1. Introduction

1.1 Purpose of this document

This is the Software Design Specification for Timeless Solutions EW^2 solution. This document will outline the software design and specification of our workflow task management system in addition to system architecture, system components, and software requirements as agreed upon by the customer and the project team.

1.2 Scope of the development project

This will be called EW^2, a website that will manage tasks determined by TAA and completed by independent proctors for a set amount of money. This website will handle the tasks and the users.

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.apache.org

www.drupal.org

www.php.net

www.taa.com

1.5 Overview of document

This document will contain specifics about the design of our software. It is a high level explanation of how the project will function, and which components will be developed to help it do so. This document will also serve to explain the technologies used to develop the website, as well as the user's interaction with various features.
2. System architecture description

2.1 Overview of modules / components

Drupal-

Users Modules

- Allows a user to log into a website with secure credentials
- Determines their credentials which will allow for a smarter search heuristic for the projects displayed on the screen
- Keeps track of user web activity on the site

Search Modules

- Uses an AJAX component (web service) that will handle dynamic searching of workflow modules to be displayed to the user
- Links the user and the workflow together

Workflow Modules

- Handles all tasks and projects
- Highly customizable
- States can be Available, In Progress, In Review and Finished

Project Modules

- A unit of work that TAA needs to have completed
2.2 Structure and relationship

When a user logs in to the system, he is presented with a list of available projects. He will be able to use the search function to sort projects by several categories. He will be able to see a complete list of available projects, or he can narrow the list according to subject. An administrator will be able to create projects and assign them to a particular workflow, as well as accept bids and approve completed work.

A user will have a configurable profile with settings such as name, address, telephone number, email address, and educational credentials. An administrator will have the ability to verify the credentials and qualify the user for particular types of work. For instance, after verifying that the user has a college degree, the administrator will be able to add the verification to the user profile. If the degree is in math education, for example, the administrator will be able to add a qualification to the user profile for the user to work on math-related projects.

An administrator will have the ability to create custom workflows. For instance, in one type of workflow, TAA may accept bids from users for a particular project, and in another workflow TAA may set a particular payment amount. Also, they may want to have multiple people involved in the review process. For instance, they may want to have a math teacher a math curriculum and then have it reviewed by the principal before accepting the work.

2.3 User interface issues

Workflow Modules
* Handles all tasks and projects
* Highly customizable
* States can be Available, In Progress, In Review and Finished

Search Modules
* Uses AJAX components
* Links the Users, Workflow and Projects together visually

Users Modules
* Allow user to log into site
* Determines credentials
* User Activity

Project Modules
* A unit of work that TAA needs to have completed

Drupal
3. Detailed description of components

This will be completed in SDSv2.

4.0 Reuse and relationships to other products

- Role in Product Design
  - Existing Drupal Modules are to be used whenever possible
  - Modules that are close to what we need will be "tweaked" using module "hooks." This is Drupal's modular ability to affect different modules through method overloading.

- Product Implementation and Reasons for Changes
  - As we discover better modules, our project's goals may change to accommodate. Different modules will require different enhancements.
  - If we discover there is no suitable module for a particular need, we will need to implement one from scratch ourselves.

- Thrown Out Reusable Information
  - Modules which are too difficult to use for our tasks will be discarded.

5.0 Design decisions and tradeoffs

The requirement of using Drupal to design the website made many of the design decisions for us. Leveraging the strengths of Drupal require the use of preexisting modules instead of creating custom modules.

Mechanical Turk was considered as a method of tracking tasks any paying employees, however Mechanical Turk failed to offer user API which was needed for the teachers to accept the tasks. Because of this Mechanical Turk was abandoned and Drupal will be used instead to manage tasks.

6.0 Pseudocode for components

The following is an example of some code used for writing "hooks" in drupal. This is how we can tweak a module's functionality:

```php
/**
 * php Drupal code example
 * This demonstrates how to remove unwanted content created
 * by another module.
 */
function moduleName_link_alter(&$links, $node)
{
    if ($links->type == "contentType")
    {
        $links->linkType = array(); // empties the links.
    }
}
```
It is still fairly early in the development process to give a large amount of pseudocode for components and modules that may have Drupal alternatives already.

The following are ideas for some custom modules if needed:

**Workflow(class)**

This class will control the Project class for access with different settings and customizability, such as specific pay procedures.

**Project(class)**

This will be a very low level implementation of our Project/Task idea. Basically a project can be a task, but a task is the lowest level. This will be controlled by Workflow.

**Search(class)**

This will be the glue that keeps the User and Workflow communicating. There will be AJAX implementations to allow for fast searching while also presenting the data in a pleasing way to the user.

**7.0 Appendices (if any)**