**User Interface Design Document**

by Jacob Briand, Jared Gonsalves, Simeon Grosu, and Samantha Hughson

Adapted for LAFS by Elliott McCrory, 2/19/2022 12:45:38pm

Copyright © 1994-1997 by Bradford D. Appleton

*Permission is hereby granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies*.

**Table of Contents**

[1. Introduction 1](#_Toc166398920)

[1.1. Document Outline 2](#_Toc166398921)

[1.2. Document Description 4](#_Toc166398922)

[1.2.1. Introduction 4](#_Toc166398923)

[1.2.2. System Overview 5](#_Toc166398924)

[2. Design Considerations 5](#_Toc166398925)

[2.1. Assumptions and Dependencies 5](#_Toc166398926)

[2.2. General Constraints 5](#_Toc166398927)

[2.3. Goals and Guidelines 6](#_Toc166398928)

[2.4. Development Methods 6](#_Toc166398929)

[3. Architectural Strategies 6](#_Toc166398930)

[4. System Architecture 7](#_Toc166398931)

[5. Policies and Tactics 8](#_Toc166398933)

[6. Detailed System Design 9](#_Toc166398934)

[6.1. Classification 9](#_Toc166398935)

[6.2. Definition 9](#_Toc166398936)

[6.3. assumptions and Constraints 10](#_Toc166398938)

[6.4. Uses/Interactions 10](#_Toc166398940)

[6.5. Processing 10](#_Toc166398942)

[6.6. Interface/Exports 11](#_Toc166398943)

[6.7. Detailed Subsystem Design 11](#_Toc166398944)

[7. Glossary 11](#_Toc166398945)

[8. Bibliography 11](#_Toc166398946)

# Introduction

## Document Outline

* **Introduction**
* **System Overview**
* **Design Considerations**:
  + Assumptions and Dependencies
  + General Constraints
  + Goals and Guidelines
  + Development Methods
* **Architectural Strategies**:
  + Particular Product Used
  + Reuse of Existing Software Components
  + Future Plans for Extending or Enhancing the Software
  + Error Detection
* **System Architecture**:
  + Screen One
  + Screen Two
* **Policies and Tactics**:
  + Products Used in Creation
  + Coding Standards
  + Design Pattern
  + Requirements Traceability
  + Plans for Testing the Software
  + Plans for Maintaining the Software
* **Detailed System Design**:
  + Classification
  + Definition
  + Assumptions and Constraints
  + Uses/Interactions
  + Processing
  + Interface/Exports
  + Detailed Subsystem
* Bibliography

## Document Description

### Introduction

* **Purpose**:
  + The goal of the User Interface Development Team is to allow the instructor to have a more streamlined, easier to use user interface and to allow the use of a multiclass report generation.
* **Scope**:
  + The software goals align with the Business Goals to help overall productivity and to allow a more streamlined user experience. With this polished version of the UI, this will greatly impact the productivity of the instructors by allowing multiple comparisons at a time.
* **Intended Audience**:
  + Academic Affairs, Department Chairs and Program Coordinators.
* **Version Number**:
  + Version 1
* **Pertinent Documents**:
  + User Interface Requirements Document
* **Important Terms**:
  + UI – User Interface
  + NCC – Nashua Community College
  + CSV – Comma-Separated values or an Excel Spreadsheet
* **Summarized Document**:
  + This Design Document will go into detail as to how the User Interface team will achieve each of the requirements for this project.

### System Overview

The User Interface is an essential element in almost all modern-day software. In this updated software system, the User Interface will be updated to not only look a cut above the previous version but outperform it. This updated system will have the User input a canvas token key that is generated from their canvas account, which in-turn, will be authenticated by the system. Once authenticated, the User will then be allowed to either narrow down or broaden their search to exactly what is needed. Finally, after being precisely chosen, the User can then generate their report an access it in a trouble-free way. Our team’s updated system will mostly be seen by the user. However, the backbone of the system will be handled by the Computation and Visualization teams respectfully. Once integrated together, the user will have a completed system that works exceptionally well as a whole.

