Aligning the Course Information System supporting Accreditation with the Student E-portfolio Evaluation System along common Program and Course outcomes to aid continuous improvement

Shashank Balasubramanian, Department of Engineering, Computing, ASU Polytechnic
December, 2011

Abstract— **Course Information System supporting Accreditation (CISA) is a web application for publishing appropriate curriculum artifacts in a manner that lends to accreditation. CISA was developed as a Master's project at the Division of Computing Studies at ASU Polytechnic. Mahara is an open source e-Portfolio and social networking web application used to automate the portfolio review process carried out by the Department of Engineering at Arizona State University [1]. It supports assessment of student generated portfolios demonstrating learning outcomes. It was also developed as part of a Master's project.**
This project integrates Mahara and CISA to provide input to continuous improvement and course information into a single system. The integration ensuring the two applications share common information and updating both applications is synchronized and meets the needs of continuous curriculum improvement.

***Index Terms*—CISA, Mahara, ABET, outcome, assessment, continuous improvement, accreditation, portfolio, course, program**

I. INTRODUCTION

A. Problem Statement

Currently, the Division of Computing Studies at the Arizona State University houses two applications, namely CISA and Mahara, for managing the course accreditation information and handling the student e-portfolio assessment respectively.
CISA is a web application which serves as a repository for viewing and managing all course and program related information for public access. It also includes curriculum information specific to a program and the mapping between course and student outcomes with respect to ABET outcomes and criteria. Thus, it lends well to continuous curriculum improvement.
Mahara is a web application which supports student e-portfolio assessment. It includes groups which are mapped to their respective course or program outcomes. The outcomes can further have sub-outcomes. The students are assigned to their respective groups and all portfolios are submitted against a particular group/outcome. The portfolios are evaluated based on a set of rubrics, which are a set of general criteria to assess a portfolio for an outcome/sub-outcome, in a level based or meets/not meets paradigm. Generally there are 4 levels in level based outcomes which are defined in the system. The portfolio is evaluated by the committee based on these set of rubrics and levels (if any), to provide a final assessment of the student for a portfolio.

However, at the core of both these applications are course level and program level outcomes to which both these applications are tightly integrated. The two applications currently have their own separate databases which house separate copies of outcome information. This causes the systems to be out of sync with respect to course and program outcome information in each of these systems as the CRUD (Create, Read, Update, Delete) operations firing on these two databases are mutually exclusive.

This project converts the database from Postgres to MySQL to allow seamless integration of the CISA system with the Mahara system. This migration involves making schema changes as well as user interface enhancements in both systems. The course content is more effectively organized in CISA by further grouping the course outcomes and activities based on their prefixes. Secondly, new screens need to be added under program outcomes to view and maintain curriculum based information such as Overview, Description, Technical Outcomes, Continuous Improvement and Major Map. Lastly, the ABET outcomes need to be redesigned to support general criteria and student outcomes for each program in the department.
On the Mahara front, the aim of this project is to integrate it with CISA in such a way that student assessments play a critical role in defining a continuous improvement plan for the program. Student outcomes for a program are considered as an integral part of the curriculum and CISA's intent is to provide a single point of reference for multiple facets defining a program. For this reason, the outcomes against which the assessments are made need to be moved to CISA so that Mahara can share the database and align its assessments based on the outcomes defined in CISA. This will synchronize the two systems and provide for a smoother program improvement plan, and avoiding replicated data.

B. Background

The Division of Computing Studies at ASU uses two custom homegrown applications, namely CISA (Course Information System supporting accreditation) and Mahara (E-portfolio system) to handle two specific functions within the

department. CISA is responsible for storing and displaying course and program related information such as outcomes, objectives and activities and their alignment with ABET specified criteria and outcomes for program accreditation. Mahara, on the other hand is an e-portfolio system which is aimed at providing assessment of student achievement of outcome based on portfolios submitted by students. The portfolios are strictly aligned with outcomes which could be at the course or program level.
The two applications are very tightly integrated with outcomes which make them an excellent candidate for merging so as to extend CISA's functionality to include a continuous improvement plan for a particular program taking into consideration even the student assessment information from Mahara.

C. *Objective*

There are two major objectives of this project:

- The first is to ensure that the information available on the CISA application is better organized and comprehensive. This includes creation of additional screens on the public as well as secured site to facilitate better dispersal and maintenance of information.
- The second is to ensure that the course and program outcome information in the CISA and Mahara applications are synchronized ensuring that the quality of information that is accessed in both these applications is current and updated.

