

Software Requirements Specification for Honor Program Web Application

Prepared by Kelly Morgan, Charbel Marche, Stephen Wilson, Tiffany Lower

January 29 2018

Table of Contents

1.1 Purpose	2
1.2 Scope.....	2
1.3 References.....	2
1.4 Intended Audience and Reading Suggestions	2
1.5 Overview of the Remainder of the Document	3
2.1 Product Perspective/System Overview	3
2.2 Client Characteristics	4
2.3 User Characteristics	4
2.4 Product Functions	4
Use Cases.....	5
Use Case 2.4.1	5
Use Case 2.4.2	5
Use Case 2.4.3	6
Use Case 2.4.4	6
Use Case 2.4.5	7
Use Case 2.4.6	7
Use Case 2.4.7	7
Use Case 2.4.8	8
2.5 Operating Environment	8
2.6 Design and Implementation Constraints	8
2.7 User Documentation	9
3 Functional Requirements and Non-Functional Requirements	9
3.1 Logging in and Account Management	9
3.2 Administrator Capabilities	9
3.3 Student Capabilities	10
3.4 Hosting the application	10
3.5 Safety Requirements	10
4 Non-Requirements	11
5 Assumptions	11
6 Appendices	11
6.1 Glossary Terms	11
6.2 Author Information	11
6.3 Additional Documents	13
Figure 1: Checksheet example	13
Figure 2: Database example	14
Figure 3: User Diagram	15

1 Introduction

1.1 Purpose

Honors Program Web Application version 1.0 is designed to deliver a functional web environment that allows UMW Honors Program students to view and keep track of their progress in the Honors Program. The application also allows Administrative access to upload a spreadsheet with data on students progress, adjust student information, and to make announcements that are visible to both students and admins.

1.2 Scope

The Honors Program Web Application benefits the student by allowing a functional environment to view cumulative progress in the UMW Honors Program. This, in turn, can be used in advisory meetings to create a goal to meet graduation and Honors Program requirements for the student. It can also be used by the student, allowing them to get updates on their own progress and stay on the correct track.

The Application will benefit administrators of the program by allowing the ability to upload a CSV (spreadsheet) file that contains all students and details on their progress. The application will automatically generate an account for newly added students and update the progress information of continuing students. Administrators will also be able to convey and share important information and announcements of events and offered courses.

1.3 References

Please see the additional documents section in the appendix for the layout of the CSV file and the layout for the current Honors Program checksheet.

1.4 Intended Audience and Reading Suggestions

This document is written for future developers, testers, users, and the customer. It provides a better understanding of what the Honors Web Application project is about and what the initial goals for this application are. We recommend reading and understanding the glossary terms before attempting to read the document.

1.5 Overview of the Remainder of the Document

In section two of this document, we will detail the project description and specifically touch on the overview of the system, the client characteristics, the user characteristics, and the product functions. Section three describes in more detail about the requirements of the web application. This includes the functional and non-functional requirements. Each requirement is labeled with their own unique number for identification with a short description of its function. Section 4 hosts non-requirements. Non-requirements section will explicitly list things that the project will not include. The non-requirements will be listed. In section 5, assumptions about the system are listed in a bulleted list. Section 6 named “Appendices” concludes the remainder of the documents. In section 6.1, there are glossary terms that defines technical terms. The authors are listed as well as their contributions in 6.2. Finally, paper documents are that were referenced in the production of this document are in 6.3.

2 Project Description

2.1 Product Perspective/System Overview

The Honors Program web application is a replacement for the current system in place within the program. Currently, student information may only be accessed by visiting the physical location of the Honors Program office on the Fredericksburg Campus of the University of Mary Washington. The web application must allow students to access their program check sheet from anywhere they may have internet access. It must allow students in the UMW Honors Program to view their progress for each semester and see what requirements they have satisfied. It also must allow students that are in the UMW Honors Program to view announcements that are posted by Honors Program administrators, allowing them to keep in touch. Administrators must be able to upload updated information for all students in the form of a CSV file. This must automatically create accounts for new students and update accounts for continuing students. Administrators also must be able to post announcements. All users must be able to change passwords and view announcements. Administrators must be able to access all student information by searching for them.

