AbOut
MTM Program Product
Software Requirements
Specification

Version 3.5
5/1/2018

Applying MTM SRS Version 3.4

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.
3.5	010518	ESOF 326 Students	Updated based on work done in ESOF 326, spring 2018.
3.59	091319	ESOF322 Students	Updated functional requirements for Login and Admin based on inspection by students, ESOF 322, fall 2019.
3.6	100619	ESOF 322 Students	Updated functional requirements for Faculty and Reports based on inspection by students, ESOF 322, fall 2019.

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

INTRODUCTION.. 7

1.1 SOFTWARE PURPOSE AND SCOPE 7
1.2 DOCUMENT PURPOSE AND CONTENTS 7
1.3 DEFINITIONS, ACRONYMS, AND REFERENCES 8
1.3.1 Definitions.. 8
1.3.2 Acronyms and Abbreviations 9
1.3.3 Technical Definitions/Data Dictionary.. 10
1.4 REFERENCES 10

GENERAL FACTORS.. 10

1.5 PRODUCT PERSPECTIVE 10
1.6 PRODUCT FUNCTIONS 10
Function Overview.. 10
Administrative Functions.. 10
Faculty Functions.. 11
Reporting Functions.. 11
1.7 ENVIRONMENTAL CONDITIONS 11
1.8 USER CHARACTERISTIC 12
1.9 DEPENDENCIES 12
1.10 ASSUMPTIONS 12

USER INTERFACES 12

SPECIFIC REQUIREMENTS.. 13

1.11 FUNCTIONAL REQUIREMENTS 13
1.11.1 System (SM) 13
SM1: AbOut login.. 13
SM2: AbOut logout.. 13
SM3: Automatic logout.. 13
1.11.2 Administrator (AM).. 13
AM1: Add user information.. 13
AM2: Edit user information.. 14
AM3: Delete user information 14
AM4: View user information.. 14
AM5: Add semester 14
AM6: View all semesters.. 15
AM7: Default semester 15
AM8: Change default semester.. 15
AM9: Administrative view 15
AM10: Add student outcome.. 16
AM11: Edit student outcome 16
AM12: Delete student outcome 16
AM13: View outcomes 16
AM14: Add course.. 17

AM15: Edit course 17
AM16: Delete course 17
AM17: View courses 17
AM18: Associate or dissociate outcomes with course .... **Error! Bookmark not defined.**
AM19: Add course offering 18
AM20: Edit course offering 18
AM21: Delete course offering 19
AM22: View all course offerings 19
AM23: Import a list of students into a course offering 19
AM24: Add student to a course offering 19
AM25: Delete a student from a course offering 19
1.11.3 Faculty (FC) 20
FC1: Semester selection 20
FC2: Export a list of students from a course offering 20
FC3: Add/remove students from course offering 20
FC4: Add assessment to a course offering 20
FC5: Edit assessment 20
FC6: Delete assessment 21
FC7: Add student score to assessment 21
FC8: Edit student score 21
FC9: View extent to which students met student outcome 21
FC10: Faculty view 21
1.11.4 Reports (RP) 22
RP1: Indicate semester(s) for Overview Course, Outcome, and Overview Outcome Reports 22
RP2: Indicate program(s) for Overview Course, Outcome and Overview Outcome Reports 22
RP3: Indicate course offering for CORE Report 22
RP4: Indicate show raw data 22
RP5: Show raw data on reports 22
RP6: Generate .pdf report file 23
RP7: Generate .csv report file 23
RP8: Generate CORE Report 23
RP9: Generate Overview Course Report 23
RP10: Generate Outcome Report 24
RP11: Generate Overview Outcome Report 24
RP12: Matrix Report 24
RP13: Report view 25
1.12 NON-FUNCTIONAL REQUIREMENTS 25
1.12.1 Design Constraints (DC) 25
1.12.2 Human Factors (HF) 27
1.12.3 External Interface Requirements (XI) 27
Hardware (HW) 27
Software (SW) 28
Communications (CM) 28

