Introduction

Motivation

The Science and Technology Entry Program (STEP), a program under the New York State Education Department (NYSED), published a set of priorities that it wanted to see solutions proposed for when determining which institutions it would award funding to. Paraphrased, these priorities are for institutions to provide programs and services to improve:

1. The recruitment and retention of male participants;

2. The recruitment and retention of Hispanic/Latino and American Indian participants;

3. Eighth grade students' NYS Math and Science Assessment examination scores [**?**].

Additionally, STEP created a set of service and data requirements that funding recipients must provide. The underlying themes of the requirements are that there should be increased collaboration between the awarded institution, the local educators, students, and parents as well as providing students with assistance in gaining skills needed to succeed in learning and pursuing STEM (Science, Technology, Engineering, and Mathematics) fields [**?**].

Data published in numerous studies [**?**, **?**] show that American 8th grade students perform consistently below those in many other countries around the world. School Districts in Northern New York can substantiate the STEP priorities and requirements. NYS Mathematics test data from 2010 revealed that 56% of the school districts in St. Lawrence County had below average proficiency levels for 8th grade [**?**]. At the Salmon River School District, where there is a majority American Indian demographic, 54% of students scored below acceptable proficiency levels on the 8th Mathematics assessment [**?**]. It is clear that STEP identified these troubling facts and is now pushing institutes of higher education to try to solve these issues.

IMPETUS (Integrated Mathematics and Physics for Entry To Undergraduate STEM), for Career Success is the result of collaboration between St.Lawrence-Lewis County BOCES,

the STEM Partnership, and Clarkson University. Its primary goal is to improve and increase opportunities for underrepresented minorities and students from economically disadvantaged rural areas to realize their potential for college entry as STEM majors and for eventual career success in technically oriented professions.

Traditionally, this collaboration has been accomplished via in-person meetings between various combinations of the relevant parties. St. Lawrence County is the largest county by area in not only Northern New York, but in the whole state, yet has a population density of only $41/\text{mi.}^2$, whereas the state average is $355/\text{mi.}^2$. Combined, these circumstances do not create an environment suited towards an efficient and effective exchange of information [**?**].

IMPETUS has proposed the creation of a Web-based Collaborative Environment that will satisfy the STEP criteria by combining different technologies into a unified solution while increasing the convenience and accessibility of collaboration. Students who are in need of assistance, when meeting in-person is not possible, will be able to engage in web-based cooperative learning with Clarkson University students. Educators will be able to survey and receive feedback from targets regardless of their location and parents will have access to their children's data.

In 2008, the National Science Foundation (NSF) created a Cyberlearning Task Force that was charged with, amongst other objectives, determining what the key research topics were surrounding cyberlearning [**?**]. In a subsequent NSF publication, it was determined that one of these topics should be collecting, analyzing, and managing data about how individuals use a given cyberlearning system [**?**].

Due to STEP's data logging requirement and the fact that the NSF has deemed cyberlearning data collection to be an area of research interest, the IMPETUS Web-based Collaborative Environment (IWCE) will collect an extensive amount of data about not only the individuals using it, but also how they are using it.

Research Objectives

Methodology

Creating the IMPETUS Web-based Collaborative Environment will be done following a modified waterfall model of software engineering: Requirements, Design, Implementation, and Verification; the modification being that feedback will be readily incorporated into the phases allowing progress to go "up" the waterfall. Being that there is only one developer on the the project, it is the most basic model to follow.

As modules becomes ready for Verification, usability tests will be done to get feedback on their usefulness and design. This will require the creation of surveys and the user studies gauge how effective the interface is at allowing a user to accomplish goals.

Background

Symfony

The Collaborative Environment is built using Symfony Standard Edition, an open-source, object-oriented PHP framework designed around the Model-View-Controller (MVC) software architecture. Using Symfony to create the Collaborative Environment encourages the use of design patterns that are well understood and allows for more of the development time to focus on application features rather than reimplementing standard components of web applications.

The MVC architecture separates an application's business logic, that is, all of the algorithms which process the exchange of data between an interface and a database, from its user interfaces. The *model* consists of object representations of application data (Entities) and a persistence layer, which will store and retrieve entities via a databases management system. The *view* will render a model object into a user interface, such as a web page. The *controller* is what mediates transactions between the user and the application. The typical

control flow of a basic MVC application is:

1. Client makes a request
2. Controller receives the request and transforms it into a manipulation of the Model
3. The updated Model is saved to the database by the persistence layer from the Controller
4. A View is generated by the Controller which makes the changes to the Model visible
5. The Controller responds to the Client's request with the generated View.

Symfony Standard Edition comes with many software bundles which extend the core Symfony framework to provide an enterprise-grade application skeleton. Three of the key bundles included are the Security, Doctrine, and Twig bundles, which provide user-rights management, an object-relational mapper, and a template engine respectively.

