CLARKSON UNIVERSITY

Overcoming Geographic Isolation: The Design and Implementation of a Web-based Collaborative Learning Environment

A Thesis By
Ryan James Lewis
Department of Computer Science

Advisor
Dr. Kathleen Fowler
Department of Mathematics

Submitted in partial fulfillment of the requirements
for the degree of
Master of Science
Computer Science

December 15, 2011

Accepted by the Graduate School

Date	Dean

The undersigned have examined the thesis entitled

Overcoming Geographic Isolation: The Design and Implementation of a Web-based Collaborative Learning Environment

presented by Ryan James Lewis,

a candidate for the degree of Master of Science,

and hereby certify it is worthy of acceptance.

Date

Dr. Kathleen Fowler, Advisor

Dr. Christopher Lynch

Dr. David Wick

Abstract

A web-based Collaborative Environment is designed and implemented that serves as both *1*) a solution to overcoming the challenges of operating a STEM outreach program in geographically isolated, rural areas; and *2*) a tool for further research into web-based (cyber) learning environments. This effort is in alignment with what the National Science Foundation has deemed to be a key research topic in the area of cyberlearning: the collection, analysis, and management of cyberlearning system usage data.

Students in need of educational assistance, when meeting in-person is not possible, can engage in personalized web-based learning with a tutor using assessments and materials backed by state standards. Educators can create surveys to receive feedback from targets regardless of their location while parents have access to their children's data and program information. Additionally, data logged about students and their usage of the system can be used in future analyses and reports.

The Collaborative Environment's feature set has been crafted to fit the requirements of IMPETUS, a STEP program serving rural Northern New York. A usability study was performed via a survey distributed to current IMPETUS participants to verify the system design and gather an initial set of qualitative data for future quantification.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The Science and Technology Entry Program (STEP) is a pre-collegiate preparation program under the New York State Education Department (NYSED). Its purpose is to help prepare minorities, historically underrepresented, or economically disadvantaged secondary school students for entry into postsecondary degree programs in scientific, technical, health-related, and the licensed professions [6]. Recently, STEP released a request for proposals in which it outlined a set of program priorities. An institute looking to receive funding through STEP would be given a high precedence should they propose to provide one or more of the program priorities. Paraphrased, these priorities are for institutions to provide programs and services to improve:

P1. The recruitment and retention of male participants;

P2. The recruitment and retention of Hispanic/Latino and American Indian participants;

P3. Eighth grade students' NYS Math and Science Assessment examination scores [13].

Additionally, STEP asserts a set of service and data requirements which its participating institutions must provide evidence of accomplishing in an annual report to continue to receive funding. These requirements are as follows:

R1. Provide evidence of formal collaborations between the funded institute, local school districts, collegiate outreach programs, and professional agencies;

R2. Provide program activities to assist students in acquiring the skills and aptitudes necessary to pursue postsecondary education leading to STEM (Science, Technology, Engineering, and Mathematics) fields;

R3. Provide services to enhance and increase students' involvement in research, internships, and college level coursework;

R4. Provide program services to enhance students' mathematical and scientific skills in accordance with the Advanced Regents Diploma; and

R5. Develop a means of involving a student's parents and give them a clearly defined relationship with the funded institution's program [13].

Data published in numerous studies [11, 9] 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 [5]. At the Salmon River School District, where there is a 58% American Indian enrollment, 54% of students scored below acceptable proficiency levels on the 8th grade Mathematics assessment [15]. 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 a STEP program resulting from the collaboration between St.Lawrence-Lewis County BOCES 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. IMPETUS woks towards their goal and addresses the STEP priorities/requirements by providing a collection of activities, programs, and services during both the academic year and summer to its participants. During the academic year, workshops are organized to occur on Clarkson's campus that are designed to improve a student's academic skills and awareness of career paths. During the academic year, students participate in a roller coaster design contest while paired with college student mentors. This contest acts as a framework for learning and applying math and science skills. High school juniors and seniors form teams to design and execute a research project about a STEM topic of their choice. This can lead to teams presenting at the annual STEP student conference.

Traditionally, collaboration in IMPETUS has been accomplished via in-person meetings between various combinations of its 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/mi.^{2}, whereas the state average is 355/mi.^{2}. The region of Northern New York constitutes nearly 20% of the state's land area, yet only 2% of its population and a mere 0.2% of the state's received federal funding. In a region of these attributes, it should not be surprising that the presence of traditional STEM learning venues such as museums, aquariums, science centers, and high-tech industries are basically non-existent. Combined, these circumstances do not create an environment suited towards an efficient and effective exchange of information [15]. As it stands now, collaboration between all IMPETUS participants only occurs seven times per academic year, once for each on-campus event.

IMPETUS proposed the creation of a web-based Collaborative Environment to aid in satisfying the STEP priorities and requirements by combining different technologies into a unified solution while increasing the convenience and accessibility of collaboration. Students in need of assistance, when meeting in-person is not possible, will be able to engage in personalized web-based cooperative learning (i.e. tutoring and mentoring) with Clarkson University students and web-based self-learning all using state-standard based materials and assessments. 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 and program information. With this system, collaboration between all IMPETUS participants would not be hindered by the geographical constraints of Northern New York. Instead, collaboration can occur at any place with a computer and an internet connection. The design and implementation of this Collaborative Environment forms the basis of this thesis.

This effort is in alignment with a national need for computer literacy in education. 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 [4]. 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 [8]. 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 Collaborative Environment will collect an extensive amount of data about not only the individuals using it, but also how they are using it.

1.2 Related Work

The proposed IMPETUS Collaborative Environment (hereafter referred to as simply the "Collaborative Environment") can be categorized as a Cyberlearning System (CLS). There are two classes of existing CLSs: the traditional, such as WebCT [10] (which has since been bought and renamed by Blackboard, Inc.[1]) and Moodle[2], and the specialized, such as Khan

[1] Blackboard, Inc.: `http://www.blackboard.com/`
[2] Moodle: `http://moodle.org/`

Academy[3].

Traditional CLSs provide access to educational resources in addition to user and course management features. Teachers can manage multiple courses, make documents available to their students, create quizzes for the students to take, and are overall very robust pieces of software, but this does not mean they are without their drawbacks. WebCT is a proprietary and licensed piece of software meaning that one is not able to create novel features that interact with the existing system, but rather one only has access to what Blackboard, Inc. provides. Moodle, being free and open source software, does allow for the creation of novel features in the form of plug-ins. Moodle is capable of providing some necessary functionality, such as user management, quizzes, and surveys, but does not offer specific data analysis and web tutoring/mentoring functionality. Additionally, traditional CLSs, because they are meant to manage an entire school, provide a substantial amount of potentially unnecessary features and complexity. Ultimately, for a small sized client organization, they provide a lot of features that are not needed and only some of those that are required.

A specialized CLS provides features specifically catered towards the requirements and needs of its developer. Khan Academy provides its users with educational videos and associated exercises organized in a hierarchical tree of topics. If someone wants to track their exercise scores or details about the videos they have watched, they must register for an account on the Khan Academy website. As a user completes exercises, their progress is tracked in the hierarchical tree to allow quick identification of problem and proficient topics; this can be used to personalize tutoring sessions with students. While this is perhaps the easiest CLS to start using, it does not allow for any flexibility, just as the proprietary WebCT, and does not provide all of the essential features that the Collaborative Environment needs.

The Khan Academy website is entirely open source, including their exercise framework. During the planning for the Collaborative Environment we initially reasoned that we could

[3] Khan Academy: `http://www.khanacademy.org/`

use this exercise framework as a drop-in solution for allowing the creation of quizzes in the Collaborative Environment, but upon further investigation, creating an exercise was not a task that could be accomplished by any type of user. Creating exercises using the Khan Academy software requires extensive use of HTML and JavaScript. In order to meet our objective of having a system that is accessible to many types of users, a technical wall such as this is unsatisfactory. This notion was the primary motivating factor behind the creation of WebCT, that all users should be able to utilize web-based learning environments, not just those with a technical background [10].

In terms of CLSs being relevant research tools, many research papers have concluded that the future of education will almost certainly include online components [17] and that, on average, students perform equally or better academically with online learning as with in-class learning [16]. However, they follow with the notion that further research is necessary to characterize and quantify these claims [16].

To summarize, the factors motivating the creation of a new CLS, the Collaborative Environment, are that it needs to be free and open source, accessible by users of technical and non-technical backgrounds, and provide a set of functionality (namely, a quiz system, a survey system, user management, tutoring/mentoring functionality, and data analysis) not wholly available in an existing solution. Additionally, because it is evident that there is a need for further CLS research [17, 16], the data-logging objective of the Collaborative Environment makes it a prime candidate.

1.3 Methodology

Creating the Collaborative Environment will be done following a modified waterfall model of software engineering: Requirements, Design, Implementation, and Verification [14]; 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 project, it is the most basic model to follow.

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

1.4 Contributions

The primary contribution of this work is a free and open source[4] web-based Collaborative Environment that combines, in a novel fashion, features of traditional and specialized CLSs. It combines the advanced user management, non-technical usability, and data analysis found in traditional CLSs with the personalized tutoring/learning and assessment of specialized CLSs. This feature set allows for the program to serve as both *1*) a solution to overcoming the challenges of operating a STEM outreach program in geographically isolated, rural areas; and *2*) a tool for further research into web-based (cyber) learning environments. The design of this software incorporates the results of usability research performed with different types of expected users.

1.5 Thesis Summary

This thesis is outlined as follows: Chapter 2 provides an overview of the Collaborative Environment. Defined are the goals of the system (i.e. what tasks should be accomplishable by using it), the primary features that will realize the goals, and what the perceived classes of users of the system will be. Chapter 3 provides a background of the preexisting software and services that were utilized in its development and continues into the detailed explanations of each primary feature's concept, design, and implementation. Chapter 4 details the

[4] Download the source code at: `https://github.com/cosmotron/Impetus`

usability study that was performed to gather an initial set of feedback on the Collaborative Environment. The study was accomplished with a user survey and described are its goals, participants, testing procedure, responses, and results. Chapter 5 discusses the conclusions of this work and where it can lead in the future.

Chapter 2

Software Overview

2.1 Goals

The primary motivations behind the creation of the Collaborative Environment are the need for *1*) increased and facilitated collaboration among a diverse set of program participants; and *2*) the need for data logging to assess program impact. The Collaborative Environment will be targeted at a variety of different user types (i.e. user classes), which will be further elaborated upon in §2.3. 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. Specifically;

- 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 contact information for teaching assistants.

- Students should have access to external educational resources.
- Teaching assistants should be able to create quizzes and surveys.

The following list of goals corresponds 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 be categorized under data logging.

- Parents should be able to view, but not modify, their children's data.
- Students should have access to their own data.
- Teaching assistants should have access to the students that they are assisting.
- Coaches should be able to be able to view reports for the students that are available to them.
- Coaches should be able to edit data for the students that are available to them.
- Application Administrators should be able to create and access the entities by which data is stored.
- Appliation 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 fundamental objects that provide the basis for all relationships in the user data.

2.2 Features

The features of the Collaborative Environment are designed and implemented to accomplish the goals outlined in §2.1. The following provide a high-level overview of each of the major features of the Collaborative Environment and describe which goals they aid in accomplishing. More specific details regarding each can be found in §3.2.

Academic Year

The Collaborative Environment will be used for multiple academic years, each of which have their own data that is dependent on it. For example, different quizzes might be given each year and students are usually in a different grade and classes. To give users access to a previous year's data, there is a simple interface to switch between them on-the-fly. This is a necessity for a robust data logging tool and to understand the longitudinal impact of IMPETUS on student aptitude and attitude towards STEM disciplines.

User Management

In order to keep track of data about individuals, each must have an account. Users that a registered as students will have data such as their gender and ethnicity, standardized test performance, and quiz results stored in this system.

District Management

Coaches, teaching assistants, and students are all associated with a district to better organize their data for analysis and to control access to private student data.

Learning Pathway

Students are provided with a visual representation of the knowledge that they should have as their school year progresses. This representation is a hierarchical tree of concepts based on both the New York State Core Content and Common Core State standards. Each node in this tree is an individual concept and has associated with it both video resources and quizzes. The tree acts as a guide to what students know and what they need more practice in. Students and their teaching assistants will be able to use these identified problem areas during collaborative-learning.

Quiz System

A system by which coaches and teaching assistants can create assessments of their student's knowledge is one of the key collaborative goals of this system. By allowing coaches and teaching assistants to create quizzes quickly and easily, it will enable them to receive valuable feedback from their students. From the student's perspective, they'll have access to a means of studying and brushing up on topics that they might feel they are struggling with.

Survey System

Coaches and teaching assistants can get additional feedback in a non-quiz like fashion. It can be used as an early intervention indicator in that if a student is not participating it might indicate that this student needs more attention. Feedback can also be garnered from students' parents thus increasing their involvement with IMPETUS.

Messaging

To facilitate communication between all of these users, a messaging system was devised that would allow for a quick means of contacting either a user or a group of users. This system keeps track of messages internally, but is capable of notifying a user of a new message via email if they have an email address associated with their account.

Analytics

A STEP funded institute must submit, annually, a report to the Education Department that summarizes data about the participants, activities, program content, and outcomes. The Collaborative Environment collects all of this data but it often needs manipulation and formatting to output the exact data that the report mandates. The analytical system is responsible for this task.

2.3 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 listed in order of least access to most (also note that any level's access to features is a superset of its preceding level's features). Further, there is a distinction between non-privileged and privileged classes: users of a non-privileged class have access to zero or one user's data (typically their own or their child's) whereas users of a privileged class have access to zero or more users data (typically all of their students).

Non-privileged

Anonymous

An anonymous user is any user that has not presented any authentication credentials. They may view announcements, calendars, and schedules.

Parent

Has access only to their student's educational data, quiz results, and attendance. They are able to participate in surveys and send messages to any user.

Student

May participate in quizzes and view their own various attempts at quizzes.

Privileged

Teaching Assistant

For any district that they are enrolled as an assistant for, they may view any student's data. They may also create surveys and quizzes.

Coach

May read and write student data for only those that belong to the same district as themselves. This data includes survey results, quiz results, and detailed student educational information. A coach may also enroll a new teaching assistant into their district and create new students and parents.

Application Administrator (aka Administrator)

Responsible for creating the fundamental entities which all other users interact with and have access to all user data and reports. They may create new academic years and districts, as well as the specific student activities, courses, and exams that may be tracked. They may also create a user of any role and assign additional coaches to their respective districts.

Chapter 3

Software Design and Implementation

3.1 Background

This section will provide details of the existing software and services that are leveraged to create the Collaborative Environment. For each, we provide the reasoning for its use as well as a technical description.

3.1.1 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 database 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. We describe these in detail below:

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, updated, 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.

3.1.2 Khan Academy

The Khan Academy is a nonprofit organization that aims to better global education by providing free, high-quality resources to anyone anywhere. [2] On their website, they provide in excess of 2700 videos that teach varying topics in STEM fields. The organization has been provided with the financial support and public accolades of large entities such as The Bill & Melinda Gates Foundation and Google [1, 3] indicating that they produce noteworthy content. The videos that Khan Academy hosts are all created by Salman Khan who holds an MBA from Harvard and three science degrees from MIT. Each is produced in a fairly consistent manner using a software whiteboard, a screen capture program, and a microphone while typically lasting about 10 minutes.

As stated in §2.1 and §2.2, students using the Collaborative Environment should have access to supplemental learning resources and will have the ability to take quizzes given by their coaches and teaching assistants. The Khan Academy videos are ideal for this use as they are concise and short enough to be associated with individual quizzes that are created in the Collaborative Environment. If a student, while taking a quiz, feels that they are in need of some assistance, a link to the an appropriate Khan Academy video will be readily available to them.

3.2 Features

We now describe the main features of the Collaborative Environment as previously outlined in §2.2. Each subsection henceforth provides the concept, design, and implementation of an individual feature. They have been arranged in such a way that each successive feature will build on the previously introduced ones.

3.2.1 Academic Year

In a traditional document-based teaching environment, student data is stored in numerous grade books or spreadsheets which are held by many different people in many different places. Each teacher, for instance, would have their own set of student records and a district will have a permanent record for each student containing information such as their standardized test scores. If an administrator managed to collect all of this data into a single place, however frustrating that may be, the next thing they would have to do to it, before any type of analysis, would be to group it all by what year the record corresponds to.

As the Collaborative Environment will be collecting data year after year, an Academic Year entity is required. Coaches, teaching assistants, and students all have data associated with them that can change on an annual basis. Students will be taking different classes, have taken different standardized tests, and participated in different activities as well as be in a different grade altogether or have graduated. Each year can also bring with it a different round of quizzes and surveys and even new coaches or teaching assistants.

Design

The Year entity is really very simple, just a primary key identifier, *id*, and the year itself as an integer, *year*. Explicitly storing the years that the Collaborative Environment will be tracking in a table allows consistency throughout the entire application and customization by an administrator. An alternative implicit design would have been to keep a *date* column in each entity that was going to have time-sensitive content added to it and the server would be responsible for getting the current year from the operating system, but problems arise if the system's time and date are not correct.

