Project Plan Document

for

The College of Education Application Station

Version 1.0

Prepared by Molly Burns, Alec Carlyle, Mary Clark, and Hannah Zontine

UMW Computer Science Department

March 1, 2017

Table of Contents

1. Introduction

1.1. Purpose

The goal of this document is to explain and outline our project plan for the College of Education Application Station web application. This document will discuss the system itself, the application interface, any constraints, and user characteristics. This document serves two additional purposes: it will show the client our plan for implementation of the CoE Application Station, as well as being a reference for the group to accomplish as many functional requirements as possible. The intended audience for this document includes our client, Dr. Venitta McCall, our Software Engineering professor, Dr. Karen Anewalt, and our peers.

1.2. Scope

This project is an application designed to allow for a streamlined student-teaching program registration experience to both the students in the program and the faculty that oversees the process. The largest benefit of this project will be the switch from physical applications to digital, as the website shall be hosted on a server provided by the University. Given that a student has a umw account, they shall be able to register and fill out any applications for the CoE program that they need to.

1.3. References

This document is derived from both sections' requirements documents as well as further collaboration with the client, Dr. McCall. The screenshots of these references are found in the reference section of the Appendix.

- REF 1: Initial Licensure Internship Application
  (1 page)

- REF 2: CoE Application for Admission/Continuation Into the Graduate Program
  (1 page)

- REF 3: Application for Undergraduate Admission to the 5-year Pathway Initial Licensure Programs
  (4 pages)

- REF 4: A screenshot of the P.O. Box, a web application developed by a prior Software Engineering team that the design team shall draw inspiration from.

1.4. Overview of the remainder of the document

This document is organized into sections, which are listed on the Table of Contents (located on page 2) for reference. Each section is signified by bold text and a section number throughout the document. Additionally, each section and subsection contains specific information pertaining to the section heading.

2. Project Description

2.1. Project Scope

The College of Education Application Station is a website designed to digitize the student-teaching application process at UMW. Upon initially loading the webpage, a user is able to either login to their CoEAS account or register for a new one. The application will support two classes of users: student users and admins. Both types of users have a distinct dashboard upon which they may interact with the system. Any UMW-enrolled student shall be able to register for a student user account in order to fill out 3 distinct forms. These forms include:

- Initial Licensure Internship Application
- Undergraduate 5-Year Program Application
- CoE Continuing into the Graduate Program Application

Once submitted, the above forms shall be stored inside of the database along with all of the accounts in the system. Users shall also be able to edit their submitted forms to make corrections, given that the form is still available to edit and/or submit. The admin of the system is able to set/extend the deadline for application submissions,

Faculty of the education department shall interact with the system as an admin. They shall be able to view application submissions and filter them by user name, type of application, and email address. They'll be able to sort a list of users in the database and delete inactive users. When the end of the semester comes around, the admin shall be able to archive the data in the database so that they may store it and then clear the database. They also shall be able to export application submissions into an excel-ready .csv file format.

2.2. User Characteristics

2.2.1. Admin/Client

The client will be an admin of the system. She is responsible for reviewing student applications, checking for errors or inconsistencies, and contacting students about these issues, all of which is typically done in excel or printed in a large paper spreadsheet. In the system, she will have full control over these tasks in the form of viewing/archiving submitted applications and student information, controlling submission openings and deadlines, and editing data if necessary. Admins will also be responsible for cleaning the database of both users and applications, as this will not be done automatically.

2.2.2. Student Users

Only students will use the application as basic users, and each will be required to created their own account to create/draft, submit, view, and edit applications. These are the only functions they will need to perform. All users will either be entering the CoE 5-year program as undergraduates or graduate (Masters) students, and thus need to complete one or more of the 3 applications. According to the client, users frequently want to submit more information than is asked (e.g. “Yes, but…” instead of a yes or no response) so users will be limited in what data they can submit in order to simplify admin tasks.

2.3. Functional Requirements

Functional requirements are the necessary functions of the CoE web application. These requirements highlight how the system shall behave when a user interacts with it and shall be explained using user stories.

2.3.1. Admin

A. As an admin, I want to be able to set and/or extend the deadline for application submissions so that students may be able to either turn in or edit their applications at a date that I see appropriate.

B. As an admin, I want to sort submitted submissions by email, name, or type of form so that I may find the application that I need.

C. As an admin, I want to delete a user account or application submissions so that the database does not become too cluttered.

D. As an admin, I want to archive the database at the end of the semester to store the data before I clear the database.

E. As an admin, I want to export a series of submissions into a .csv file format so that I may view and manipulate those forms in excel.

F. As an admin, I want to be able to login to my account so that I may perform actions unique to my account privileges.

G. As an admin, I want to view any submitted form's contents on the website so that I may see if as student is good to go on their submissions.

H. As an admin, I want to reset my password so that I can login to the website in case I forget it.

I. As an admin, I want to create additional admin accounts so that I am not the only administrator in the system.

J. As an admin, I want to clear the database at the end of a semester so that I have a fresh database to work with for upcoming students.

2.3.2 Student User

A. As a student, I want to be able to fill out up to 3 distinct applications so that my position as a student teacher at UMW may be accepted or maintained.

B. As a student, I want to be able to register for a CoE application account so that I may be able to recover any progress on either submitted or unsubmitted forms during the entirety of my participation in the student teaching program.

C. As a student, I want to verify my account so that I can confirm my identity using my email address.

D. As a student, I want to reset my password so that I can login to the website in case I forget it.

E. As a student I want to be able to login so that I can perform actions unique to my account.

2.4. Constraints

This system is limited by many different types of constraints. Constraints are restrictions on the degree of freedom that the developers have when implementing the system.

2.4.1. Performance Constraints

This software must be accessible to at least 10 simultaneous users. During a semester, it must be able to receive an estimated twenty to forty form submissions. In addition, the system must be able to handle multiple administrators accessing its data at the same time.

2.4.2. Security Constraints

All personal student information must be encrypted in the database. Only administrator accounts will have access to a student's private information, such as GPA, test results, criminal history, and drug-use history.

2.4.3. Software Constraints

This system must not be hosted on the UMW network. The client prefers not to pay hosting fees. Together, we decided that the solution is to host it on Domain of One's Own. DoOO is a project at UMW that allows students and staff to register a domain name and associate it with a hosted web space for free. The domain must be hosted under the account of Dr. Venitta McCall.

2.4.4. Time Constraints

This project's final deadline is May 1st, 2017. At least two weeks before the deadline, the system must be ready to be tested by the project's testing team. By the deadline, it must be online and free of bugs.

2.4.5. University Constraints

This system is under the jurisdiction of the University of Mary Washington. The software can not use the University's Banner system for aid in authenticating faculty administrators.

3. Project Schedule

3.1. Approach

Our approach to this project encompasses three values the client would like to see in her software:

1. Functionality over appearance.

2. User limitation and categorization over free input.
3. Admin control over automation.

First, the client's current application for student practicums, P. O. Box, was "very sleek", but encountered application-breaking errors after deployment that prevented her from using the application in her department, resulting in a financial loss for the school. Because of this, we want to ensure that our current application prioritizes functionality, simplicity, and thorough testing over creating the ideal user interface.

