

# Software Design

Version 1.2



## Team: Excelsior

February 21st, 2020

### Project Sponsor:

Dr. Toby Hocking

### Mentor:

Dr. Eck Doerry

### Team Members:

Austin Torrence, Matthew Rittenback, Brandon Thomas

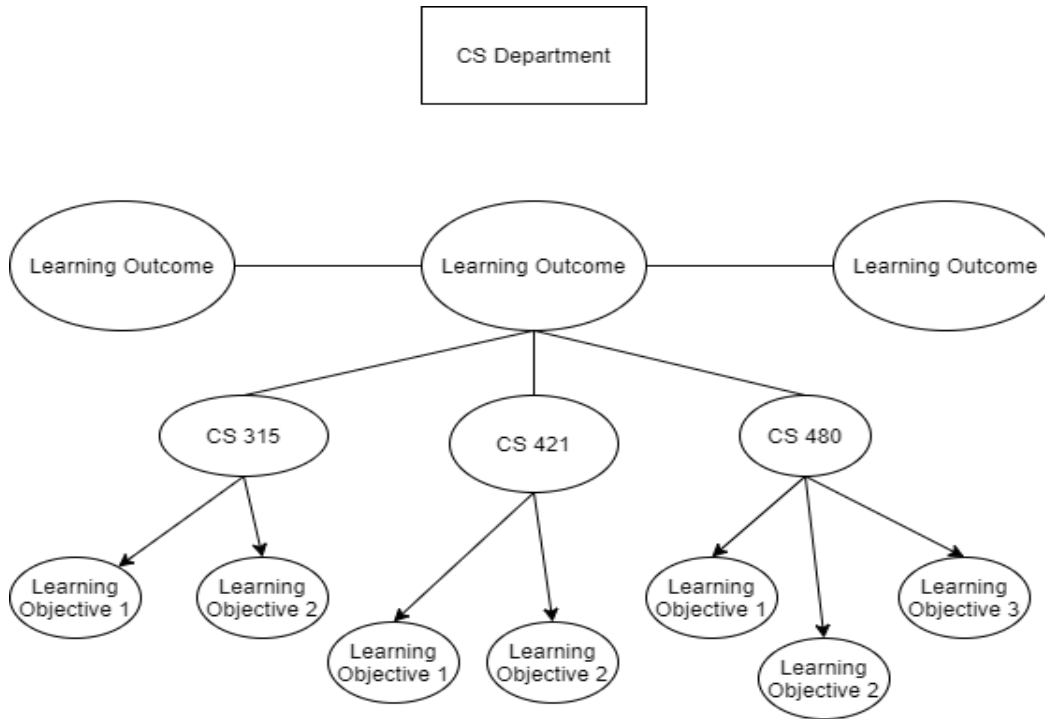
## Table of Contents

<b>1. Introduction</b>	<b>2</b>
<b>2. Implementation Overview</b>	<b>4</b>
<b>3. Architectural Overview</b>	<b>5</b>
<b>4. Module and Interface Descriptions</b>	<b>7</b>
4.1 File Handling	7
4.2 Data Monitoring	7
4.3 Notifications	7
4.4 Data Visualization	8
4.5 GUI	9
<b>5. Implementation Plan</b>	<b>10</b>
<b>6. Conclusion</b>	<b>11</b>

# 1. Introduction

Technology runs most of the modern world and it's the engineers of the world that design, build, or maintain technologies ranging from personal cell phones to the airliners that move millions of people across the world every day. Flaws in engineered products can often result in catastrophic failure sometimes resulting in the loss of human life. To ensure the quality of the engineering workforce, universities make sure their students are completing programs that are accredited by agencies who set the global standards for their major. These agencies provide guidelines to make sure that each and every engineer who graduates from an accredited program knows the necessary information to excel in their field and produce safe and high-quality engineered products.

