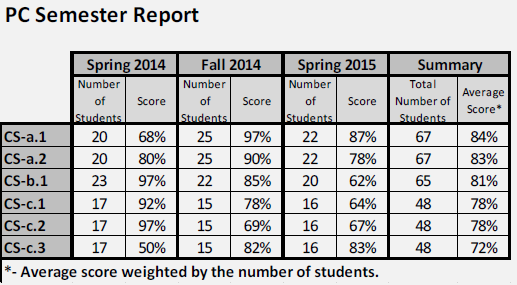
DC5: C.O.R.E. Report shall appear similar to the following:



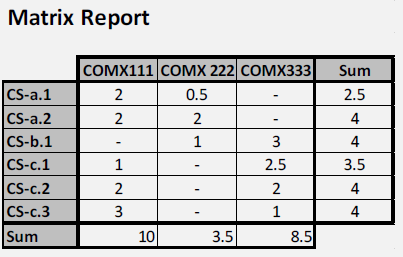
DC6: Course PC Report shall appear similar to the following:



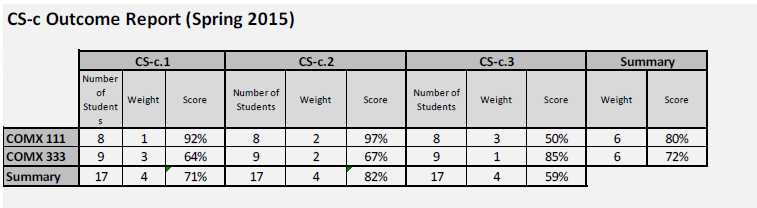
DC7: PC Semester Report shall appear similar to the following:



DC8: Matrix Report shall appear similar to the following:



DC9: Outcome Report shall appear similar to the following:



### Human Factors (HF)

No provisions related to human factors are required by this application.

### External Interface Requirements (XI)

#### Hardware (HW)

HW1: The application is accessible on any hardware connected to the Internet which supports one or more of the browsers listed below (SW1).

#### Software (SW)

SW1: The application must render correctly on the following browsers: Firefox 3 and Google Chrome 4. It is expect that as new versions of browsers become the norm, the system will be updated to run on those versions.

#### Communications (CM)

CM1: Connection to CAS is required.

### Security (SC)

SC1: The application shall only be accessible to MTECHs users within the AbOut system.

SC2: The application shall not contain Montana Tech student numbers. It may contain student names and will contain assignment, project and/or test question scores.

SC3: AbOut must comply with FERPA regulations.

### Development Environment (DV)

No provisions related to the development environment are required by this application.

### Standards (ST)

ST1: All HTML code adheres to XHTML 1.0 Strict.

ST2: All CSS code adheres to CSS level 2.1

ST3: A set of coding standards will be used so that the format and character of the code is consistent. These coding standards shall include the W3C standards (<http://www.w3.org/standards/>) for web-content development.

### Delivery Environment (DL)

#### Site (SI)

No site requirements are placed on the application.

#### Operations (OP)

No operations requirements are placed on the application.

### Performance (PR)

PR1: All pages of this application must load on average within 6 seconds when accessed from a computer connected to the campus network. Measurements will be taken using the iWebTool Speed Test ([www.iwebtool.com](http://www.iwebtool.com/), website speed test).

### Deliverable Items, Dates and Conditions (DD)

DD1: A current version of this SRS, design document, test document, and maintenance manual will be delivered with the application.

### Cost (CT)

No cost requirements are placed on the application.

### Quality (QL)

#### Reliability (RL)

No reliability requirements are placed on the application.

#### Availability (AL)

No availability requirements are placed on the application.

#### Maintainability (ML)

No specific maintainability requirements are placed on this application.

#### Usability (UL)

UL1: No faculty or staff member spends longer than 10 minutes figuring out how to complete a task of the application.

UL2: The user shall be able to exit the system at any time.

#### Enhanceability/Extendibility (EN)

No specific enhanceability/extendibility requirements are placed on this application.

#### Portability (PT)

No specific portability requirements are placed on this application.

### V&V Activities (VV)

VV1: As future users of this system will be readily available, the application must be validated by users a minimum of two times during application development.

### Database (DB)

DB1: MySQL shall be used for this application.

### Adaptability (AD)

No changes to the above non-functional requirements are expected.