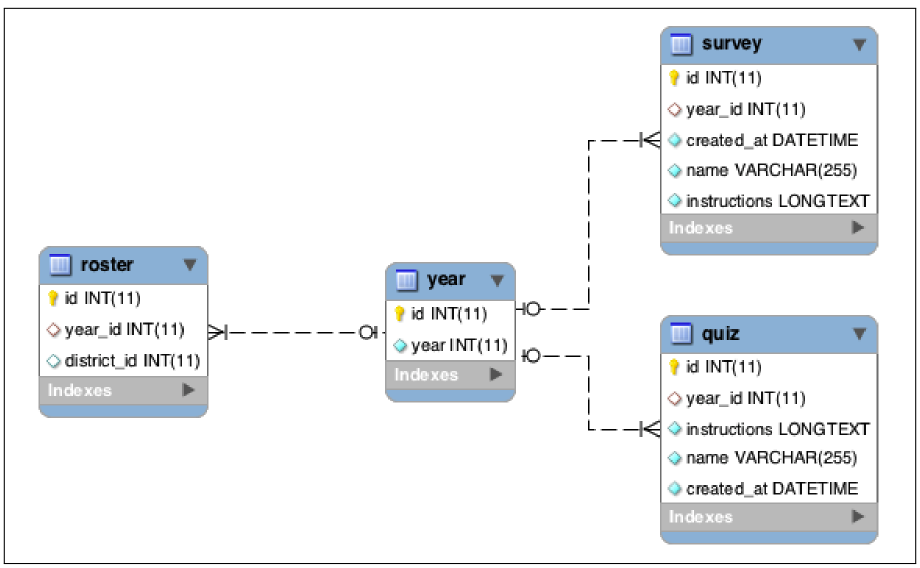


Figure 3.1: Year Enhanced Entity-Relationship (EER) Diagram

We see in Figure 3.1, that the year entity appropriately maintains a one-to-many relationship with each of the tables that reference it. In an EER diagram such as this, each box represents a table in a database and each row in a box corresponds to a column in that table. A connecting line indicates that a relationships exists between its connected tables. If we take the relationship between the year and survey tables as an example, the line, which is drawn using Crow's Foot notation, indicates that a survey can be given during one year and that a year can have many surveys given during it. This diagram additionally indicates that many quizzes (see §3.2.5) and many rosters (see §3.2.3) can also only be associated with only one year.

The user's choice of which year they want to have as their current context for data viewing and manipulation will have to persist between each page of the Collaborative Environment. Having to constantly select what year to interact with would be cumbersome, thus, once a user initially visits the web site, a PHP session variable will be established that tracks the year which will then be used by subsequent SQL queries to retrieve the requested data.

Implementation

As established in the previous section, the selected year must be persisted from page-to-page as well as be accessible in such a way that it will not require a user to constantly have to interact with a selector.

This was implemented by having a year selector list appear in the upper-right corner of each page in the Collaborate Environment, as shown in Figure 3.2. The list is populated by making an asynchronous HTTP GET request to the URL `/year` which will both provide all available academic years that have been configured and indicate which is the current year as selected by the user.

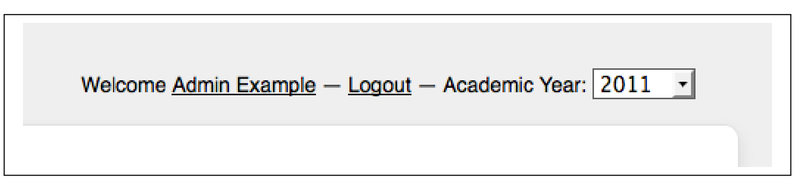


Figure 3.2: Year Selector Highlight

Upon choosing a year from the selection, if it differs from the currently selected year, then an HTTP POST request will be sent to `/year/change/`*`selected-year`*. This will cause the year PHP session variable to be updated to *selected-year*. If the server responds with a success message, then the current web page will be refreshed and any changes to the data that are dependent on the year will be reflected.

3.2.2 User Management

A User entity is perhaps the most fundamental object in the Collaborative Environment. Users of each class, as defined in §2.3, with the exception of anonymous, will be required to have an account with this system, and that account is what the User entity represents. The set of users minus anonymous will be referred to as "registered users".

This entity facilitates the fulfillment of the Collaborative Environment's goal of logging user data by allowing the storage of a student's personal data such as gender, ethnicity, if they've graduated, and what college they are attending, in addition to student educational data such as standardized test scores, courses enrolled in, and after-school activities.

The collaborative functionality of the system also requires a User entity. Quiz attempts, survey submissions, and messaging need a way of differentiating one user from another.

Design

The Collaborative Environment internally refers to the set of user classes as "roles". Roles are easily stored in the database by using a role table such that its *name* is stored in a column. The User entity, as modeled in Figure 3.3 stores two types of data: required information that applies to all registered users and personal/educational data that applies strictly to students. Required information consists of a user's *username*, *password*, *salt*, *firstName*, and *lastName*.

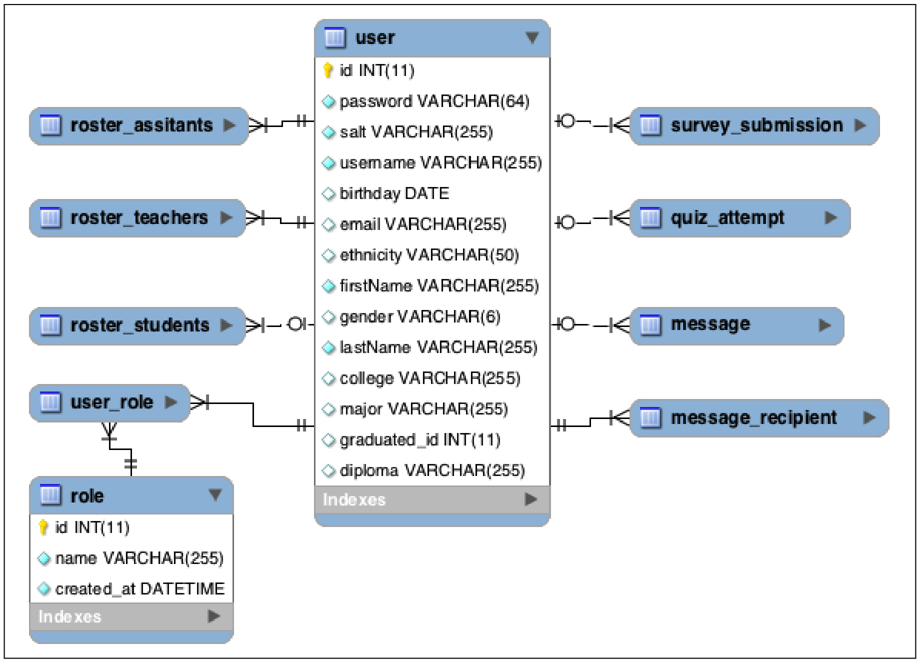


Figure 3.3: User and Role EER Diagram

Each of these attributes has an obvious use except, perhaps, for *salt* which is a precautionary security measure. All user passwords are hashed using using the SHA-256 cryptographic hash function. This means that if two users had the same password, then they would obviously result in identical hashes. This fact can be exploited by people that have large masses of precomputed hashes. A salt, which is a randomly generated string, is append to the plaintext of a password before it is hashed. Now, even if the user's password is something common, what is hashed is, most likely, something unique.

For the sake of simplicity, the personal/education attributes *birthday*, *email*, *ethnicity*, *gender*, *college*, *major*, *diploma*, and *graduated* as year-independent data. This means that if and when these attributes have a value, that it will not vary based on what academic year the client has selected to be the current year; their values will persist through all years. On the other hand, there is strictly educational data that we do treat as year-dependent: activities, courses, exams, and grade level.

Figure 3.4, outlines the general model to achieve a year-dependent relationship between

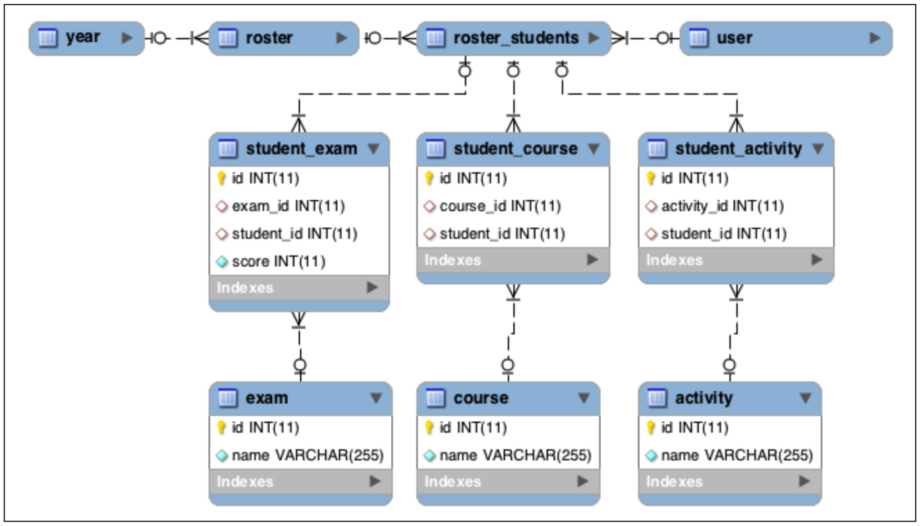


Figure 3.4: Student user EER Diagram

a student and their educational data and will be explained in depth in §3.2.3. Any given year is associated with many rosters, each of which contain a set of students in varying grade levels. Each student in a roster can then be associated with many exams, courses, and activities.

Implementation

User management is broken up into three processes: creation, updating, and reading. As an interface to each of these processes, privileged users have access to a user management interface seen in Figure 3.5. This interface will only display the users that the currently authenticated user has access to. For administrators, this will be all users, whereas a coach or a teaching assistant will only see the students that belong to the district that they themselves belong to. For example, if a coach is listed as such in the district "Public School #1" for the year 2011, then the user list will only show students that also enrolled at the district during that year.

This idea of coaches and teaching assistants only having access to the students that are on the same roster as them is persistent throughout the application. The details regarding districts, rosters, and permissions will be explained in §3.2.3.

With this data now being tracked by the Collaborative Environment, trends and reports will be able to be generated. Tracking student participation in activities, courses, and exams is now possible for the set of all students or even subsets such as minorities, males, or females.

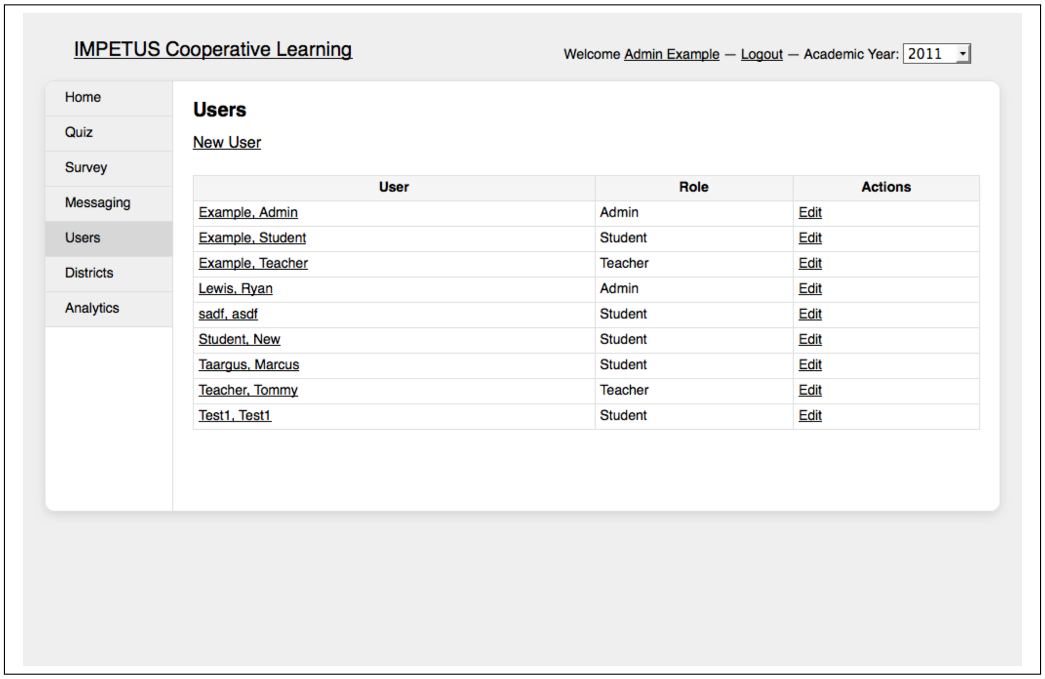


Figure 3.5: A privileged user's view of all the system's users.

When a privileged user must add or edit a user, the interface appears as it does in Figure 3.6, except all of the fields are simply blank in the case of adding a new user. The interface has has been split horizontally into two sections, the top is for personal/educational data whereas the bottom is for strictly educational data. In the top half, input fields have been grouped into sets of required and supplemental (optional). This allows the user to readily identify which fields they must have a value for and not have to wait for the program to return an error.

The supplemental fields for *ethnicity*, *diploma*, and *major*, provide the user with a limited selection of options rather than a freely enterable text input because this will allow for consistent data analysis. If users were able to enter whatever they wanted for these, it would

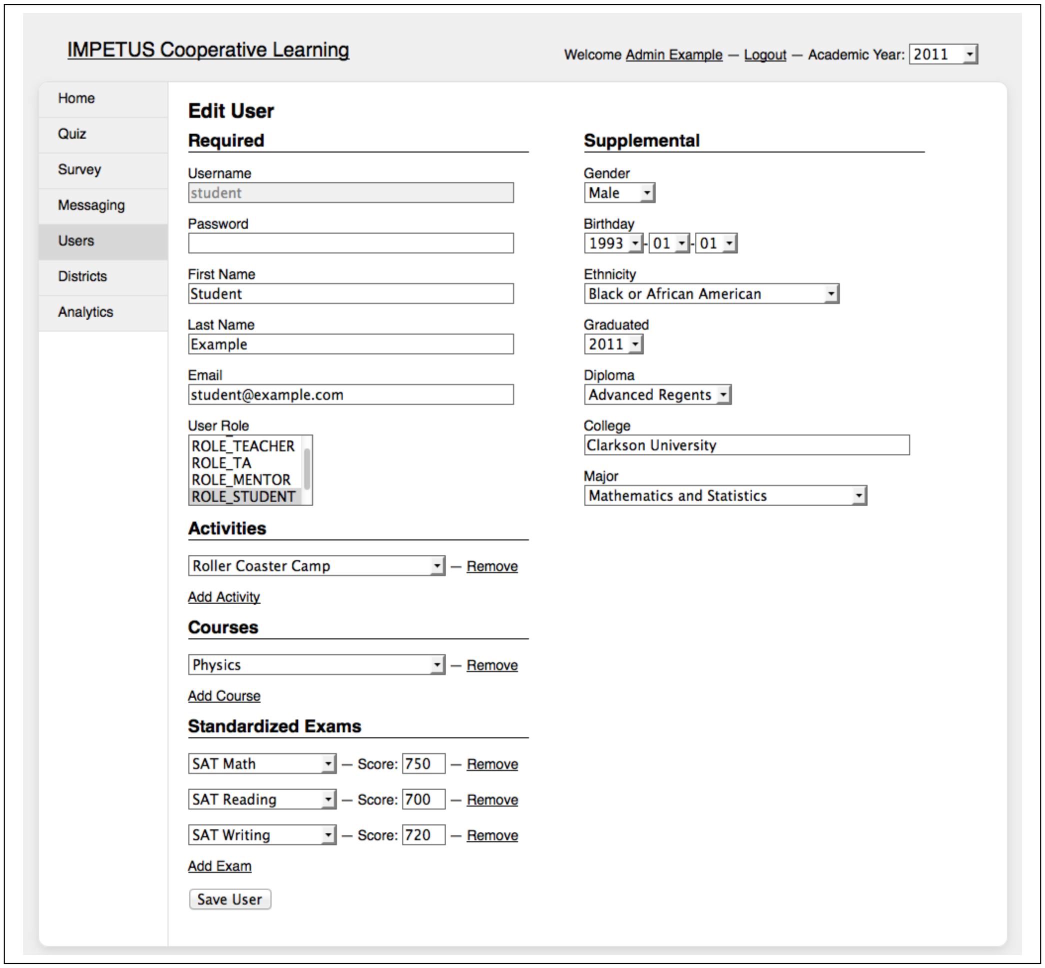


Figure 3.6: A privileged user editing a student's account.

become difficult to use that data to generate accurate reports. For example, if one person were to enter that a student was going to major in "Mathematics" and another were to enter "Math", then the system would have to either know what all synonyms of "Mathematics" are and group them or, more likely, ignore some data because it was not clearly related.

Since the bottom half of the screen only applies to students, where strictly educational data is entered, it will only appear if the user is assigned the Student role. As stated in the Implementation section, *activities*, *courses*, and *exams* are year-dependent. This means that if a user were to choose different academic years while editing a student's data, they would see these fields change appropriately. A student can be associated with as many year-dependent fields as necessary by clicking the Add button for the appropriate section.

Lastly, viewing user data in a more organized fashion is shown in Figure 3.7. This simply provides a view into the data stored for a user. Just as the add/edit screen would update its data when the current academic year is changed, so will the view screen.

