# Honors Program Degree Evaluation

Test Plan

CPSC-430: Writing Assignment 3

Dr. Anewalt

10 November 2017

Aaron Dyke, Adam Hurnyak, Andrew Woodruff, Daniel Adams

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to define and organize the test strategy for the University of Mary Washington (UMW) Honors Program Degree Evaluation (HPDE) program for Computer Science (CPSC)-430. The audience of this document includes the writers' professor Dr. Karen Anewalt, the client Jeanne Campbell, and the team testing the software application.

## 1.2 Scope

The scope of this project is to allow UMW honors students to more easily view their honor's checklist and to improve communication between the client and the student. The client desires this program in order to lower her workload and allow for her to more easily manage her honors students. This is accomplished by automating some tasks normally performed by the client, such as the providing of honors checklists to students. For the students this would be an easy and user friendly way to check their progress within the honors program without having to contact the client directly. It would also be a resource for help and information about the UMW honors program in general.

## 1.3 Overview of this Document

Section 2 of the document contains a description of the project. It describes a high-level overview of the product's functionality, functional requirements.

Section 3 is the test plan, which will describe our overall testing strategy and the project management issues that are required to properly execute effective tests.

Section 4 is the testing procedure, which describes a detailed test procedure including test tactics and test cases for the software.

Section 5 is the appendix. This section contains the glossary for the document, the contributions, and references used to create the document.

# 2. Project Overview

## 2.1 Product Description

This product should meet the needs of the client to provide direct information to students within the UMW Honors Program. Currently, the client tracks UMW Honors Program candidates' progress manually using a hard-copy checksheet, which follows each student through their course of study. She also uses an Access Database which must be concurrently maintained using the checksheet, Banner information, and data imports from a Microsoft Excel Comma Separated Values (CSV) format. This new application must provide students with direct access to their Honors Program status and progress. For the purpose of this project, UMW's

account system, Banner, will continue to maintain student identification numbers and Grade Point Averages (GPAs). These items will not be tracked in this application.

The system must be a web application accessible over the Internet with a simple user interface, CSV reading, and data storing capabilities. Students and administrators must be able to securely log into and out of their accounts using their UMW email as their login name. There should also be a password changing functionality within the system. Password recovery should be implemented if time permits. The system should provide error handling if users enter wrong login, search, or Uniform Resource Locator (URL) information. It should deny access to pages users do not have permission to visit, or any pages apart from the login screen if proper login credentials have not been entered.

After login, users will observe a landing page containing announcements posted by the administrator along with any other relevant information. There will be a menu bar with buttons linking to an honors FAQ screen, password change screen, and a screen where a student can view their honors checklist. The system displays different options in the menu bar depending on if the user is a student or administrator. If the user is a student, they will see an option to download their honors checklist as a PDF. If user is an administrator, they will have options in their menu bar to create and post announcements, search for a specific student's checklist, or upload a CSV file.

Administrator roles may see menu options to Search Checksheet, Broadcast Announcement, Import Data, and Manage Accounts. Search Checksheet should provide a textbox to search for a student's email or student name and display that person's checksheet data. The Broadcast Announcement page should display options to turn on or off an announcement to all users, with the option to type and save one announcement. Import Data should offer either web page checksheet or an Excel import which provides a workflow that allows an admin to upload, parse, and update user data from an Excel or CSV file format. If importing a file, after the file is read into the system, the page should display the results of its readings and users to match the data. Once the results are verified, the user may submit the form.

## 2.2 Functional Requirements

- 2.2.1 As a User, I want to be able to securely log in.
  - 2.2.1.1 Make passwords and transport encrypted.
  - 2.2.1.2 Make a login page.
- 2.2.2 As an Administrator user, I want to be able to upload a .csv file to the database.
  - 2.2.2.1 Make page for uploading .csv file.
  - 2.2.2.2 Make page only reachable to Administrator users.
  - 2.2.2.3 Add new users from the .csv file to users table.
  - 2.2.2.4 Add new student checklists from .csv to checklist table.
  - 2.2.2.5 Send email to create account for new users.

2.2.3 As a Student user, I want to be able to view my current records.
    2.2.3.1 Make page to view checklists.
    2.2.3.2 Populate page with checklist information for the user logged in.
2.2.4 As an Administrator user, I want to be able to search for any student's records.
    2.2.4.1 Make a checklist search page.
    2.2.4.2 Make checklist information searchable by name or email.
    2.2.4.3 Make page only reachable to Administrator users.
2.2.5 As an Administrator user, I want to be able to post announcements to the landing screen.
    2.2.5.1 Make an announcement form page.
    2.2.5.2 Create a table for storing announcements in the database.
    2.2.5.3 Display a number of the most recent announcements to the landing screen.
2.2.6 As a Student user, I want to be able to view a help window with FAQs.
    2.2.6.1 Create the help page.
    2.2.6.2 Display contact information and FAQs.
    2.2.6.3 Allow Administrator users to update the FAQs.
2.2.7 As a Student user, I want to be able to download my Honors checklist as a PDF.
    2.2.7.1 Create download option on the checklist view page.
    2.2.7.2 Create a human readable PDF version of the checklist for the download.
2.2.8 As a User, I want to be able to change my password through the application.
    2.2.8.1 Create a password recovery system for users.

# 3. Test Plan

## 3.1 Testing Strategy

The overall strategy for testing will be black box validation tests for each system of the software. Features that will be tested via validation include the login of admin and student users, posting/editing/deleting announcements and frequently asked questions, uploading of the checklist .csv file, searching checklists, and viewing checklists.