Second and third, the client wants to be in full control of both the application's data and what data the users submit. Due to the variability in responses that often occurs when students are asked to submit paper forms, category dropdown is preferred over free user input in any case where it is possible. Furthermore, nothing in the application should be automated and must be confirmed by the admin before the action is completed (ex. the database should not archive or clear data automatically).

These standards for scheduling and building our progress should ensure that the client receives a product she and her department can use and navigate confidently.

3.2. Milestones and Deliverables

Below is a table that represents each milestone we must complete and a deliverable we can present to our client. Each milestone has a list of tasks and subtasks that must be completed in order to implement the deliverable.

Milestone	**Deliverable**	**Tasks (and subtasks)**
Complete initial interface	GUI interface on DoOO	- Set up DoOO - Design interface - Select BootStrap template - Implement BootStrap template
Complete database	Database with 5 tables	- Create database - Create user table - Create admin table - Create necessary table(s) to contain input information from the Initial Licensure Internship Application form - Create necessary table(s) to contain input information from

		the Admission/Continuation Into the Grad Program form - Create necessary table(s) to contain input information from the 5-Year Program Admissions form
Complete login	Login interface	- Design registration page - Design login page - Implement registration page - Email verification - Add users to users table - Add admins to admins table - Implement login page - Verify username and password - Determine whether user or admin and display appropriate dashboard
Complete user interface	User Dashboard	- Design layout of user dashboard - Implement user interface - Select a form - Save a draft of a form - Submit a form - View submitted or saved forms from the current semester - Change password
Complete admin interface	Admin Dashboard	- Make an admin account for the client - View submitted applications from the current semester - Filter submissions by form type - View an individual student’s application and private information - Remove users from the user table who have been inactive - Copy data into an Excel file - Clear data from all form tables in the database

		- Change password - Add new admin accounts

3.3. Work Breakdown Structure

The project schedule consists of 5 incremental sprints. Each will last roughly 2 weeks, depending on client meeting schedule, and have its own milestone and deliverable at the sprint deadline. Project milestones and deliverables are joint components of each sprint: the former will serve as our development team's goal, while the latter will be presented and reviewed in a meeting with the client prior to beginning the next sprint. Each sprint will also be accompanied by respective unit, integration, or usability testing. Under this breakdown, the project is expected to take 7-8 weeks to complete. Extra padding of 2 hours has been added to each time estimate.

3.3.1. Sprint 1: Interface on DoOO

- Time: 23 hours

During this sprint, the team will set up the production server and create an example interface to display to the client on that server. Doing this first will give the team and stakeholders confidence that Domain of One's Own is a dependable solution.

We estimate that setting up DoOO for our purposes will take 2 hours because, although some group members are familiar with the application, we expect to encounter difficulties. Since we are all creating and experimenting with interfaces individually, we expect that merging our designs and creating a simple layout with Bootstrap will take roughly 3 hours. The main template that will be created is the user dashboard, which all other templates will be integrated with. We estimate that merging the designs for this template specifically will take 2 hours, and implementing it will take 6. A sum of 2 hours will be added to subsequent sprints to account for linking new templates to the user template (thus 8 hours is the estimate given in the Gantt chart).

During this time, we also plan to setup our database securely, create tables, and seed the database. Database design and creation is estimated at 8 hours due to the complexity of the forms.

3.3.2. Sprint 2: Registration and Login

- Time: 16 hours

During this sprint, the team will create the functionality for user registration and login. We estimate that creating the interfaces for these two pages will take 2 hours, since we will have already completed most of the template. Creating the functions for user login and registration should take 4 hours, since this is a pretty basic part of web applications and something we're familiar with, but still a thorough process that requires testing.

Finally, the most difficult part of this sprint will be adding email confirmations. Researching this should take one hour, and since this process is unfamiliar to all members of our group but still has few functional steps involved, we estimate that implementing this should take 5 hours. An extra 2 hours has been added to this estimate for testing.

3.3.3. Sprint 3: Form Submission

- Time: 15 hours

During this sprint, the team will focus on creating, storing, and viewing the College of Education forms. Each form may share similar inputs, but will each be unique and require their own tables (created in sprint 1). For this sprint, we will focus on creating just one form for the client to review before we make others.

We estimated here that each form's functionality could take up to 9 hours. This includes the time needed to create functions to store the data in the database (5 hours), and the time needed to create a more "interactive" form that will hide unused elements on complicated forms (4 hours). These will also require thorough unit tests, since the previous group's failure to test "NA" inputs broke the application, so an extra 3 hours is added to account for our first unit and usability testing on the forms.

Lastly, another hour is added to integrate this form into the application and link it in the user dashboard.

3.3.4. Sprint 4: Admin Initial Dashboard

- Time: 34 hours

During this sprint, the team will continue creating the functionality to accept and process forms, completing any that were not completed in the previous sprint, and develop the administrator dashboard that the client will use.

The remaining forms are expected to take 11 hours each for the same reasons given in the previous sprint (but with 1 hour shaved off for code reusability). Someone in the group will also create the front-end of the admin dashboard, which will will present to the client for review. This is estimated to take 3 hours to design and, while the full implementation of this dashboard will take 20 hours. Only about half of this time will be completed during this sprint. Those 10 hours will be spent creating and testing admin displays of submitted forms, which users will not need to see.

3.3.5. Sprint 5: Admin Functional Dashboard

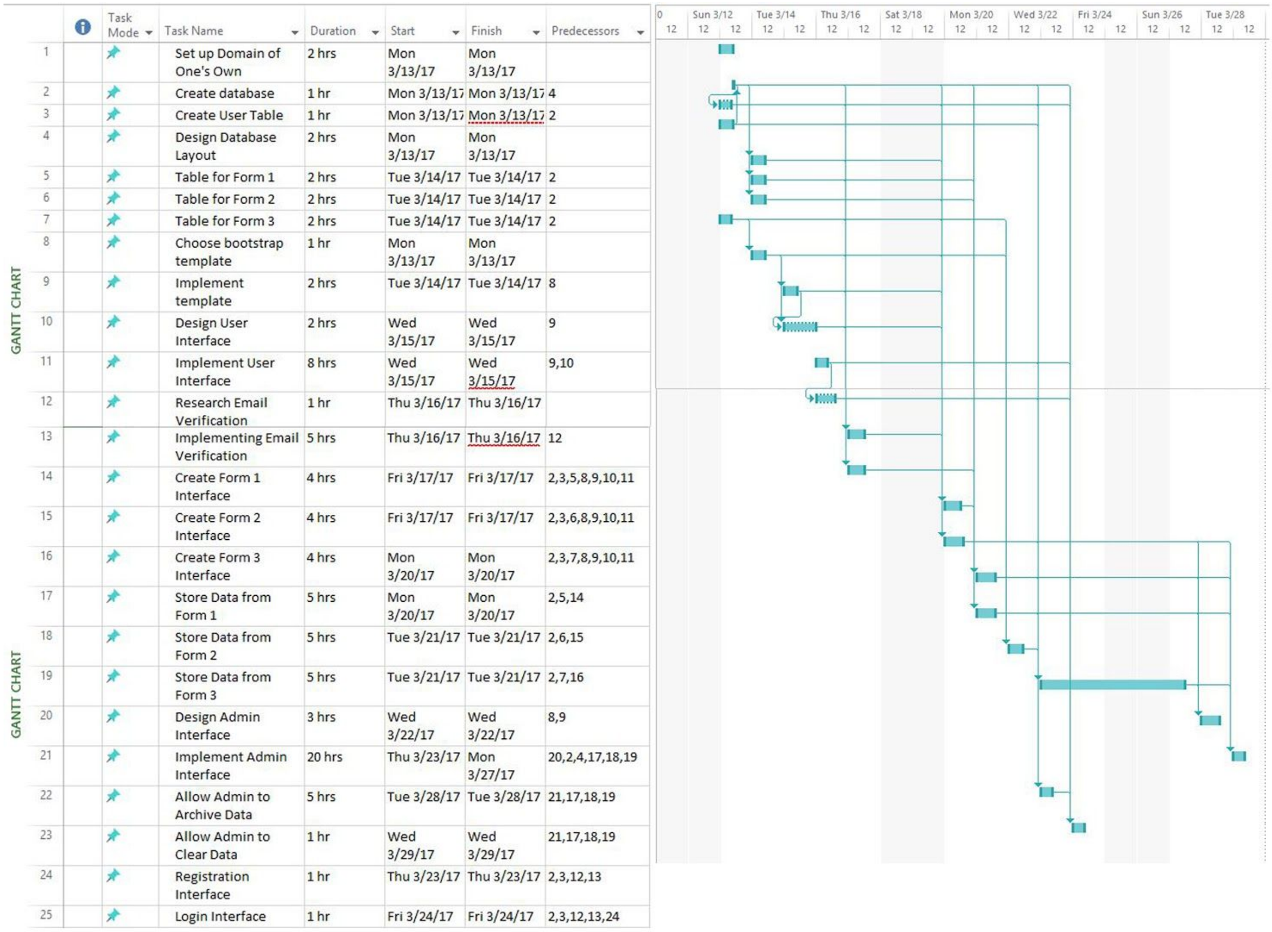
- Time: 18 hours

During this sprint, the team will add the remaining functionality to the administrator dashboard, including archiving and clearing the database. Archiving data in an excel spreadsheet should require 5 hours, and creating the functionality to clear the database should take just 1 hour, as this is simple to implement and test.

Lastly, the remainder of the admin dashboard will be created and the admin functions created in the first paragraph will be added. This brings the total estimate to 16 hours before padding.

3.4. Gantt Chart

The Gantt Chart is a project management tool that is created from a Task Duration and Dependencies Table, which is shown on the left in the image below. The Gantt Chart emphasizes tasks that can be completed in parallel and is shown on the right in the image below. This table and chart provide a schedule which we can use to ensure that we are working at maximum efficiency.