# Design Considerations

## Assumptions and Dependencies

* The User has previous knowledge of how to work with a Windows machine and how to access its file system.
* Assume how the User knows how to use a basic User Interface.
* Assuming the user has a general idea of what courses they are currently teaching.

## General Constraints

* No code that has been developed is currently present.
* We have base code, but it does not compile at the current moment.
* It will take time to develop an understanding of the current code once compiled.

## Goals and Guidelines

* **Simple by design and very intuitive**:
  + We want the user to have an easy time using the application.
* **Fast run time of the application**:
  + Waiting times for the reports to be generated is time consuming and costly. Shortening these wait times can save not only time but money.
* **Generated reports are easily Accessible in the file directory**:
  + Being able to find the generated reports will allow the User to have an overall easier experience.
* **The application does not consume memory**:
  + Not consuming too much memory is essential for allowing the User to use the application as much as they please with little issue.

## Development Methods

The method used is known as the Waterfall Development Method. This method consists of sequential phases that each focus on specific tasks. Before the next phase in the project can be focused on, the current phase must be completed.

# Architectural Strategies

* **Particular Product Used**:
  + C++ is the Main Programming Language
  + Boosted Libraries U sing C++
  + Visual Studio Integrated Libraries
  + BitBucket/Git Hub
* **Reuse of Existing Software Components**:
  + The current software update will include the previous team’s code.
* **Future Plans for Extending or Enhancing the Software**:
  + There are no future plans for extending or enhancing the software past the current team’s project time.
* **Error Detection**:
  + Error detection will be included every step of the way not only during the development time but also during the user’s experience to accurately give feedback.
* **Communication Mechanisms**:
  + Canvas

# System Architecture

As a whole, the system’s User Interface can be broken up into Two Screens with Nine Major Components:

* **Screen One**:
  + The First Screen consists of the Canvas API Key Text Field (First Major Component) and an Authentication Button (Second Major Component). The system will require a Canvas token key to be generated on their website. Once generated, the key will be inputted into the text field and then be authenticated once the authentication button is pressed. Fundamentally, the text field will communicate to the button that a key is inputted. Once the button is pressed, the system will then communicate with Canvas to authenticate the key; help will be provided to the user and an error will be thrown if any arise. After being authenticated, the system will then close screen one and move to screen two.
* **Screen Two**:
  + The Second Screen consists of the Add Course Number Text Field (Third Major Component), Add Instructor Text Field (Fourth Major Component), Add Semester Text Field (Fifth Major Component), Added Courses Text Field (Sixth Major Component), Search/Generate Report Button (Seventh Major Component), Select All (Eighth Major Component), and Remove All (Ninth Major Component). After the key is authenticated and moved onto the second screen, the user will then be presented with a slew on text fields, the first of which being the Add Course Number Text Field. This section is limited to a list of courses, related only to the user, returned from Canvas. The user will be presented a list of classes from a drop-down menu that can be chosen from or can enter their own course numbers, if previously known from the user. Each class that is selected will then be saved and added into the “Added Courses” text field. Another option to refine a user’s generated results is the “Add Instructor Text Field”. This field will be used to display a dropdown of all available instructors in relation to the currently selected courses. In the case of a department head, the dropdown of all professors within that department will be available for the user to select. However, if the user is any other professor or NCC faculty, the only available instructor will be themselves, making the field frayed out and unmodifiable. To refine the user’s generated results, an Add Semester text field has been added. This section will display a drop down of all available semesters, and once one of the semesters has been selected, all corresponding class within that semester will automatically be added to the Added Courses Text Field. In addition, if a semester is unselected from the dropdown, all classes corresponding to that semester will be removed from the Added Courses Text Field. After all desired information is inputted into the system, the user can then generate their report/reports through the Search/Generate Report Button. This button works by taking all the inputted information from the user and sending it to Canvas. Once sent, all the desired information will then be returned, and a report will then be formatted and generated. The second screen will stay up with all the previously selected options, to allow the user to continue creating reports. If wanted, the user can use the “Remove All” button to delete all previously selected options to essentially restart the second screen. Furthermore, the user can also use the “Select All” button to select every available course allowed to be accessed by that user.

# Policies and Tactics

* **Products used in creation**:
  + Visual Studio 2019
  + Boosted Libraries
  + Canvas API
  + Excel Spreadsheet
  + Windows 10 File Architecture
  + BitBucket / GitHub
* **Coding Standards**:
  + Coding Standards stated by NCC
* **Design Pattern**:
  + The Behavioral Design Pattern
* **Requirements Traceability**:
  + User Interface Requirements Document
* **Plans for Testing The Software**:
  + The current plan for testing the updated software is to not only test features as needed throughout the development phase, but to test the entire system once completed.
* **Plans for maintaining the software**:
  + There are no current plans to maintain the software past the project’s end date.

# Detailed System Design

## Classification

* Canvas API Key Text Field
* Canvas API Key Authentication Button
* Add Course Number Text Field
* Add Instructor Text Field
* Add Semester Text Field
* Added Courses Text Field
* Search/Generate Report Button
* Select All Button
* Remove All Button

## Definition

* **Canvas API Key Text Field**:
  + Holds the Canvas generated key inputted by the User.
* **Canvas API Key Authentication Button**:
  + Verifies that the Canvas generated key inputted into the “Canvas API Key Text Field” is valid. If it isn’t, an error is sent back to the user describing exactly what happened and how to fix it. After being authenticated, the screen is then switched, allowing the user to use the program.
* **Add Course Number Text Field**:
  + This field allows the user to select all of the classes they would like to have a report generated on. This field is required and cannot be left empty. All of the selected courses are then added to the “Added Courses Text Field”.
* **Add Instructor Text Field**:
  + This field allows the user to select any instructors that are of importance or related to the selected courses. In the case that the user is a department head, a dropdown of all professors within that department will be available to be selected. If the user is any other professor or NCC faculty, the only available instructor will be themselves, making it an unmodifiable field.
* **Add Semester Text Field**:
  + This field allows the user to refine their search and choose exactly what semester/semesters they would like the report generated from. This is a mandatory field.
* **Added Courses Text Field**:
  + All of the selected courses from the “Add Course Number Text Field” will appear in this field. This field will simply act as a holder for the fields and will not allow for input from the user.
* **Search/Generate Report Button**:
  + This button applies all of the selections from above into a predetermined format and generates a report from the information sent back from canvas.
* **Select All Button**:
  + This button selects all of the courses available to the user and adds them into the “Added Courses Text Field”.
* **Remove All Button**:
  + This button removes all changes that the user applied.

## Assumptions and Constraints

* **Canvas API Key Text Field**:
  + We are currently unsure how the Canvas API key works.
  + We are assuming it can be implemented easily.
* **Canvas API Key Authentication Button**:
  + We are currently unsure how the Canvas API key works.
  + We are assuming it can be implemented easily.
* **Add Course Number Text Field**:
  + Canvas might not grab all of the available course numbers.\
  + We currently do not have access to courses on Canvas through their API
  + We are assuming that the implementation of this won’t be too hard.
* **Add Instructor Text Field**:
  + We are currently unsure how the Canvas API key works.
  + We are assuming it can be implemented easily.
* **Add Semester Text Field**:
  + We are currently unsure how the Canvas API key works.
  + We are assuming it can be implemented easily.
* **Added Courses Text Field**:
  + The team is unsure if there is a way inside visual studio to duplicate text from one field to another. We are assuming that there is but are unsure at this time.
* **Search/Generate Report Button**:
  + The report being generated could not be generated.
  + The generated report could be lost and harder to find than wanted.
  + Canvas could be down and that would allow for the reports to not be generated.
  + We are assuming that this will be implemented with little to no problems.
* **Select All Button**:
  + This is one of the later items to be implemented so there is potential for this item to not be fully implemented by the due date.
  + The team is assuming we can implement it by the due date.
* **Remove All Button**:
  + This is one of the later items to be implemented so there is potential for this item to not be fully implemented by the due date.
  + The team is assuming we can implement it by the due date.