1.12.4 Security (SC) ... 28
1.12.5 Development Environment (DV) ... 28
1.12.6 Standards (ST) ... 28
1.12.7 Delivery Environment (DL) ... 28
Site (SI) ... 28
Operations (OP) ... 28
1.12.8 Performance (PR) ... 28
1.12.9 Deliverable Items, Dates and Conditions (DD) ... 28
1.12.10 Cost (CT) ... 29
1.12.11 Quality (QL) ... 29
Reliability (RL) ... 29
Availability (AL) ... 29
Maintainability (ML) ... 29
Usability (UL) ... 29
Enhanceability/Extendibility (EN) ... 29
Portability (PT) ... 29
1.12.12 V&V Activities (VV) ... 29
1.12.13 Database (DB) ... 29
1.12.14 Adaptability (AD) ... 29
1.13 REQUIREMENTS MODELS ... 29
FUTURE ENHANCEMENTS (FE) ... 30

Introduction

1.1Software Purpose and Scope

The goal of this web application is to simplify and standardize how faculty members in the Computer Science (CS) Department at Montana Tech assess their courses. Specifically, this software helps CS faculty members determine the extent to which students in their courses have met accreditation outcomes. In most cases, the software streamlines 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 department. The web application is called AbOut, for Accreditation Board of Engineering and Technology (ABET) Outcomes.

1.2Document 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 developed by the software engineering (SE) students in the course Requirements and Specifications (ESOF 328) in the spring semester of 2011.

1.3Definitions, Acronyms, and References

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

1.3.1 Definitions

Administer	See "Authorized Administrator"
Assessment	An assessment associated with a course offering. The assessment 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.
Authorized Administrator	Montana Tech staff member in the MTECHS domain who is registered in the AbOut system.
Authorized Faculty Member	Montana Tech faculty member in the MTECHS domain who is registered in the AbOut system.
Authorized User	Montana Tech faculty member or staff in the MTECHS domain who is registered in the AbOut system.
Assessment 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 assessment goal can be changed easily. Throughout this document, it is assumed that the assessment goal is 70%.
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.
Overview Course Report	Report showing the extent to which students met the student outcomes during the semester(s) of interest. This report is divided by student outcomes and courses. It is similar to the Outcome Report only that report divides the information by semesters rather than 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 and outcomes. It is used by faculty members for purposes other than that of presentation to accreditation committees or the IAB.
Modern Browser	Any web browser which reliably implements the latest (as of 2011) standards in HTML and CSS, with complete support for JavaScript.
Outcome Abbreviation	Uniquely identifies a student outcome. Begins with prefix CAC or EAC followed by a letter or number.
Outcome Report	Report showing the extent to which students met the selected student outcome during the semester(s) of interest. This report is divided by courses and semesters.
Overview Outcome Report	Report showing the extent to which students met the student outcomes during the semester(s) of interest. This report is divided by student outcomes and semesters. It is similar to the Course Report only that report divides the information by courses rather than semesters.
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 an assessment.
Student Outcome	A program criterion defined by the CS department. These are identified by a prefix followed by a letter or number. The prefixes are CAC or EAC. Each student outcome has descriptive text.
User	See “Authorized User”

1.3.2 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
SE	Software Engineering
SRS	Software Requirements Specification

1.3.3 Technical Definitions/Data Dictionary

Item Name	Type	Brief description of data item
SSL	Protocol	Secure Sockets Layer; secures transmitted data through encryption of data.

1.4References

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

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

General Factors

1.5Product Perspective

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

1.6Product 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:

- associate ABET student outcomes with specific CS and SE courses
- associate faculty and students with courses
- enable faculty to record assessments of the course offerings they teach
- enable faculty to record the score which a student earned on an assessment
- generate a variety of reports indicating the extent to which the outcomes were met

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 outcomes
- Add, edit, delete and view courses, along with the student outcomes 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 access the assessments associated with a course offering. Only faculty members can add, edit, delete and view assessments.

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 assessments 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 assessments in the course offering which they taught or are teaching
- Export a list of the students in an offering they have taught or are teahing.

Reporting Functions

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

- Generate a Matrix report showing the mappings between courses and the outcomes that those courses assess, for the selected program (CAC and/or EAC).
- Generate CORE statistics showing the extent to which students enrolled in a course offering met the outcomes associated with that offering.
- Generate an Outcome Report showing the extent to which students met a student outcome during the semester(s) of interest. This report is divided by courses and semesters.
- Generate an Overview Outcome Report showing the extent to which students met the student outcomes during the semester(s) of interest. This report is divided by student outcomes and semesters.
- Generate an Overview Course Report showing the extent to which students met the student outcomes during the semester(s) of interest. This report is divided by student outcomes and courses.