ABET is a non-profit agency that accredits programs across all engineering and technology fields. ABET is focused on maintaining the quality of education for engineers enrolled in their accredited programs by (a) providing quality standards that programs at universities like Northern Arizona University need to adhere to to stay accredited and recognised; and (b) ensuring that programs monitor and enforce those standards. Graduating from a program accredited by ABET informs employers that this potential new hire has the ability to perform in their duties at a professional level.



*Figure 1: ABET Model showing Learning Outcomes supported by courses that all contribute to various Learning Objectives*

A department follows a specific guideline given by ABET which is shown in Figure 1. Departments will create ABET-approved learning outcomes that are to be supported by a variable number of classes. These classes will then have specific learning objectives related to the course that directly support the overall learning outcome. These learning objectives are measured in a variety of ways depending on the course, and examples of these measurements include questions on a final exam, a portion of a presentation, or any other academic benchmark. These learning objectives will then be manually recorded in an excel spreadsheet by the professor on a scale of 1 to 5 depending on the individual student's performance regarding that learning objective. These excel files then produce a short summary of the percentage of students who are performing to ABET standards. ABET requires that at least 70% of the students are at or above that 3 level.

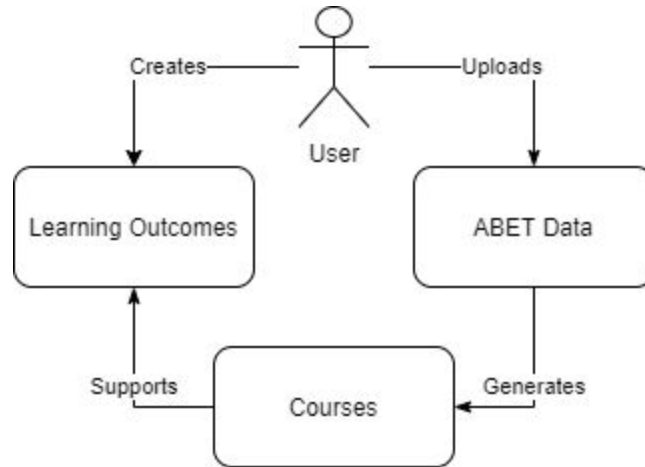
Currently for the ABET coordinator, the process of monitoring of ABET data collection across multiple learning outcomes is inefficient and error-prone. There is no technologies in place at the moment for the coordinator to check which professors have completed uploading ABET data for a given course or semester. This can lead to professors forgetting to collect and input data for a course from a past semester or even past year, which makes it difficult for the

coordinator to collect the data. With 6 year intervals, the ABET coordinator is responsible for compiling all of the collected ABET data from that time period, to prove and show that the department is successfully supporting it's learning outcomes through the scores from the learning objectives that students achieve. With missing or lacking data from professors, it can be difficult for the coordinator to successfully pull together the necessary data at the end of these intervals to support the department in proving they are meeting ABET standards through the courses being taught.

## 2. Implementation Overview

Our team was tasked with designing a product to help improve the overall ABET data monitoring process. While researching the requirements for our product to achieve, we worked closely with Dr. Doerry, as well Dr. Hocking and Dr. Palmer to learn each professor's personal experience with ABET and what they would hope to get out of our product, ABET Monitoring Tool & Visualization (AMTV). Through this discussion with the professors, we determined our product needs to be able to be run on any platform, parse excel files in the format that already exists, graphically visualize the performance of a class for one semester or multiple semesters using different types of graphs, and notify professors of any missing or incomplete data.

The solution revolves around a desktop application that can be easily downloaded and run by professors or other university staff, have the ability to load ABET data from importing excel files, and graphically displaying data on the outcomes that have been achieved and the areas for improvement. The application will have a dashboard where configurations for the visualisations can be edited and changed. Users will be able to load excel files from the dashboard which will then be used to display data on the courses. Learning outcomes from ABET will have specific learning objectives that will be connected to all the different courses that contribute to the overall outcome. Once the data is loaded, the dashboard will be able to display indicators on the classes and outcomes such as whether or not all the data for a class has been collected, and if the desired learning outcomes were achieved or not.