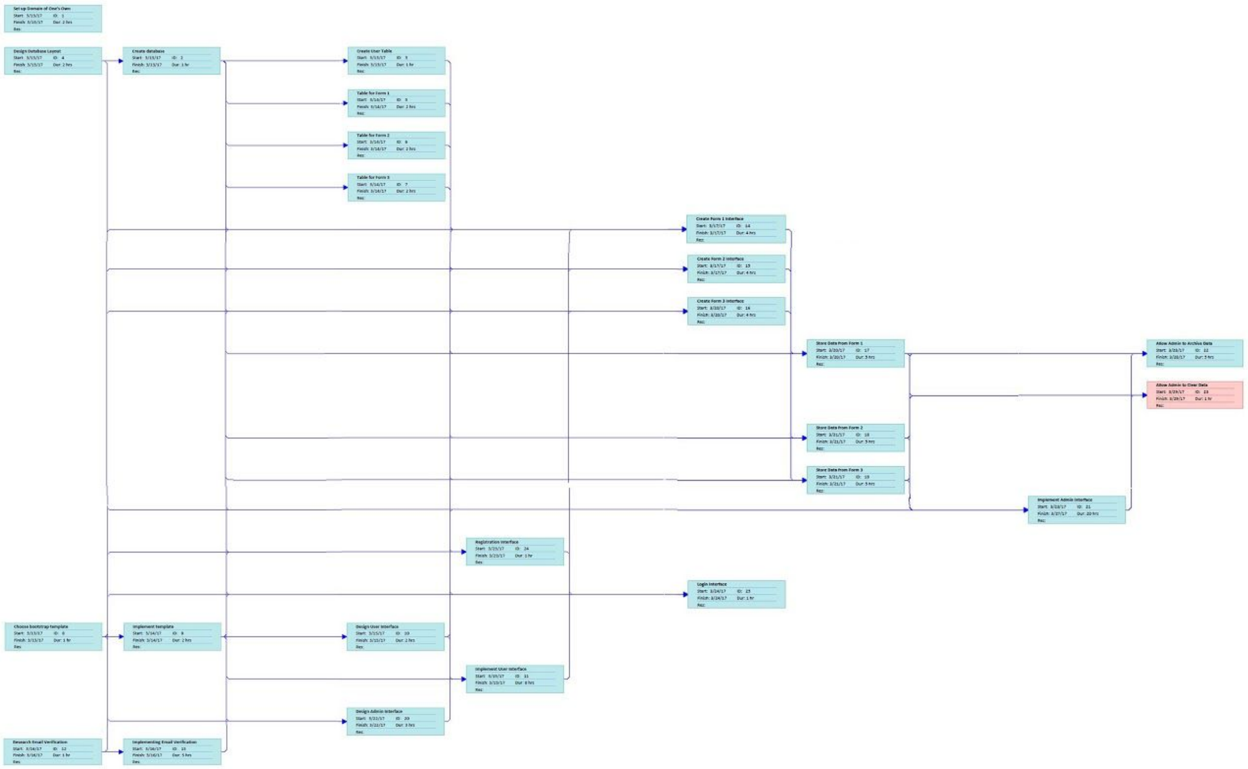


	ℹ	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		📌	Set up Domain of One's Own	2 hrs	Mon 3/13/17	Mon 3/13/17	
2		📌	Create database	1 hr	Mon 3/13/17	Mon 3/13/17	4
3		📌	Create User Table	1 hr	Mon 3/13/17	Mon 3/13/17	2
4		📌	Design Database Layout	2 hrs	Mon 3/13/17	Mon 3/13/17	
5		📌	Table for Form 1	2 hrs	Tue 3/14/17	Tue 3/14/17	2
6		📌	Table for Form 2	2 hrs	Tue 3/14/17	Tue 3/14/17	2
7		📌	Table for Form 3	2 hrs	Tue 3/14/17	Tue 3/14/17	2
8		📌	Choose bootstrap template	1 hr	Mon 3/13/17	Mon 3/13/17	
9		📌	Implement template	2 hrs	Tue 3/14/17	Tue 3/14/17	8
10		📌	Design User Interface	2 hrs	Wed 3/15/17	Wed 3/15/17	9
11		📌	Implement User Interface	8 hrs	Wed 3/15/17	Wed 3/15/17	9,10
12		📌	Research Email Verification	1 hr	Thu 3/16/17	Thu 3/16/17	
13		📌	Implementing Email Verification	5 hrs	Thu 3/16/17	Thu 3/16/17	12
14		📌	Create Form 1 Interface	4 hrs	Fri 3/17/17	Fri 3/17/17	2,3,5,8,9,10,11
15		📌	Create Form 2 Interface	4 hrs	Fri 3/17/17	Fri 3/17/17	2,3,6,8,9,10,11
16		📌	Create Form 3 Interface	4 hrs	Mon 3/20/17	Mon 3/20/17	2,3,7,8,9,10,11
17		📌	Store Data from Form 1	5 hrs	Mon 3/20/17	Mon 3/20/17	2,5,14
18		📌	Store Data from Form 2	5 hrs	Tue 3/21/17	Tue 3/21/17	2,6,15
19		📌	Store Data from Form 3	5 hrs	Tue 3/21/17	Tue 3/21/17	2,7,16
20		📌	Design Admin Interface	3 hrs	Wed 3/22/17	Wed 3/22/17	8,9
21		📌	Implement Admin Interface	20 hrs	Thu 3/23/17	Mon 3/27/17	20,2,4,17,18,19
22		📌	Allow Admin to Archive Data	5 hrs	Tue 3/28/17	Tue 3/28/17	21,17,18,19
23		📌	Allow Admin to Clear Data	1 hr	Wed 3/29/17	Wed 3/29/17	21,17,18,19
24		📌	Registration Interface	1 hr	Thu 3/23/17	Thu 3/23/17	2,3,12,13
25		📌	Login Interface	1 hr	Fri 3/24/17	Fri 3/24/17	2,3,12,13,24

For a more reader-friendly view of this table and chart, please visit: https://goo.gl/57CkU3
to view the image file itself which allows you to zoom in on any portion of the image.

3.5. Task Dependency Diagram

A task dependency diagram illustrates which tasks are dependent upon another, or rather which tasks must be completed before moving on to a subsequent task. Tasks that are not dependent on another task are shown at the left side of the diagram. Any tasks that are dependent upon these tasks are to the right of this column, with the dependencies between tasks shown by lines with arrows pointing to the dependent task. The task dependency diagram for the CoE Application Station is shown below.



For a more reader-friendly version of this diagram, please visit https://goo.gl/nV9dDN
to view the image file which allows you to zoom in on any portion of the image.

The following bullet points describe each of the dependencies and and why the tasks are dependent upon one another.

- The first dependency is between Design Database (Task 4), and Create Database (Task 2). The reasoning behind this dependency is that the Database itself must first be designed or laid out before being created.
- Subsequently, creating tables for the Users (Task 3), Form 1 (Task 5), Form 2 (Task 6), and Form 3 (Task 7) are all dependent upon Task 2: Create Database. The database needs to have been created in order to create tables for storing data within the database.
- Implementing the Bootstrap Template (Task 9) is dependent upon Choosing a Bootstrap Template (Task 8). This is because the group must agree upon what particular Bootstrap template will be used when implementing the CoEAS.
- Designing the User Interface (Task 10) is dependent upon choosing and implementing the Bootstrap template (Tasks 8 & 9). The user interface could not be designed without already having a template in place.
- Implementing the User Interface (Task 11) is directly dependent on designing the user interface (Task 9) and designing the user interface (Task 10). The User Interface must be designed before being implemented.

- Implementing Email Verification (Task 13) is dependent upon Researching Email Verification (Task 12). Since our team is not familiar with email verification systems, we need to research how email verification systems work before implementing them in the CoEAS.
- Creating the Form 1 Interface (Task 14) is dependent on multiple tasks: creating the database (Task 2), creating the user table (Task 3), creating the table for Form 1 (Task 5), choosing a Bootstrap template (Task 8), implementation of the Bootstrap template (Task 9), designing the user interface (Task 10), and implementing the user interface (Task 11). Creating the interface for Form 1 wouldn't be possible or make any logical sense if it was to be completed before all of the tasks it is dependent on.
- Creating the Form 2 Interface (Task 15) is dependent upon many tasks: creating the database (Task 2), creating the user table (Task 3), creating the table for Form 2 (Task 6), choosing a Bootstrap template (Task 8), implementation of the Bootstrap template (Task 9), designing the user interface (Task 10), and implementing the user interface (Task 11). Creating the interface for Form 2 would not be possible without the completion of the tasks it is dependent upon.
- Creating the Form 3 Interface (Task 16) is dependent on multiple tasks: creating the database (Task 2), creating the user table (Task 3), creating the table for Form 3 (Task 7), choosing a Bootstrap template (Task 8), implementation of the Bootstrap template (Task 9), designing the user interface (Task 10), and implementing the user interface (Task 11). Creating the interface for Form 3 would not be possible without the completion of the tasks it is dependent on.
- Storing the Data for Form 1 (Task 17) is dependent upon a few tasks: the creation of the database (Task 2), creating the table for Form 1 (Task 5), and creating the Form 1 interface (Task 14). In order to store the data for Form 1, the database and table that the data will be stored in must be created first, as well as the interface in which the data will be entered.
- Storing the Data for Form 2 (Task 18) is dependent on a few tasks: the creation of the database (Task 2), creating the table for Form 2 (Task 5), and creating the Form 2 interface (Task 15). In order to store the data for Form 2, the database and table that the data will be stored in must be created first, in addition to the interface in which the data will be entered.
- Storing the Data for Form 3 (Task 19) is dependent upon a few tasks: the creation of the database (Task 2), creating the table for Form 3 (Task 7), and creating the Form 3 interface (Task 16). In order to store the data for Form 3, the database and table that the data will be stored in must be created first, as well as the interface in which the data will be entered.
- Designing the Admin Interface (Task 20) is dependent on choosing a Bootstrap template (Task 8) and implementing the Bootstrap template (Task 9). The Admin Interface cannot

be implemented without first having chosen and implemented the Bootstrap template which is the foundation of the CoEAS web application.

- Implementing the Admin Interface (Task 21) is dependent upon multiple tasks: designing the database (Task 4), creating the database (Task 2), storing the data for Form 1 (Task 17), storing the data for Form 2 (Task 18), storing the data for Form 3 (Task 19), and designing the admin interface (Task 20). The Admin Interface itself allows the admins to perform tasks such as viewing applications, archiving and clearing data, all of which are dependent on having the back-end of the web application established.
- Allowing Admins to Archive Data (Task 22) is dependent on a few tasks: storing the data for Form 1 (Task 17), storing the data for Form 2 (Task 18), storing the data for Form 3 (Task 19), and implementing the admin interface (Task 21). The ability to export the data stored within the database would not be possible without first having implemented the data storage process and admin interface itself.
- Allowing Admins to Clear Data (Task 23) is dependent upon storing the data for Form 1 (Task 17), storing the data for Form 2 (Task 18), storing the data for Form 3 (Task 19), and implementing the admin interface (Task 21). The ability for admins to clear the data from the database would not be possible without first having implemented the data storage process and admin interface itself.
- The Registration Interface is dependent on creating the database (Task 2), creating the User table (Task 3), researching email verification (Task 12), and implementing email verification (Task 13). Creating and implementing the Registration Interface would not be possible without creating the database and table for storing user data, as well as researching and implementing the email verification system to ensure that only UMW students and CoE faculty members will be able to register as users of the CoEAS.
- The Login Interface is dependent upon creating the database (Task 2), creating the User table (Table 3), researching email verification (Task 12), implementing email verification (Task 13), and the creation of the registration interface (Task 24). Creating and implementing the Login Interface would not be possible without creating the database and table for storing user data, in addition to the research and implementation of an email verification system, therefore only allowing registered users to log in to the CoEAS.

4. Appendix

4.1. Glossary

- .csv file: simple file format to store tabular data using a comma (,) as a delimiter
- Archive data: move data that is no longer needed/used to a separate means of storage.

- BootStrap: An open-source framework designed for web development. Allows for mobile compatability by automatically resizing the browser to scale with the dimensions of the screen of the user.
- CoE: College of Education.
- CoEAS: College of Education Application Station.
- Dashboard: The visual set of tools available either to the student user or the admin. Each type of user has a dashboard unique to their usages of the system.
- Database: An ordered, stored arrangement of data inside of the web application. Holds onto user accounts and submitted form information.
- DoOO: Domain of One's Own. A UMW-based website host that allows for faculty or enrolled students to host a webpage in their own name.
- Functional Requirements: The functions that a system shall perform.
- Gantt Chart: Horizontally-oriented chart that displays the planned schedule of a project.
- GUI: Graphical User Interface.
- Interface: A display highlighting how the user and the system will interact.
- P.O. Box: A web application designed for submission of practicum forms.
- Practicum: A course for teachers that involves them gaining real-life experiences as educators.
- Sprint: A two week period of work that will result in a deliverable to show to the client.
- Task Dependency Diagram: Displays all of the tasks of the project and which tasks require other tasks for completion.
- UMW: University of Mary Washington.

4.2. Contributions

This section details the individual contributions that each group member made to this Project Plan Document.

Introduction

Purpose - Molly
Scope - Alec
References - Alec
Overview of the remainder of the document - Molly

Project Description

Project Scope - Alec
User Characteristics - Mary
Functional Requirements - Alec
Constraints - Hannah

Project Schedule

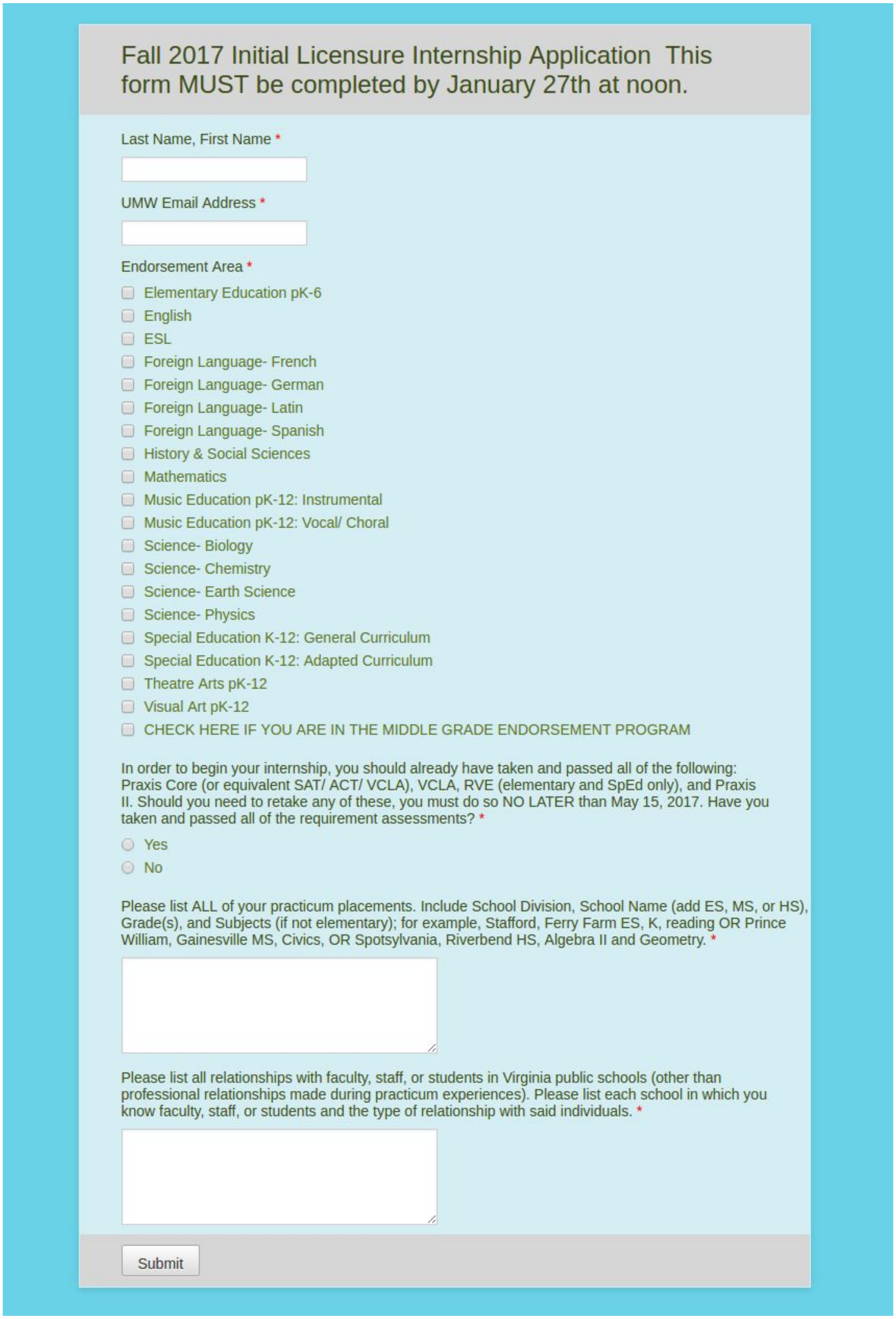
Approach - Mary
Milestones and Deliverables - Hannah
Work Breakdown Structure - Mary
Gantt Chart - Together/Molly
Task Dependency Diagram - Molly

Appendix

Glossary - Alec
Contributions - Mary
References - Alec

4.3. References

REF 1, a screenshot of the Initial Licensure Application:



Fall 2017 Initial Licensure Internship Application This form MUST be completed by January 27th at noon.