1.7Environmental 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.

1.8User Characteristic

The primary users of this system are the faculty and staff of the CS Department. An understanding of the assessment process, a familiarity with web browsers, and proficiency completing forms on a computer, is assumed.

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

1.9Dependencies

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.

1.10 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.

User Interfaces

This document originally contained sample user interface screens, however, once development started, multiple user interfaces were trialed and an interface entirely different from what was given in this document was chosen. Thus, that sample interface was removed from this document.

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.

1.11 Functional Requirements

The following requirements describe the functionality that AbOut will provide.

1.11.1 System (SM)

SM1: AbOut login

The user shall input a Montana Tech login name and password to enter the AbOut system. The login name and password shall be verified by 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: To keep sensitive information secure, users must be allowed to log out of the system, as long as the user isn't in the middle of an operation.

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 when they are not actually using the software.

1.11.2 Administrator (AM)

Authorized faculty members are able to perform all administrative functions. Administrative functions are adding, updating, removing and listing users, semesters, outcomes, courses and offerings, and adding students to courses.

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, and an indicator if the person is an administrator or a faculty member.

Rationale: New users may join the department and need access to AbOut. A list of current faculty is needed when the administrator creates new course offerings.

AM2: Edit user information

Administrators shall be able to edit the name that AbOut associates with a user, the MTECHS username and the role (active or not). When the name that AbOut associates with a user is changed, the new name will appear in subsequently created course offerings. Users will only be allowed to log into AbOut if the MTECHS username that AbOut associates with the user matches the users name in the MTECHS domain. The role of a user can be active or inactive. Active users are assumed to be faculty members currently teaching courses. Inactive users are faculty members that are no longer teaching courses, or administrators.

Rationale: It must be possible to edit user information since that information may have been entered incorrectly. Also, the real world name, MTECHS name, or role of a faculty member may have changed.

Note: The notion of active/inactive is too simplistic and will be changed in the future. Faculty members no longer working at Tech, shall no longer be considered administrators.

AM3: Delete user information

Administrators shall be able to delete users who are no longer authorized to use AbOut. If a user is deleted, assessments 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.

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 in ascending alphabetical order by first name.

Rationale: Administrators may need to know all users which AbOut knows about.

AM5: Add semester

Administrators shall be able to tell the system to generate the next chronological semester. The system shall create the semester and ask the user if the newly created semester should be the default semester (see “Change default semester”).

Rationale: Much of the data in the system is tied to semesters, so semesters can't be edited or deleted. Furthermore, semesters must be generated in order, so it is best that the system handle it.

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 know what semesters AbOut knows about.

AM7: Default semester

At all times, AbOut shall have a default semester.

Rationale: Most AbOut operations are dependent on a semester. Work is typically done on the current semester, so it is helpful if that is a default semester.

Note: This functionality is not needed. AM7 and AM8 will eventually be removed. Right now a default semester can be designated, but that designation is ignored. In the future, the notion of a settable default semester will be removed. The most recent semester will be the default semester.

AM8: Change default semester

Administrators shall be able to change the default semester.

Rationale: It was expected that AbOut administrators would perform operations relative to a particular semester, so having a settable default semester would be helpful. However, in use of the system, administrators typically do operations relevant to the most recent semester. Thus, in the current system, the most recent semester is always the current semester.

Note: This functionality is not needed. AM7 and AM8 will eventually be removed. Right now a default semester can be designated, but that designation is ignored. In the future, the notion of a settable default semester will be removed. The most recent semester will be the default semester.

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 outcome abbreviation, textual description, the program area to which it refers (CS or SE), and the semester when the outcome came into 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.

AM11: Edit student outcome

Administrators shall be able to edit the text that AbOut associates with a student outcome, and program that AbOut associates with the outcome, the start semester when that student outcome went into effect, and the last semester in which that outcome is in effect. Student outcomes will be considered "expired" in semesters occurring after the end semester.