2.2 Client Characteristics

Our client is Dr. Anewalt, a computer science professor at the University of Mary Washington. She is requesting this system because currently, students are only able to see their progress in the Honors Program through progress reports sent physically by mail every semester. Upon creation of the Honors Program Web Application, students will be able to access reports on their progress at will. Administrators will be able to post announcements and upload a CSV file to update student information. This will allow for easier communication between the professors and staff that are in charge of the Honors Program and the students that are a part of it.

2.3 User Characteristics

The application will have Student and Administrator user classes. Both user classes must be included in the final product. Administrators and Students must be able to have a separate login. To differentiate between appropriate logins, the username will be used.

Administrative privileges will include uploading a large CSV file, viewing all students' progress, looking up individual students, and creating posts such as announcements and future course offerings.

Student privileges will include viewing posts by Administrators and either viewing or downloading their respective check sheet.

2.4 Product Functions

The web application must include a Student and an Administrator login. The Administrator must be able to upload and edit spreadsheets of student data, then generate a Comma Separated Value (CSV) file. The Administrator must also be able to make announcements or posts containing pertinent information.

The application must also be hosted through a free service, such as the University of Mary Washington Computer Science department server. Other options include UMW's Domain of One's Own, or Google's Firebase App.

Use Cases:

See Figure 3 in section 6.3 for User Diagram.

Use Case 2.4.1

Title: Logging in

Description: A user wants to log into their account

Main Flow:

1. User accesses web page
2. User types in login name
3. User types in password
4. User submits and logs in

Alternative Flow A:

3. User types in wrong login name
4. System prompts the username is wrong and to try again. Flow returns to standard step

Alternative Flow B:

4. User types in wrong password
5. System prompts the password is wrong and to try again. Flow returns to standard step 3.

Use Case 2.4.2

Title: Change password

Description: User wants to change password

Main Flow:

1. User selects that they want to change password
2. System asks for their current password
3. User enters current password
4. System verifies that password is correct
5. System asks for the new password (typed twice)
6. User enters new password twice and submits form
7. System makes sure that the passwords match
8. System updates password and redirects the user to the home page

Alternate Flow A:

4. System sees that password incorrect
5. System prompts user to try again, and the flow returns to 3

Alternate Flow B:

7. System sees that the passwords don't match
8. System prompts user to reenter passwords and flow returns to 6

Pre-Conditions:

User must be logged in.

Use Case 2.4.3

Title: Uploading CSV file

Description: Administrator wants to upload CSV file of student information

Main Flow:

1. Administrator logs in
2. Administrator prompts system they want to upload a new CSV
3. System prompts user to upload the file
4. User uploads file
5. System updates with new CSV file

Alternative Flow A:

5. User uploads wrong file format
6. System prompts user to upload correct format. Flow returns to standard step 5.

Use Case 2.4.4

Title: Student Views Progress

Description: Follows flow of student accessing view of spreadsheet/progress

Main Flow:

1. Student logs in
2. Student is redirected to main page
3. Student clicks link to see spreadsheet/progress

Alternate Flow:

2. Student prompted for download or browser view.
3. Student selects download link
4. Student receives file through download

Precondition:

1. Student must have login credentials
2. Student must be logged into website

Use Case 2.4.5

Title: Download CSV check sheet

Description: The student wants to download a CSV check sheet of their progress.

Main Flow:

1. The student logs in.
2. The student clicks on a check sheet link.
3. The check sheet is downloaded to the student's local machine.

Alternate Flow A:

3. The check sheet opens in a new window within the student's browser.
4. The student downloads the check sheet from the new window.

Use Case 2.4.6

Title: Post and edit an announcement

Description: An administrator wants to post and edit an announcement

Main Flow:

1. An administrator logs in.
2. The user indicates they want to post an announcement.
3. The system displays the proper interface to type announcement.
4. The system allows the user to enter necessary details.
5. The user tells the system the announcement is finished.
6. The system publishes the announcement.