Last Name, First Name *

UMW Email Address *

Endorsement Area *

- □ Elementary Education pK-6
- □ English
- □ ESL
- □ Foreign Language- French
- □ Foreign Language- German
- □ Foreign Language- Latin
- □ Foreign Language- Spanish
- □ History & Social Sciences
- □ Mathematics
- □ Music Education pK-12: Instrumental
- □ Music Education pK-12: Vocal/ Choral
- □ Science- Biology
- □ Science- Chemistry
- □ Science- Earth Science
- □ Science- Physics
- □ Special Education K-12: General Curriculum
- □ Special Education K-12: Adapted Curriculum
- □ Theatre Arts pK-12
- □ Visual Art pK-12
- □ CHECK HERE IF YOU ARE IN THE MIDDLE GRADE ENDORSEMENT PROGRAM

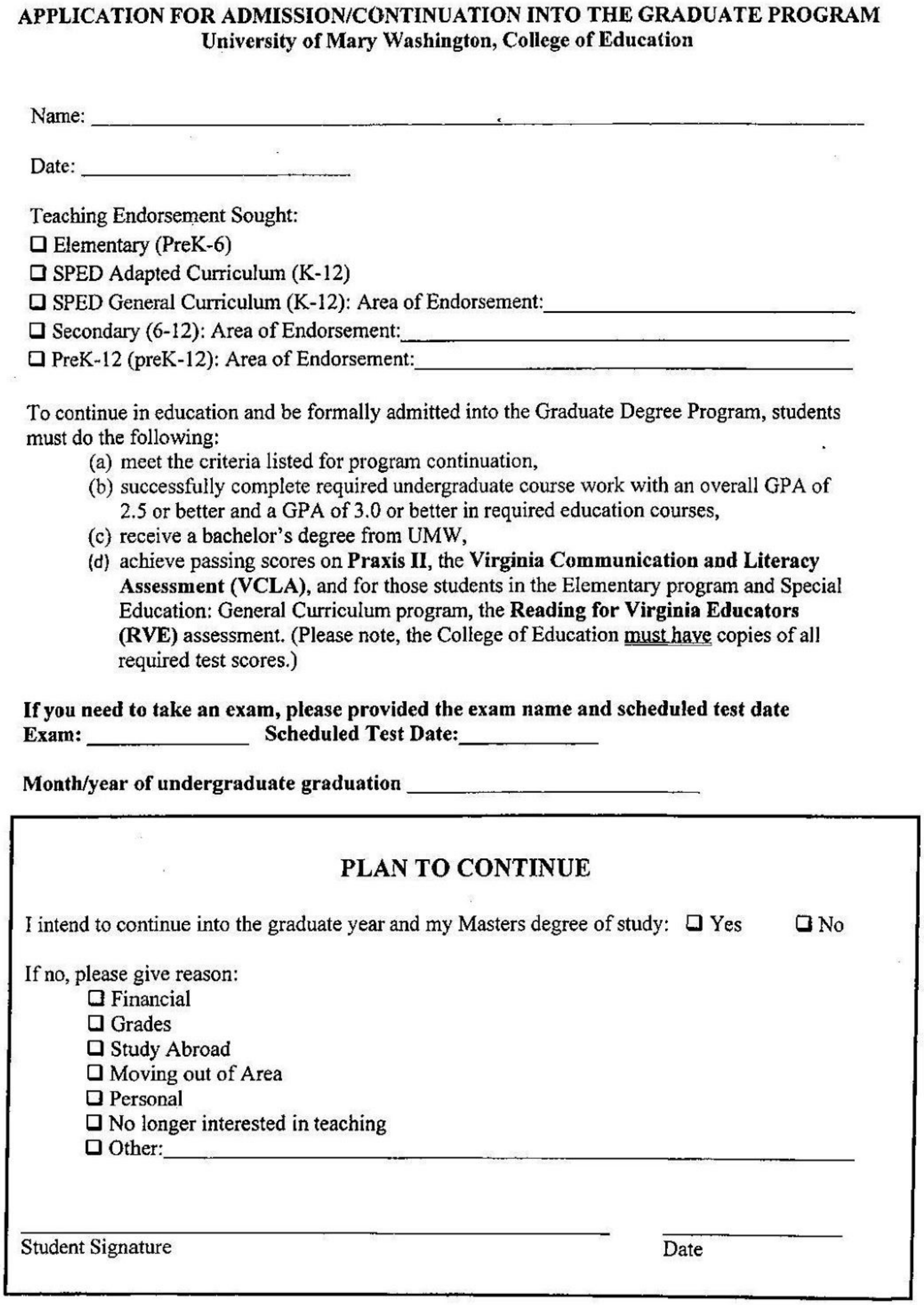
In order to begin your internship, you should already have taken and passed all of the following: Praxis Core (or equivalent SAT/ ACT/ VCLA), VCLA, RVE (elementary and SpEd only), and Praxis II. Should you need to retake any of these, you must do so NO LATER than May 15, 2017. Have you taken and passed all of the requirement assessments? *

- ○ Yes
- ○ No

Please list ALL of your practicum placements. Include School Division, School Name (add ES, MS, or HS), Grade(s), and Subjects (if not elementary); for example, Stafford, Ferry Farm ES, K, reading OR Prince William, Gainesville MS, Civics, OR Spotsylvania, Riverbend HS, Algebra II and Geometry. *

Please list all relationships with faculty, staff, or students in Virginia public schools (other than professional relationships made during practicum experiences). Please list each school in which you know faculty, staff, or students and the type of relationship with said individuals. *

REF 2, screenshots of the application for graduation continuation admission:



APPLICATION FOR ADMISSION/CONTINUATION INTO THE GRADUATE PROGRAM
University of Mary Washington, College of Education

Name: ______________________________

Date: ______________

Teaching Endorsement Sought:

□ Elementary (PreK-6)

□ SPED Adapted Curriculum (K-12)

□ SPED General Curriculum (K-12): Area of Endorsement: ______________

□ Secondary (6-12): Area of Endorsement: ______________

□ PreK-12 (preK-12): Area of Endorsement: ______________

To continue in education and be formally admitted into the Graduate Degree Program, students must do the following:

(a) meet the criteria listed for program continuation,

(b) successfully complete required undergraduate course work with an overall GPA of 2.5 or better and a GPA of 3.0 or better in required education courses,

(c) receive a bachelor's degree from UMW,

(d) achieve passing scores on **Praxis II**, the **Virginia Communication and Literacy Assessment (VCLA)**, and for those students in the Elementary program and Special Education: General Curriculum program, the **Reading for Virginia Educators (RVE)** assessment. (Please note, the College of Education must have copies of all required test scores.)