When the text that AbOut associates with a student outcome is changed, the new text will appear in all reports subsequently generated. Similarly, when AbOut associates a student outcome with a different program, the change will be reflected in all subsequently generated reports.

Rationale: It must be possible to edit student outcome information since that information may have been entered incorrectly. It is assumed that the new information is a correction, and the new change needs to be reflected in subsequent reports.

AM12: Delete student outcome

Administrators shall be able to delete student outcomes from the AbOut system if that student outcome has no assessments measuring it.

Rationale: Student outcomes may be mistakenly added to the system and administrators need to be able to remove outcomes mistakenly added. Student outcomes associated with a course and /or an offering, can be "expired" by setting the outcome's end semester.

AM13: View outcomes

Administrators shall be able to a view all student outcomes in the system, even old student outcomes.

Rationale: Administrators must know what student outcomes AbOut knows about.

AM14: Add course

Administrators shall be able to add courses to the AbOut system. Course information includes an abbreviation, course name, and the semester when the course became part of the curriculum. Administrators can also associate outcomes with a course (see “Associate or disassociate outcomes with course”).

Rationale: New courses will be added to the curriculum and the system will need to know when these new courses were added.

AM15: Edit course

Administrators shall be able to edit the name that AbOut associates with a course, the program or programs associated with the course, the start semester when the course became part of the programs, the last semester in which that course was part of the programs, and the student outcomes that AbOut associates with this course. Courses will be considered “expired” in semesters occurring after the end semester.

When the name that AbOut associates with a course is changed, the new name will appear in all reports subsequently generated. Similarly, when AbOut associates a student outcome with a different program, the change will be reflected in all subsequently generated reports. When the student outcomes that AbOut associates with a course are changed, the new student outcomes will only appear in those course offering created subsequent to the change.

Rationale: It must be possible to edit course information since that information may have been entered incorrectly. For the course name and programs, it is assumed that the new information is a correction, and the new change needs to be reflected in subsequent reports. Changes are expected in what outcomes the course measures, and these changes may not reflect corrections, but rather changes in what is actually covered in the course. These changes should effect those course offerings created after the change.

AM16: 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 be mistakenly added to the system and administrators need to be able to remove courses mistakenly added. Courses associated with an offering can be “expired” by setting the course’s end semester.

AM17: View courses

Administrators shall be able to a view all courses in the system, even old courses.

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

AM18: View, associate and dissociate outcomes with course

Administrators shall be able to view the outcomes that AbOut associates with a course, and to change those associations.

Rationale: Course material will change over time so the outcomes associated with that course must change. Administrators need to view what outcomes are currently associated with the course, so that the changes can be made.

AM19: 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. They shall also select a course, a section number, and the current faculty member who will teach this course offering. The name of the faculty member shall be copied into this course offering. The student outcomes that this course offering will measure shall be automatically connected to this course offering. Note that the student outcomes associated with this course offering cannot be changed here. In order to change which student outcomes are associated with this offering, an administrator must delete this offering, make the new associations between the course and outcomes, and recreate the offering.

Rationale: When the name of a faculty member changes in reality, the new name should only appear in offerings created subsequent to the change. Course offerings taught before the name change, should not reflect the new name.

AM20: Edit course offering

Administrators shall be able to edit the section number and faculty member that AbOut associates with a course offering. They shall also be able to add and remove student names associated with the course offering. The semester AbOut associates with the offering cannot change, nor can the student outcomes that AbOut associates with the offering.

When the section or faculty member associated with the course are changed, these changes will appear on all subsequently generated reports.

Rationale: It must be possible to edit the section number or faculty member associated with a course as that information may have been entered incorrectly. For these fields it is assumed that the edits are corrections, and the new values should be reflected in subsequent reports.

The semester and outcomes associated with a course offering are considered intrinsic to the offering and cannot be changed. If mistakes to these are discovered, the course offering is to be removed from the system and created again.

AM21: Delete course offering

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

Rationale: Course offerings may have been mistakenly added to the system and administrators need to be able to remove those offering mistakenly added.

AM22: View all course offerings

Administrators shall be able to a view all courses offerings in the system, ordered from the most recent offering to the older offerings.