*Figure 2: Flow Diagram of ABET Data from the user being processed in the application*

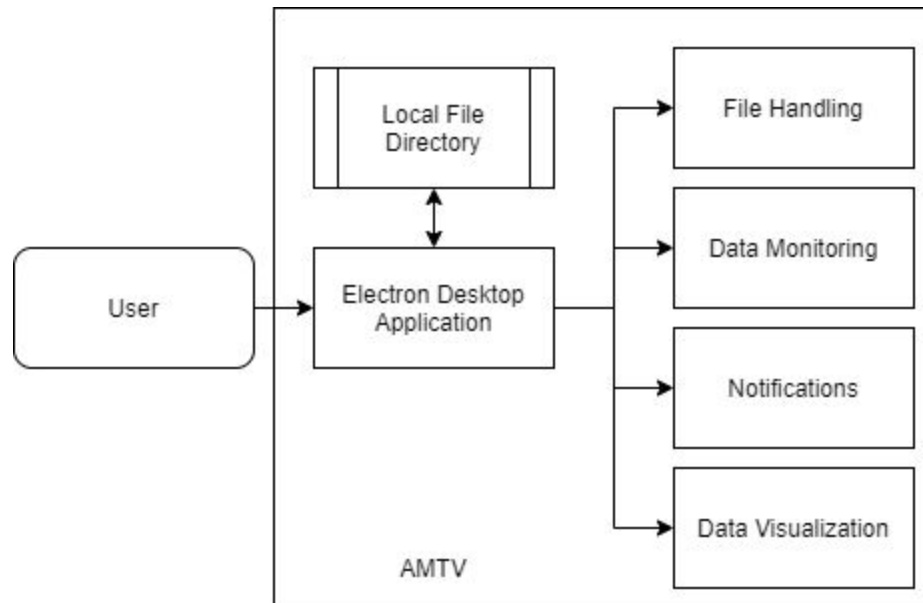
In the above figure, the basic workflow of the application is outlined for how the user interacts with the application. The workflow is entirely self sufficient in that it only requires the input of the user to function, and for the user to maintain a minimal amount of maintenance. The team decided to create a desktop application in Javascript using Electron to package up the web app into an easily executable program that can be downloaded and run on any operating system. The bulk of the program will be written in Javascript and the design will be done using HTML and CSS. The libraries we will be using to assist the team include:

- Electron: We choose this as our library to package our web app inside of a desktop application that can be run cross-platform and allow users to run it on most personal machines.
- SheetJS: This library was chosen because it provided us the easiest way of extracting the data we need from excel files.
- Plotly: This library was chosen because it provided us all of the features we were looking for in data visualization and integrates well inside our desktop app.

### **3. Architectural Overview**

The AMTV will be a system with four key modules that are contained and run inside of a desktop application. The user will launch and interface with the desktop application to access

and utilize the modules of the AMTV to enable them to monitor their ABET data. This section will provide a high level overview of the architecture involved with the AMTV.



*Figure 2: The modules the user can access through AMTV*

The figure above shows how the four main portions of AMTV, File Handling module, Data Monitoring module, Notifications module, and Data Visualization module, will interact with one another. It starts with the user launching AMTV and providing a local file directory containing all the ABET data. Then AMTV will provide information to the user based on current settings and what is currently provided in the file directory.

Once the directory has been identified, the tables are then parsed using the File Handling and Data Monitoring modules to gather the required information for the Data Visualization module. Once the Data Visualization module receives the necessary data, it can display accurate and helpful visualizations. These visualizations will include stacked bar charts of a certain class of one or more learning objectives showing the percentages of students at each level (1 through 5) as well as a line chart showing the performance of a specific class's learning objectives over time to better show how a class has improved or deteriorated. Our product will also include indicators on whether or not a class has succeeded under ABET's standards. This would be shown as a green, red, or yellow circle/box indicating whether or not a class has shown successful outcomes, failed to achieve its goal in terms of the learning objectives, or if a class does not have enough required data to make a conclusion, respectively.

With the overall architecture of AMTV outlined, we will now go into detail about the functionality of each module.