## Uses/Interactions

* **Add Course Number Text Field**:
  + This field will be used and analyzed by the Computation team to access the available courses.
* **Add Instructor Text Field**:
  + This field will be used and analyzed by the Visualization team to display the instructors name on a generated report if needed. In the case that the user is a department head, a dropdown of all professors within that department will be available to be selected. In the case of a professor or NCC faculty, the only available instructor will be themselves, changing the field to be unmodifiable.
* **Add Semester Text Field**:
  + This field will be used and analyzed by the Visualization team to display the selected semester and to allow the semester to be included in graphs. A dropdown box will be included to display all available semesters, and once one of the semesters is selected, all corresponding classes within that semester will automatically be added to the Added Courses Text Field. In addition, if a semester is unselected from the dropdown, all classes corresponding to that semester will be removed from the Added Courses Text Field.
* **Search/Generate Report Button**:
  + This field will be used by the Computation team to allow the save of data into a .csv file.

## Processing

A description of precisely how this component goes about performing the duties necessary to fulfill its responsibilities. This should encompass a description of any algorithms used; changes of state; relevant time or space complexity; concurrency; methods of creation, initialization, and cleanup; and handling of exceptional conditions.

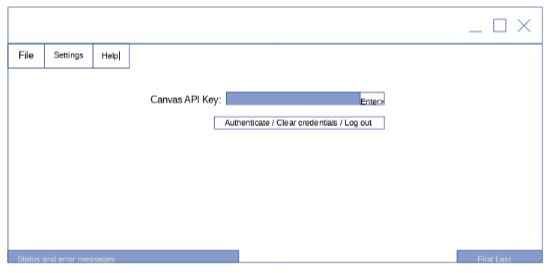
* **Canvas API Key Text Field**:
  + The duty of the Canvas API Key Text Field to hold the Canvas Key generated by the User’s Canvas Account. This key is an essential part to the entire system (without this key the program does not work). This key text field will interact with the Canvas API Key Authentication Button.
* **Canvas API Key Authentication Button**:
  + The duty of the Canvas API Key Authentication Button is to interact and authenticate directly with canvas. A request will be sent back to canvas through its API and will either send a true or false back as to whether the key has been authenticated or not. The authentication process will happen once the button is pressed. The process will interact directly with the Canvas API Key Text Field.
* **Add Course Number Text Field**:
  + The duty of the Course Number Text Field is to allow the user to input all desired courses from this field into the Added Courses Text Field. The user is allowed to manually type in each course number or select from a list that will be generated from the Canvas key. This field will have direct communication with Canvas to grab all available courses. There will also be active error checking to ensure all of the Course Numbers, either selected or entered manually are valid.
* **Add Instructor Text Field**:
  + The duty of the Add Instructor Text Field is to allow the user to select all available instructors that relate to that canvas account. In the case that the user is a department head, a department will be available to be selected. If the user is any other professor or NCC faculty, the only available instructor will be themselves, making the field unmodifiable. This information will be grabbed from the inputted Canvas key and the courses selected. There will be error checking to ensure each instructor inputted is valid. If an instructor is chosen and a course has not been taught by that instructor, no report will be generated, and the user will be notified.
* **Add Semester Text Field**:
  + The duty of the Add Semester Text Field is to allow the user to select one available semester from a drop-down menu of all available semesters. The information will be grabbed from the inputted Canvas key and the courses selected. This field is not a required field and will be manually inputted with the User’s most recent semester. There will also be error checking to ensure each class lines up with their respected semester. If a semester is chosen and a course has not been held during that time, no report will be generated, and the user will be notified.
* **Added Courses Text Field**:
  + The duty of the Added Courses Text Field is to display to the user the active courses selected by the user. This field will have direct contact with the Add Course Number Text Field and simply mirror, on a larger scale, what has been inputted by the User. This will also allow User’s to check each of their selected courses one last time before the report is generated.
* **Search/Generate Report Button**:
  + The duty of the Search/Generate Report Button is to allow the user to take all of their inputted information and to generate a report based off of the inputted information. This button will directly interact with the Visualization and Computation teams efforts to generate a formatted a streamlined experience for the user. This generated report will be allowed to be found easily by the user.
* **Select All Button**:
  + The duty of the Select All Button is to add every single Course Number available to the user. Once the key has been inputted, the button will then get a list of all available classes.
* **Remove All Button**:
  + The duty of the Remove All Button is to remove all the information inside of the Add Course Number Text Field, Add Instructor Text Field, Add Semester Text Field and Added Courses Text Field. This will happen by having the button directly have access to all the fields and delete everything once inside of the fields.

## Interface/Exports

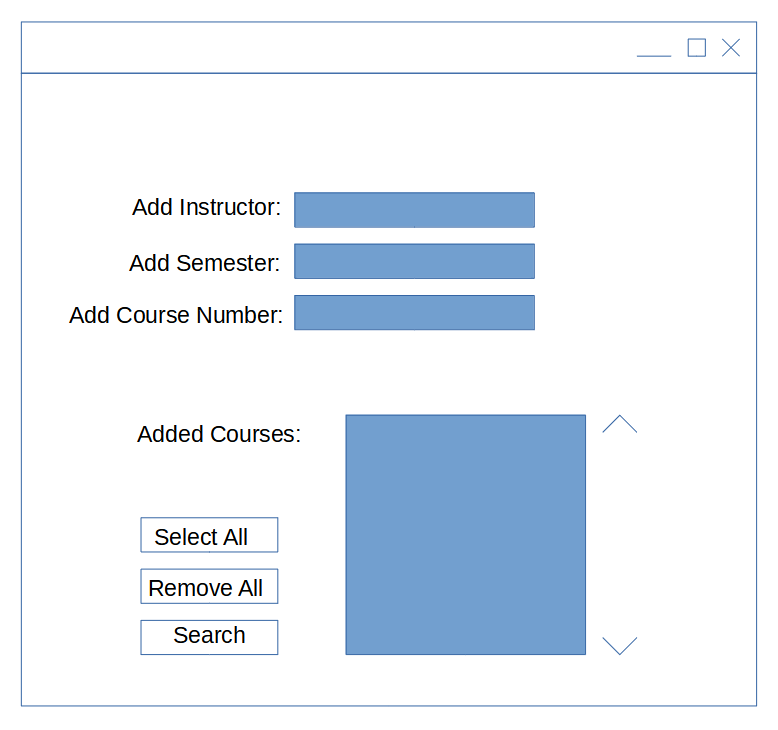
There is no Current code developed since the previous team. The code that was provided from the previous team has had little to no comments added. This point will be updated once the previous team code has been commented.

## Detailed Subsystem Design

* **Screen One**:

****

* The first screen has two key components, the Canvas API Key text field, and the Canvas API Key Authentication Button. The user will grab their Canvas API Key from their Canvas account and input it into the text field. After the key is inputted, the user will then press the Canvas API Key Authentication Button to authenticate their key. After the key has been authenticated, the user will then be sent to the second screen, closing the first.
* **Screen Two**:



* The second screen contains seven key components, the Add Instructor Text Field, Add Semester Text Field, Add Course Number Text Field, Added Courses Text Field, Select All Button, Remove All Button, and Search/Generate Report Button. The user can add any related Instructors, Semesters and courses they would like to be added to the generated report. The added courses will be displayed in the “Added Courses Text Field”. The user can choose to select all available courses or to remove all available changes. After the user gets the exact selections they would like, they can then Search/Generate the report.

# Bibliography

A list of referenced and/or related publications.

Brad Appleton <brad@bradapp.net>

http://www.bradapp.net