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

AM23: 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. The system shall provide a file picker to facility finding the file.

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

AM24: Add student to a course offering

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

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.

AM25: Delete a student from a course offering

Administrators or the faculty member teaching a course offering shall be able to remove a student from a course offering, causing all assessment scores for that student to be removed. If the course offering had one or more assessment, 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, fail the course, or may not be CS or SE majors. It needs to be possible to remove these students from the assessments for that course offering.

1.11.3 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.

Note: This functionality is not needed. It will be removed along with “Default semester” AM7 and “Change default semester” AM8.

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: 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 drop or fail the course, and will need to be removed from the offering. Alternatively, a student may add the course.

FC4: Add assessment to a course offering

The faculty member teaching a course offering shall be able to add an assessment to the offering. Assessment information includes a short textual description of the assessment item, the maximum number of points that a student can achieve on the item, and a list of those student outcomes that this assessment measures. The student outcomes being measured must come from the list of student outcomes associated with this course offering.

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

FC5: Edit assessment

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 assessment will measure. The student outcomes being measured must come from the list of student outcomes associated with this course offering.

Rationale: Faculty members may reconsider an assessment description and the student outcomes it assesses, 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.

FC6: Delete assessment

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

Rationale: Assessments may be mistakenly added to the system and the faculty member teaching the offering must be able to delete these assessments.

FC7: Add student score to assessment

The faculty member teaching a course offering shall be able to add a student score to an assessment. Student scores must be in the range of 0 to the maximum number of points for the assessment. 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.

FC8: Edit student score

The faculty member teaching a course offering shall be able to edit a student score on an assessment provided the new score is within the range of 0 to the maximum number of points for the assessment. 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.

FC9: 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 assessment(s) which measures a student outcome, the percentage will be 0%.

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

FC10: 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.

1.11.4 Reports (RP)

RP1: Indicate semester(s) for Overview Course, Outcome, and Overview Outcome 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 these reports are relevant to a semester or a range of semesters.

RP2: Indicate program(s) for Overview Course, Outcome and Overview Outcome Reports

Users shall be able to indicate the program(s) (CS, SE of both) for which the report is to be generated.

Rationale: Reports on CS courses are needed when reporting to the ABET Computing Accreditation Commission (CAC), reports on SE courses are needed when reporting to Engineering Accreditation Commission (EAC), and CS and SE courses are needed when reporting to the Computer Science Department Industry Advisory Board.

RP3: 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.

RP4: Indicate show raw data

Users shall be able to indicate that raw data is to be shown on any report except the Matrix Report.

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

RP5: 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 who earned the assessment goal over the total number of students assessed), in addition to the corresponding percentage representation.

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

RP6: 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.

RP7: 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. The filename for the .csv file will be the report name, along with the course, program, or outcome that is the focus of the report. If allowed by the browser and operating system, this report can be saved at a location chosen by the user.

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

RP8: Generate CORE Report

Users shall be able to prompt the system to generate statistics for a CORE Report which displays the percentage of students enrolled in a course offering who earned the assessment goal or higher. 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 assessment 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.

RP9: Generate Overview Course Report

Users shall be able to prompt the system to generate an Overview Course Report which displays the percentage of students, by course and student outcome, who earned the assessment goal or higher, for all assessments within the range of semesters indicated. If there are no student scores for any assessment of an outcome, a dash shall be displayed. The report shall also display an average, across all outcomes, for each course, and an average, across all courses, for each outcome.

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

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

RP10: Generate Outcome Report

Users shall be able to prompt the system to generate an Outcome Report which displays the percentage of students, by course and semester, who earned the assessment goal or higher, for all assessments within the range of semesters indicated. If there are no student scores for any assessment of an outcome, t a dash shall be displayed. The report shall also display an average, across all outcomes, for each semester, and an average, across all semesters, for each outcome.

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 Overview Outcome Report will allow AbOut users, ABET evaluators and IAB members to quickly see the extent to which each student outcome was met during a semester range.

RP11: Generate Overview Outcome Report

Users shall be able to prompt the system to generate an Overview Outcome Report which displays the aggregate percentage of students, by student outcome and semester, who earned the assessment goal or higher, for all assessments in all course offerings within the range of semesters indicated. If there are no student scores for any assessment of an outcome, a dash shall be displayed. The report shall also display an average across all semesters for each outcome.