## **4. Module and Interface Descriptions**

With an understanding of the overall architecture of the application, each of the major modules of the application will be detailed below, providing a description of the module and its purpose. The major modules that will be covered in this section include the File Handling, Data Monitoring, Notifications and Data Visualization. Following the discussion of the modules, we will break down the interface to explain where the features are located in the interface and how they interact together.

### **4.1 File Handling**

The File Handling module is responsible for populating AMTV with Learning Outcomes and populating these LO tables with courses. Courses are created by the user providing a local file directory containing all of the user's ABET excel files. This then allows AMTV to scan the file directory and generate/update data inside the AMTV.

### **4.2 Data Monitoring**

The Data Monitoring module is responsible for scanning the local file directory provided by the user and performing numerous operations. First it's responsible for detecting all files and determining if the AMTV is up-to-date with the files provided. It will compare the list of detected files to a list of saved data in the AMTV to determine if it needs to perform an update to it's local data.

### **4.3 Notifications**

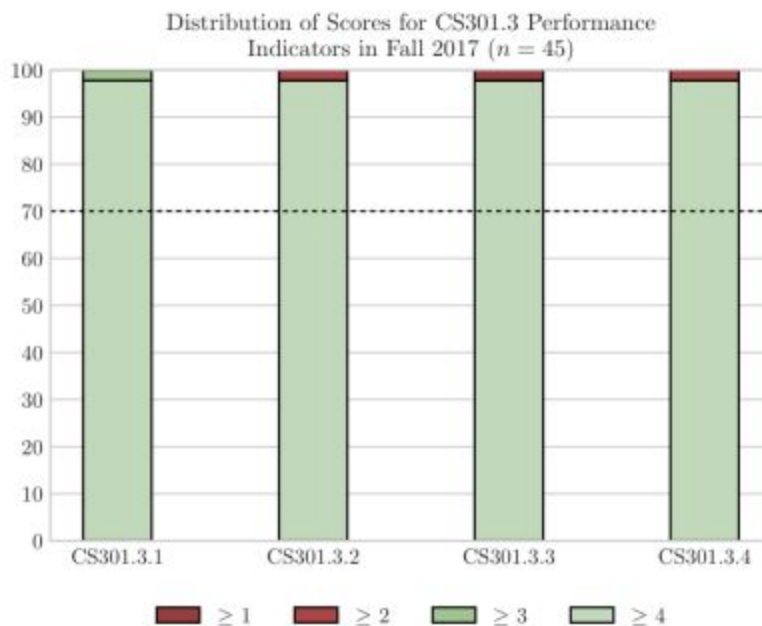
The Notifications module is responsible for taking in a user's contact information (email) and providing both local and email notifications to the user based on entered settings from the file handling and data monitoring modules. These notifications will occur on regular time intervals entered by the user and will only be sent after AMTV detects the upload date for a



course has passed. Notifications will be purely text reminders and will depend on the user to navigate to AMTV on their own to locate the problem course.

## 4.4 Data Visualization

The Data Visualization module is responsible for displaying graphical visualizations of the uploaded data after the user has handled the file handling portion of the application. The module will also be responsible for displaying different forms of graphs to accommodate for different data ranges. For example, the module will need to be able to display data graphs for both individual tables of data as well as groups of tables to show trends over time.