If you need to take an exam, please provided the exam name and scheduled test date
Exam: __________ **Scheduled Test Date:** __________

Month/year of undergraduate graduation ______________

PLAN TO CONTINUE

I intend to continue into the graduate year and my Masters degree of study: □ Yes □ No

If no, please give reason:

□ Financial

□ Grades

□ Study Abroad

□ Moving out of Area

□ Personal

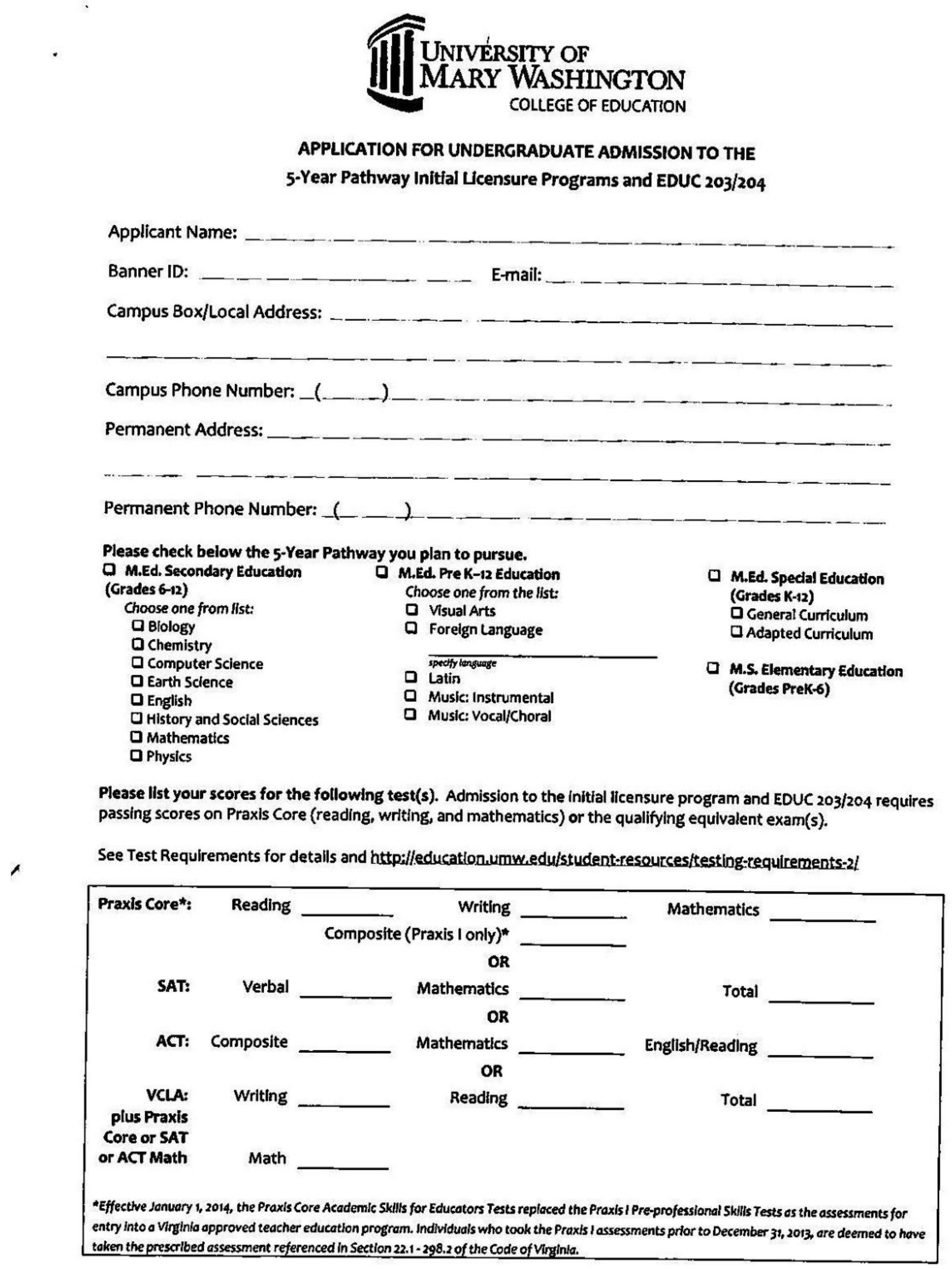
□ No longer interested in teaching

□ Other: ______________________________

______________________________ Student Signature

______________ Date

REF 3, four screenshots of the application for undergraduate admission:



UNIVERSITY OF
MARY WASHINGTON
COLLEGE OF EDUCATION

APPLICATION FOR UNDERGRADUATE ADMISSION TO THE
5-Year Pathway Initial Licensure Programs and EDUC 203/204

Applicant Name: ______________________________

Banner ID: ______________________ E-mail: ______________________

Campus Box/Local Address: ______________________________

____

Campus Phone Number: _(_____)______________________

Permanent Address: ______________________________

____

Permanent Phone Number: _(_____)______________________

Please check below the 5-Year Pathway you plan to pursue.

□ **M.Ed. Secondary Education (Grades 6-12)**
Choose one from list:
- □ Biology
- □ Chemistry
- □ Computer Science
- □ Earth Science
- □ English
- □ History and Social Sciences
- □ Mathematics
- □ Physics

□ **M.Ed. Pre K–12 Education**
Choose one from the list:
- □ Visual Arts
- □ Foreign Language ______________ *specify language*
- □ Latin
- □ Music: Instrumental
- □ Music: Vocal/Choral

□ **M.Ed. Special Education (Grades K-12)**
- □ General Curriculum
- □ Adapted Curriculum

□ **M.S. Elementary Education (Grades PreK-6)**

Please list your scores for the following test(s). Admission to the initial licensure program and EDUC 203/204 requires passing scores on Praxis Core (reading, writing, and mathematics) or the qualifying equivalent exam(s).

See Test Requirements for details and http://education.umw.edu/student-resources/testing-requirements-2/

Praxis Core*:	Reading	________	Writing	________	Mathematics	________
			Composite (Praxis I only)*	________		
			OR			
SAT:	Verbal	________	Mathematics	________	Total	________
			OR			
ACT:	Composite	________	Mathematics	________	English/Reading	________
			OR			
VCLA:	Writing	________	Reading	________	Total	________
plus Praxis Core or SAT or ACT Math	Math	________				

**Effective January 1, 2014, the Praxis Core Academic Skills for Educators Tests replaced the Praxis I Pre-professional Skills Tests as the assessments for entry into a Virginia approved teacher education program. Individuals who took the Praxis I assessments prior to December 31, 2013, are deemed to have taken the prescribed assessment referenced in Section 22.1 - 298.2 of the Code of Virginia.*

Current UMW Status – *please check the corresponding box*

□	1st Semester Freshman	□	1st Semester Sophomore	□	1st Semester Junior
□	2nd Semester Freshman	□	2nd Semester Sophomore	□	2nd Semester Junior

TYPE OF STUDENT: □ BA □ BS □ BLS □ Undeclared

Undergraduate Major

Name of Academic Major Program: ______________________________ Declared? □ Yes □ No

Name of Major Advisor: ______________________________

Month/year of undergraduate graduation: ______________________________

GPA: For admission and permission to take the first education course, EDUC 203/204, current UMW students must have attained a cumulative 2.5 GPA or better in at least 12 UMW credits. (Important – If you have not completed 12 credits at UMW, leave section blank. We will confirm your grades and GPA at the end of the semester).

Current UMW GPA ________________ based on ____________ credit hours

Group Information Session (Optional):
Students interested in seeking credentials to teach should contact the College of Education as early in their college career as possible and *are encouraged* to attend a group information session about the teacher education program. Group information sessions are offered throughout the academic year.

Have you attended a Group Information Session? □ Yes □ No

If Yes, please list the month/year attended ______________________________

TRANSFER STUDENTS
Transfer students who have a 3.25 GPA, have submitted passing scores on Praxis Core or qualifying equivalent assessments, and attended a Group Information Session may apply to take EDUC 203/204. in their first semester at UMW.