Security

Security in Symfony is handled by a two-step process: authentication and authorization. The authentication step identifies a user and is typically accomplished by having a user first visit the website, at which point they are authenticated as an anonymous user, and then having them provide a set of credentials to authenticate them as a specific user. Authorization occurs when a user attempts to access a resource of an application. Symfony allows for a set of user roles to be defined, of which each user has a set of, as well as an access control list, to determine exactly what resources any given user has access to.

Controllers are aware of what the current user's authentication is and, as such, can be configured to only allow processing to occur if and only if the current user has a given role. While an Entity will generically describe all instances of a piece of data in an application, if the currently authenticated user should only have access to an exact instance of an entity, an entry can be added to an access control list for that particular user-entity instance pair.

These features provide the Collaborative Environment with a powerful and robust mechanism that is essential to securing the Student data being tracked.

Doctrine

In a dynamic web application, Entities are constantly being created, read, update, and deleted. To keep track of these changes, it is natural to consider using a database management system (DBMS) such as MySQL or PostgreSQL. In an object-oriented program, these Entities often contain non-scalar data (e.g. lists, arrays) that must be persisted by a DBMS, which can be problematic because a DBMS can often only store scalar data (e.g. strings, integers). Using an object-relational mapper (ORM), such as Doctrine, solves this problem by managing the transformation to and from scalars/non-scalars when persisting an object.

As an example of how an ORM operates, let's consider an object-oriented system where a Student is enrolled in a set of Courses. Naturally, we would have two Entity objects, Student and Course. The fields of the Student entity could be an ID, a name, and a list of Courses that they're enrolled in (consider this to be a many-to-many relationship). The fields of the Course Entity could be a name and number. When a new Student has been created and they are enrolled in, perhaps, two different Courses and the ORM is told to persist this data, it will store the scalar data in the DBMS as its native types (names map to varchar, IDs and numbers map to integer), but for the Student's list of enrolled courses, since we established that there is a many-to-many relationship, the ORM will manage a Student ID-Course number tuple (Student 1 is enrolled in Course 1, Student 1 is enrolled in Course 2). Of course, when this data is requested from the DBMS, the ORM will convert the data back into Entity objects.

Twig

When a Controller has to return a new response, it is common for the View to be defined as a template that will be rendered though an engine, such as Twig, rather than explicitly crafted in the desired output format. For example, this could be accomplished by designing a Twig template that will output specific Model Entity variables and have it be processed into HTML by the Twig engine instead of just writing a mixture of HTML and PHP directly.

There are many benefits to using a template engine, such as:

- Template inheritance: Defining a parent or a layout template that can be inherited from children allowing for a consistent theme across an application.
- Syntax: Provides numerous shortcuts for applying common patterns to variables such as escaping output and modifying control flow.
- Speed: After compiling a template, the result will be optimized and low-overhead PHP when compared to freehand coding.

Overview

The primary motivations behind the creation of the Collaborative Environment are the need for increased collaboration and the need for data logging. The Collaborative Environment will be targeted at a variety of different user types (user classes), it follows that each will be permitted access to a different set of features and data. This means, that while an increase in collaboration is an overall goal for all user classes, specific collaborative goals vary based on how much access each class has:

- All users should have read access to announcements and an event calendar.
- All users should have access to an internal messaging system for one-to-one and one-to-many communication.
- All users should be able to answer surveys that are available to them.
- Students should be able to answer quizzes that are available to them.
- Students should have access to an availability schedule for teaching assistants.
- Students should have access to external educational resources.

- Teaching assistants should be able to create their own availability schedule.
- Teaching assistants should be able to create quizzes and surveys.
- Teachers should be able to create announcements and events for the calendar.

The next set of goals correspond to data logging. There is the possibility of some conceptual overlap between whether a goal is collaborative or data logging related, so the line has been drawn at whether or not the collaborative aspect is simply incidental to having to log data. If that is the case, then the goal will categorized under data logging.

- Parents should have access to their children's data.
- Students should have access to their own data.
- Teaching assistants should have access to the students that they are assisting.
- Teachers should be able to be able to view reports for the students that are available to them.
- Teachers should be able to edit data for the students that are available to them.
- Administrators should be able to create and access the entities by which data is stored.
- Administrators should be able to access user tracking data.

In the above list, there are two terms which require definition: *data* and *entities*. *Data* is defined as all personal information, survey results, and quiz results for the user it is in reference to, whereas *entities* are the core objects that provide the basis for all relationships in the user data.

Features

User Classes

In the Collaborative Environment, there exists a hierarchy of roles where each user will be a member of a particular level. These roles naturally correspond to the type of user that they describe and are presented in descending order of access to data: