Honors Program Degree Evaluation

Project Plan

CPSC-430: Writing Assignment 2
Dr. Anewalt
9 October 2017
Aaron Dyke, Adam Hurnyak, Andrew Woodruff, Daniel Adams

1. Introduction **2**
1.1. Purpose 2
1.2. Scope 2
1.3. References 2
1.4. Overview of this Document 2

2. Project Description **3**
2.1 Product Overview Functions 3
2.2 User Characteristics 4
2.3 Functional Requirements 4
2.4 General Constraints 5

3. Project Schedule **6**
3.1 Approach 6
3.2 Milestones and Deliverables 6
3.3 Work Breakdown Structure 7
3.3.1 Milestone 1 8
3.3.2 Milestone 2 8
3.3.3 Milestone 3 8
3.3.4 Milestone 4 9
3.3.5 Milestone 5 9
3.4 Gantt Chart 10
3.4.1 - Task List 10
3.4.2 - Gantt Chart 10
3.5 Task Dependency Diagram 11

4. Appendix **11**
4.1 Glossary 11
4.2 Contributions 12
4.3 References 12

1. Introduction

1.1. Purpose

The purpose of this document is to define and organize the requirements 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 implementing the desired 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. References

This document contains multiple acronyms that may be confusing to non-technical readers. Definitions of these acronyms can be found in Appendix B, and their definitions were quoted directly from their respective wiki pages, courtesy of wikipedia.org.

1.4. 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, user characters, functional requirements, and general constraints.

Section 3 is the project schedule which describes the team's approach to the project, milestones and deliverables, and a gantt chart and task dependency diagram.

Section 4 is the appendix. This section contains the glossary for the document, the contributions, and references used to create the document.

2. Project Description

2.1 Product Overview Functions

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 (Appendix B.2), 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 User Characteristics

This application must serve the interests of two classes of users, administrators and students. Administrators and students will have separate tools available on the site, along with specific shared functions. Students and administrators should have the ability to log into the system with a username and password combination and be able to change their password. Both users may view general announcements that the application displays and view checksheet information from uploaded Excel sheets. Both user classes can view an About or Help page with information about the UMW Honors Program itself.

Once logged in to the system, the administrative user can create and post announcements through the application which are seen by all users on the landing screen. The administrator can also navigate to a screen where they alone can upload a CSV file containing the information of all UMW honor students into a database accessible by the application. The system will detect new users based on this upload and send them emails to create an account in the system. The administrator can also search for and display the honors checklist for specific students using a search feature..

The students have access to a limited range of functions in the application. Once logged in, student users can view their own honors checklist within the application. Depending on time, students may also be able to download copies of their checksheet in the form of a PDF for their personal record.

2.3 Functional Requirements

2.3.1 As a User, I want to be able to securely log in.

2.3.1.1 Make passwords and transport encrypted.

2.3.1.2 Make a login page.

2.3.2 As an Administrator user, I want to be able to upload a .csv file to the database.

2.3.2.1 Make page for uploading .csv file.

2.3.2.2 Make page only reachable to Administrator users.

2.3.2.3 Add new users from the .csv file to users table.

2.3.2.4 Add new student checklists from .csv to checklist table.

2.3.2.5 Send email to create account for new users.

2.3.3 As a Student user, I want to be able to view my current records.

2.3.3.1 Make page to view checklists.

2.3.3.2 Populate page with checklist information for the user logged in.

2.3.4 As an Administrator user, I want to be able to search for any student's records.

2.3.4.1 Make a checklist search page.

2.3.4.2 Make checklist information searchable by name or email.

2.3.4.3 Make page only reachable to Administrator users.

2.3.5 As an Administrator user, I want to be able to post announcements to the landing screen.

2.3.5.1 Make an announcement form page.

2.3.5.2 Create a table for storing announcements in the database.

2.3.5.3 Display a number of the most recent announcements to the landing screen.

2.3.6 As a Student user, I want to be able to view a help window with FAQs.

2.3.6.1 Create the help page.

2.3.6.2 Display contact information and FAQs.

2.3.6.3 Allow Administrator users to update the FAQs.

2.3.7 As a Student user, I want to be able to download my Honors checklist as a PDF.

2.3.7.1 Create download option on the checklist view page.

2.3.7.2 Create a human readable PDF version of the checklist for the download.

2.3.8 As a User, I want to be able to change my password through the application.

2.3.8.1 Create a password recovery system for users.

2.4 General Constraints

Due to the nature of this class, the largest constraint is the short time table for development. The development team will have only a few weeks to design, and program the requirements listed in section 2.3. Similarly, the budget for this project is $0. Any program or software locked behind a paywall will not be utilized. Due to security concerns, UMW's existing Banner resources will not be integrated nor will the system use the UMW single sign-on system.

3. Project Schedule

3.1 Approach

The University of Mary Washington Honors Program Degree Evaluation website will be developed using two major stages, “barebones” creation and feature adding. “Barebones” creation refers to creating a basic website with the minimal functionality that the client desires. Account creation, login, view records, and uploading of CSV files will be features included in the “barebones” stage. These three features create the minimal features for the project. The feature adding stage will include implementing the features stated by the client as medium-low priority.

The development of the project will start on the free cloud hosting platform Cloud9. This hosting platform allows one to use a small cloud server to develop web applications. It has a collaboration feature which allows teams to simultaneously edit files avoiding the need to merge using version control. Development will begin using Cloud9 due to restrictions on the CS server not allowing students to make/host databases. The goal will be to migrate the “barebones” website from Cloud9 to the CS server once the team receives database permissions on the CS server. This migration will take time, however, in the long run, development of all features can start earlier when using Cloud9.

From the start of the project, the team will be using the distributed version control known as Git. This version control is widely used in industry and the UMW computer science curriculum. The team chose to use Git due to our familiarity with it as well as the possibility to open source our code by hosting a repository on the website GitHub.

The team will be working collaboratively using a modified version of the Scrum framework for development. Our goal is to meet one to two times a week for standups, where we discuss our progress on sprints. A sprint is is a set of tasks that lead to a deliverable. We will be tracking sprints and all the associated tasks with each sprint using the project management platform Taiga. This service lets the team create and manage a backlog of tasks related to the functional requirements of the project. Using these organizational tools the team hopes to create deliverables roughly every 2 weeks.

3.2 Milestones and Deliverables

Our first milestone, after we get a skeleton of the website, will be to allow the admin to upload a .csv file into the database that we will set up prior. To set this up we will need to create a specific page that only the administrator will have access to, from which they will be able to upload a .csv file to our database. To get to reach this page from the landing screen, we will also create a button that can only be viewed if logged in as an administrator. Once the page and button are set up, our next step will be to allow the admin to add new users from the .csv file to the users table. If the transfer to the database recognizes a new entry, meaning a new student, then an email will be sent to the student, informing them that an account has been created for them in the system.

Our second milestone will be to allow the users to securely log in. This milestone should be fairly simple compared to the previous milestone. We will create a login html page which a user will see when they initially visit the site. After doing this, we will want to make sure that when a user types in a password, the password will be salted and encrypted using SHA256 in the database.

The third milestone will allow the student user to view their current records and allow the administrator user to search for any student's Honors checklist. The reason we are grouping these two in the same milestone is because both of them will just require some database statements and will return a simple query. Also, both of these rely on having the first milestone setup. The first three milestones will be a part of our first deliverable because these milestones consist of the bare minimum functionality of the program requested by the client.

The fourth milestone will be adding the ability for the administrator to post announcements of upcoming events/deadlines to the landing screen. This will allow the administrator to get important information out faster to the students without having to email each one individually. Along with posting announcements, this milestone should also include the ability for the users to change their login password. Both of these are of lower priority from the previously mentioned task, which is why they are included in a later milestone. This will also be our second deliverable.

The final milestone and deliverable will be to allow a student user to download their Honors checklist as a PDF and to implement a help window with Honors checklist FAQ in it. These two task are near the bottom of the priorities list, so they will be implemented if time permits.