## 3.2 Testing resources and staffing

A computer with access to the internet is necessary for testing. Preferably each tester should have their own computer with internet access for the purposes of stress testing the system. At least one tester will need access to a text editor on their computer such as Notepad or Sublime. All testers will need an email account that they can access for the purposes of testing certain functionality of the application.

To test the application, testers must navigate to *[website url]*. Administrator login credentials are provided in the table in section 4.1. The integration team will also supply a template .csv file that the test team will modify and use to perform certain tests. Instructions on how to modify the .csv file will also be provided.

### 3.3 Test work products

Work products that will be produced as a result of testing are new user accounts and new entries in the student database. New user accounts will be created when uploading the test .csv files. For each new user account, a respective entry in the student database will be created. The successful creation of these accounts and entries will determine if the product's functionality is working as intended.

Bugs and other errors will be tracked using extensive error checking which has already been implemented by the integration team. A server log will be maintained during the testing process which will contain errors for any bugs or issues that have yet to be encountered by the integration team. We also ask that the testing team make note of actions that they performed which rendered an error so that we can properly fix the code which caused the error.

### 3.4 Test record keeping

Record keeping of the tests performed shall be entered, by each tester, into a corresponding google form (found at this URL https://goo.gl/forms/OBWyb76eUmfulOBD3). Each tester should open up the form and fill out the questions associated with the test they are performing. The test number section of the form corresponds to the test number found in the table in section 4.1 of this document. The results of the form will be exported to an excel sheet for the implementation team to review.

### 3.5 Test schedule

For our test schedule, a total of two days will be allotted for the testing phase. The first day, November 13th, we will exchange information that is required to start testing, such as username information required to test certain aspect and the test plan document. Wednesday, November 15th, will be reserved for the actual testing since we do not predict this to take too long.

## 4. Test Procedure

### 4.1 Unit Tests

**Bolded** test numbers denote tasks that can only be completed using the provided administrator account. *Italicized* tasks can only be completed when logged in with a student user account.. Tasks that are not bolded or italicized can be completed using either account class.

| Test Number | Related Requirements | Purpose | Test Case Data | Expected Results |
|---|---|---|---|---|
| 1 | 2.2.1 | Test Login | Username: | Redirected to |

| | | | | |
|---|---|---|---|---|
| | | | test_user<br>Password:<br>430Honors | personalized landing screen |
| 2 | 2.2.6 | Test view *About* page | None | Proper display of *About* page |
| 3 | 2.2.6 | Test view *FAQ* page | None | Proper display of *FAQ* page |
| 4 | 2.2.5 | Test viewing all announcements | None | Proper display of all announcements |
| **5** | 2.2.5 | Test *Post Announcements* | None | Announcement is posted successfully |
| **6** | None | Test *Edit Announcement* | None | Announcement is changed successfully |
| **7** | None | Test *Remove Announcement* | None | Announcement is removed successfully. |
| **8** | 2.2.2 | Test *Upload Degree Evaluation Checklists* | File: test_plan.csv | Upload successful message appears |
| **9** | 2.2.4.1, 2.2.4.2 | Test *Search for Student Checklists* | Search a user from the provided .csv file. | Proper display of checklist for searched for student |
| 10 | 2.2.2.5 | Verify emails sent to users in provided .csv file | None | An email with login credentials was sent to all users in the .csv file |
| 11 | 2.2.1 | Test logging out | None | Redirected to the login screen upon clicking the *Logout* button |
| *12* | 2.2.1 | Test logging in as student user | None | Redirected to landing screen with *Administrator* |

| | | | | |
|---|---|---|---|---|
| | | | | *Resources* not visible, but *View Checklist* is visible |
| *13* | 2.2.2.2, 2.2.4, 2.2.5.1 | Test student restricted access | None | No access to any administrator pages, specifically */upload, /announcements, /search, /editannouncement* |
| *14* | 2.2.3 | Test student view of records | None | Display student checklist for xxxxxxx |
| 15 | 2.2.8 | Test *Password Change* by changing password, logging out, and attempting to log in with new password | Username: test_user Password: [new user-chosen password] | Password was changed successfully |

# 5. Appendix

## 5.1 Glossary

| | |
|---|---|
| CPSC | Computer Science |
| CS Server | The server run by the UMW computer science department. |
| CSV | Common Separated Values. File format commonly used for Microsoft Excel spreadsheets. |
| FAQ | Frequently Asked Question |
| GPA | Grade Point Average |
| HPDE | Honors Program Degree Evaluation |

| | |
|---|---|
| SHA256 | Type of cryptographic hash algorithm used to obfuscate passwords in the database. |
| UMW | University of Mary Washington |
| URL | Uniform Resource Locator. Address belonging to a website (e.g. https://google.com) |

### 5.2 Contributions

Portions of sections 1-2 were borrowed from the previous groups' requirements documents; however, significant changes were made to some of the sections as indicated by the contributions section in the table below.

| Team Member | Section Contributions |
|---|---|
| Daniel Adams | 1.1, 1.2, 1.3, 2.1, 2.2, 3.2, 3.3, 4.1 |
| Aaron Dyke | 1.1, 1.2, 1.3, 2.1, 2.2, 3.4 |
| Adam Hurnyak | 1.1, 1.2, 1.3, 2.1, 2.2, 3.5 |
| Andrew Woodruff | 1.1, 1.2, 1.3, 2.1, 2.2, 3.1, 4.1 |

### 5.3 References


1. https://docs.google.com/document/d/1S-1STP_sCuxmb5ILrMOsNZD6cxxZBznr1JCnEUN3-Cw
2. https://docs.google.com/document/d/1PNGKO7v_7MehQq2s-IzD2_T3wb6pVc7qHqGBTg0dIDA
3. Tester form https://goo.gl/forms/OBWyb76eUmfulOBD3