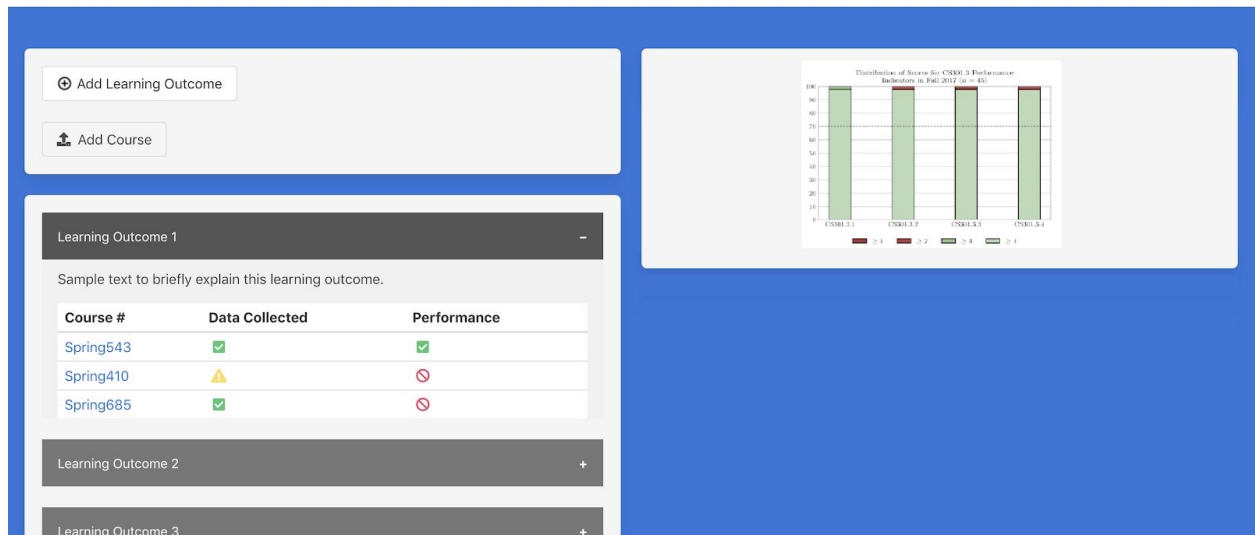


*Figure 3: Example Visualization Graph of ABET data*

Upon clicking a specific course, graphs will be displayed in the application as shown above. The graph is a bar graph with each bar representing a specific learning objective that was measured in that course. Each bar is also color coordinated based on the performance value indicated in the application settings. Below the graph will be settings available for the user to modify in order to adjust the graph in a desired manner. Adjustments include time range, learning object constraints and specific course constraints.

## 4.5 GUI

With the application being run inside of Electron, all pages of the application will be basic HTML pages that will be utilizing the Bulma front-end framework.



*Figure 4: Prototype layout of GUI*

After launching the application, the user will see 2 major columns of information displayed inside of the application's window.

The left column begins with a section for the user to add learning outcomes to their list of tracked learning outcomes. This section also allows the user to create a new course which includes uploading a file from their local machine to the application, and attaching the course to a relevant learning objective in the application. Below this section, is where the learning objectives are listed and this list will be populated by the user. Each learning objective in this list is a collapsible element which after clicking, will expand and display a table listing all of the courses that support that learning objective. Each course listed in the table, has a course id, data completeness indicator, and a performance indicator.

The right column is the space of the application dedicated to the Data Visualization module with plenty of space for graphs to be displayed and for graph options to be modified. Some graph options include adjustments to time range, learning object constraints and specific course constraints.

With the design of the modules and interface being outlined and explained above, this demonstrates an effective split of tasks to implement the project. Although each module may be simple to implement by themselves, some are built on each other and thus requires an effective agile development plan. As a team we have drafted an implementation plan to coordinate the development of the modules in an effective manner, as explained in the next section.

## **5. Implementation Plan**

This capstone project was designed to be completed in one semester as opposed to a normal two-semester capstone project. The team will have to be proactive and in order to successfully implement the plan and get the work done, the semester is broken up into phases of progression toward that goal. The first week started with establishing team relations by getting to know one another. Afterwards, with proper communication among the team members, meetings for the coming semester were established, with the team meeting twice a week, meeting the team mentor once a week, and meeting the client once a week as well. After the basic group dynamics and standards were established, the team sat down and outlined all the major deadlines and programming tasks necessary for the project.

As shown in figure 5, the projects major programming tasks are split and grouped into two major phases. The first phase of the project will take place in the first half of the semester preceding spring break and the major goal of this first phase will be to accomplish a hastened requirements acquisition and implement the alpha demo prototype. Then after spring break, we will move on to the second phase of the project, with the goal of refining the alpha prototype that was developed with feedback received from users and implement more advanced features. The team is confident that following the schedule outlined above and meeting deadlines on time, they can produce AMTV.