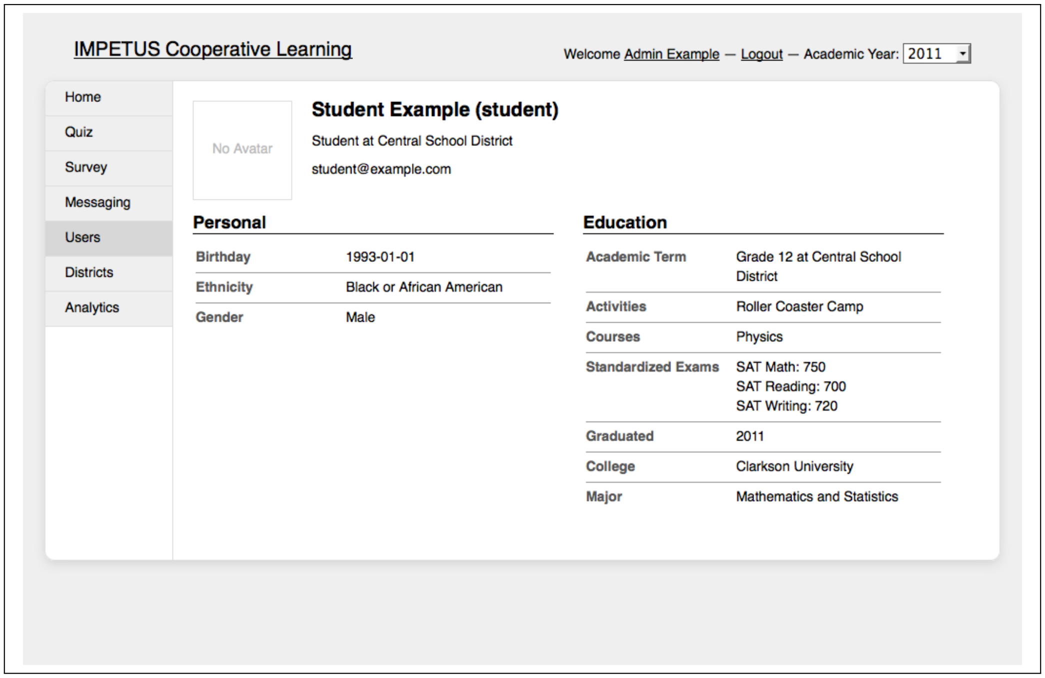


Figure 3.7: A privileged user's view of a student's profile.

3.2.3 District Management

In order to more accurately group the Collaborative Environment's users together and to control access to private student data, a district entity is necessary. If a district was to only ever have the same users associated with it, then a district entity alone would be sufficient, but this is not true; a district's relationship with its coaches, teaching assistants, and students is year-dependent. To allow for this, we introduce the idea of a roster entity where a district can have many rosters such that each roster is associated with a single year.

Using this grouping, it is now also possible to allow access to data based upon which roster a user belongs to. For instance, coaches could be given access to only the students in the roster that they themselves are a part of for a given year, thus they would not be allowed to see student performance in another district. Having a district associated with a student for any given year also allows for a wider range of data analysis as it becomes possible to, for instance, compare student performance between districts.

Design

Using the concept of a roster, it becomes a join table (a table which links common attributes between two or more tables, in this case *year_id* and *district_id*) for the district's many-to-many relationship with a year. In other words, a district will be associated with many years because it will have a different set of users each year. Likewise, a year will be associated with many districts because each academic year, the Collaborative Environment will be tracking multiple school districts.

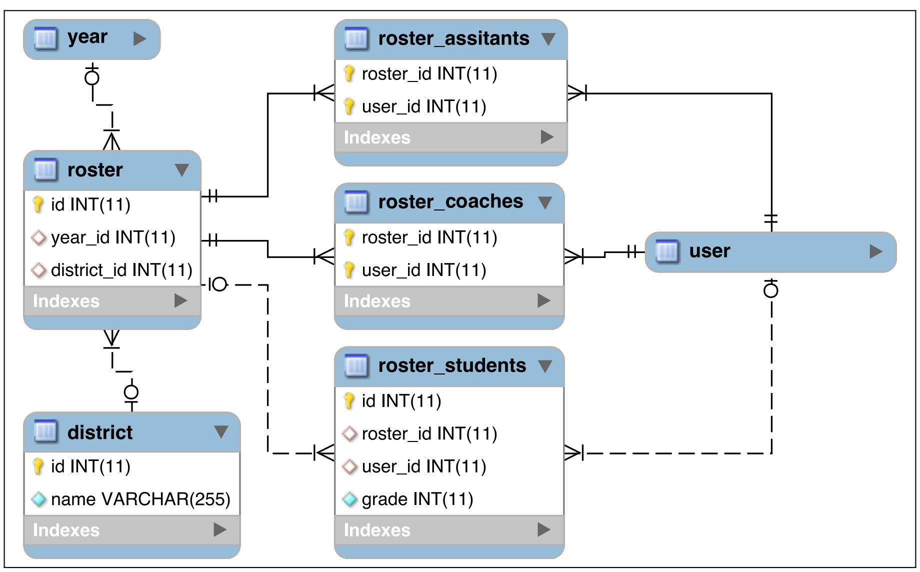


Figure 3.8: District and Roster EER

Figure 3.8 shows the many-to-many relationship between district and year via roster. A roster must be uniquely identifiable since it is in a many-to-many relationship with the user table, thus it has its own *id*. To keep track of a district's three groups of users for a given year (coaches, teaching assistants, and students), its roster maintains three different relationships with the user table. If a user is a coach and is to be in a district's roster for a given year, then they will be added to the roster_coaches table. The same logic applies to assistants and students, except a student can additionally have a *grade* associated with them (e.g. grade 9, grade 10).

Implementation

Privileged users have access to the list of districts, as show in Figure 3.9, which they are in the rosters of for the currently selected year where they can a. An exception to this is for administrators, which have access to all districts. From this list, privileged users can edit any district in the list and administrators have the additional ability to create a new district.

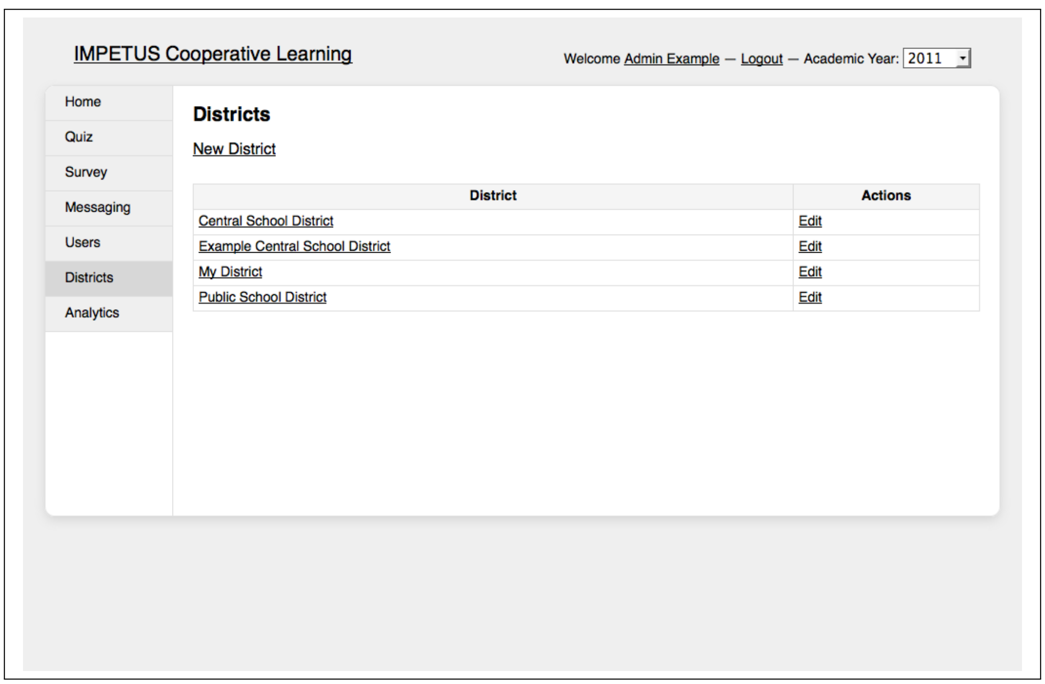


Figure 3.9: An administrator's view of all the system's districts.

When editing or adding a district, the user will use the interface displayed in Figure

3.10. One is able to add users of different classes to the roster by searching for their name in the appropriate search boxes. In this initial implementation, there is a constraint such that coaches and students can only be in one district's roster for a given year. This was set because, in most cases, coaches and students generally do not need to be associated with multiple districts. Due to this limitation, users that are already in a different roster for the current year will not appear in the search results for a user. If a student, for example, switches school districts, they must first be removed from their current district and then they will appear in the search results when being added to their new district.

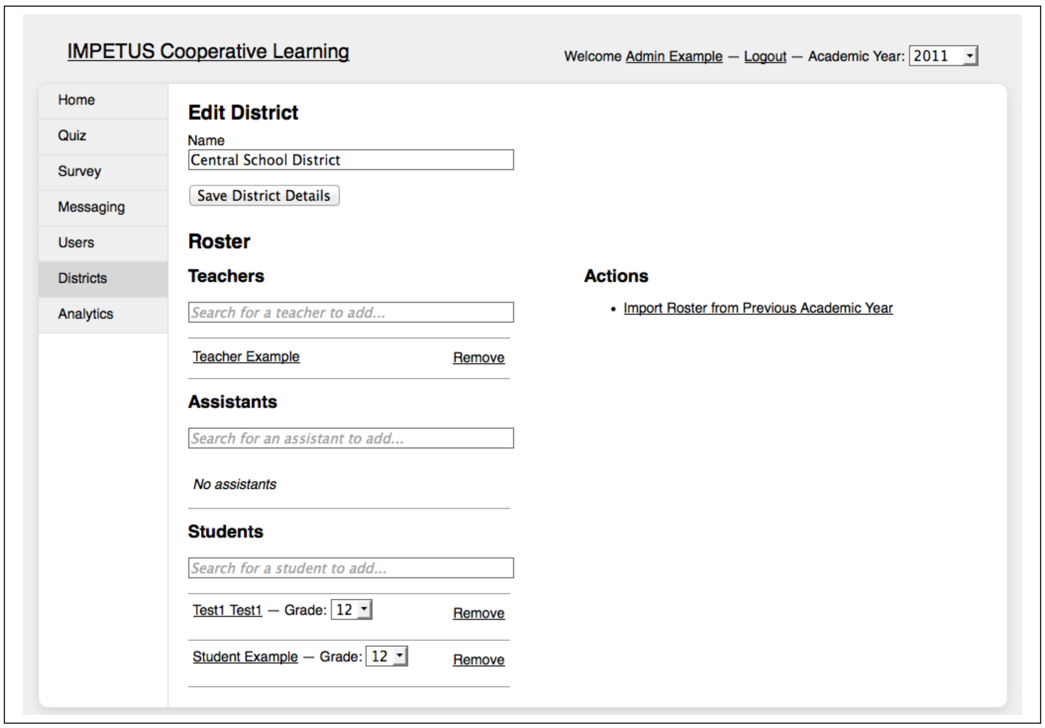


Figure 3.10: A privileged user editing a district.

Each new academic year, a roster will need to be created that ties the districts to that new year. This can be a tiresome activity, especially if a district has a lot of users that need to be associated with it. To try to ease this process, there is an import feature which will simply look at the previous year's roster for the current district and copy it to the current year's roster. It will also automatically increment the student's grades by one; this is due to the assumption that a majority of students will advance to the next grade. Those that do not advance can be manually edited.

3.2.4 Learning Pathway

While exploring the use of the Khan Academy video and exercise educational resources, we took note of a tool which they call a Knowledge Map (see Figure 3.11). They advertise this map as an organization of all the site's exercises that is capable of suggesting the best exercises for the student to work on and can remind them when they seem to need a review [2]. Due to the open source licensing of the Khan Academy, we adopted this idea to better suit the needs of IMPETUS.

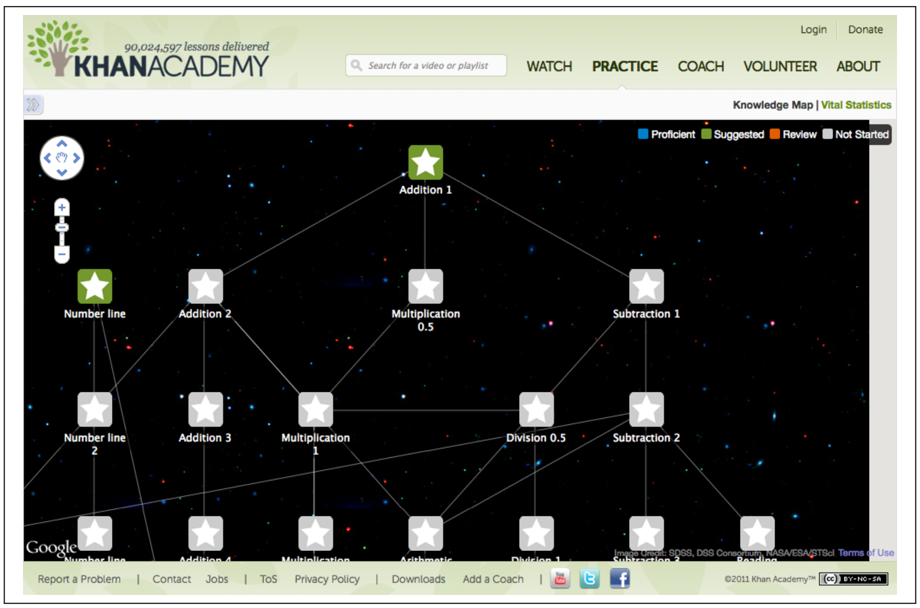


Figure 3.11: Khan Academy Knowledge Map

In the Khan Academy Knowledge Map, clicking on a node takes the student directly to the exercise that is associated with it. For the Collaborative Environment, we wanted one layer of abstraction to sit between the clicking on a node and going to an assessment. This abstraction (i.e. a "node details" screen) would be able to provide a link to the state standard justifying its existence in the map, links to quizzes built using the Collaborative Environment, and links to educational resources (such as videos at the Khan Academy) that will aid the student in achieving mastery in the quizzes.

Design

The learning pathway is a tree structure containing nodes and edges. This can be translated into a database by having a node table with a many-to-many relationship with itself (see Figure 3.12). The join table for this relationship holds any given node's parent node(s). To keep track of where a node should appear, it has a horizontal and vertical position (*h_pos* and *v_pos*). Lastly a node needs a link to its details which are, as described above, links to educational resources and quizzes.

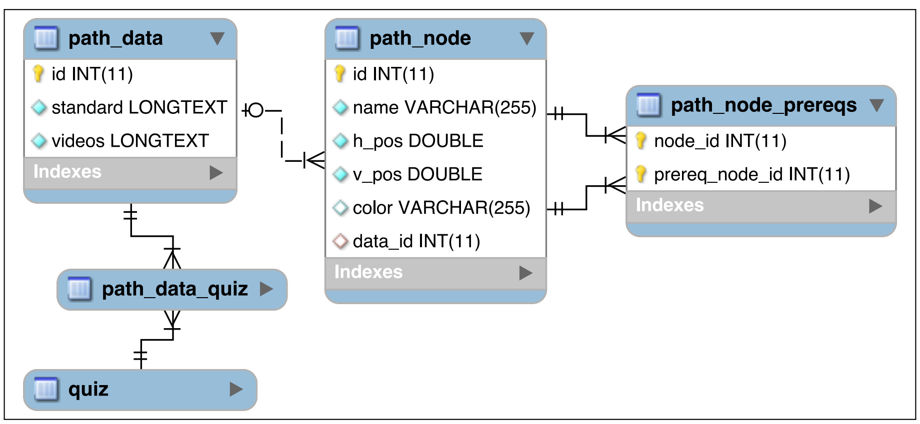


Figure 3.12: Learning Pathway EER

Implementation

The tree itself is drawn using the Google Maps API such that each node is a map marker and its edges are line segments. All the nodes are retrieved from the database and drawn to the screen at the their appropriate coordinates. Each node is given one of three color templates: neutral, completed, and incomplete. Completed nodes appear as green and indicate that the student has achieved a perfect score on the quizzes associated with the node. Incomplete nodes appear as red and used when a student has attempted the quizzes, but has not achieved a perfect score yet. Neutral nodes appear as white and are used when a student has made no attempts at the node's quizzes. Next, all of the node relationships are queried for and then drawn to the screen. Using the Google Maps API gives some nice flexibility such as panning around a large tree when it gets too large.

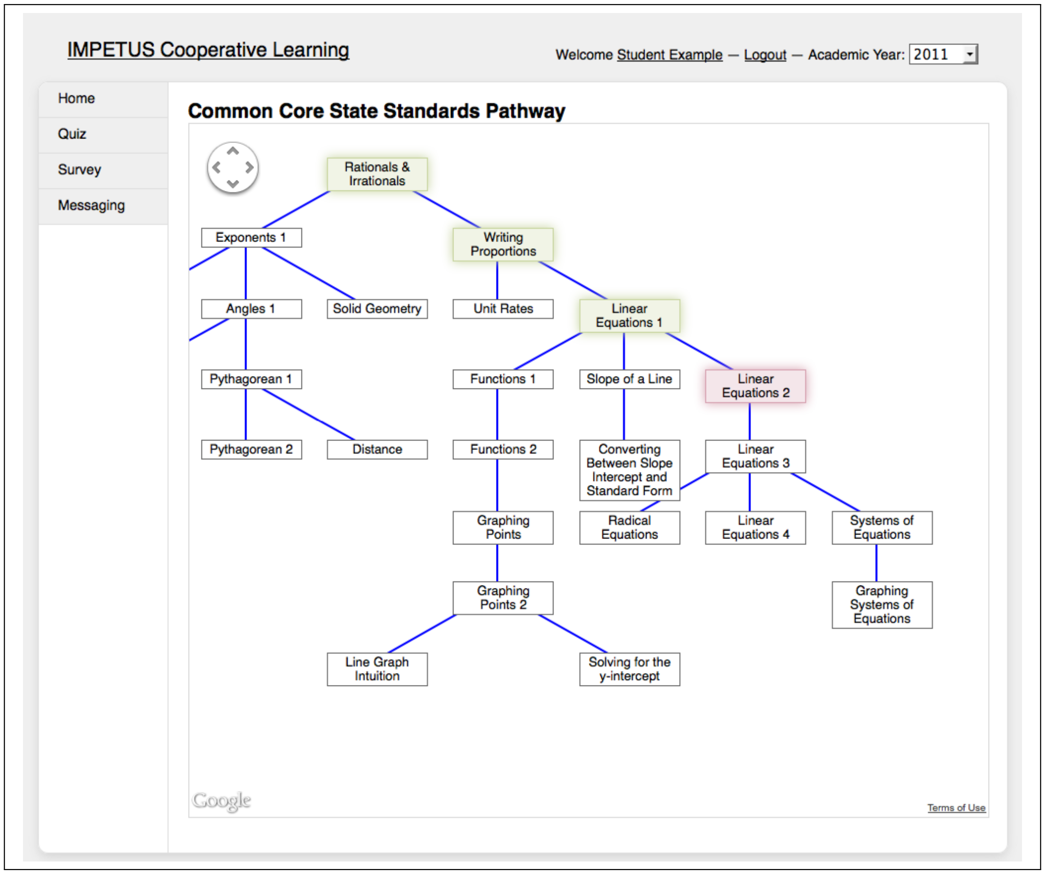


Figure 3.13: A student views their current learning pathway.

In Figure 3.13, we see the final output of one student's learning pathway. The layout will appear the same for all students, but the coloring of nodes is, of course, dependent on their individual performance. The graph in this figure represents IMPETUS' interpretation of the Common Core State Standards [12].

Clicking on a node will take the student to a details page that contain links to: *1*) the state standard providing justification for learning this topic; *2*) quizzes that coaches or teaching assistants make for this topic; and *3*) educational resources to aid in learning the material, such as links to videos at the Khan Academy.

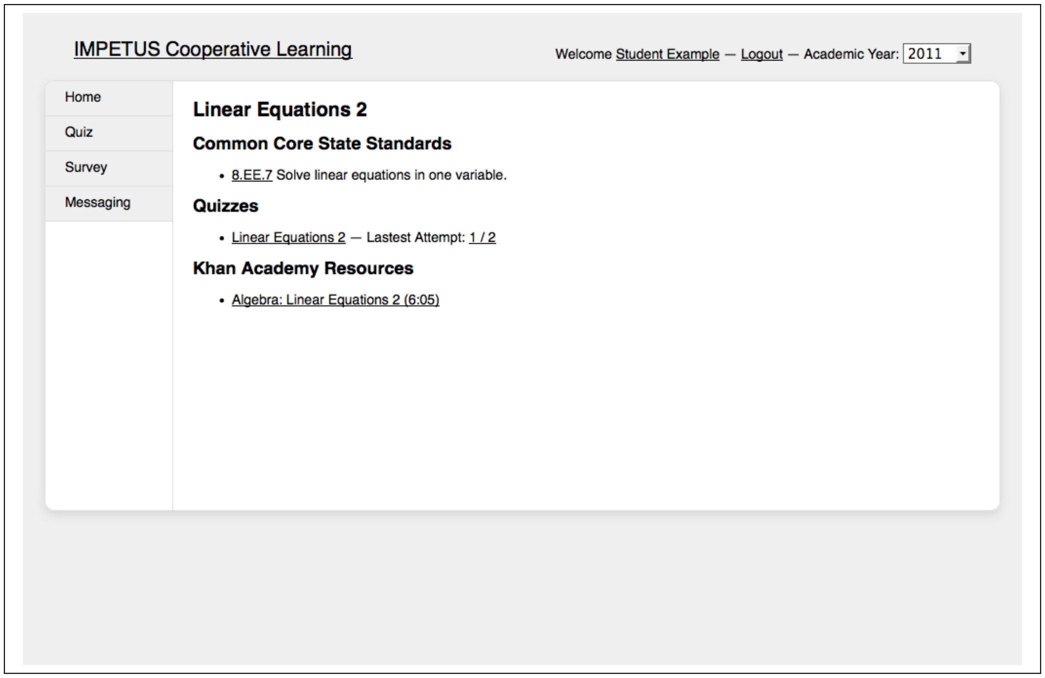


Figure 3.14: A student views their progress in the "Linear Equations 2" pathway node.

In Figure 3.14 each of these link varieties can be seen. The quiz section also lists and links to the last attempt the student made at this. Since it is not a perfect score, the node for "Linear Equations 2" is marked as incomplete. Details of creating this quiz and making that attempt can be found in §3.2.5.

A current limitation of this system is that adding new nodes and rearranging existing ones in the tree is a tedious and manual process.

3.2.5 Quiz System

A system by which coaches and teaching assistants can create assessments of their student's knowledge is one of the key collaborative goals of this system. By allowing coaches and teaching assistants to create quizzes quickly and easily, it will allow them to get valuable feedback from their students. From the student's perspective, they'll have access to a means of studying and brushing up on topics that they might feel they are struggling with. A quiz can be made about any topic, but when used in conjunction with the learning pathway, as described in §3.2.4, it creates a rich means of showing students the topics they need to know (based on state standards) while providing them with the resources to succeed.

Design

A quiz in the Collaborative Environment can be broken up into two sets of relationships: *1*) a quiz, its questions, and its question's answers; and *2*) a student, their attempt at a quiz, and how they performed on each question during that attempt. Figure 3.15 shows that one quiz may have multiple questions and that a question may have multiple answers. This approach allows for two different types of questions to be used in a quiz: short answer and multiple choice. If a quiz question only has one answer, then it is a short answer, whereas if it has multiple answers then it is clearly a multiple choice question.

To allow students the ability to learn from their mistakes, quizzes may have multiple attempts made at them to achieve a perfect score. This requires that a user have a many-to-many relationship with a quiz itself via an attempts join table. To be able to store exactly which questions a student got correct or incorrect along with their actual answer, every quiz attempt has a many-to-many relationship with a quiz's questions called result. The idea is that an individual attempt at a quiz is comprised of a series of results, or a result of answering each quiz question. Each result is able to store the exact answer that a student

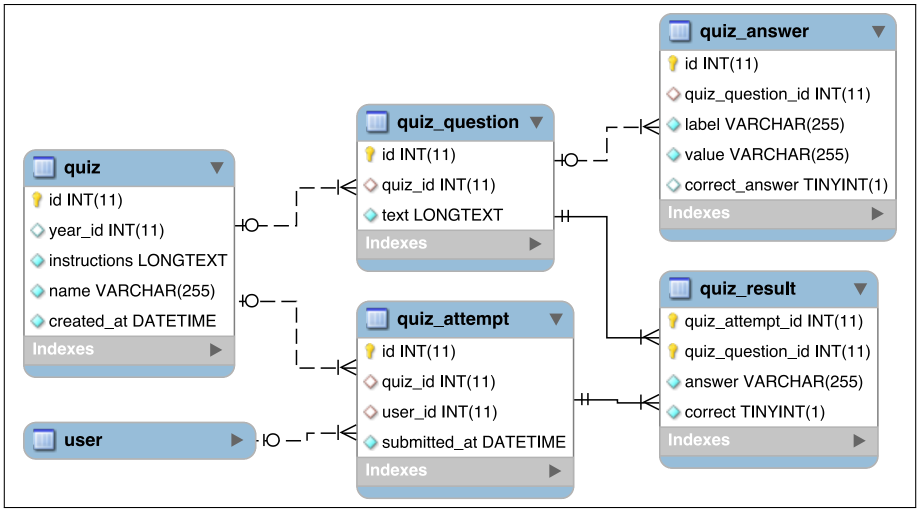


Figure 3.15: Quiz and Quiz Result EER

gave along with if it was correct.

Implementation

When a privileged user wants to create a quiz, they will be presented with the interface seen in Figure 3.16. First, the creator must choose a name for the quiz and may enter instructions for the student to follow. Next, the creator may add as many problems to the quiz as they want, each of which they may add one or more answers to.

When adding the text for the question, the creator may opt to use the LaTeX displayed math shorthand (i.e. `\[ ...\]`). Should they choose to, then their question will be rendered by a JavaScript library called MathJax[1] that displays equations in LaTeX markup in all web browsers. If they choose not to, then whatever is entered will simply appear as the question. An example of each can be seen in the questions in Figure 3.16.

As stated in the design section, a question can have one or more answers associated with it. If it has only one answer, it makes it a short answer question allowing the student to freely type in an answer. The creator must include a "text label" for the answer which is what

[1]MathJax: `http://www.mathjax.org/`

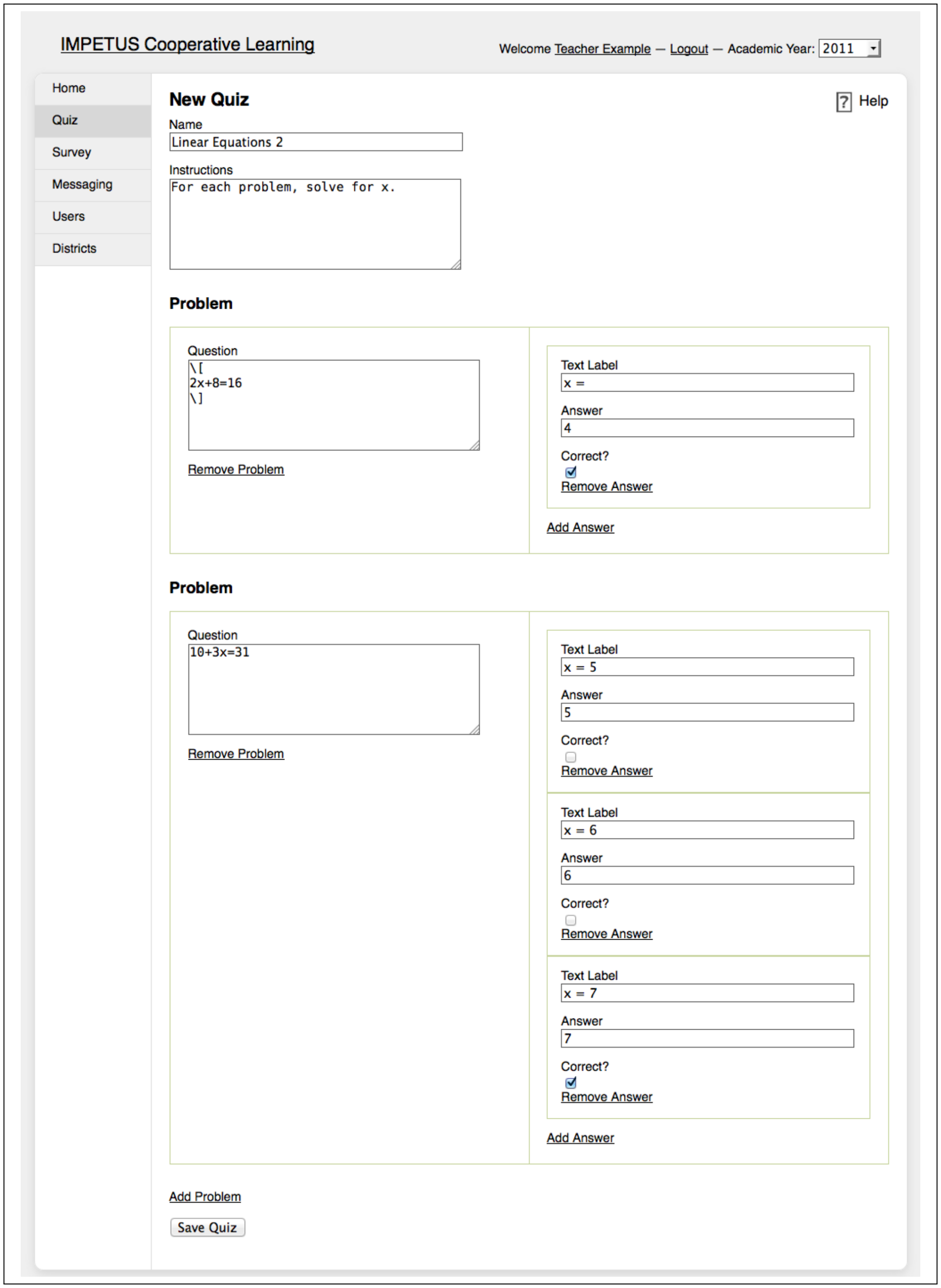


Figure 3.16: A privileged user creating a quiz.

will appear before the text input when attempting the quiz (e.x. "$x =$"). If the question has multiple answers (making it a multiple choice question), then this same text label becomes the visible text identifying one answer from another.

Switching to the perspective of a logged in student, they would see the interface shown in Figure 3.17 which displays the available quizzes, some of which the student has attempted and one of which that they have not.

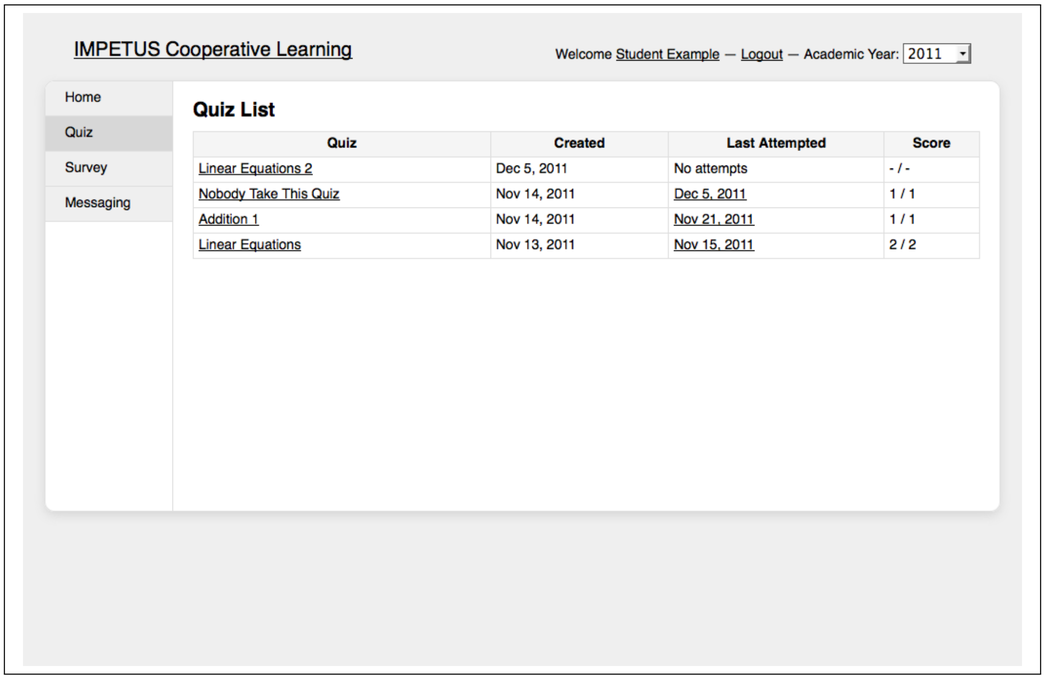


Figure 3.17: A student's list of available quizzes.

If the student were to attempt the "Linear Equations 2" quiz, they would then be presented with what is shown in Figure 3.18; a rendered version of what was being constructed in Figure 3.16. One can see that "Problem 1" has a styling that looks identical to what one would expect to see in a LaTeX rendered document as well as a plaintext question in "Problem 2". "Problem 1" is a demonstration of a short answer question and allows the user to freely type in an answer, whereas "Problem 2" is clearly a multiple choice question.

