**ACAT Software Requirements Specification Document**

**CS 5530**

**ABET Course Assessment Tool**

Joel Snider

Stephan Maxi

Wyatt Landers

Patrick Evans

Jorge Ruiz

Langdon Boxxe

**Version: 1.0** **2/24/2019**

**Table of Contents**

**1. Introduction**

*1.1 Purpose*

*1.2 Scope*

*1.3 Definitions, Acronyms, and Abbreviations*

*1.4 References*

*1.5 Overview*

**2. The Overall Description**

*2.1 Product Perspective*

*2.2 Product Functions*

*2.3 User Characteristics*

*2.4 Constraints*

*2.5 Assumptions and Dependencies*

*2.6 Apportioning of Requirements*

**3. Specific Requirements**

*3.1 General Requirements*

*3.2 Assessment Report Requirements*

*3.3 Course Outcome Requirements*

*3.4 Direct Assessment Requirements*

*3.5 Report Functional Requirements*

*3.6 Non-Functional Requirements*

**4. Management Process**

# **1. Introduction**

## **1.1 Purpose**

The purpose of this document is to present a detailed description of the ABET Course Assessment Tool (ACAT). It will outline the function and purpose of the tool and describe how the tool is intended to be used. This document is to be used by both users of the tool and those interested in its development.

## **1.2 Scope**

ACAT is a tool intended for use by professors of universities approved by ABET accreditation. It is designed to assist faculty in the creation of course assessment reports for ABET accreditation. The tool allows the user to enter course learning outcomes, mission objectives, evaluation instruments and assignments, and grades in order to generate a report.

It will allows individual users to log in to the system and create a portal for any number of courses they need to create a report for. All files and information would be saved on a local database that can be backed up into a cloud based server.

## **1.3 Definitions, Acronyms, and Abbreviations.**

|  |  |
| --- | --- |
| TERM | DEFINITION |
|  |  |
|  |  |
|  |  |

## **1.4 References**

*IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specification. IEEE Computer Science, 1998.*

## **1.5 Overview**

The following chapter provides a more detailed description of the ACAT Tool. Chapter 3 lists the specific requirements for our software. The final chapter will detail the process of how we created the software and how our team divided up the work.

# **2. The Overall Description**

## **2.1 Product Perspective**

ACAT is a self-contained software application designed for a user to load on their local machine. Upon start up, the user will enter their login credentials and will be given access to their personal portal. This allows the user to view previously made courses or update any necessary information. There is also a button which allows the user to create a new course from scratch. When a new course is created, the tool will ask the user for a variety of information. This includes the course name, semester, number of students, number and names of all the assignments for the course. Any of this information can be edited later if need be.

## **2.2 Product Functions**

The tool allows the user to input any information and parameters required to create an ABET appropriate report and store any class information needed.

# **3. Specific Requirements**

## **3.**1General Requirements

The general functional requirements for ACAT are as follows:

1. ACAT shall require the user or administrator to log in to use the system.
2. ACAT shall not allow a user to view or modify another user’s assessments.
3. ACAT shall allow a user to change their password or personal information.
4. ACAT shall allow an administrator to add and delete users, and can reset passwords.
5. ACAT shall not allow an administrator to view a user’s password.
6. ACAT shall allow a user or administrator to log out of the system.

## 3.2 Assessment Report Requirements

The requirements for the assessment reports created by ACAT are as follows:

1. ACAT shall allow a user to create a new assessment report.
2. ACAT shall allow a user to modify an existing assessment report.
3. ACAT shall create only one assessment report for each unique combination of user, course, and semester.
4. ACAT shall allow a user to enter past changes that were made since the report information was last accessed.
5. ACAT shall support one or more changes to an assessment report at a time.
6. ACAT shall support one or more comments for each assessment, viewable only to the user.

## 3.3 Course Outcomes Requirements

The requirements for the course outcomes are as follows:

1. ACAT shall allow a user to input desired course outcomes and mission objectives.
2. ACAT shall allow a user to edit course outcomes.
3. ACAT shall allow a user to enter comments for each course outcome.

## 3.4 Direct Assessment Requirements

The direct assessment requirements for ACAT are as follows:

1. ACAT shall allow the user to define one or more instruments of assessment for each course outcome.
2. ACAT shall allow a user to enter a score associated with each assessment for each student in the course.
3. ACAT shall allow a user to input comments for each assessment.
4. ACAT shall allow a user to edit entered scores.
5. ACAT shall allow a user to remove any or all of the students’ scores from an assessment report.

## 3.5 Report Functional Requirements

The functional requirements for reports created using ACAT are as follows:

1. ACAT shall allow a user to generate an assessment report once all necessary has been entered.
2. ACAT reports shall contain instructor’s name, course name, semester, and CRN.
3. ACAT reports shall contain a summary table.
4. ACAT reports shall contain a score table.
5. ACAT reports will show all course outcomes, mission objectives, ABET requirements, assessment average scores, assessment standard deviations, and score averages per learning outcomes.
6. ACAT shall allow a user to save the report to a file.
7. ACAT shall a user to edit reports as necessary.

## 3.5 Non-Functional Requirements

All other non-functional requirements for ACAT are as follows:

1. ACAT shall support local database integration using SQLite.
2. ACAT shall be a locally run and hosted computer application.
3. ACAT shall allow multiple users to have accounts in the system.

# **Management Process**

Our team has chosen to use Trello as a team labor management tool. Joel Snider is the team lead and handles relating with the client, Dr. De. Wyatt Landers and Jorge Ruiz are handled the application interface and ensured that all data is processed correctly. Patrick Evans and Stephan Maxi were responsible for creating and managing the database that stores all necessary information for report creation.