## 6. Conclusion

In today's world, technology is increasing at a rapid speed, and the demand for engineers is increasing at an equally rapid speed. However, some of the technologies that engineers can build can pose a serious danger if built incorrectly. Universities that educate these engineers use ABET accreditation to prove that they provide a certain level of quality and that engineers know the necessary tools/skills to build safe functional products. While effective, the data collection portion of ABET is arduous involving potential for missing or incomplete data. This process makes it hard for professors to remember to collect data, to know if data is complete, and to understand and analyze the data to see outcome achievement progress or failure.

Our desktop application, available on any OS, will be able to take these ABET-formatted Excel files and provide detailed visualizations of classes, learning objectives, and learning outcomes monitoring for completeness of data and being able to send email notifications regarding missing/incomplete data. The team is confident with the overall approach to visualizing and monitoring ABET data, the team will be able to create a fine tuned product that will provide professors at many universities with a unique solution to process ABET data to improve data collection, monitoring, and visualization.

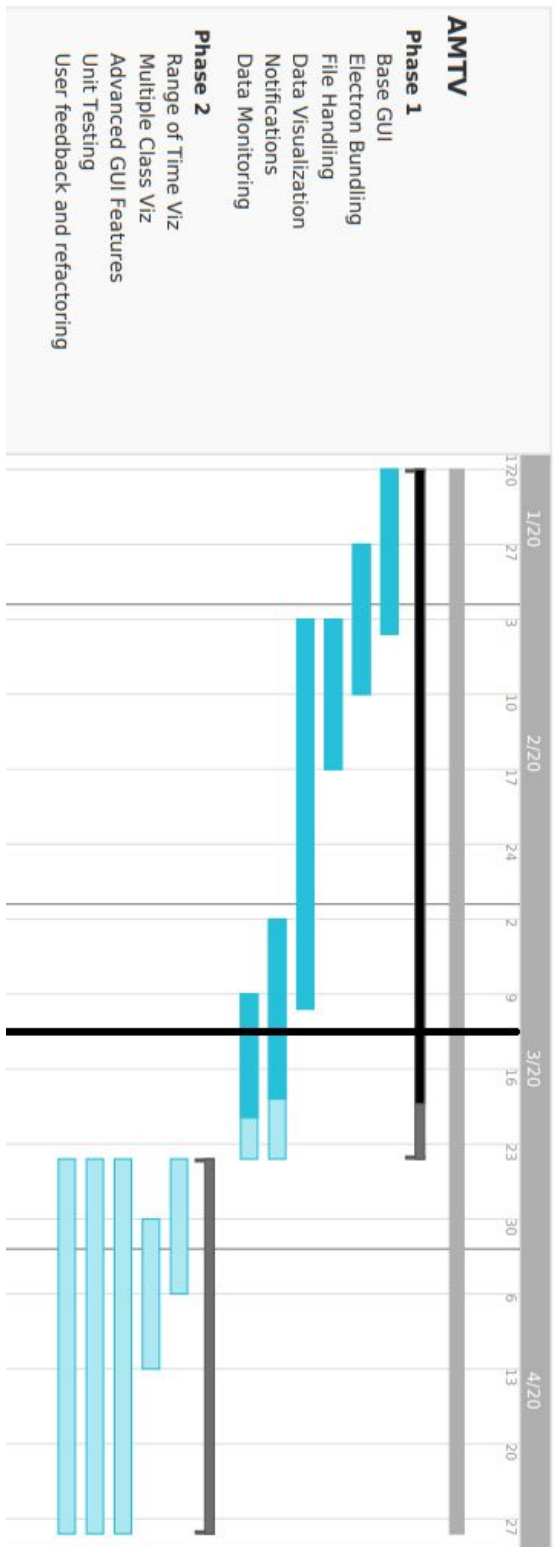


Figure 5: Gantt Chart displaying the team's major coding phases