When a student enters their answers for the quiz, the system will simply compare their answers to answers that were set during quiz creation. This implementation can only exactly

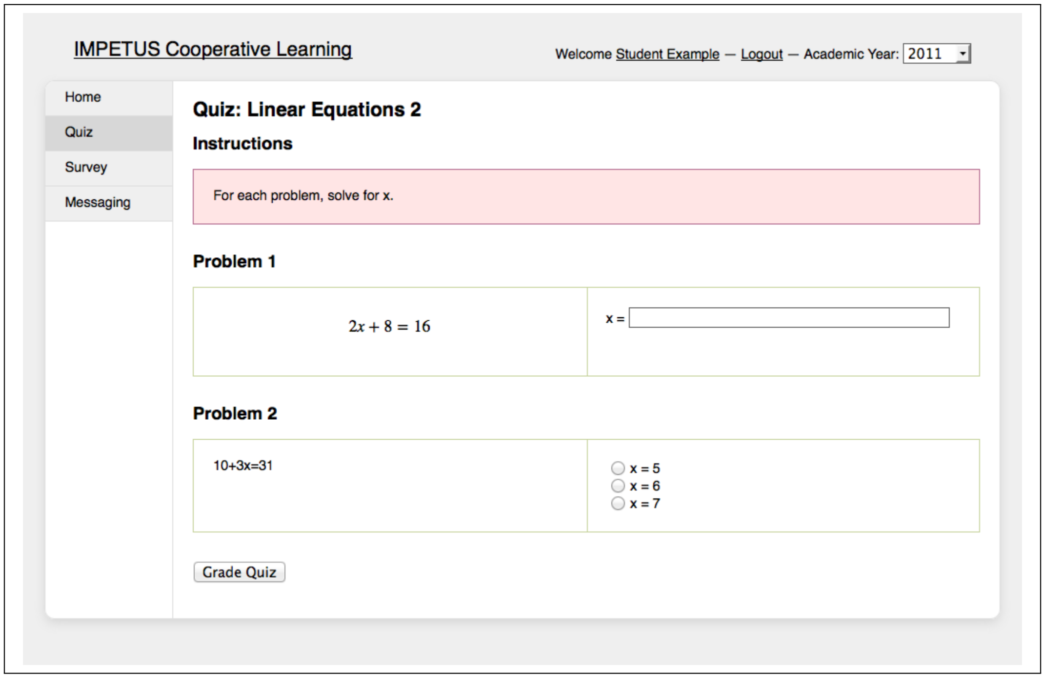


Figure 3.18: A student attempts a new quiz.

compare a student's input to the values that were given during a quiz's creation (e.g. it can not round or reduce answers). The results of submitting a quiz for grading can be seen in Figure 3.19.

As previously stated, students have the option of taking a quiz as many times as they see fit and the system will, in turn, keep track of those attempts. This allows for analyzing student performance on individual topics and for assessing the quality of quizzes. We see the list of attempts that was made on the "Linear Equations 2" quiz in Figure 3.20.

Once students start answering quizzes, coaches and teaching assistants are able to see their progress on them from the results interface show in Figure 3.21. The viewing policy on quiz results is that coaches and teaching assistants may only see the students that are in the same district roster as them for the currently selected year (see §3.2.3 for details of the district and roster design).

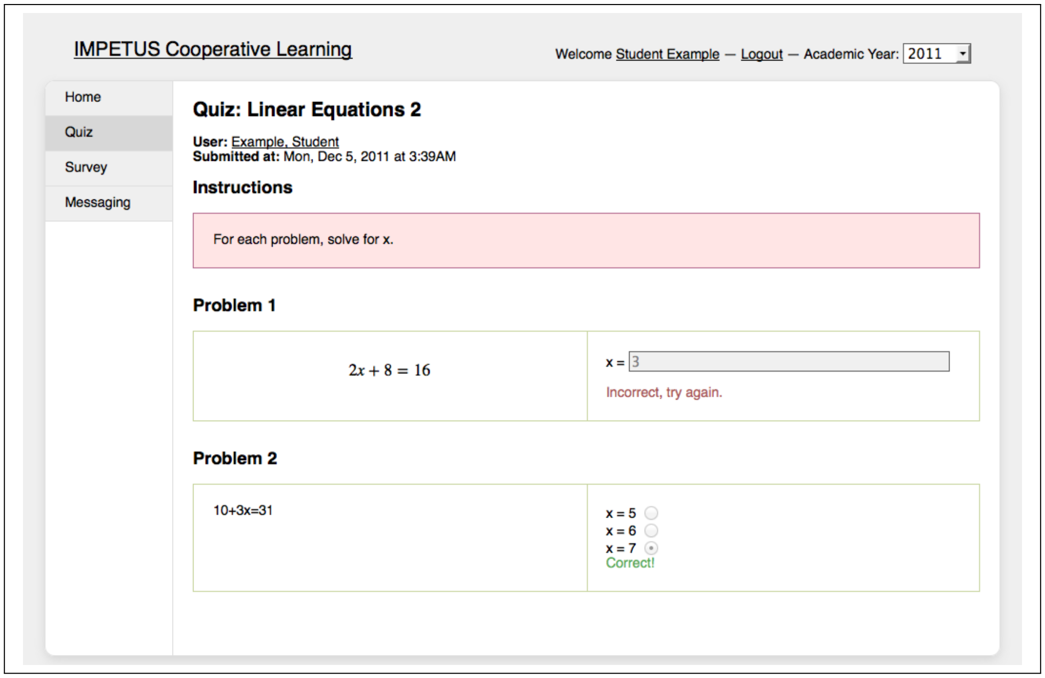


Figure 3.19: A student receives the results of their quiz attempt.

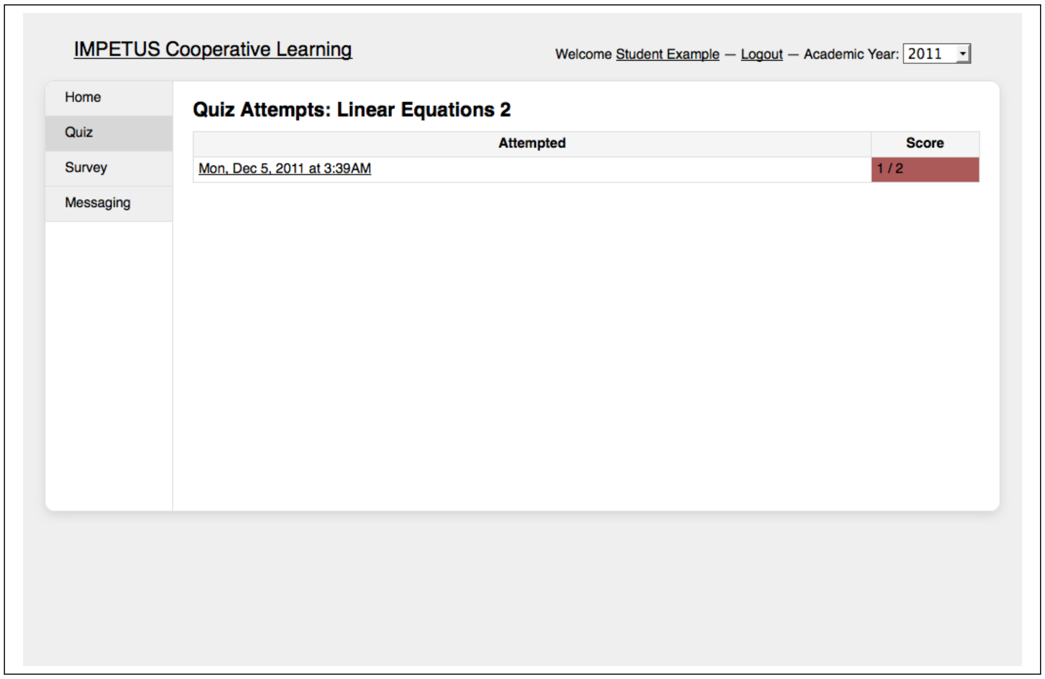


Figure 3.20: A student views all of their attempts at a quiz.

3.2.6 Survey System

Having an integrated survey system as part of the Collaborative Environment is critical not only for the collection of data, but also for tracking student participation. If a student is

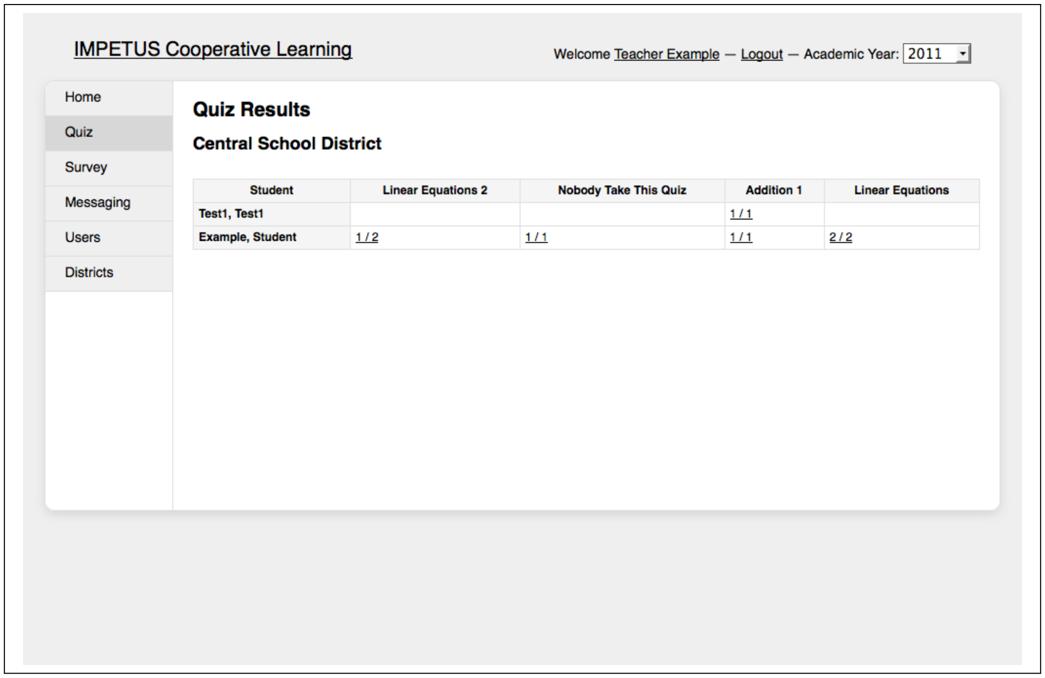


Figure 3.21: A coach views the quiz results of the students in their district.

not actively participating, then this can be treated as an early intervention indicator thus allowing them to receive more attention. Having surveys be integrated with the rest of the Collaborative Environment, as opposed to outsourcing this functionality to another service (e.g. Google Docs Spreadsheet forms), allows for all centralization of student data. Had this data been available in several spots, it would then require aggregation which might result in a loss of data.

Design

A survey is designed in a nearly identical fashion to that of a quiz. The only difference being that a survey is not allowed to have multiple submissions by the same user (whereas with quizzes, a single user may attempt the same quiz numerous times). Surveys can also be broken up into two sets of relationships: *1*) a survey, its questions, and its question's answers; and *2*) a user, their submission of a survey, and what they answered for each question in that submission.

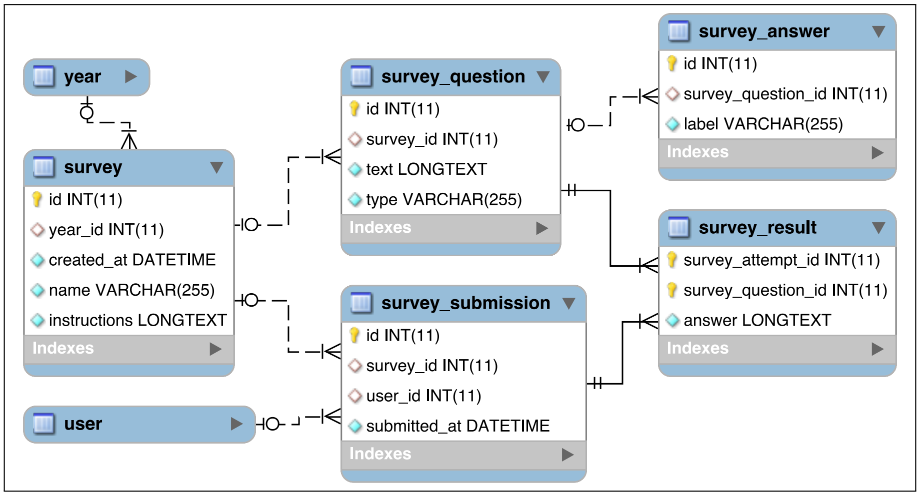


Figure 3.22: Survey and Survey Submission EER

Figure 3.22 shows that one survey may have multiple questions and that a question may have multiple answers. A survey question may be one of three types: multiple choice, short answer, and a scale. A scale question is a shorthand for making a multiple choice question with these answers: "disagree strongly", "disagree", "disagree somewhat", "no opinion", "agree", "agree somewhat", and "agree strongly".

Storing what each user answers for a particular question in a survey requires that there exist a many-to-many relationship between a survey submission and a survey question called result. Each result is able to store the answer a user provided.

Implementation

Privileged users are permitted to create surveys and do so using the interface seen in Figure 3.24. The creator must supply a name for the survey and, optionally, instructions for the survey participants to follow. Following this, they may add as many of the three possible question types (multiple choice, short answer, and scale) as are necessary.

The first question in Figure 3.24 is a multiple choice question. Once the text is added, the

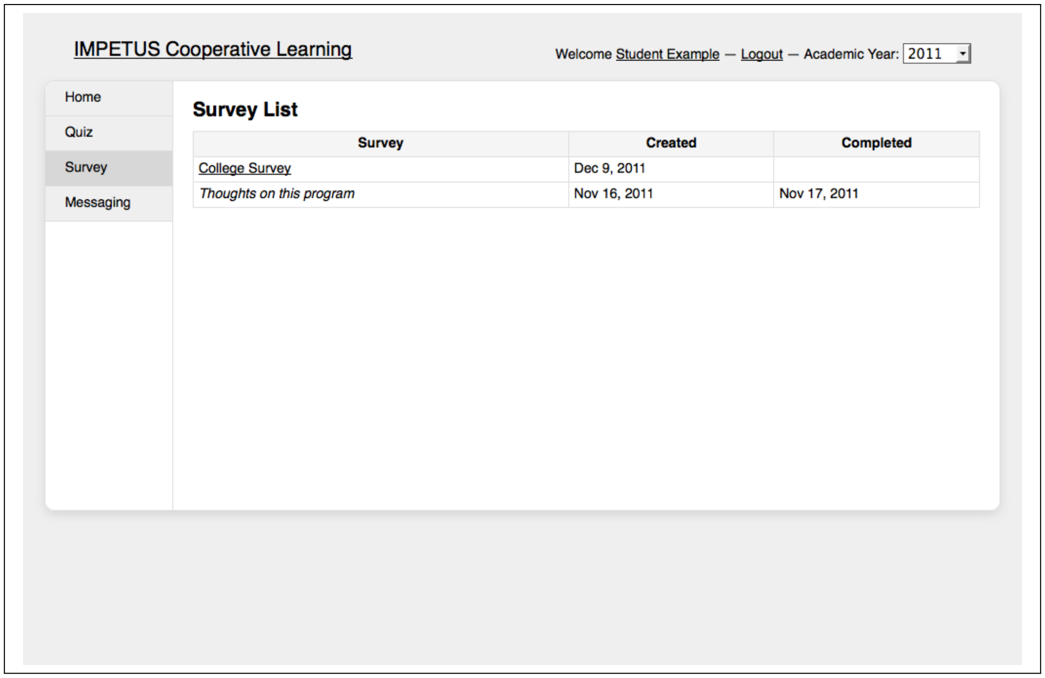


Figure 3.23: A student views a list of incomplete and completed surveys.

creator may add as many possible answers as necessary to the question. The next question in a short answer, allowing a participant to freely type in an answer. Lastly, there is a scale question. The agreement scale will automatically be rendered when someone views the survey.

As a student that is interested in participating in a survey, they would first encounter a list of surveys organized by their completion state. A survey can only be completed once so the list of italicized entries is just to act as proof of submission. In Figure 3.23, the survey being created in Figure 3.24, “College Survey”, is now available to be completed.

Upon choosing to participate in the “College Survey”, the student will complete it using the interface shown in Figure 3.25. Once a submission occurs, a student will not be allowed to see either the results or what they submitted. This is reserved for privileged users.