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 Overview Outcome Report will allow AbOut users, ABET evaluators and IAB members to quickly see the extent to which each student outcome was met during a semester range.

RP12: 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 an 'X', while courses that do not will be unmarked.

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.

RP13: 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.

1.12 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.

1.12.1 Design Constraints (DC)

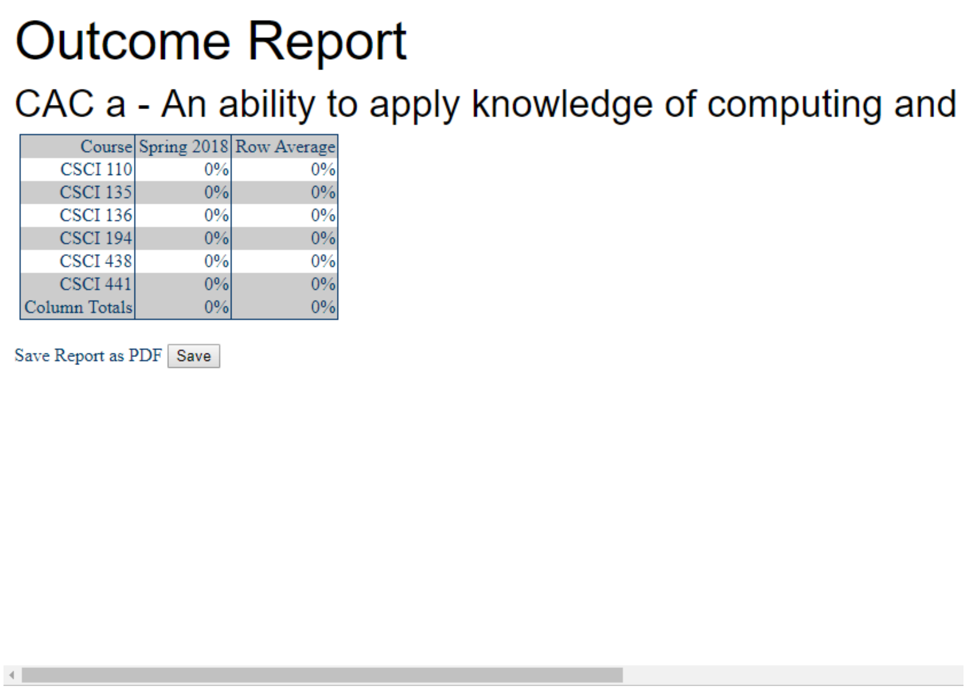
DC1: This application is to be developed using MySQL and PHP.

DC2: It must be easy to change the value of the assessment goal. A programmer should only need to change the code in one place, and all reports will be generated using the new assessment 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: Outcome Reports shall appear similar to the following:

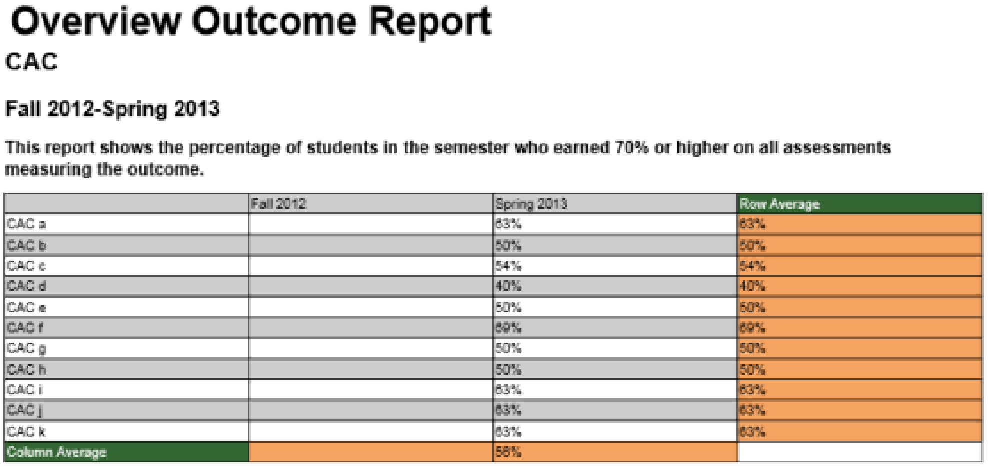


Outcome Report

CAC a - An ability to apply knowledge of computing and

Course	Spring 2018	Row Average
CSCI 110	0%	0%
CSCI 135	0%	0%
CSCI 136	0%	0%
CSCI 194	0%	0%
CSCI 438	0%	0%
CSCI 441	0%	0%
Column Totals	0%	0%