Official Transcripts from all schools listed are required for licensure and will need to be obtained once admitted into the 5-Year Pathway Program..

College or University	Location-City/State	Dates of Attendance	Degree Earned / GPA	Name when enrolled
College or University	Location-City/State	Dates of Attendance	Degree Earned / GPA	Name when enrolled
College or University	Location-City/State	Dates of Attendance	Degree Earned / GPA	Name when enrolled

Have you ever applied to the College of Education? □ Yes □ No

If Yes, when? ______________________________

Have you ever been convicted of a crime other than a traffic violation? □ Yes □ No
(If yes, attach a letter of explanation and a copy of the court proceedings)

Have you ever been convicted of a felony? □ Yes □ No
(If yes, attach a letter of explanation and a copy of the court proceedings)

Have you ever been convicted of a misdemeanor involving children or drugs? □ Yes □ No
(If yes, attach a letter of explanation and a copy of the court proceedings)

Have you ever been found guilty of a criminal offense in another country? □ Yes □ No
(If yes, attach a letter of explanation and a copy of the court proceedings)

OPTIONAL - Demographic Information

Gender: ____________ **Birth Date:** ______________________________

If you wish to be identified with a particular race or ethnic group, please select the one that best describes you:

□ African American, Black
□ Asian or Pacific Islander
□ Hispanic or Latino
□ Native American, Alaska Native
□ White, Non-Hispanic
□ Multi-racial

LEADERSHIP AND INTERPERSONAL SKILLS

List offices held and experiences in extra-curricular and related experiences (work, social, high school, college, etc.)

PREVIOUS EXPERIENCES WITH YOUTH

List and describe experiences in which you have been involved in supervising, sponsoring, or teaching school-age children:

ESSAY

In essay form, please describe your motivational, academic, and personal qualifications that have influenced your decision to become a teacher. (This essay must be typed and will be reviewed for content, style, and effective expression.)

I followed the *UMW Honor Code* when creating this application.

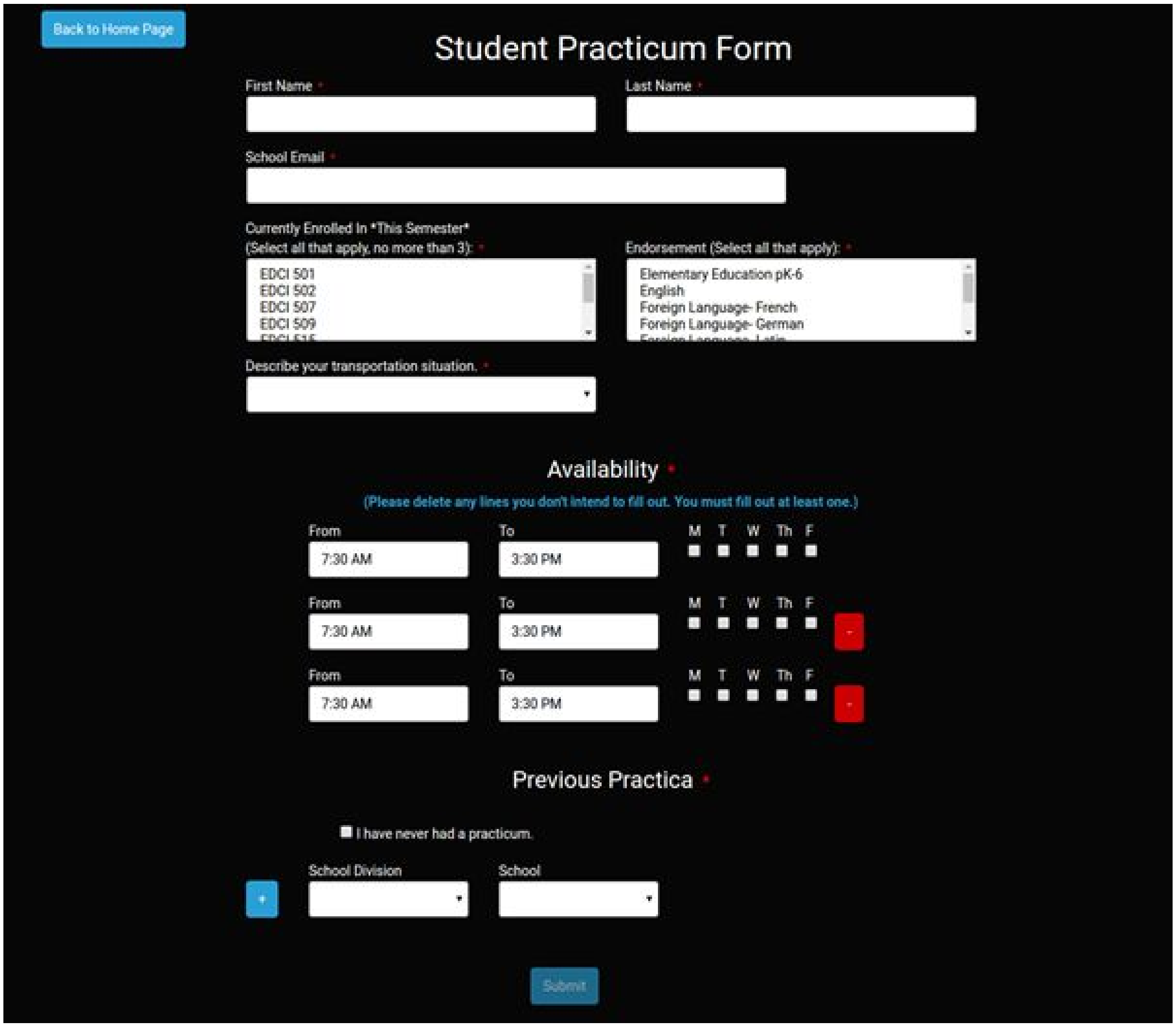
Applicant's Initials _____

I certify that the information given in this application is correct to the best of my knowledge and belief.

Applicant's Signature ________________________________ *Date* ________________

IMPORTANT: *Please re-read this application and make sure that all blanks have been filled in. Incomplete forms will be returned and the processing of your application may be delayed.*

REF 4, a screenshot of the P.O. Box:



Student Practicum Form

First Name *

Last Name *

School Email *

Currently Enrolled In *This Semester*
(Select all that apply, no more than 3): *

EDCI 501
EDCI 502
EDCI 507
EDCI 509
EDCI 515

Endorsement (Select all that apply): *

Elementary Education pK-6
English
Foreign Language- French
Foreign Language- German
Foreign Language- Latin

Describe your transportation situation. *

Availability *

(Please delete any lines you don't intend to fill out. You must fill out at least one.)

From	To	M	T	W	Th	F	
7:30 AM	3:30 PM	□	□	□	□	□	
7:30 AM	3:30 PM	□	□	□	□	□	-
7:30 AM	3:30 PM	□	□	□	□	□	-

Previous Practica *

□ I have never had a practicum.

School Division

School