At any time after a survey has been released, its live results can be viewed by privileged users via an interface such as the one seen in Figure 3.26. For each question in the survey, if the question was multiple choice or a scale, then the number and percentage of each answer

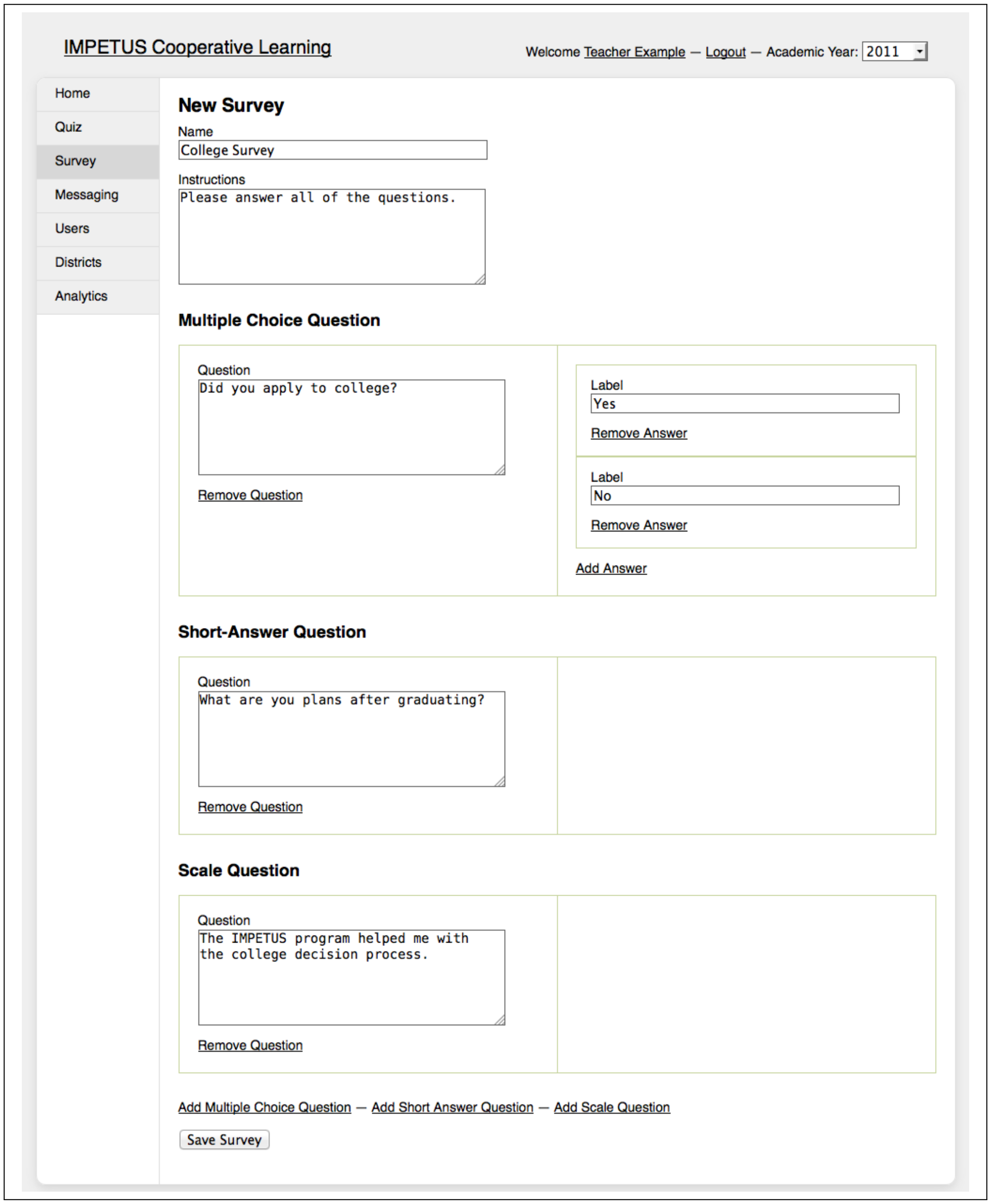


Figure 3.24: A coach creates a new survey.

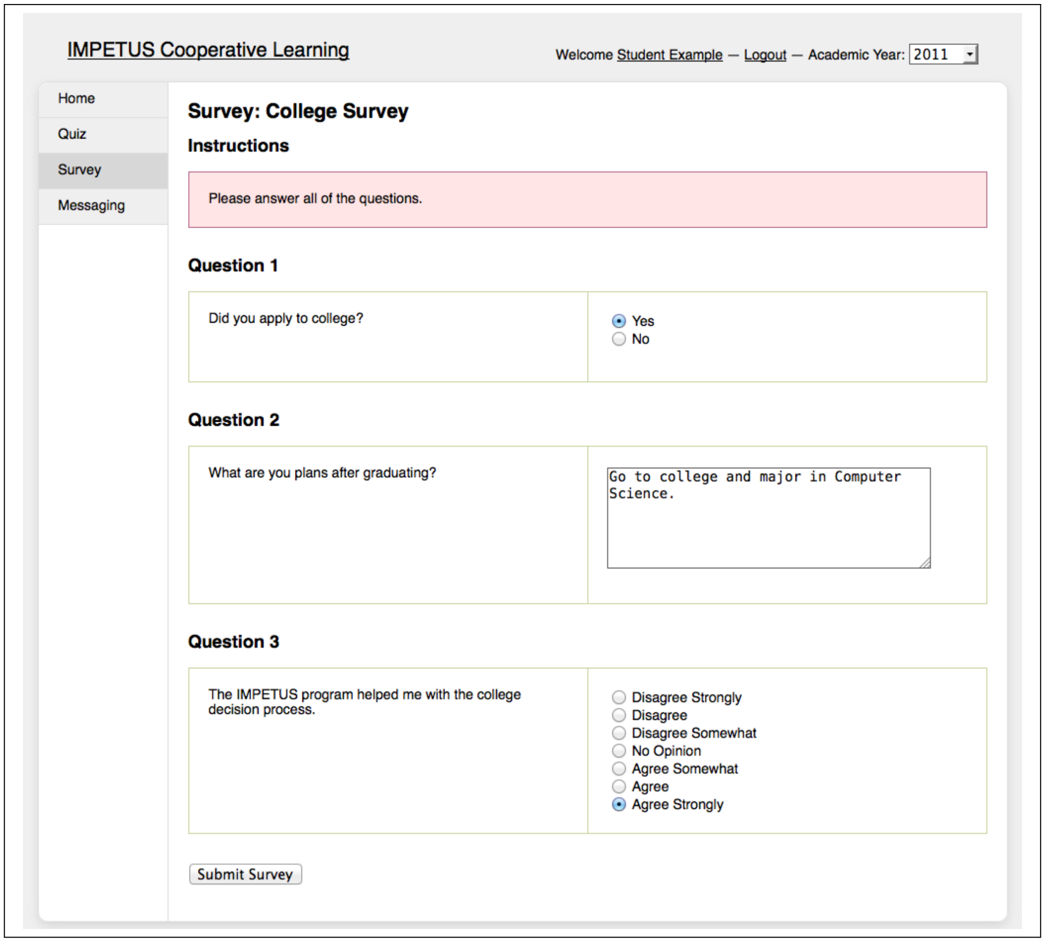


Figure 3.25: A student attempts a new survey.

is provided. If the question was a short answer, then an accumulation of the answers is provided. Additionally, a list and total number of participants is shown below the question data.

3.2.7 Messaging

To facilitate communication between all of these users, a messaging system was devised that would allow for a quick means of contacting either a user or a group of users. A problem that IMPETUS has continuously faced is that of organizing the contact information of all its associated people. There was no quick way to contact students and contacting coaches

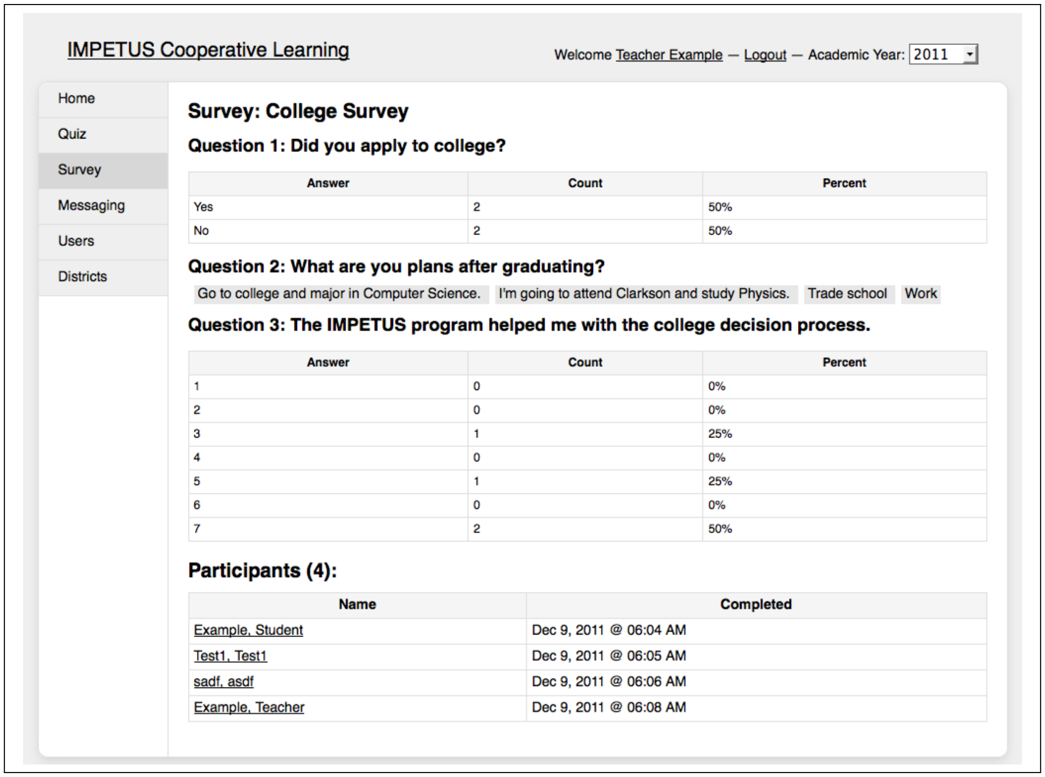


Figure 3.26: A coach views the results of a survey.

involved sorting through contact lists.

As a solution to this, the Collaborative Environment has a simple messaging system that allows for contacting any of the system's registered users. One may send a message to just an individual, a group of individuals, or even an entire class of users quickly. The system keeps track of messages internally, but is capable of notifying a user of a new message via email if they have an email address associated with their account.

Design

The messaging system treats messages and their replies as a correspondence thread, that is, one message is a parent and then replies may be added after that parent. When a new message is created, a single instance of that message is stored in the message table (see Figure 3.27). All of the designated recipients are then added to the recipient table, including the sender themselves. This means that even though a message will be sent to at least two people, there is only ever one instance of the message itself and all recipients are basically given viewing rights to that message via the recipients table. When a user has opened a message or deletes it, their entry in the recipients table for that message is simply modified.

If a message is a reply to a parent, then its entry in the message table is marked as such and all the recipients of the parent message will then receive a copy of the reply.

Having a message system built like this allows for an archive of messages and reduces message text duplication in the database.

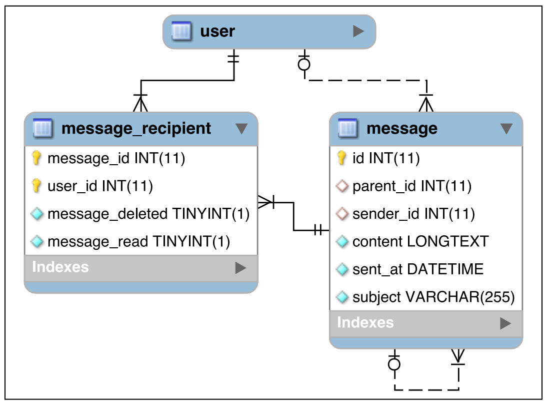


Figure 3.27: Message and Message Recipient EER

Implementation

As stated previously, any registered user in the system has access to the messaging feature and, as such, can message any other user in the system. In Figure 3.28, a student is sending their coach a message regarding a homework assignment. The “To” field allows for any number of recipients to be added to it via an autocompleting search feature. Recipients can also be classes of users such as “Students” as a quick way of addressing all the users in the system that have the student role. Upon sending the message, it will appear in both the sender’s and the recipient’s message lists (but in the case of the sender, it will be marked as read). Additionally, if the recipients have an email address associated with their account, a copy of the message will be sent to that address along with a link to view the message in the Collaborate Environment. The message list acts as a combination of an inbox and an outbox.

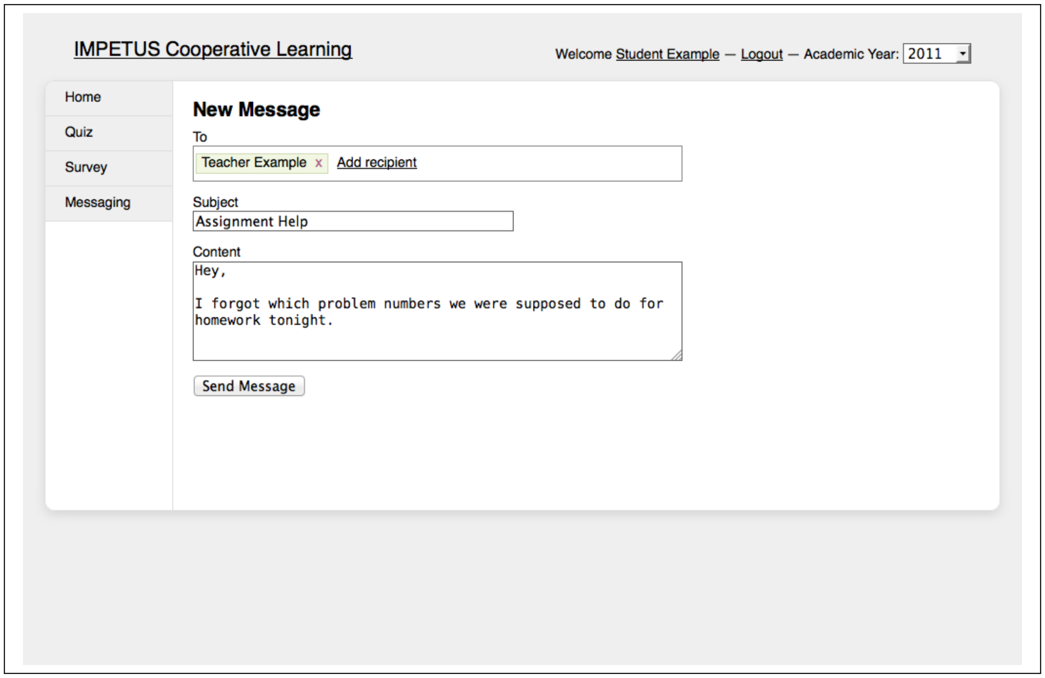


Figure 3.28: A student sends a new message to their coach.

When the coach that was marked as the recipient of the student's message checks their message list (Figure 3.29), they will see it at the top and highlighted as it is an unread message.

Viewing a message will open the chain of messages with the parent at the top. In this case, since the student was the original sender, their message will appear at the top of the list and replies will be appended. The coach responds to the student's inquiry, as shown in Figure 3.30, and an email will be sent to the student, if appropriate, notifying them of the response. If the coach were to return to their message list, the thread of messages will now be marked as unread.

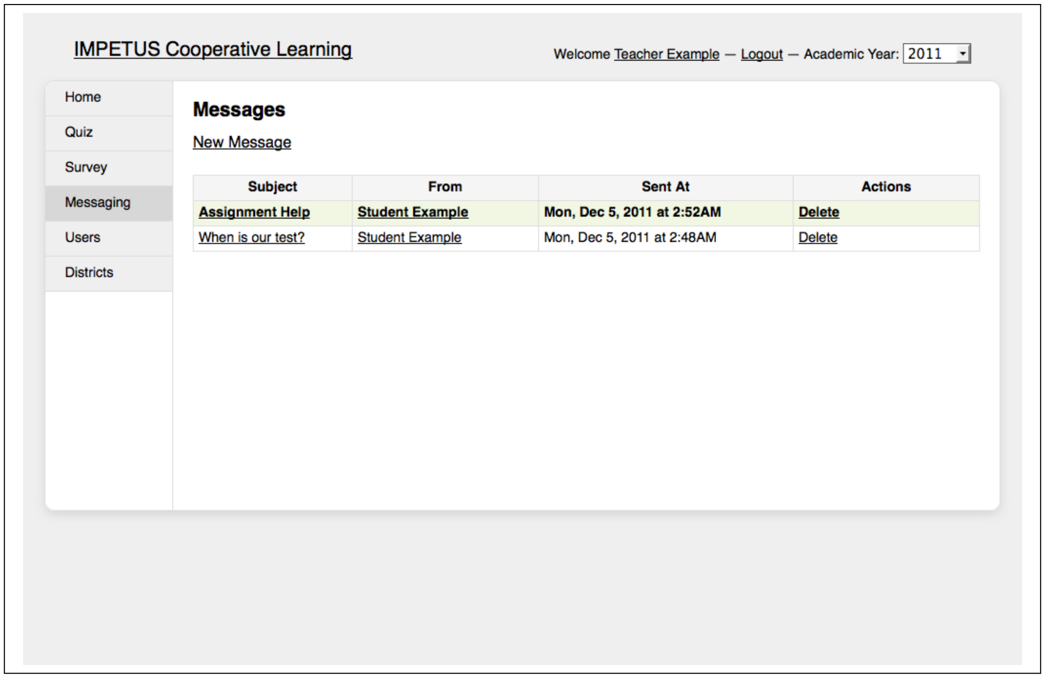


Figure 3.29: A coach views their message list.

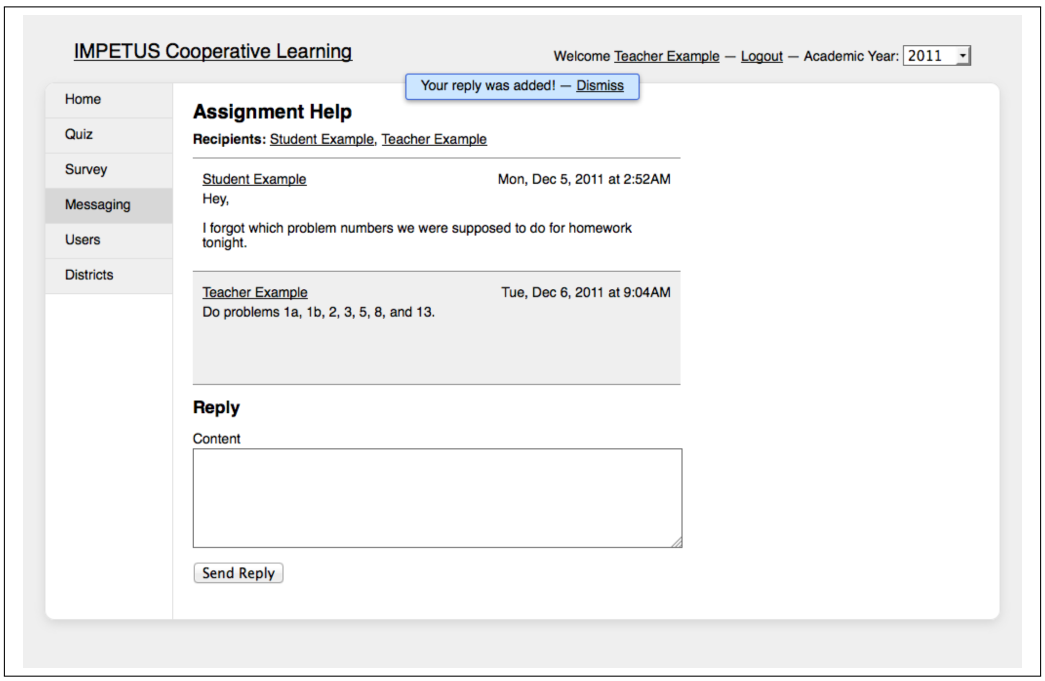


Figure 3.30: A coach replies to a student's message.