D. *Scope*

The scope of this project spans across two applications and includes the following changes:

- Converting the CISA database from PostgreSQL to MySQL to make sure the Mahara application can retrieve information from the CISA database without significant rework on the Mahara application code front.
- Organizing the course related information more effectively by introducing another filter criteria based on the course prefixes
- Adding additional screens to display and manage information for a curriculum with respect to a particular program
- Adding screens for supporting separate ABET criteria for the Computing and Engineering programs
- Moving all the current outcome information into the CISA database which would also be used by Mahara to fetch outcome information
- Displaying and manipulating outcome information in the Mahara application by connecting to the CISA database

E. *Project Assumptions*

Following are the assumptions considered while working on this project:

- The applications are used by the computing facility in the Department of Engineering at the Arizona State University [2]
- Converting the CISA database to MySQL is a better option as opposed to changing the Mahara code base to connect to a PostgreSQL instance. Mahara being a significantly larger application, the effort required to migrate the CISA codebase to MySQL would be lesser than converting the Mahara codebase to adapt with PostgreSQL.
- There is no provision to display CISA/ABET specific information in Mahara or assessment specific information in CISA. They are two distinct separate applications

F. *Technologies and Software*

Technologies involved:
Microsoft ASP.NET 2.0 Framework
ADO.NET, ADODB
Languages: C#, PHP
Database: MySQL

Tools involved:
Apache ANT (1.8.1), Mono (2.6.7)

Administration and deployment:
Servers: apache2, webdev.webserver, mod_mono server
Operating Systems: LINUX, Windows, MacOSX

II. LITERATURE REVIEW

A. *Technologies*

ASP.NET: It is a development platform and framework which was developed by Microsoft Corporation and used to build dynamic web sites, web applications and web services [3]. As it is built on the Common Language runtime, it allows developers to write code in any one of the supported .NET languages such as C# and Visual Basic [3].
The .NET architecture follows an MVC (model-view-controller) paradigm in which the functionality is delineated into three separate aspects.
The view is composed of .aspx (active server page extension) pages also called web forms which are ultimately rendered on the browser. It may also include client side validation on pages which involve user input. In addition, the view component may include .ascx (web controls) which act as fragments which could be included and reused in multiple .aspx files.
The controller includes the codebehind classes which handle user requests. Based on these requests, the controller interacts with the model layer, fetches content and returns the results back to the view layer which is then rendered on the browser as a .aspx page.
The model houses the business logic of the application which may include access to one or more databases as well as the classes representing type and behavior of the model objects.

C#: is one of the supported .NET languages and is intended to be a simple, modern, general-purpose, object-oriented

programming language [4]. The CISA application has been developed on the .NET platform using the C# language. C# is used extensively in the controller and model sections of the MVC architecture of CAPISA. In other words the codebehind classes which provide data to the .aspx are all C#. The code-behind file is compiled into a Dynamic Link Library (.dll) and both the .aspx page and its associated .dll are made available to the web server containing the ASP.NET engine [2].

PHP: It stands for PHP Hypertext Preprocessor) and is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML [5]. The E-portfolio application has been developed using this technology. The application is based on the open source Mahara framework. PHP is an interpreted language and all the code is executed on the server side in the web server. It is also a platform independent language and compatible with most of the modern servers.

MySQL: It is an open source database server which is ideal for small and large scale applications [6]. Both CISA and E-portfolio applications have used a MYSQL database instance as its back-end to store data. Mysql is freely available, compiles on a number of platforms and supports standard SQL commands [6].

ADO.NET: ADO.NET (ActiveX Data Object for .NET) is a set of computer software components that programmers can use to access data and data services [7]. ADO.NET provides consistent access to data sources such as MySQL Server [8]. ADO.NET includes .NET Framework data providers for connecting to a database, executing commands, and retrieving results. Those results are either processed directly, or placed in an ADO.NET DataSet object in order to be exposed to the user in an ad-hoc manner, combined with data from multiple sources, or used remotely between tiers [8].

ADOdb: ADOdb is a database abstraction library for PHP [9]. PHP's database access functions are not standardised. This creates a need for a database class library to hide the differences between the different database API's (encapsulate the differences) so we can easily switch databases [10]. This role is played by ADOdb. Mahara employs ADOdb to perform its data access functions.

B. *Related Work*

There are various other tools and methodologies which are used to map course and outcomes based assessments to the program accreditation process [1]. Some of them may be summarized below.