3.3 Work Breakdown Structure

The Work Breakdown Structure displays the time and dependencies required for each of the separate tasks for each milestone. The milestones in the breakdown structure have a start date, which is the date at which all dependencies required to start tasks in that milestone are scheduled to have been completed, an end date, which is the date at which work on that milestone is scheduled to conclude based on time required to complete each task, and a number of workdays, where a day constitutes 2 to 4 hours of work. Milestones 1 through 3 are all linear in implementation and rely on all the previous milestones and tasks to be completed prior to the start of the next milestone. Milestones 4 and 5 have no dependency between each other and can be implemented in any order or concurrently. As stated in section 3.2 the features being implemented are considered additional to the core features that will be completed by milestone 3.

3.3.1 Milestone 1

Start date 10/9/2017, End date 10/16/2017; 6 days

3.3.1.1 Setup web server, 1 day

3.3.1.2 Setup version control, Create HTML for landing screen, Create database containing tables for user info, user login, and announcements; 1 day

3.3.1.3 Create admin account, Create button in HTML to reach .csv upload screen; 1 day

3.3.1.4 Create page in HTML where administrator uploads .csv file; 1 day

3.3.1.5 Write back end code which takes tables from the .csv file and populates a table in the database; 2 days

3.3.2 Milestone 2

Start date 10/17/2017, End date 10/19/2017; 3 days

3.3.2.1 Write back end code which determines which users are new to the database; 1 days

3.3.2.2 Write back end code which sends an email with an invitation to log into the system to users who are new to the system; 2 days

3.3.3 Milestone 3

Start date 10/19/2017, End date 10/25/2017; 4 days

3.3.3.1 Create HTML for search page, Create HTML button which links to search page, Write front end code which takes the user input in the form of a string which is the student to search for; 1 day

3.3.3.2 Write back end code which queries the database using the string entered by the administrator, Write front end code which displays honor’s checklist for the user who was search for, Write back end code to handle errors arising from searching for a user who does not exist, Write front end code which displays an error message when no results are found for the user who was searched for, Create HTML page where Honor's checklist is displayed; 1 day

3.3.3.3 Create button in HTML which takes user to a screen which displays their Honors checklist, Write back end code which fetches information from the database tables depending on the logged in user; 1 day

3.3.3.4 Write front end code which is populated by the information retrieved from the database; 1 day

3.3.4 Milestone 4

Start date 10/25/2017, End date 11/7/2017; 5 days

3.3.4.1 Create button in HTML which takes user to password reset screen; 1 day

3.3.4.2 Write HTML for password reset screen which contains two text fields where the user enters a new password, Write back end code which ensures passwords entered by the user match and if so, update the user table with the new passwords, Write Html which indicates to the user that password change was successful, Write HTML that indicates to the user that the passwords entered don't match; 2 days

3.3.4.3 Create announcement form page, Create announcements table in database, Create button in HTML what allows the administrator to create their announcement; 1 day

3.3.4.4 Display X number of the most recent announcements on frontpage; 1 day

3.3.5 Milestone 5

Start date 10/25/2017, End date 11/3/2017; 6 days

3.3.5.1 Create button in HTML which user can click to download Honors checklist as a PDF; 2 days

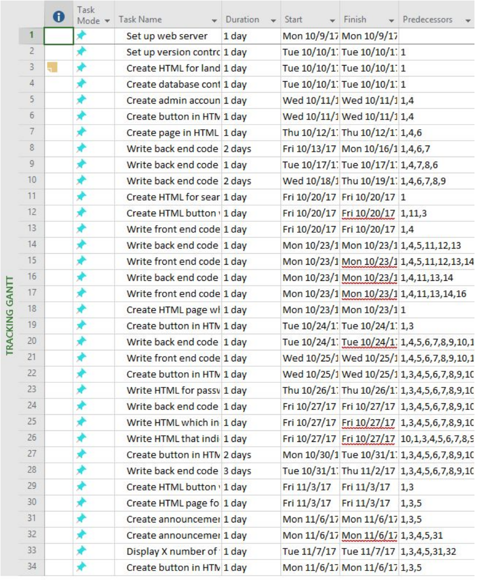
3.3.5.2 Write back end code which fetches the user's information from the database and converts it into a PDF which automatically downloads; 3 days

3.3.5.3 Create HTML button which takes user to Honors checklist FAQ screen, Create HTML page for Honors Checklist FAQ which will be populated by information provided by either the Client or the UMW Honor's website; 1 day

3.4 Gantt Chart

3.4.1 - Task List

Below is a tentative list of all tasks necessary to complete the project. The table contains the name and description of each task, its estimated duration, the day it will be completed, and a list of predecessor tasks if any exist.