3.2.8 Analytics

As stated in §1.1, a motivation behind creating the Collaborative Environment is to collect data about the individuals using it, in particular, the students. The methods that have been described for storing and organizing student personal data (§3.2.2), educational data (§3.2.2 and §3.2.3), and quiz results (§3.2.5) also need retrieval methods that will combine said data into more meaningful representations.

A STEP funded institution must submit, each year, a report to the Education Department that summarizes data about the participants, activities, program content, and outcomes. This summary data is required in a tabular form as specified by a master report form which the Education Department provides [7], examples of which are a participant roster, participating school roster, student assessment scores, SAT scores, and student graduate placement.

The Collaborative Environment uses the data it collects to output a web based version of the state-requested information in a format that matches what is expected by the annual report. Additionally, it will output the same data as a comma-separated value (CSV) file for manipulation and more advanced processing via a spreadsheet program.

Design

Gathering the data to output is a straightforward process when using the tables described in the previous sections, as it is just a matter of constructing database queries to get the result you want. Once a result set of a data has been generated, outputting it depends on how it was requested.

A result set can be returned to the requesting user as either an HTML table on a webpage or as a CSV file to be downloaded. Both of these formats are basically tables, except that one has its data wrapped by HTML tags and the other by commas. This is an ideal scenario for the use of abstraction which Figure 3.31 outlines.

Since all analytical queries will have a result set of data that is tabular in nature, we can abstract that notion into `AbstractResult`. It represents the idea that results will have to be outputted based on one of two formats, either HTML or CSV, but how exactly that is done

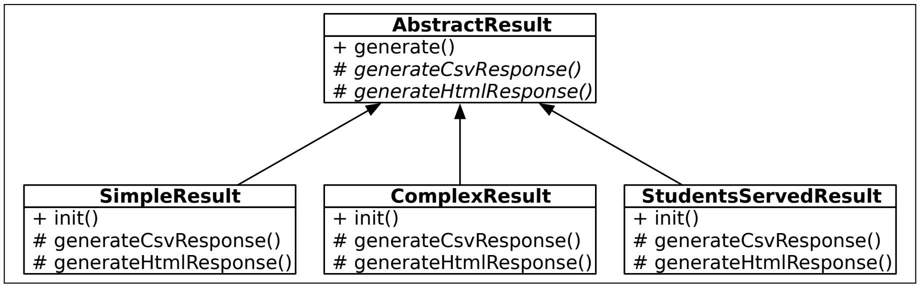


Figure 3.31: Analytics UML Object Diagram

is left up to a specific implementation (e.g. a table of participants in STEP will look and be implemented differently than a table of how many males in each grade of each ethnicity participated in STEP for a given year).

The `SimpleResult` implementation is for outputting a result set where each row's cells are independent of each other (i.e. the value of one will not influence the value of another). An example of this would be outputting a list of users: the result set is every row of the user table. One the other hand the `ComplexResult` implementation is used for outputting a result set where some columns are dependent on others. For example, the Collaborative Environment keeps track of an arbitrary set of activities, each of which monitor the number of student participants. If one were to query how many participants are enrolled in each activity, that number is, obviously, dependent on the activity and, as such, two queries must occur: first for all activities, then for each activity, query for how many students participate in it.

Sometimes a result set must be generated that does not fit the simple or complex implementations, in which case a more specialized implementation is needed, as is the case with `StudentsServedResult`. In this case, it can handle outputting two result sets at once.

The abstract design allows for a lot of code reuse as well as the flexibility to handle unique circumstances.

Implementation

The current implementation of the Collaborative Environment only allows administrators to access the analytics menu, which can be seen in Figure 3.32. This list provides access to the current analytical tables HTML and CSV formats. Adding support for all of the tables that the Education Department requires be filled out annually is part of the project's future work. It is worth nothing that all of data that will be outputted is year dependent and based on the currently selected academic year. Screenshots can be found in Appendix A.

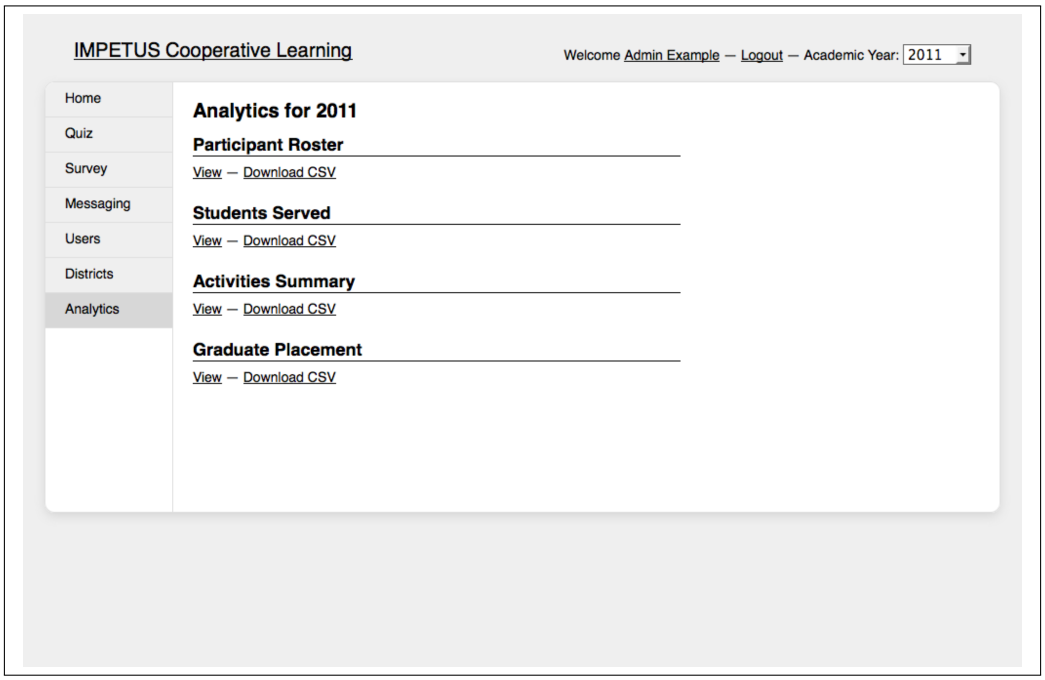


Figure 3.32: The currently implemented analytical tools available to an administrator.

Chapter 4

User Survey

In this chapter, the details and results of a user survey that was conducted are described.

4.1 Study Goals

As stated in §1.3, the Collaborative Environment is being developed using a modified waterfall model. This means that the verification stage can lead back to the requirements or design stage. The idea is that the feedback received while verifying very often leads to uncovering flaws in the design.

There are two motivations behind performing this study:

1. To gather an initial set of qualitative data for future quantification; and
2. To get comments and suggestions regarding the features.

Even though only one usability test was conducted, the answers to its qualitative questions should provide some insight into the effectiveness of the Collaborative Environment.

4.2 Participants

IMPETUS administrators, teachers, and graduate fellows were given the opportunity to participate in this user study. These classes of users were chosen because they would have access to the most breadth of implemented features in the Collaborative Environment. The candidates had no prior hands-on experience with this system and only knew of its development (i.e. there was no familiarity bias). They were asked to participate during an IMPETUS campus event hosted at Clarkson University. Participants were given a unique set of credentials at the start of the survey to use in the demo installation of the Collaborative Environment. This was to ensure that all participants would have access to an identical set of data in the system.

4.3 Procedure

The survey was created using a Google Docs Spreadsheet form and designed to be used as a guide through a series of typical interactions with the Collaborative Environment. In particular, the participants played the role of a teacher and, as the survey instructed them on what tasks to perform (but not *how* to perform them), it also acted as a way for them to leave feedback as they proceeded. To accomplish this, the participant needed the survey open in one browser tab/window and the Collaborative Environment open in a second so that they could switch between them. The survey was split into eight sections primarily focusing on a site feature (Login, Academic Year Switching and Student Data, Editing Users, Adding Users, District Management, Messaging, Surveys, Quizzes) and finished with a generalized set of questions focusing on the site as a whole.

4.4 Responses

Throughout the survey, participants faced four varieties of questions, each of which served a different purpose:

Likert Scale – Agreement

These opinion questions have the participant rate their own responses. In this survey they are used to gather opinions on the clarity, difficulty, and intuitiveness of features.

Likert Scale – Satisfaction

These broad questions allow for a generalization of opinion regarding a large system feature.

Qualitative Multiple Choice

Questions of this type have a correct answer. They are used to ascertain whether or not users are actually able to perform a task rather than just provide an opinion.

Short Answer

Gives the participant the flexibility to say whatever they would like. Scale questions are often accompanied by at least one short answer question to allow exact impression/opinions to be stated.

Due to the agreement and satisfaction question being subjective and qualitative in nature, it would be ideal to perform numerous usability surveys to track the *changes* to responses. These changes would provide quantitative data regarding the redesigns and implementations after each validation stage of the software's development.

4.5 Results

In total, the survey had 18 participants. As described in the preceding sections, the survey was broken up into eight sections, each of which contained a mixture of questions. These questions were designed to provide insight into how some of the Collaborative Environment's target users perceive its utility. The proceeding subsections describe each of the eight sections responses. Table 4.1, Table 4.2, and Table 4.3 provide the accumulated sets of the survey's scale and multiple choice questions and answers.

Login

The first part of the survey involved having the user simply log into the system using the provided credentials. When asked if this process was too complex, 84% disagreed to some extent and an additional 11% had no opinion on the matter. One participant stated that they believe most users will see this as a familiar process while another offered the suggestion to include a way to recover a forgotten password.

Academic Years and Student Data

Participants were given a brief description of the concept behind on-the-fly switching of academic years and what types of data it could be used to compare. They were then asked what grade a student named Sally was in during the year 2010. This required switching from academic year 2011 to 2010 and then viewing Sally's profile. 83% of the participants got this correct. 11% incorrectly chose Sally as being in 12th grade, indicating that the participant did not realize they were not viewing 2011 data to begin with. Additionally, when asked if switching between academic years was easy, 84% were in agreement.

While looking for what grade Sally was in, users were also asked to look at the types of data they could see about a student and asked what they felt should be added, if anything. The most common response was the request to include the number of years a student has been a participant in the the IMPETUS program.

Editing a User

After ensuring that academic year 2011 was the current academic year, participants were told to edit a user by adding Physics to their list of enrolled courses and adding an SAT Math score of 750 to their list of standardized exams. When asked if navigating to the user editor was easy, only 72% agreed. Participant comments informed us that the link to the user editor was not in a spot that necessarily made sense. They would have preferred it to be in the profile viewer rather than only in the list of users. Overall, 89% of the participants expressed satisfaction with this feature

Adding a User

When asked to add a new user to the system, 84% agreed that this task could be easily accomplished. A comment was made suggesting that the navigation required to add users be modified to require fewer clicks. This way, when performing the task in batch, it will reduce the overall time needed to finish. Just as with editing users, 89% were satisfied to some degree with this feature.

District Management

Participants were asked to add the user created in the last step to the district they had access to. Unfortunately, there was a change in how the add user feature worked due to a software bug and resulted in the directions for this section to be incorrect. This undoubtedly led to confusion for the participants as 23% noted. Comments regarding this were the most prevalent that were recorded.

Messaging

The messaging system was explained to the participants who were asked to send a message to a user and view a thread of existing messages. Again, 84% of users felt this message system was intuitive and expressed a degree of satisfaction.

Surveys

Participants were told to explore this feature by creating, taking, and viewing the results of surveys. Explanations were not explicitly given in the instructions as this feature has a built-in help feature. When asked if creating surveys was confusing, 89% disagreed. Most comments mentioned a need for editing surveys once they have been created. This is a current limitation of the survey system. Participants were also asked their opinions on the data presented in the survey results viewer, to which providing graphs for the data was the most common suggestion. About 84% stated satisfaction with the survey system.

Quizzes

Just as with surveys, participants were told to explore the quiz feature by creating, attempting, and viewing the results of a quiz. A help page was also featured that explained the quiz creator in detail. Only 45% of the users felt that the quiz creator was not confusing and 55% expressed satisfaction with it. The most prevalent comment made about this involved not understanding how adding multiple answers to generate a multiple choice problem worked, namely due to poor labeling of the text fields.

Overall

As a conclusion to the survey, the participants were asked their overall satisfaction with the program as it was presented to them and what they would like to see added. 89% stated they were satisfied while the remaining 11% were neutral. The most requested additional features were quiz/survey editing and adding recent news events and pictures.

Question	Disagree strongly	Disagree somewhat	Disagree	Neither agree nor disagree	Agree	Agree somewhat	Agree strongly
Logging in is too complex.	**67% (12)**	11% (2)	6% (1)	11% (2)	6% (1)	-	-
Switching between Academic Years is easy.	11% (2)	-	-	6% (1)	6% (1)	22% (4)	**56% (10)**
Navigating to the User editor is easy.	6% (1)	-	-	22% (4)	6% (1)	22% (4)	**44% (8)**
I intuitively knew how to add a Course and a Standardized Exam score to a User.	6% (1)	-	-	17% (3)	-	**50% (9)**	28% (5)
Students can be added to the system easily.	6% (1)	-	-	11% (2)	11% (2)	22% (4)	**50% (9)**
Students can easily be added to a District's roster.	6% (1)	-	6% (1)	22% (4)	22% (4)	11% (2)	**33% (6)**
I find Districts to be confusing.	**44% (8)**	6% (1)	6% (1)	22% (4)	17% (3)	-	6% (1)
It is easy to add recipients to a message	-	-	6% (1)	11% (2)	17% (3)	11% (2)	**56% (10)**
The messaging system is intuitive.	-	-	6% (1)	17% (3)	6% (1)	22% (4)	**50% (9)**
Creating a Survey is confusing.	**39% (7)**	17% (3)	33% (6)	6% (1)	6% (1)	-	-
Creating a Quiz is confusing.	17% (3)	6% (1)	22% (4)	**28% (5)**	11% (2)	11% (2)	6% (1)

Table 4.1: Likert Scale – Agreement: Questions and Results ($N = 18$)

Question	Grade 9	Grade 10	Grade 11	Grade 12	Unsure
What grade is Sally in during Academic Year 2010?	-	-	**83% (15)**	11% (2)	6% (1)

Table 4.2: Qualitative Multiple Choice: Academic Year Proficiency; Grade 11 is correct. ($N = 18$)

How satisfied are you with ... ?	Extremely dissatisfied	Dissatisfied	Neutral	Satisfied	Extremely satisfied
The live user search feature	6% (1)	6% (1)	22% (4)	28% (5)	**39% (7)**
The types of data that the Survey Results provides you with	-	11% (2)	17% (3)	**56% (10)**	17% (3)
The types of data that the Quiz Results provides you with	-	6% (1)	**50% (9)**	33% (6)	11% (2)
User creation	-	-	11% (2)	33% (6)	**56% (10)**
User editing	-	-	11% (2)	33% (6)	**56% (10)**
District editing	-	6% (1)	22% (4)	22% (4)	**50% (9)**
Messaging	-	-	17% (3)	39% (7)	**44% (8)**
Survey creation	-	-	17% (3)	**56% (10)**	28% (5)
Survey results	-	-	17% (3)	**44% (8)**	39% (7)
Quiz creation	-	11% (2)	**33% (6)**	**33% (6)**	22% (4)
Quiz results	-	6% (1)	28% (5)	**44% (8)**	22% (4)
Overall system	-	-	11% (2)	**61% (11)**	28% (5)

Table 4.3: Likert Scale – Satisfaction: Questions and Results ($N = 18$)

Chapter 5

Conclusions

5.1 Conclusions

The primary contribution of this work is a free and open source web-based Collaborative Environment that ultimately provides a means of satisfying the STEP priorities and requirements by way of aiding in overcoming the issues of operating a geographically isolated STEM outreach program. By basing the Collaborative Environment on state-determined criteria, it can be utilized by any STEP funded institution that has a desire to provide web-based collaborative learning to its participants. Additionally, its open source nature allows for local customization and adoption for state programs outside of New York which have goals similar to that of STEP's. It has been shown that the currently implemented Collaborative Environment's feature set addresses the STEP priorities and requirements. Specifically:

User Management

The storage of student data is facilitated by this system. It allows for personal data, such as gender and ethnicity, and educational data, such as standardized test scores, courses enrolled in, and after-school activities participated in, to be stored on a yearly basis. This data can be used to show improvement in the STEP program priorities: the recruitment/retention rates of male users and underrepresented minorities as well as tracking NYS Math and Science assessment examination scores.