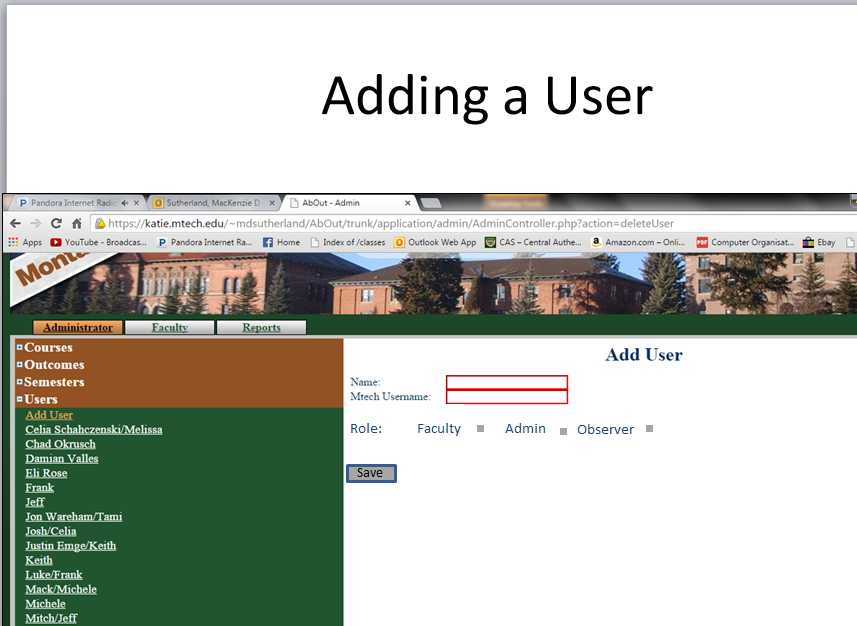
## Requirements Models

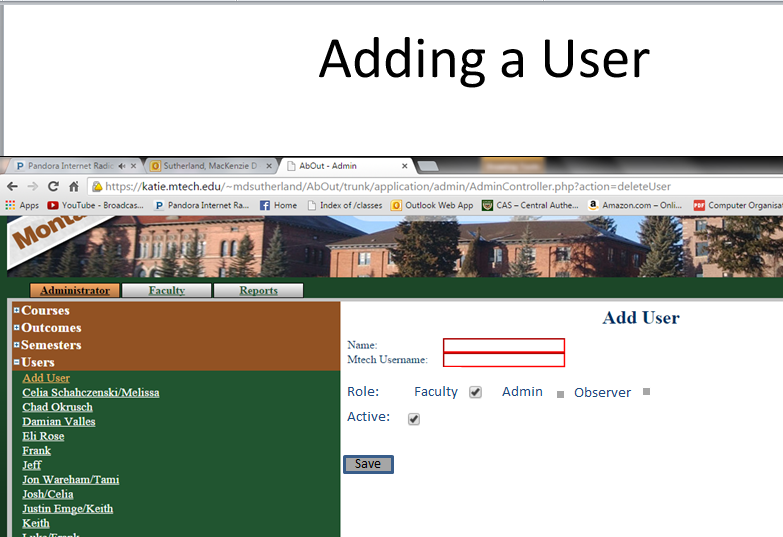
No requirements models are included in this specification.

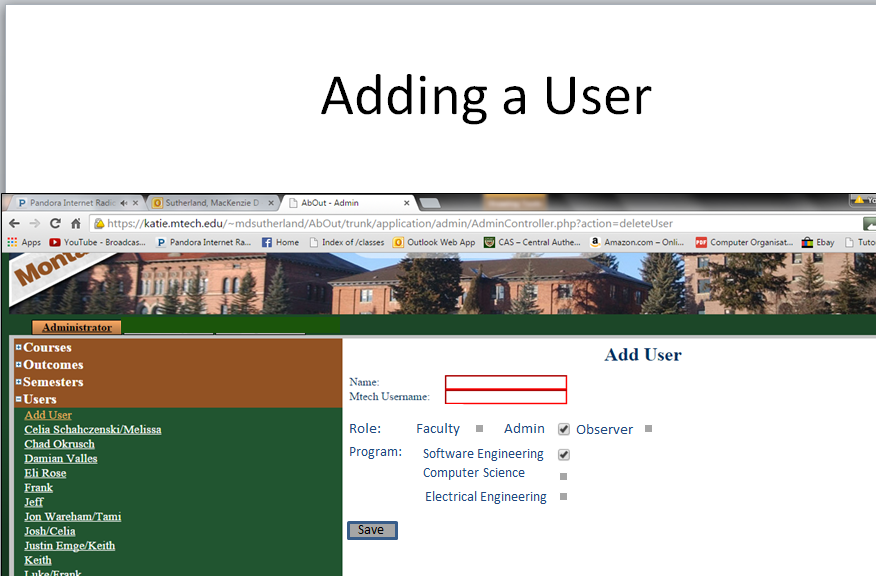
# 6. Sample User Interface

The following screen demonstrate possible screens for the enhanced system.