Save Report as PDF



Overview Outcome Report

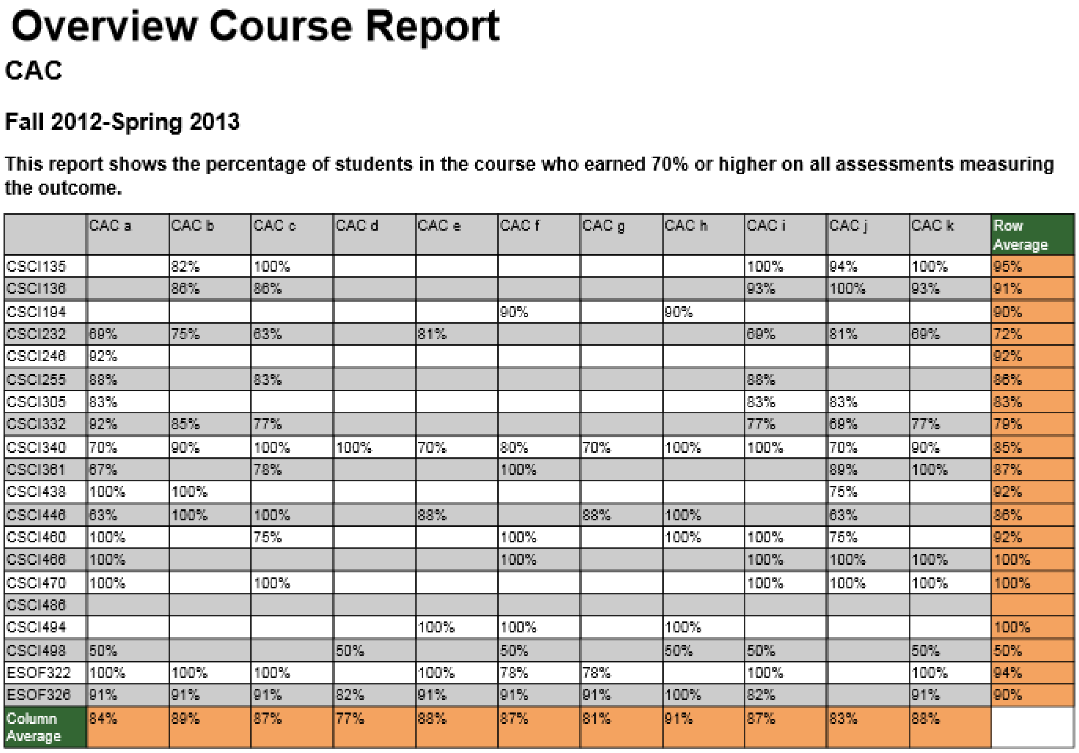
CAC

Fall 2012-Spring 2013

This report shows the percentage of students in the semester who earned 70% or higher on all assessments measuring the outcome.

	Fall 2012	Spring 2013	Row Average
CAC a		63%	63%
CAC b		50%	50%
CAC c		54%	54%
CAC d		40%	40%
CAC e		50%	50%
CAC f		69%	69%
CAC g		50%	50%
CAC h		50%	50%
CAC i		63%	63%
CAC j		63%	63%
CAC k		63%	63%
Column Average		56%	

DC6: Course Reports shall appear similar to the following:



Overview Course Report

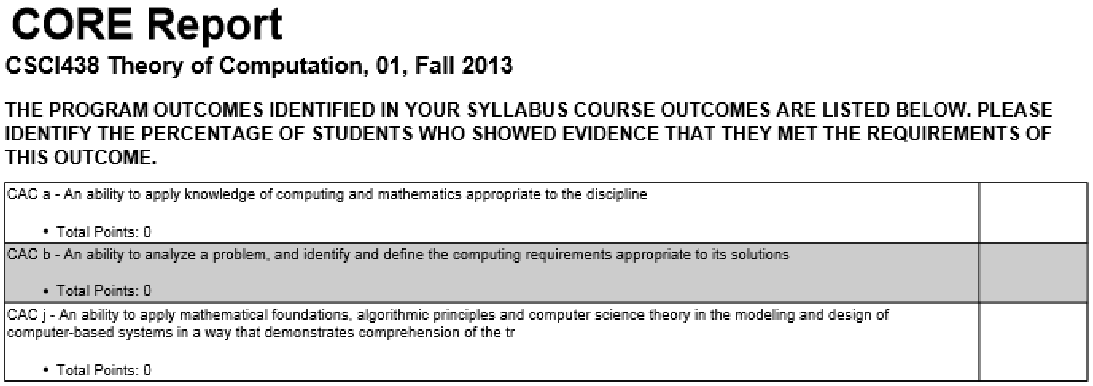
CAC

Fall 2012-Spring 2013

This report shows the percentage of students in the course who earned 70% or higher on all assessments measuring the outcome.

	CAC a	CAC b	CAC c	CAC d	CAC e	CAC f	CAC g	CAC h	CAC i	CAC j	CAC k	Row Average
CSCI135		82%	100%						100%	94%	100%	95%
CSCI136		86%	86%						93%	100%	93%	91%
CSCI194						90%		90%				90%
CSCI232	69%	75%	63%		81%				69%	81%	69%	72%
CSCI246	92%											92%
CSCI255	88%		83%						88%			86%
CSCI305	83%								83%	83%		83%
CSCI332	92%	85%	77%						77%	69%	77%	79%
CSCI340	70%	90%	100%	100%	70%	80%	70%	100%	100%	70%	90%	85%
CSCI361	67%		78%			100%				89%	100%	87%
CSCI438	100%	100%								75%		92%
CSCI446	63%	100%	100%		88%		88%	100%		63%		86%
CSCI460	100%		75%			100%		100%	100%	75%		92%
CSCI466	100%					100%			100%	100%	100%	100%
CSCI470	100%		100%						100%	100%	100%	100%
CSCI486												
CSCI494					100%	100%		100%				100%
CSCI498	50%			50%		50%		50%	50%		50%	50%
ESOF322	100%	100%	100%		100%	78%	78%		100%		100%	94%
ESOF326	91%	91%	91%	82%	91%	91%	91%	100%	82%		91%	90%
Column Average	84%	89%	87%	77%	88%	87%	81%	91%	87%	83%	88%	

DC7: CORE statistics shall appear similar to the following:



CORE Report

CSCI438 Theory of Computation, 01, Fall 2013

THE PROGRAM OUTCOMES IDENTIFIED IN YOUR SYLLABUS COURSE OUTCOMES ARE LISTED BELOW. PLEASE IDENTIFY THE PERCENTAGE OF STUDENTS WHO SHOWED EVIDENCE THAT THEY MET THE REQUIREMENTS OF THIS OUTCOME.

CAC a - An ability to apply knowledge of computing and mathematics appropriate to the discipline • Total Points: 0	
CAC b - An ability to analyze a problem, and identify and define the computing requirements appropriate to its solutions • Total Points: 0	
CAC j - An ability to apply mathematical foundations, algorithmic principles and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tr • Total Points: 0	

1.12.2 Human Factors (HF)

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

1.12.3 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.

1.12.4 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.

1.12.5 Development Environment (DV)

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

1.12.6 Standards (ST)

ST1: 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.

1.12.7 Delivery Environment (DL)

Site (SI)

No site requirements are placed on the application.

Operations (OP)

No operations requirements are placed on the application.

1.12.8 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, website speed test).

1.12.9 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.

1.12.10 Cost (CT)

No cost requirements are placed on the application.

1.12.11 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.

1.12.12 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.

1.12.13 Database (DB)

DB1: MySQL shall be used for this application.

1.12.14 Adaptability (AD)

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

1.13 Requirements Models

No requirements models are included in this specification.

Future Enhancements (FE)

This product could be modified so that other Montana Tech programs which are accredited by ABET can use it.