**iGrad On-Line Application Progress Bar**

**System Requirements Specifications**

**Version 1.0**

**May 15th, 2018**

**Project Team – Explosive Salsa:**

Duc Nguyen

Jeff Pratt

Kevin Locke

**Document Author(s):**

Duc Nguyen

**Customer Representative(s):**

Carol Cleveland, Principal

Mary Anderson, Principal assistant

1. **Introduction**

This system is an online application for iGrad, it provides an efficient and simplified substitution to the original 40-pages paper application for students applying to iGrad. The progress bar acts as a visual representation of student’s completion status on a form. This document describes the requirements of this progress bar.

1. **Functional Requirements**

**FR0. Progress Bar Initialization**

* FR0.1. On New Form

The progress bar shall be empty in new forms on load. The progress bar starting value shall be 0, it shall not go lower than 0 and shall not increase (+1) until user make valid changes to a required input field.

* FR0.2. On Form in Progress of Completing

The progress bar shall not be empty in forms that are currently in progress of completing. The progress bar starting value shall be calculated based on the completed required fields on ready state of the form. The progress bar value shall have the same value as the total completed required fields, it shall not increase (+1) until user make valid changes to a required input field, and it shall not decrease (-1) until user make invalid change to a required input field.

* FR0.3. On Completed Form

The progress bar shall have full value and shall be fully colored. The progress bar full value shall be equal to the total validly completed required fields in the form. The progress bar value shall not decrease until user make invalid change to a required input field, and the progress bar shall not increase until user make valid change to a required input field.

**FR1. Progress Bar Value Calculation Equation**

FR1 describes the equation for calculating the progress and its components

* FR1.1. Progress Calculation Components

FR1.1.1. Total Required Fields (TRF)

TRFs value shall be the total number of the HTML class “required-marker” that are *visible* in the current form.

FR1.1.2. Valid Inputs Count (VIC)

This value shall be the total number of valid user inputs associating with the “required-marker” fields. IVC shall not be lesser than 0 nor shall it be greater than TRFs. IVC shall increase by 1 when user make valid change to a required input field, and IVC shall decrease by 1 when user invalid change to a required input field.

FR1.1.3. Current Progress (CP)

CP shall be calculated using TRF and VIC.

* FR1.2. Progress Calculation Equation

The progress bar current value (CV) shall be calculated using the equation below, it calculated CP value shall be in percentage (See FR2). CP value shall not be more than 100%.

CP = (VIC / TRF) \* 100

**FR2. Displaying Progress Bar Value**

FR2 describes the rules for displaying the progress

* FR2.1. Progress Bar Components

FR2.1.1. Bootstrap Progress Bar

The Bootstrap Progress Bar shall act as a container.

FR2.1.2. Completed Progress Indicator

The Completed Progress Indicator shall be contained by the Bootstrap Progress Bar. It shall increase or decrease base percentage value of CP. The maximum value of the Completed Progress Indicator shall be 100%.

FR2.1.3 Completed Fields Progress Indicator

The Completed Fields Progress Indicator shall showcase how many fields user have completed out of the TRF.

* FR2.2 Displaying Progress Bar Components

FR2.2.1. Bootstrap Progress Bar

The Bootstrap Progress shall be displayed on top of the form. Its width shall be the full length of the form and it shall have a grey background color.

FR2.2.2. Completed Progress Indicator

The Completed Progress Indicator have a green background color. It shall be contained within the Bootstrap Progress Bar and Its width shall be equal to calculated value of CP. It shall be visible only when the calculated value of CP is greater than 0% and its width shall not be greater than 100%.

FR2.1.3 Completed Fields Progress Indicator

The Completed Fields Progress Indicator shall be displayed as white text. It shall not be displayed until calculated CP value is greater than 0% and Completed Progress Indicator width has been assigned probably base on CP.

**FR3. Tracking Changes of Required User Inputs**

FR3 describes the “validity” of user inputs and how they are tracked

* FR3.1. Valid user inputs

User inputs shall be qualified as “valid” when its length is greater than 0 and it passes jQuery valid() function. User shall be qualified as “invalid” when its length is lesser than 0 and it does not pass JQuery valid() function.

* FR3.2 Tracking User Inputs

FR3.2.1. On page ready state

Each field with “required-marker” or “required-checker” (See FR3.3) HTML class shall have its input be verified for validity. Each valid field shall increment VIC by 1. When all fields are checked, then CP shall be calculated to display on Progress Bar.

FR3.2.2. On input change state

A change to a required input shall be verified for validity. If the current state of the field is invalid and user make valid change then VIC shall increment by 1. If the user current of the field is valid and user make invalid change then VIC shall decrement by 1. If the current state of the field is valid and user make valid change then VIC shall not change.

* FR3.3 Tracking different required input types

FR3.3.1. Required fields

A field shall be considered as required base on iGrad representatives and thus shall be made required for user to provide that information. These fields will have label with “required-marker” HTML class, and depends on its type shall be tracked differently (See FR3.3.2)

FR3.3.2. Required input types

FR3.3.2.1. Text, textbox, email, date pickers, number, and URL Types

Input fields with these types shall be validated by accessing their current value and verified using JQuery valid() function. These fields shall have required attribute.

FR3.3.2.2. Radio buttons and Check boxes Type

These two types shall have hidden input field with “required-checker” HTML class associating with the input group, of which current value attribute shall be updated based on user’s selection of the radio button group.

FR3.3.2.3. Partial views and Fields that may have multiple inputs

These types shall have hidden input field with “required-checker” HTML class and the hidden field shall be assigned with a specific name. An array shall be used to store the user inputs, and said array shall be set as the hidden field’s value based on user’s changes. User changes to these types shall be add/remove appropriately from the array, and its length shall be used to verify if the hidden field value is valid or not.

1. **Nonfunctional Requirements**

**NR1. Performance**

The system shall track all changes user make on input fields within a form, be It required or not. Upon change is detected, CP value shall be recalculated; the newly calculated CP value shall be used to display Completed Progress Indicator and Completed Fields Progress Indicator.

NR1.1. Input on change:

The system shall perform the calculation based on change of a required field within 0.01 seconds after the change was made.

NR1.2 Update Completed Progress Indicator and Completed Fields Progress Indicator

The system shall update the Completed Progress Indicator and Completed Fields Progress Indicator within 0.01 seconds after the change has been made and the changed field is out of focus.

**NR2. Usability**

A user shall be able to determine quickly what are their current progress in completing the form

NR2.1. Completed Progress Indicator

The green background of the Completed Progress Indicator shall provide the sense of completion for the user. Its percentage shall let user visualize their completion status.

NR2.2. Completed Fields Progress Indicator

The white text on the green background shall show user exactly how many fields they have completed out of the total required fields. Its numeric value shall let user know exactly how many fields they have completed.

NR2.3. User errors

The system shall catch improper input from all text fields in the system. It shall prompt them with error messages after having verified these inputs as invalid

1. **Constraints**

jQuery codes have to work with the C# .NET framework Razor Notation

All testing shall be done via Selenium and user testing

1. **Requirements Dependency Traceability**

* No action to progress bar and the calculations of its value shall be done until a change has been made to a required input field
* When a change has been made to a required input field
  + Verify the validity of the change
  + Increment/decrement VIC accordingly
  + Calculate new CP
  + Update Completed Progress Indicator and Completed Fields Progress Indicator
  + Display Completed Progress Indicator and Completed Fields Progress Indicator

1. **Development and Target Platforms**

* Web platform Chrome, Firefox, and Safari
* Visual studio 2017