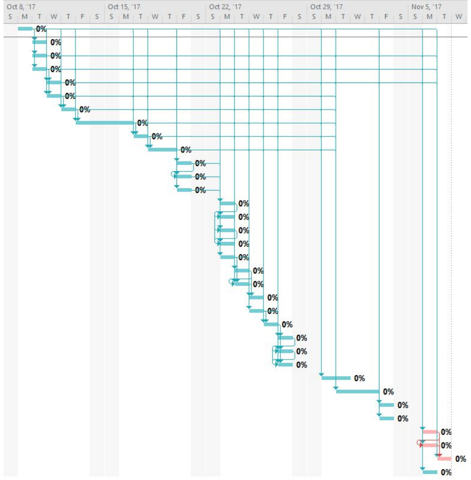


TRACKING GANTT

	ⓘ	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		📌	Set up web server	1 day	Mon 10/9/17	Mon 10/9/17	
2		📌	Set up version contrc	1 day	Tue 10/10/1	Tue 10/10/1	1
3	🗒	📌	Create HTML for land	1 day	Tue 10/10/1	Tue 10/10/1	1
4		📌	Create database cont	1 day	Tue 10/10/1	Tue 10/10/1	1
5		📌	Create admin accoun	1 day	Wed 10/11/1	Wed 10/11/1	1,4
6		📌	Create button in HTN	1 day	Wed 10/11/1	Wed 10/11/1	1,4
7		📌	Create page in HTML	1 day	Thu 10/12/1	Thu 10/12/1	1,4,6
8		📌	Write back end code	2 days	Fri 10/13/17	Mon 10/16/1	1,4,6,7
9		📌	Write back end code	1 day	Tue 10/17/1	Tue 10/17/1	1,4,7,8,6
10		📌	Write back end code	2 days	Wed 10/18/1	Thu 10/19/1	1,4,6,7,8,9
11		📌	Create HTML for sear	1 day	Fri 10/20/17	Fri 10/20/17	1
12		📌	Create HTML button	1 day	Fri 10/20/17	Fri 10/20/17	1,11,3
13		📌	Write front end code	1 day	Fri 10/20/17	Fri 10/20/17	1,4
14		📌	Write back end code	1 day	Mon 10/23/1	Mon 10/23/1	1,4,5,11,12,13
15		📌	Write front end code	1 day	Mon 10/23/1	Mon 10/23/1	1,4,5,11,12,13,14
16		📌	Write back end code	1 day	Mon 10/23/1	Mon 10/23/1	1,4,11,13,14
17		📌	Write front end code	1 day	Mon 10/23/1	Mon 10/23/1	1,4,11,13,14,16
18		📌	Create HTML page wl	1 day	Mon 10/23/1	Mon 10/23/1	1
19		📌	Create button in HTN	1 day	Tue 10/24/1	Tue 10/24/1	1,3
20		📌	Write back end code	1 day	Tue 10/24/1	Tue 10/24/1	1,4,5,6,7,8,9,10,1
21		📌	Write front end code	1 day	Wed 10/25/1	Wed 10/25/1	1,4,5,6,7,8,9,10,1
22		📌	Create button in HTN	1 day	Wed 10/25/1	Wed 10/25/1	1,3,4,5,6,7,8,9,10
23		📌	Write HTML for passv	1 day	Thu 10/26/1	Thu 10/26/1	1,3,4,5,6,7,8,9,10
24		📌	Write back end code	1 day	Fri 10/27/17	Fri 10/27/17	1,3,4,5,6,7,8,9,10
25		📌	Write HTML which in	1 day	Fri 10/27/17	Fri 10/27/17	1,3,4,5,6,7,8,9,10
26		📌	Write HTML that indi	1 day	Fri 10/27/17	Fri 10/27/17	10,1,3,4,5,6,7,8,9
27		📌	Create button in HTN	2 days	Mon 10/30/1	Tue 10/31/1	1,3,4,5,6,7,8,9,10
28		📌	Write back end code	3 days	Tue 10/31/1	Thu 11/2/17	1,3,4,5,6,7,8,9,10
29		📌	Create HTML button	1 day	Fri 11/3/17	Fri 11/3/17	1,3
30		📌	Create HTML page fo	1 day	Fri 11/3/17	Fri 11/3/17	1,3,5
31		📌	Create announcemer	1 day	Mon 11/6/17	Mon 11/6/17	1,3,5
32		📌	Create announcemer	1 day	Mon 11/6/17	Mon 11/6/17	1,3,4,5,31
33		📌	Display X number of	1 day	Tue 11/7/17	Tue 11/7/17	1,3,4,5,31,32
34		📌	Create button in HTN	1 day	Mon 11/6/17	Mon 11/6/17	1,3,5

