1. Introduction

1.1 Purpose of this Document

The purpose of the Software Requirements Specification is to outline the requirements for The American Academy's Education With an Elastic Workforce (EW2) product . EW2 will be built on Apache, PHP and MySQL using the Drupal content management system. It will be operating system independent and accessible with any standard compliant browser.

1.2 Scope of the Development Project

Education With an Elastic Workforce (EW2) will be a web platform for managing tasks and employees for The American Academy.

1.3 Definitions, Acronyms, and Abbreviations

TAA - The American Academy

EW2 - Education With an Elastic Workforce

Drupal - Content Management System

PHP - A server scripting language used by Drupal

MySQL - The database that will be used for this project

Apache - A web service

Task - Unit of work for which TAA offers payment

1.4 References

www.drupal.org

www.php.net

www.sun.com

www.apache.org

1.5 Overview of Document

This document contains all of the software requirement specifics. It contains a general description of the types of users who will be using our application, how it is going to work, and what technologies we are using to make it work. We will also outline and describe specific components of the project.

2. General Description

2.1 User Personas and Characteristics

Teacher/Proctor - The teacher is a TAA contracted employee who not only teaches a class but also wants to do additional work for TAA. This person can log into the website and view any tasks available.

TAA/Admin - This is an employee of TAA. This person can create tasks, view current proctors/teachers, approve work and assign pay for work done.

2.2 Product Perspective

This product is designed to run an any system that is capable of running Drupal. This covers a wide variety of machines, including operating systems Mac OS X, Windows XP and Vista and Linux. The sole requirement for the user is a web browser (Safari, Firefox or Internet Explorer) with an active internet connection.

2.3 Overview of Functional Requirements

- Must be capable of storing tasks that can be done by several people at the same time.
- Must have a system for tracking payment.
- Must contain an easy interface with little to no learning curve.

2.4 Overview of Data Requirements

All the data will be stored and retrieved within MySQL. We will not implement stored procedures. Drupal will be handling a majority of the database calls through the MyPHPAdmin API.

2.5 General Constraints, Assumptions, Dependencies, Guidelines

This program will work with virtually any system that can connect to the internet and browse web pages.

2.6 User View of Product Use

The user comes to the website:

Registration:

- After entering in a user name, email address and password, the user can register with our site.

Main Menu:

- The user can view all tasks in the middle of the screen.
- The user can accept and reject tasks within this menu.
- The user/admin can modify personal information here as well.
- The admin can create new tasks

Objectives:

- The purpose of this website will be to allow Proctors to do tasks for money and Admins able to create tasks for the Proctors to receive money.

3. Specific Requirements

3.1 External Interface Requirements

- operator/user interface characteristics from the human factors point of view
  - This will interface with most browsers, especially Internet Explorer, Firefox and Safari. As long as the user has knowledge to operate a web browser, they will be able to operate our website efficiently.
- characteristics required of the interface between the software product and each of the hardware components
  - The hardware will be accessed indirectly, both from the client side (via a web browser) and server side via the various program APIs (MySQL, Apache, PHP)
- interfaces with other software components or products, including other systems, utility software, databases, and operating systems
  - We will be accessing Drupal via the creation of our own modules.
  - The software will be using Drupal to communicate with MySQL via MyPHPAdmin, Apache and PHP.
  - MySQL, PHP, and Apache will all be accessing the hardware through their own APIs.

3.2 Detailed Description of Functional Requirements

3.2.1 Template for describing functional requirements

This is a typical template you can use to describe each of the functional components that were identified in Section 2.3. These sections should be at least the following:

purpose	a description of the functional requirement and its reason(s)
inputs	which inputs; in what form/format will inputs arrive; from what sources input will be derived, legal domains of each input element
processing	describes the *outcome* rather than the *implementation*; include any validity checks on the data, exact timing of each operation (if needed), how to handle unexpected or abnormal situations
outputs	the form, shape, destination, and volume of the output; output timing; range of parameters in the output; unit measure of the output; process by which the output is stored or destroyed; process for handling error messages produced as output

3.2.2 Registration/Login/Logout UI

purpose	This will explain how a user can register with the site
inputs	The user will be expected to input their user name (must be unique), password and a valid email address.
processing	This will be done via Drupal, with Drupal storing the username, password and email address in the MySQL database with encryption.
outputs	The output will be a success and log the user in the website.

3.2.3 Admin UI

purpose	This will explain how TAA employees can manage, assign, create, and approve tasks.
inputs	The user will be expected to input their user name (must be unique) and password to log in.
processing	This will be done via Drupal, with Drupal storing the username and password in the MySQL

	database with encryption. All other information about specific users, tasks, and money will also be stored in the MySQL database.
outputs	The admin will be able to successfully see when they add tasks, approve work done etc.

3.2.4 Teacher/Proctor UI

purpose	This will explain what the user can do once the successfully login to the site.
inputs	Mouse clicks and keyboard web browser hot keys will allow the user to navigate the tasks, accept them, do them, receive payment, cancel tasks, assign tasks, setup new tasks and many more to be defined.
processing	This will be done with Drupal/PHP as well, processing any clicks etcetera. All information is stored via Drupal's MyPHPadmin API in the MySQL database. Any modules needed will access this database as well.
outputs	The output will be a sequentially correct webpage determined by the given task.

3.3 Performance Requirements

Since the project involves using a content management system as requested by TAA, some decisions on performance are highly correlated to Drupal. Issues such as the number of connections to the website are based on TAA's current settings for their website and will be modifiable in their PHP settings file. Other decisions like these are also decided by TAA after our project is done and delivered to their company.

However, one part we can define is the efficiency of our own Drupal modules. We will try to utilize what is standard and modular about Drupal, with an emphasis on what works correctly first, followed by an emphasis on code clarity, and then on speed. No page on the website should take more than a two seconds to load completely, and file uploads will limited to a reasonable size of 15 MB, although it should be easily configurable for TAA's purposes.

Searches will create tables with a configurable number of Tasks included, much like Google allows a varying number of results. The range will span from as few as 1 to up to 100.

Code will be clear and well commented, preferably following the guidlines established by the Drupal community for easy integration should any module be submitted to the official project. This is less important as the modules will likely deal to specific TAA needs.

The number of different simultaneous users and connections per second should only be limited by PHP's configuration, specified by TAA and irrelevant to our project.

3.4 Quality Attributes

The website should be as hacker proof as possible, protecting from injection attacks, robots, and other threats. This includes protecting user-enterable forms from clients. Otherwise, this involves PHP settings that will be at TAA's descrection.

The site will only be available to TAA employees and affiliates who are Teachers, Proctors, or otherwise affiliated with education. The code will be clear and well commented to allow modification if necessary. All commonly changed settings should provide an administrator interface to apply the changes. The code should be modular in nature and incremental so that changes and tweaks are easy.

The webpages we work on should provide good error output on forms the clients will complete. Warning messages should appear when something is incomplete, and Error messages will appear in the unlikely event that something doesn't work as intended, as well as information on how to resolve the issue relevant to the user.

3.5 Other requirements

None at this time.