Quiz System

Allows coaches and teaching assistants to create quizzes to measure student performance and act as a supplemental educational tool during student collaborative and self-learning. Students can use these quizzes as a means of brushing up on topics they are struggling with, while teaching assistants can use quizzes as a way to ensure their students are learning. This system assists in the fulfillment of the STEP program requirements stating that there must be formal collaborations between the funded institute and the local school districts in addition to providing activities that will assist students in acquiring the skills and aptitudes necessary to pursue collegiate STEM disciplines.

Learning Pathway

Using a visualization of state math standards in the form a hierarchical tree of dependencies, students can readily identify which topics they are proficient in and which they need assistance with. Each node in this tree represents a state standard and has associated with it both external video resources via the Khan Academy and quizzes created using the Quiz System. Nodes change color based on a student's performance on the quizzes so that a quick look at the tree will identify all the areas that an individual student does and does not need help with. This ability to know what students need help with on a individual and personalized basis via a tool backed by actual state standards will aid in accomplishing the STEP program requirement which states there must exist services to enhance the math and science skills in accordance with the Advanced Regent Diploma.

Survey System

Coaches and teaching assistants can create surveys to be distributed to the Collaborative Environment's user base. This system can be used to gather valuable data and feedback from all of the participating bodies. Additionally, this can be used as a way to identify which students might need early intervention to keep them succeeding. If a user does not fill out a survey, it might be a sign that the student needs some additional attention.

Messaging

Prior to the creation of this tool, there was no central resource for being able to communicate with any of the members of the IMPETUS program (students, coaches, assistants). Since all members will have an account (with an email address associated with it) in the Collaborative Environment, using a private message system which also notifies recipients that they have new mail is ideal for maintaining connections and collaboration.

Since the Collaborative Environment is open source and built using a set of well-defined existing tools, such as Symfony and Google Maps, it is a reasonably sustainable application for future work and maintenance. A future maintainer would have to be familiar with the concepts described in §3.1.1, SQL and database modeling, and object-oriented programming in PHP. Expected annual upkeep entails creating a new academic year, transitioning each district's roster to the newly created academic year, and ensuring that the standards-based learning pathway is accurate.

Additionally, a user survey was performed to gather some initial feedback regarding the Collaborative Environment. Although a lot of helpful comments were received, it is important to remember that the user study experiment was only run once and had a limited number of responses (18). Therefore, the qualitative responses that were received could not be used to concretely assert facts regarding the Collaborative Environment's usability. That being said, in an anecdotal sense, the results of this initial research have been primarily positive with 89% of the participants indicating satisfaction with the system.

5.2 Future Work

There exist several areas of future work for this project:

- The Analytics feature discussed in §3.2.8 was designed to make completing the annual STEP summary report easier by formatting the data collected by the Collaborative Environment into that expected by the report. As of the current implementation, only 4 of the 16 data tables required by the annual report are generated.

- Conduct another usability study to obtain qualitative data regarding the redesigned features based on the first usability study. Once another usability test has been administered, it can be compared to the one presented in this paper to have quantitative data indicating, for instance, an improvement between designs. Future studies must involve IMPETUS tutors, mentors, and students as they were a missed in the initial one.

- Due to the Collaborative Environment's data logging capabilities, it can be used as a powerful research tool. For example, student quiz results for multiple academic years can be analyzed to determine which quiz questions resulted in different levels of student academic achievement.

- The open source nature of the Collaborative Environment allows it to be potentially utilized by other STEP programs. Making these other programs aware of its existence could lead to better software and, consequently, better STEP programs by building a community around the Collaborative Environment. Additionally, reaching out to the Khan Academy could lead to a mutually beneficial exchange of information regarding collaborative and educational software.

Bibliography

[1] The tech awards: Salman khan. Website, 2009. http://thetechawards.thetech.org/the-laureates/stories/1492.

[2] Khan academy. Website, 2011. http://www.khanacademy.org/.

[3] Project 10 to the 100. Website, 10 2011. http://www.google.com/onceuponatime/project10tothe100/.

[4] Christine L. Borgman, Hal Abelson, Lee Dirks, Roberta Johnson, Kenneth R. Koedinger, Marcia C. Linn, Clifford A. Lynch, Diana G. Oblinger, Roy D. Pea, Katie Salen, Marshall S. Smith, Alex Szalay, and NSF Task Force on Cyberlearning. Fostering Learning in the Networked World: The Cyberlearning Opportunity and Challenge. Report, National Science Foundation, 08 2008.

[5] New York State Education Department. District and school performance: Grades 3 - 8 english language arts and mathematics assessments, 2010. http://www.p12.nysed.gov/irs/pressRelease/20100728/home.html.

[6] New York State Education Department. Science and Technology Entry Program :KIAP:NYSED. Website, 2011. http://www.highered.nysed.gov/kiap/step/.

[7] New York State Education Department. Step project 2010-2011 final report form. Website, 2011. http://www.highered.nysed.gov/kiap/step/.

[8] National Science Foundation. Cyberlearning: Transforming Education Cyberlearning, 09 2011. Solicitation 11-587.

[9] J. Glenn. Before it's too late: A report to the nation from the national commission on mathematics and science teaching for the 21st century. Report, National Commission on Mathematics and Science Teaching, 2000.

[10] Murray W. Goldberg, Sasan Salari, and Paul Swoboda. World wide web - course tool: An environment for building www-based courses. *Computer Networks*.

[11] P. Gonzales, T. Williams, L. Jocelyn, S. Roey, D. Kastberg, and S. Brenwald. Highlights from TIMSS 2007: Mathematics and science achievement of U.S. fourth- and eighth-grade students in an international context (NCES 2009-001 revised). National Center for Education Statistics, Institute of Education Sciences, U.S. Department of Education, Washington, D.C., 2008.

[12] Common Core State Standards Initiative. Common core state standards for mathematics. Report.

[13] New York State Education Department. STEP Operations Manual. Website, 2011. `http://www.highered.nysed.gov/kiap/step/`.

[14] Thomas Stober. *Agile software development best practices for large software development projects*. Springer, Heidelberg New York, 2010.

[15] Clarkson University. Northern New York Partnership for Rural Integration of Science and Mathematics, 2010.

[16] John F. Watson. A national primer on K-12 online learning. Report, North Central Regional Educational Laboratory, 2005.

[17] John F. Watson. Keeping pace with K-12 online learning. Report, North American Council for Online Learning, 04 2007.

Appendix A

Analytical Data Screenshots

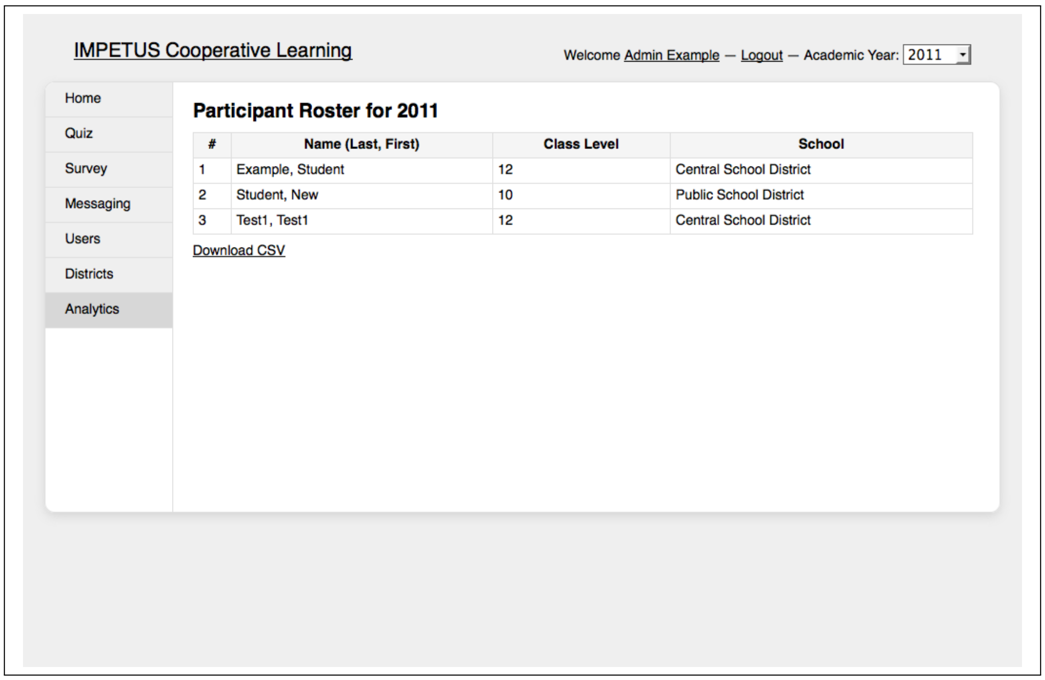
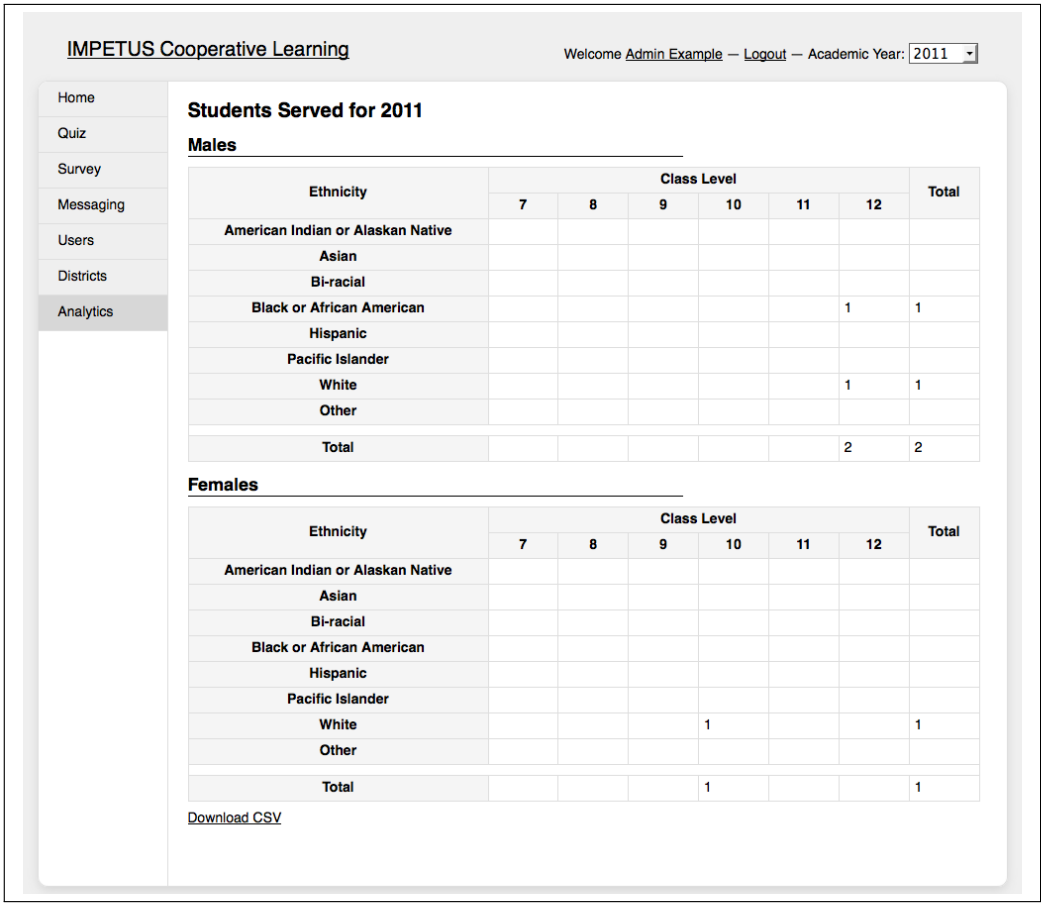


Figure A.1: Sample listing of all STEP student participants.



IMPETUS Cooperative Learning

Welcome Admin Example — Logout — Academic Year: 2011

Home
Quiz
Survey
Messaging
Users
Districts
Analytics

Students Served for 2011

Males

Ethnicity	Class Level						Total
	7	8	9	10	11	12	
American Indian or Alaskan Native							
Asian							
Bi-racial							
Black or African American						1	1
Hispanic							
Pacific Islander							
White						1	1
Other							
Total						2	2

Females

Ethnicity	Class Level						Total
	7	8	9	10	11	12	
American Indian or Alaskan Native							
Asian							
Bi-racial							
Black or African American							
Hispanic							
Pacific Islander							
White				1			1
Other							
Total				1			1

Download CSV

Figure A.2: Sample listing of the number of students of each ethnicity in grades 7-12 grouped by gender.

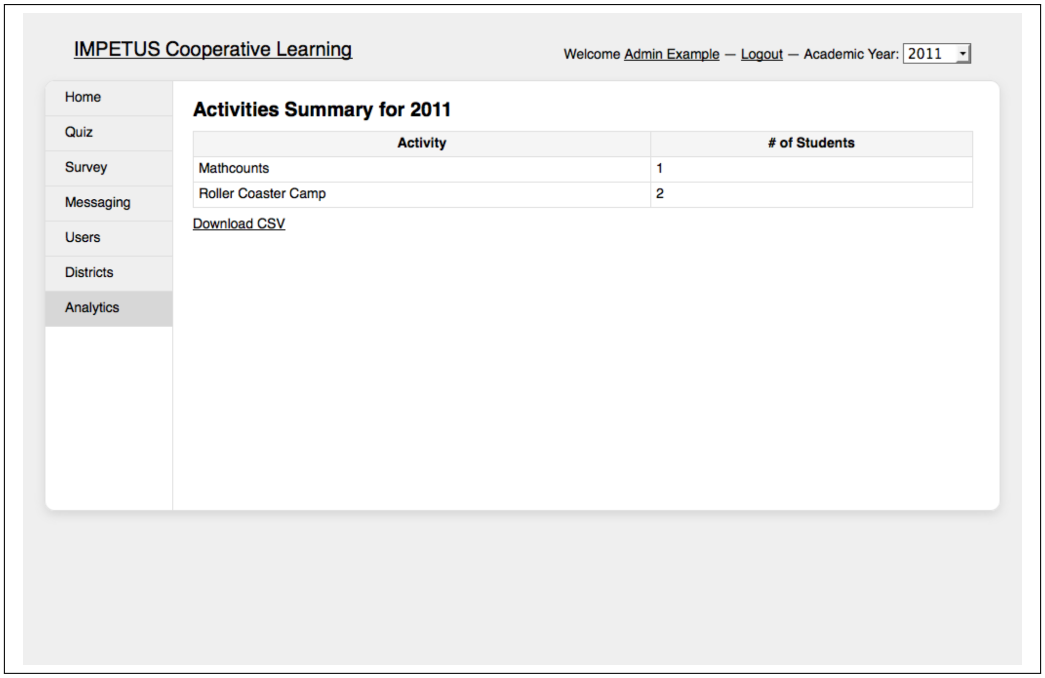


Figure A.3: Sample listing of the number of students participating in each activity.

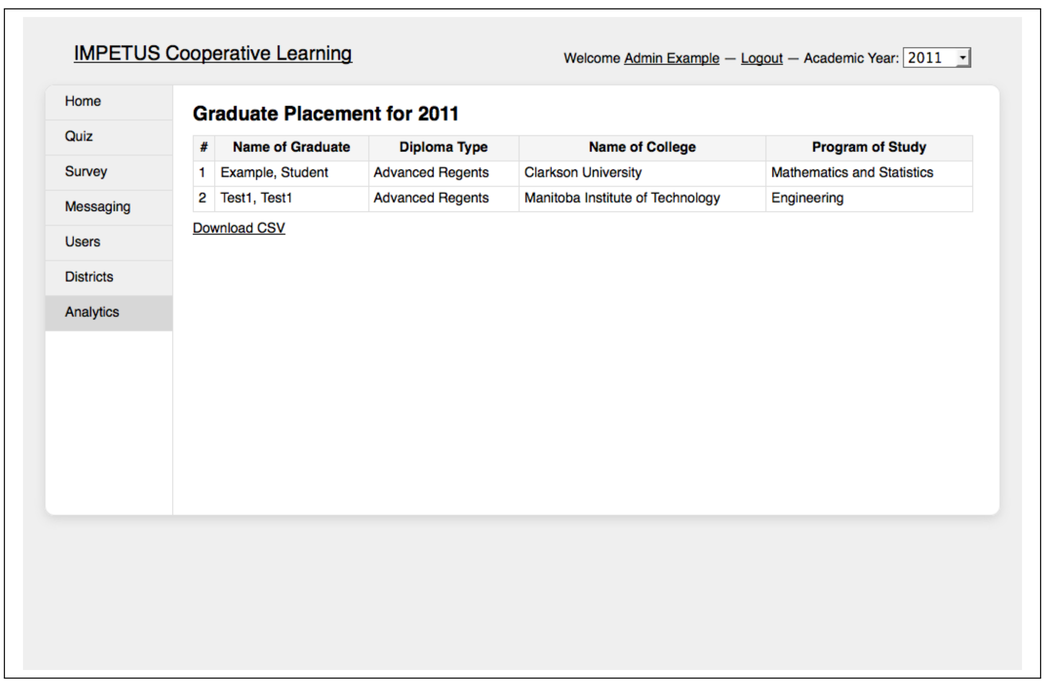


Figure A.4: Sample listing of each graduating student and details of the college they will attend.