Alternate Flow A:

7. The user sees an error in the announcement or the information has changed.
8. The user indicates to the system an edit is needed.
9. The system allows the user to edit the announcement.
10. The system publishes the changed announcement.

Use Case 2.4.7

Title: Student Views Announcement.

Description: Goes through the flow of a Student logging in to view announcements.

Main Flow:

1. Student Logs in
2. Student is redirected to main page
3. Student Visibly sees announcements feed on main page

Alternate Flow:

1. Student enters incorrect password
2. Student is prompted to see administrator

Precondition:

1. Student must be enrolled in program.
2. Student must be able to log in.

Use Case 2.4.8

Title: Look up student information

Description: The administrator wants to look up information on student progress.

Main Flow:

1. The administrator logs in.
2. The administrator navigates to a search page.
3. The administrator inputs student information such as first and last name or NetID.
4. The system retrieves the necessary student information.
5. The system displays the information within the application.

Alternate Flow A:

6. The system downloads the CSV file to the administrator's local machine.

2.5 Operating Environment

The web application will be accessible via any computer with the ability to connect to the internet. The web application will be hosted on a free online service.

2.6 Design and Implementation Constraints

The University of Mary Washington does not allow outside applications to be used with the Banner or Single Sign-on capabilities. The Honors Program web application will serve as an independent entity. Updates to student check sheets will be provided by a large, single CSV file. The primary language will be in English. The preferred hosting should be on Domain of One's Own, UMW Blogs or a similar free service. Administrators know basic HTML.

2.7 User Documentation

The web application will follow the basic steps of logging onto UMW accounts, so Students and Administrators will be familiar with the process. A user manual will be provided to explain the tasks and features available through the application.

3 Functional Requirements and Non-Functional Requirements

3.1 Logging in and Account Management

This section contains all the different actions that users (as specified in the specific requirements) are able to carry out with regards to logging in or managing their account.

1. Any student that is in the Honors Program and thus a part of the uploaded CSV file, must be able to log in using their email address as a username and their Honors Program ID as their password.
2. All admins must be able to also log in, using their emails and a created password.
3. All users must be able to change their password.
4. Accounts for new students in the Honors Program must be automatically created when encountered in the uploaded CSV file.

3.2 Administrator Capabilities

This section contains all the different actions that the administrator can carry out with the Honors web application.

1. Administrators must be able to post announcements that both other administrators and students can see on their homepage.
2. Administrators must be able to upload a CSV file containing all the Honors student information for the Honors students and that must update all existing accounts with the new information and create new accounts as they are encountered in the file.
3. Administrators must be able to view the records and progress of every student by searching for the student's name.

3.3 Student Capabilities

This section contains all the different actions that the student can carry out with the Honors web application.

1. Students must be able to view announcements.
2. Students must be able to view their progress for the current semester within the Honors Program, and this should be presented on the website.
3. Students may be able to download a PDF of their progress for the semester.
4. Students must be able to browse Honors courses on the Honors Web Application.

3.4 Hosting the application

This section contains all the requirements with regards to hosting the web application.

1. The Honors Web Application must be handled through a free service.
2. The Honors Web Application may be hosted on the UMW cs server or on domain of one's own.

3.5 Safety Requirements

This section contains suggested safety guidelines covering issues of file corruption and credential protection.

1. It is expected that an administrator would have a local copy of the CSV file.
2. It is recommended that the uploading administrator shares/backs up the CSV file with other administrators should the live copy of the file become corrupted or lost.
3. Passwords must be encrypted via hash.

3.6 User Interface Requirements

This section contains all the requirements in regards to the user interface.

1. The user interface should be similar in design to the University of Mary Washington site.
2. The user interface must be functional.
3. The user interface should be aesthetically pleasing.

4 Non-Requirements

This section contains the items that the application will specifically not contain.

1. The user interface is not required to be pretty.
2. Being able to view previous semesters individually will be not required.
3. Users will not be able to change their username.
4. Hosting will not be a paid service.
5. Students may not create new users.
6. Students may not post announcements.

5 Assumptions

This section contains assumptions that have been made about the product.

1. It is assumed that Administrators will have basic HTML skills.
2. It is assumed that as the feel of the application will be similar to the UMW website, navigation should be simple.
3. All Honors Program students have an Honors Program ID.
4. All Honors Program students and Administrators have a UMW email.
5. All users have access to a computer with an internet browser.

6 Appendices

6.1 Glossary Terms

Document Conventions

Throughout the document, “must” and “need” signify a requirement that is necessary, while “should” and “may” signifies a requirement or feature that is not necessary. When the text refers to “the project” or “the application” it is referring to the Honors Web Application.

6.2 Author Information

Charbel Marche wrote sections 1.4, 1.5, 2.1, 2.2, 2.4.2, 3.1, 3.2, 3.3, 3.4, 3.6.

Tiffany Lower wrote sections 1.3, 1.6, 2.5, 2.5 and contributed Use Cases 2.4.1 and 2.4.3.

Stephen Wilson wrote sections 1.1, 1.2, 3.5 and contributed Use Cases 2.4.4 and 2.4.7

Kelly Morgan wrote sections 2.3, 2.4, 2.7, 4, and 5, and contributed Use Cases 2.4.5 and 2.4.6

6.3 Additional Documents

Honors Program Checksheet					
Last Name	First Name	ID	email		
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>		
Status	Comments	Term	Admitted	Major	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
FSEM HN	<input type="text"/>	Co-curricular 1	<input type="text"/>	Initial Essay	<input type="text"/>
FSEM HN Date	<input type="text"/>	Date 1	<input type="text"/>	Mentoring 001	<input type="text"/>
HN Course 1	<input type="text"/>	Co-curricular 2	<input type="text"/>	Leadership 002	<input type="text"/>
HN 1 Date	<input type="text"/>	Date 2	<input type="text"/>		
HN Course 2	<input type="text"/>	Co-curricular 3	<input type="text"/>	HONR Portfolio 1	<input type="text"/>
HN 2 Date	<input type="text"/>	Date 3	<input type="text"/>	HONR Portfolio 2	<input type="text"/>
HN Course 3	<input type="text"/>	Co-curricular 4	<input type="text"/>	HONR Portfolio 3	<input type="text"/>
HN 3 Date	<input type="text"/>	Date 4	<input type="text"/>	HONR Portfolio 4	<input type="text"/>
		Co-curricular 5	<input type="text"/>	Exit Interview	<input type="text"/>
HONR 201	<input type="text"/>	Date 5	<input type="text"/>	GPA Fall	<input type="text"/>
HONR 201 Date	<input type="text"/>	Co-curricular 6	<input type="text"/>	GPA Spring	<input type="text"/>
		Date 6	<input type="text"/>	GPA Fall	<input type="text"/>
Research Course	<input type="text"/>	Co-curricular 7	<input type="text"/>	GPA Spring	<input type="text"/>
Research Date	<input type="text"/>	Date 7	<input type="text"/>	GPA Fall	<input type="text"/>
Capstone Course	<input type="text"/>	Co-curricular 8	<input type="text"/>	GPA Spring	<input type="text"/>
Capstone Date	<input type="text"/>	Date 8	<input type="text"/>		

*Indicates course/requirement in progress

3/3/2016

Figure 1: Checksheet example

B.2. CSV Template

The table below encapsulates column and order of the current import CSV file maintained by the client's Access database.

Column Letter	Column Title	Data Type	Track in System?
A	Last Name	String	Yes
B	First Name	String	Yes
C	ID	Number	No
D	email	String (unique)	Yes
E	Admitted	Term	Yes
F	duPont Code	Number	No
G	Status	Status	Yes
H	Term	Term	Yes
I	Co Cur 1	String	Yes
J	Date 1	Date	Yes
K	FSEM HN	String	Yes
L	FSEM Date	Term	Yes
M	HN course 1	String	Yes
N	HN 1 Date	Term	Yes
O	HN course 2	String	Yes
P	HN 2 Date	Term	Yes
Q	HN course 3	String	Yes
R	HN 3 Date	Term	No
S	HN course 4	String	No
T	HN 4 Date	Term	No

Figure 2: Database example

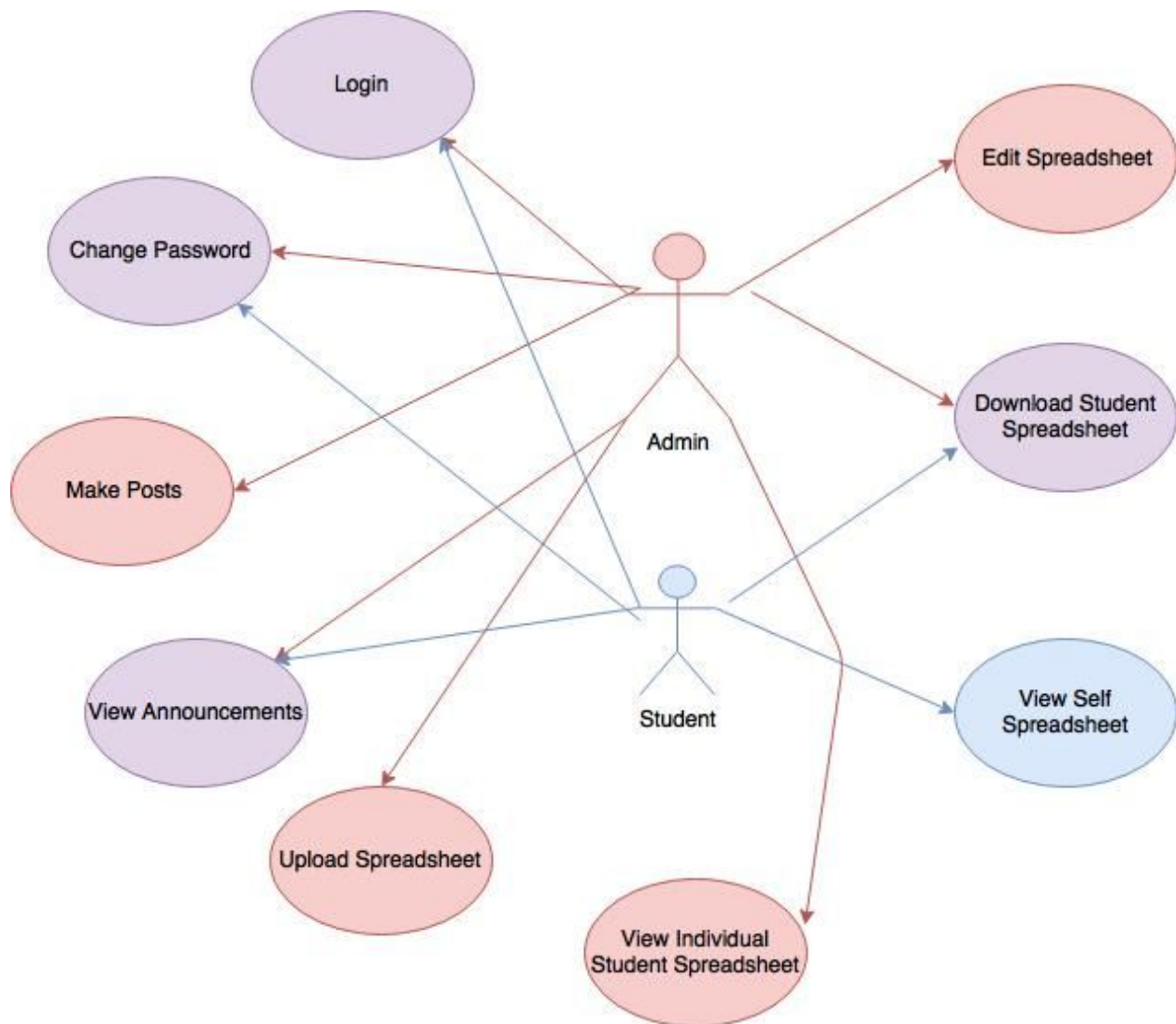


Figure 3: User Diagram

Blue nodes refer to Student-only use cases.

Red nodes refer to Administrator-only use cases.

Purple nodes refer to both Student and Administrator use cases.