1. Introduction

SLOScraper 1.0.0: This program will crawl the Butte-Glenn Community College website for the Program Learning Outcomes of each academic program offered by the college and consolidate them into a database.

1.1 Purpose

Each academic program offered by Butte College has defined expectations for what the graduates of any given program are expected to have learned. These are called Program Learning Outcomes (PLOs). Unfortunately, these PLOs are not stored in any single authoritative database, a situation which is prone to data redundancy, misinformation, and confusion. The purpose of the SLOScraper, therefore, is to go through the program descriptions from the Butte College website, extract each program's name and PLOs, and store all of this in an authoritative database.

1.2 Document Conventions

This document is made with the RUP development framework in mind. As a result, each feature of SLOScraper is assigned a priority value between 1 and 5, with 5 being top priority. Highest priority is given to the features that are likely to be the most challenging, and lowest priority will be given to the features that are in a more comfortable and familiar area of expertise.

1.3 Intended Audience & Reading Suggestions

This document is intended for use by the customer (SLO Coordinator), the team boss (April Browne), and each member of the team as the program is developed. For the customer and other non-technical readers, feel free to skip Section 3.

1.4 Product Scope

As previously stated, SLOScraper will enhance communication surrounding PLOs and their respective academic programs by crawling the college website for this information and storing it in a single, authoritative database. SLOScraper is not intended for use beyond the Butte College website, and is not designed to scrape any information from the site beyond the program name and the PLOs.

1.5 References

This project will be manifested in a way that reflects the values of The Zen of Python: valuing simplicity, pragmatism, and straightforwardness. For style guides, please refer to PEP 8 and The Hitchhiker's Guide to Python style guides (Python).

2. Overall Description

2.1 Product Perspective

SLOScraper is designed to fill the need of an organized system to check and track the updates made to the SLOs. SLOScraper is to replace the current system of Excel spreadsheets to track PLOs. The intent of SLOScraper is not to integrate something with butte.edu but use it as a resource for program data. SLOScraper will store its data in a database specifically designed for this.

2.2 Product Functions

SLOScraper will be capable of distinguishing the PLO and program name on the school website for any and all academic programs Butte College offers. This information is to be stored in a relational database as to accommodate staff and department chairs. This relational database also will offer an option to scrape for updates to the PLOs and programs from the Butte College website.

2.3 User Classes and Characteristics

The SLO coordinator and department chairs are the current defined users for the end product. This project is going forward with the assumption that the user has basic computer skills.

2.4 Operating Environment

SLOScraper and its database will be designed to run as a standalone piece of software in an environment with a valid network connection.

2.5 Design and implementation constraints

The major constraint this project faces is to ensure compatibility and ease of access for those who plan on continue the project after the current team has completed the course.

2.6 User Documentation

A short but complete description of each file needed for SLOScraper and its database to operate as intended and a short but complete description of the operation of the SLO Scraper and its database. will be included with the final version.

2.7 Assumptions and dependencies

It is assumed that SLOScraper is contributing to a larger project and is not necessarily an end product but in fact a piece of a larger entity. It is also assumed that code will be written in the future to use SLO Scraper directly as a resource by an entirely different team and CSCI 36 class. SLO scraper will be entirely dependent on the butte college website, more specifically http://butte.edu/academicprograms/.

3. External Interface Requirements

3.1 User Interfaces

There are no current plans to implement a user interface.

A command line interface may be used for development purposes. The interface will prompt the user to type a command from a list of possible commands, then carry out an action based on the input, and display a message to the user based on the result. Possible commands may include

- downloading the academic programs and storing them
- outputting a document with a selection of the stored data

This requirement is met if

1. The interface displays without errors.

2. When the user types a command, the action is either carried out successfully or an appropriate error message is printed.

3.2 Hardware Interfaces

The program must be able to access a persistent storage device on the computer to create a database file for storing information and access the file to retrieve information.

This requirement is met if the program is able to create and access a file when data must be stored and access that file again when data must be retrieved.

3.3 Software Interfaces

This software must be able run in a windows desktop environment.

This requirement is met if the program can start without errors in a desktop environment running windows.

The program must be able to interface with a local database file to retrieve data.

This requirement is met if the program is able to retrieve the correct data from a database file when it is requested.

3.4 Communications Interfaces

When the program is first run, it will need an internet connection to download the program name and PLOs through http. After the initial download, a connection will only be required when the data needs to be updated.

This requirement is met if the program is able to access the butte college website through http to download the required data when requested.

4. System Features

Web Scraping for Student Learning Objectives (Priority: 5)

1. SLOScraper's primary function is to scrape the student catalog for *Program Learning Objectives (PLOs)*. This will be accomplished with Python, using Beautiful Soup to help scrape the PLOs from the web page.

2. SLOScraper should accept a URL on the command line pointing to the academic program catalog to scrape; alternatively the URL may be hard-coded if the website is not expected to change within a few years time (at which point this program may have been replaced entirely).

3. Functional requirements include basic networking support and HTML parsing support.

Database Storage (Priority: 4)

1. SLOScraper will require some system for storing scraped academic program data. We will use MySQL to meet this requirement, as it is free and fast, and should be more than sufficient for our purposes.

2. Scraped data is useless if there is no means to export it into a user-friendly report. These reports should convey data clearly to the reader.

Automated operation (Priority: 3) - Optional

1. Sloscraper should not necessarily require human intervention to scrape changes barring significant or fundamental changes to the website; The scraper itself should be kept as simple and minimalistic as possible (within reason) to help facilitate automated usage.

2. Automated deployments require some way to specify scraping behavior without having a user present. Support for some kind of local config file will be necessary.

3. Functional requirements include filesystem I/O support and a basic facility for reading/writing a config file format.

5. Other Nonfunctional Requirements:

5.1 Performance Requirements

The application is focused scraping a web page to populate a database through manual execution. As the application is not constantly running in real-time but through a scheduled update or user-request, the execution performance requirements do not need to maximize timing, i.e. O(log n). Accuracy preferred over performance. The command-line interface derived from the scraper requires high performance.

5.2 Safety Requirements

The web scraper will be relied on to convey information critical to the Butte College institution and its faculty/students. Potential opportunities for loss while utilizing this application include inaccurate/omitted information which may cause students to forego taking a required course, thus needing to spend additional time and capital to take the course in future as it's necessary for accreditation.

5.3 Security Requirements

Due to the level of importance of such information, the web scraper execution and database access/manipulation must be private and unexecutable by non-administrators.

5.4 Software Quality Attributes

The software must be accurate as significant temporal and economic consequences will follow inaccuracy. Quality assurance must have a method to verify that the data the scraper is collecting is completely in line with what the website publicizes as students and faculty alike will rely on such data. The application must be made with reusability in mind and accommodate changes so that new data can be appended and old data removed/archived.

5.5 Business Rules

The individuals that can access the information supplied by the scraper is open to any user that desires as the information provided is publicly accessible to anyone already. Running the application itself is strictly limited to the Department Chair to maintain reliability of such information. The application will only be approved to execute and acquire new information when there are new changes/requirements of data, thus needing to re-scrape to account for new policy updates.