CSCI 5530 - Software Engineering

ABET Course Assessment Tool

SOFTWARE DESIGN DOCUMENT

Joel Snider, Stephan Maxi, Jorge Ruiz, Wyatt Landers, Patrick Evans, Langdon Boxxe

April 5, 2019

Dr. Andrew Allen

Abstract

The ABET Course Assessment Tool (ACAT) is a software designed with the intention to assist engineering professors of ABET accredited universities in the creation of mandatory ABET reports. It allows the user to input parameters for each course and assessment tool used throughout a semester, save this data, and generate an assessment report with relative ease. Within the tool, whenever a new course is created, the user is asked to enter the course’s learning outcomes, mission objectives, ABET learning objectives, number of evaluation instruments, the average score and standard deviation for each evaluation instrument, and program outcomes for the course. Once the user has input all data necessary to generate a report, they can do so at the click of a button.

The ACAT application itself will be locally hosted on the user’s device. It will run on Java so that it is easy to access on most devices. When the user first opens the program, it will establish a local database to hold all the data entered. It was also include an external database that will allow the user to access their course information from different devices. Each user will have a group of courses associated with their login credentials, which will be created by an admin account.

Table of Contents

Page

Abstract………………………………………………………………………………………………………………...2

Table of Contents……………………………………………………………………………………………………....3

CHAPTER 1 - INTRODUCTION 4

1.1 Purpose of System……………………………………………………………………...………………………….4

1.2 Design Methodology……………………………………………………………………………………………....4

1.3 Definitions, Acronyms, and Abbreviations………………………………………………………………………..5

1.4 Overview of Document…………………………………………………………………………………………....5

CHAPTER 2 - PROPOSED SOFTWARE ARCHITECTURE 6

2.1 Overview…………………………………………………………………………………………………………..6

2.2 Metamodel………………………………………………………………………………………………………....6

2.3 UML Profiles……………………………………………………………………………………………………....X

2.4 Generative Architecture..………………………………………………………………………………………….X

2.5 Subsystem Decomposition………………………………………………………………………………………...X

CHAPTER 3 - OBJECT DESIGN X

3.1 Overview……………………………………………………………..…………………………………………....X

3.2 Object Interaction …………………………………………………………………………………………............X

3.3 Detailed Class Design ………………………………………………………………………………………….....X

CHAPTER 4 - GLOSSARY X

CHAPTER 5 - APPENDICES X

Chapter 1

INTRODUCTION

Many universities across the United States seek the approval of the Accreditation Board for Engineering and Technology, or the ABET, for their engineering related courses. In order to do this, professors of these engineering courses must create a report for each of the courses they teach that detail the learning outcomes, mission objectives, and assessment grades to be submitted each semester. This process can be long and tedious, and takes up much of the time that a professor could use to focus instead on instructing students. The ABET Course Assessment Tool, or ACAT, is created to help make this process more simple. This chapter provides an overview of what ACAT is and how it will work.

1.1 Purpose of System

The reason for the create of ACAT is to simplify the process of ABET accreditation reports for engineering courses. ACAT will allow a user to create a module for any number of different courses being taught in a certain semester and enter different parameters for each course that are required for create an ABET report. These include learning outcomes, mission objectives, assessments with grade samples ABET requirements, and averages for program outcomes. The user can edit of this information at any times as needed. Once all necessary information for a certain course has been entered, ACAT will generate an accreditation report at the click of a button.

1.2 Design Methodology

ACAT was designed with usability and security as two primary factors. We understand that accreditation report creation is tedious, but not difficult. As such, we understood that our software for generating these reports should also be as straightforward as possible. Also, since student records and grade are crucial to the use of our software, we understood that the security measures related to our content and ensuring proper encryption methods was extremely important.

1.3 Definition, Acronyms, and Abbreviations

* ABET - Accreditation Board for Engineering and Technology, Inc.
* ACAT - ABET Course Assessment Tool
* Alphanumeric - A string consisting of letters and numbers
* API - Application Programming Interface
* Functional requirement - All absolutely necessary requirements our system must satisfy
* GUI/UX - Graphical User Interface
* Nonfunctional requirements - all constraints on the system that are not absolutely necessary
* System - The entirety of the ACAT software and all included code

1.4 Overview of Document

The following chapters go into further detail over the specifications and design of our ACAT system. Chapter 2 details the overall architecture of ACAT and chapter 3 discusses its object design in detail.

Chapter 2

Proposed Software Architecture

Software architecture describes the overall design of a piece of software. Architecture is by far the most important information when it comes to the creation of any software. The following sections detail the chosen architecture details for the ACAT system.

2.1 Overview

ACAT is designed using a simple three-tier architecture schema. The system is primarily a stand-alone application that allows users to generate reports, but also encompasses two databases to enhance the system design. The first tier is that application that allows the entering of information and generation of reports. The second is a local database to be loaded onto a user’s machine that allows any and all information necessary in the stand-alone application to be saved so that it can be used or modified at a later time. The third tier is a cloud-hosted database that the user can synchronize their local system to with relative ease. The cloud database shares the same schema as the local one, and the application allows for the two databases to stay in sync with one another whenever the user decides to back one up with the other.

2.2 Metamodel