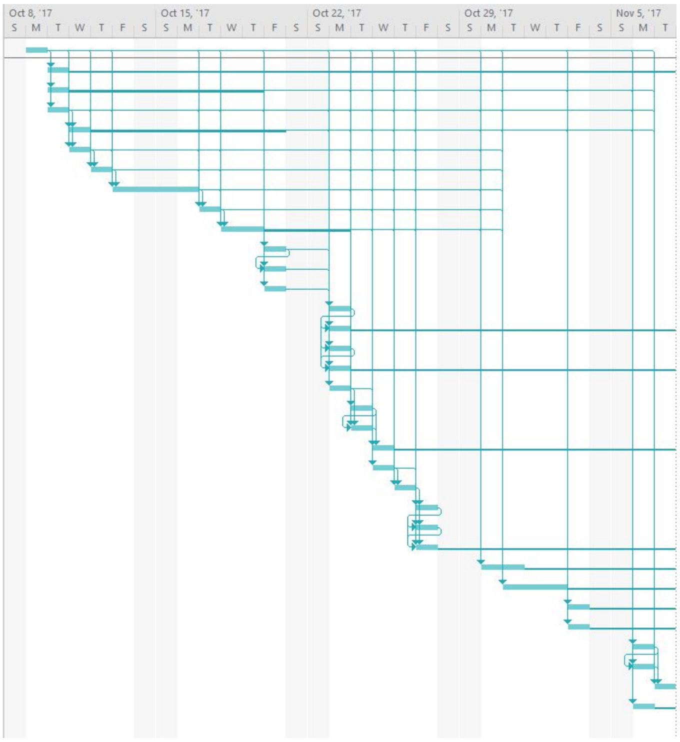
3.4.2 - Gantt Chart

Below is a visual representation of the tasks in the table above. The x-axis of the graph represents the timeline of the project, while the y-axis represents each task. Arrows between tasks represent dependencies between tasks.



3.5 Task Dependency Diagram

Below is a diagram showing each task and the relationships between them. Each rectangle represents a different task, while each arrow represents a dependency between two tasks. A task is dependant on another task if it cannot be completed without the other task being completed first.


Oct 8, '17
Oct 15, '17
Oct 22, '17
Oct 29, '17
Nov 5, '17
S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T

4. Appendix

4.1 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)

4.2 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	1.2, 1.4, 2.1, 2.2, 3.2, 3.4, 3.5, 4.1
Aaron Dyke	1.1, 2.3, 3.1, 4.1
Adam Hurnyak	1.1, 1.2, 2.1, 2.2, 3.2
Andrew Woodruff	1.3, 2.2, 2.3, 2.4, 3.3, 4.1

4.3 References

1. https://docs.google.com/document/d/1S-1STP_sCuxmb5ILrMOsNZD6cxxZBznr1JCnEUN3-Cw/edit