Assessments - Computations - Requirements

Assessments: Computations

2/2/2022

2.1

By: Mitchell Boone, Michael Quinn, and Liam Costello

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
2.1	Mitchell Boone, Michael Quinn, and Liam Costello	The computations part of the software would be able to receive a list of Canvas Course IDs and a Canvas API key from the user interface group, use them to retrieve the outcomes from Canvas, and create CSV files to be used to compare the outcomes.	2/2/2022

Review & Approval

Requirements Document Approval History

Approving Party	Version Approved	Signature	Date

Requirements Document Review History

Reviewer	Version Reviewed	Signature	Date

Table of Contents

1. Introduction

1.1 Purpose

The goal of the Computations task is to deliver outcome and assessment data as a CSV for one or more courses in the Canvas Learning Management System.

1.2 Document conventions

- CSV file - Comma-separated values file
- REST – Representational State Transfer
- API – Application Programming Interface

1.3 Intended audience

- Section 2 is intended for the stakeholders Assistant Vice President of Academic Affairs Barry Garside, Professor Jayne Barnes, Professor Jennifer Tripp, Professor Virginia Egan, and Professor Christine Morris.
- Sections 3, 4, and 5 are intended for the developers and project managers.

1.4 Scope

The goal of the Computations task is to create a CSV file to send to the visualizations group so they can use the data to create visual comparisons. This project will compare the results of multiple courses to see where they need to improve and where they are teaching well.

1.5 References

N/A

2. General Description

2.1 Product perspective

The project originated from a previous version created last year which finds the average outcome of a single class.

2.2 Product features

The computations part of the software will be able to receive a list of Canvas Course IDs and a Canvas API key from the user interface group, use them to retrieve the outcomes from Canvas, and create CSV files to be used to compare the outcomes.

Example CSV file:

	A	B	C	D	E	F
1	Professor	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5
2	Student 1	4/3	3/3	4/3	1/3	3/3
3	Student 2	4/3	3/3	3/3	2/3	3/3
4	Student 3	3/3	2/3	1/3	4/3	2/3
5	Student 4	4/3	3/3	3/3	2/3	4/3
6	Student 5	4/3	3/3	1/3	2/3	3/3
7	Student 6	3/3	3/3	1/3	3/3	2/3
8	Student 7	4/3	3/3	2/3	1/3	4/3
9	Student 8	3/3	2/3	4/3	4/3	4/3
10	Student 9	2/3	3/3	3/3	1/3	3/3
11	Student 10	4/3	3/3	4/3	3/3	3/3
12	Student 11	3/3	2/3	2/3	2/3	1/3
13	Student 12	4/3	3/3	3/3	4/3	3/3
14	Class Average	3.3/3	3/3	2.3/3	2/3	2.6/3
15	Class Median	3/3	3/3	2/3	2/3	3/3
16	% of Class 3 or Above	70%	60%	45%	65%	70%

2.3 User class and characteristics

The software is intended for use by Nashua Community College staff, but may be used by any organization that uses the Canvas Learning Management System.

2.4 Operating environment

The software is going to be designed to work in a Windows 10+ environment.

2.5 Constraints

The main constraint with the project is time.

2.6 Assumptions and dependencies

We are assuming the users will know how to work the Windows operating system and file system.

3. System Requirements

3.1 Functional requirements

We are requiring the Canvas Course IDs and Canvas API key for retrieval of outcome data.

4. External Interface Requirements

4.1 User Interfaces

N/A

4.2 Hardware Interfaces

A computer with the Windows operating system will be required to run the software. An internet connection will also be required to access the Canvas data.

4.3 Communications Interfaces

The software will be using Canvas REST API.

4.4 Software Interfaces

The software will retrieve the Canvas Course IDs and Canvas API key from the user interface improvements group to be used to retrieve outcome data from Canvas. The CSV files will look as specified in 2.2 and be written to the disk.

5. Non-Functional Requirements

5.1 Performance requirements

The only requirements are that the software can be used on a computer running the Windows operating system and that the software will execute in a reasonable amount of time.

5.2 Safety requirements

N/A

5.3 Security requirements

The software does not show names and therefore keeps the individual student’s information private.

5.4 Software quality attributes

The software will adhere to the current best practices.

5.5 Other requirements

N/A