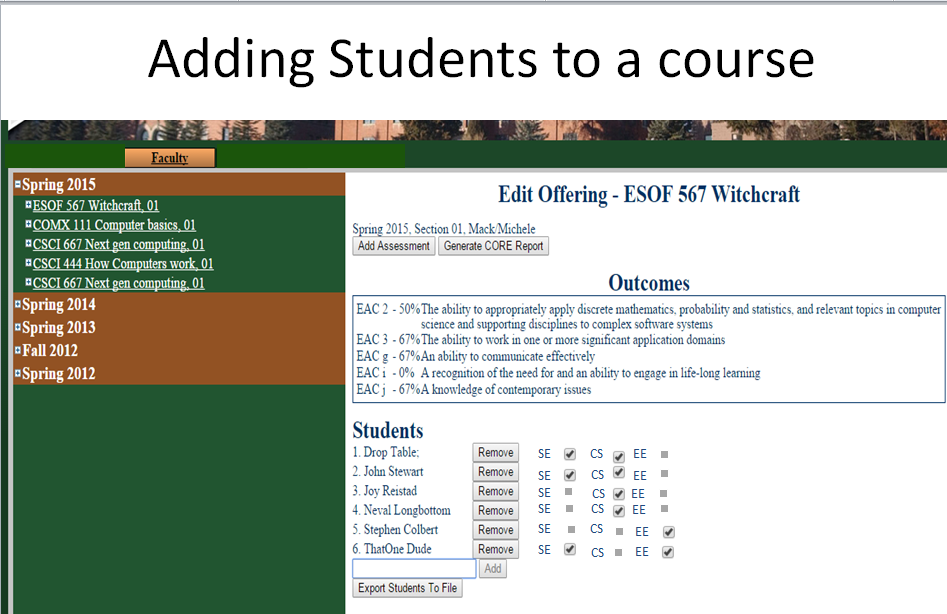
## Add User Screens

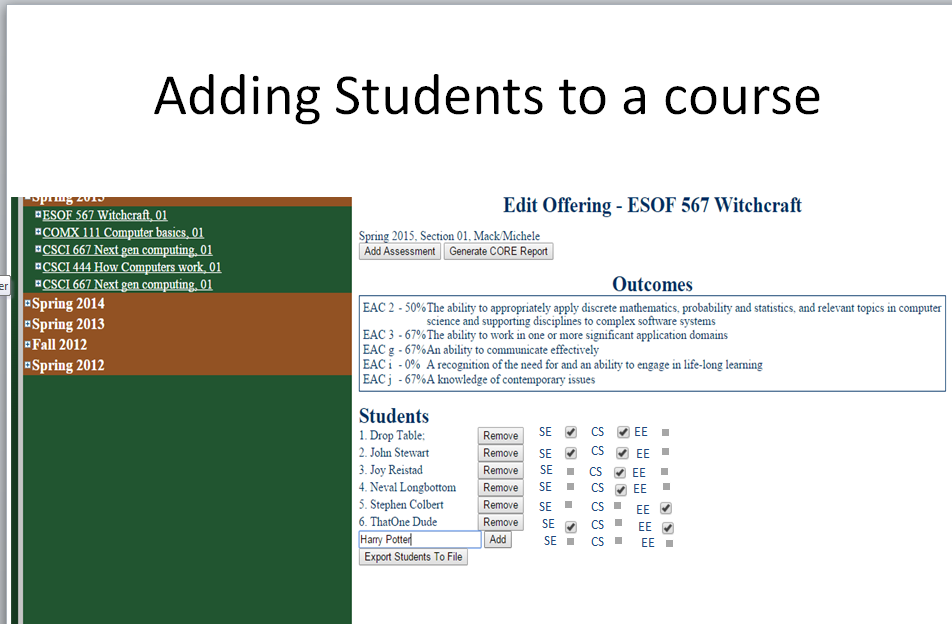






## Add Student to a Course





# 7. Future Enhancements (FE)

This product could be modified so that other Montana Tech programs which are accredited by ABET can use it. This product could also be modified to allow the users to upload scanned files that pertain to the metrics so digital copies of the material can be easily accessible for review and preserved.