The Web Course Evaluation System (WCES) developed by Columbia University's Foundation School of Engineering and Applied Sciences allows students to complete course surveys prepared by faculty and generates reports for curricula improvement [11].

The University of Bridgeport, School of Engineering has developed an electronic web based assessment system that provides an assessment presentation website and courses material website allowing the evaluator to browse from the objectives and outcomes to specific course materials along with course versus outcomes matrix and individual course grids [12].

EnableOA, developed by Douglas Walcerz, York College of Pennsylvania is a web-based software driven outcome assessment process designed to meet the assessment needs of the institutions [13]. Based on EnableOA, a web based assessment commercial software product called TrueOutcomesTM has been developed for different universities and their departments. This product stores data about courses and programs from university departments and generates a matrix of course versus outcomes for each academic major for accreditation [14].

CourseWork is Stanford University's learning management system. Using CourseWork, instructors and TAs can set up a course web site that displays announcements, course materials, a syllabus, a schedule, online assignments and quizzes, a discussion forum for students, and a gradebook. CourseWork is designed both for faculty with little web experience, who can use CourseWork to develop their web site quickly, and for expert users, who can use CourseWork to organize complex, web-based materials and link them to web communication tools [15].

MIT has implemented a course management system, called Stellar. It is a platform for learning and course management serving the MIT community. It includes support for course curriculum, materials as well as a grade book. Users have to create accounts to have access to course specific information [16].

ASU has evaluated the TrueOutcomesTM to help in its accreditation process. In the evaluation process, integration of the TrueOutcomesTM software with ASU's Course Management Software (CMS) BlackBoard has been performed by the students of DCST [17] [18]. However, it has been concluded that the integration still requires possible extensions like mappings between course outcomes and program outcomes and course activities and course outcomes [17].

Similarly, there are also other tools and software available for managing student portfolios and assessments based on the outcomes of the program. They can be summarized as follows.

Desire2Learn: Desire2Learn develops an e-Learning enterprise suite that serves as the learning management system for schools, colleges, universities, virtual schools, governments, associations, and organizations. Clients include University of Guelph, Calgary Board of Education, Red River College of Applied Arts and many more [19]. It provides features such as course conversion, learning object design, multimedia development, portal design and course development [20].

Digication: Digication is the leading e-Portfolio provider for K-12 and Higher Education schools across the U.S. Our e-

Portfolio Editions are tailored to meet the needs of individual teachers and students, classrooms, departments, and campuses [21]. It comes in various editions such as Personal Edition, Classroom Edition, Campus Edition and Admissions Edition. It also provides an Assessment Management System which is a web-based tool for tracking, comparing, and reporting on student progress and performance [21].

Foliotek: Foliotek is an electronic portfolio web application that supports Standards-Based Assessment, Career Portfolios, College and University Accreditation, and Faculty Tenure Review & Promotion. The system provides tools for both formative assessment and summative assessment [22]. Foliotek provides students with an assessment portfolio, a scrapbook portfolio (file repository), and a presentation portfolio and provides administrators with the ability to create their own portfolio structures, evaluations, scoring guides, forms, surveys, and standards. Administrators also have access to a wide-range of reports [22].

MAPS: MAPS is a proprietary web-based Assessment or EPortfolio service. The software is entirely web services based and provides an array of evidence recording tools which are structured around a blog or learner diary. Users can belong to multiple institutions and individual portfolios are owned entirely by the individual. The individual then chooses which institution or institutions to share their portfolios with. Institutions get an aggregated view of portfolios which are members of the institution [23].

Pass-port: PASS-PORT is a web-based electronic portfolio assessment and course management system operated by Innovative Learning Assessment Technologies, LLC. PASS-PORT is designed to help colleges and universities meet accreditation standards for their academic programs and student services. It assists administrators in collecting, aggregating, and reporting performance based assessment data aligned to accreditation standards from a number of sources such as file artifacts, professional development experiences and field experiences [24].

PebblePAD: PebblePAD is a proprietary web-based Personal Learning Environment or EPortfolio system. PebblePad uses moving 'pads' which guide the user through the process of creation of a variety of record types. These include records like action plans, abilities, achievements, experiences and thoughts. Records may be presented in webfolio (which are a type of webpage) and journal entries as part of a blog. Videos and Photos can be imported from social networking sites like You Tube, Flickr. Assets can be tagged, which allow them to be re-used in different parts of the software, e.g. the same asset can be part of a Webfolio and a Journal. Users can share the records and assign differential permissions to view, comment, copy, collaborate permissions. These can be granted to both PebblePad and external users. To support work based learning, field trip type activities and the collection of evidence via camera phones the system has mobile versions for Windows Mobile devices, an iPhone app and a lightweight web version for use on Internet enabled mobile phones [25].

RCampus: RCampus is a comprehensive Education Management System and a collaborative learning environment. At RCampus, you can do all your school-related work from building personal and group websites to managing your courses, eportfolios and academic communities. It supports several features such as learning and course management, ePortfolios, rubrics, student, faculty and group websites, tutor connection and book exchange [26].

III. SOFTWARE DESIGN AND ARCHITECTURE

A. System Design:

CISA

Database: A MySQL database is used which stores all the course, program and abet outcome related information.

Public View: CISA offers a public view which can be accessed by any user without logging in to view the various course and program outcomes along with their alignment with the ABET outcomes for different programs. The public view can be depicted diagrammatically in Fig 1.

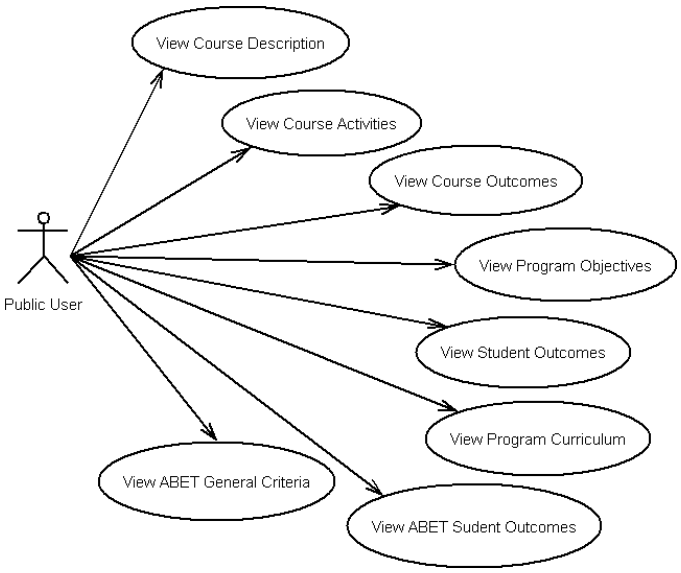


Fig. 1. Public view of CISA use case for the general user

Secured View: CISA offers a secured view for administrators to update the contents of the CISA application. These set of screens are secured and require valid user credentials to login and make the modifications. The secured view can be depicted diagrammatically in Fig 2.

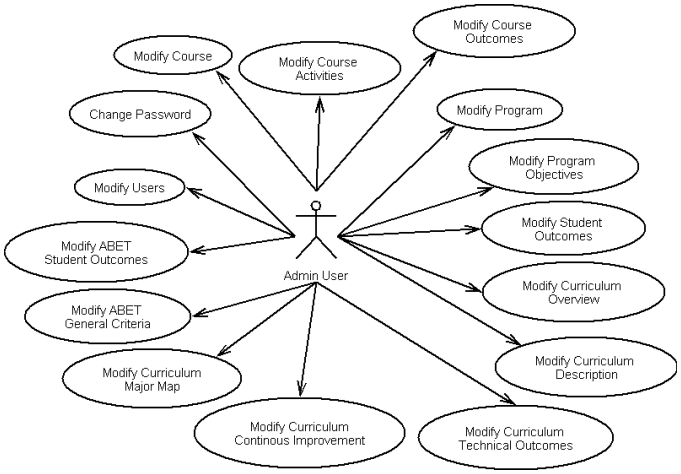


Fig. 2. Secured view of CISA use case for the admin user

Security: Security is achieved by adopting two strategies. Firstly, unauthorized usage is prevented by employing Application Security which forces users to login to access non-public pages. Secondly, the passwords stored in the database are stored as a hashed value as opposed to plain text. This would prevent theft of passwords at a database level and called Data Security.
Authorized access: CISA currently identifies the administrator and faculty roles. Therefore as part of this requirement, access to non-public pages must only be provided to users who are administrators.

Mahara
Database: A MySQL database is used which stores information such as user information, portfolio, group and assessment related information.
Admin view: The administrator is the power user of the application and has access to all parts of the system including site administration screens. The view for admin can be depicted diagrammatically in Fig 3.

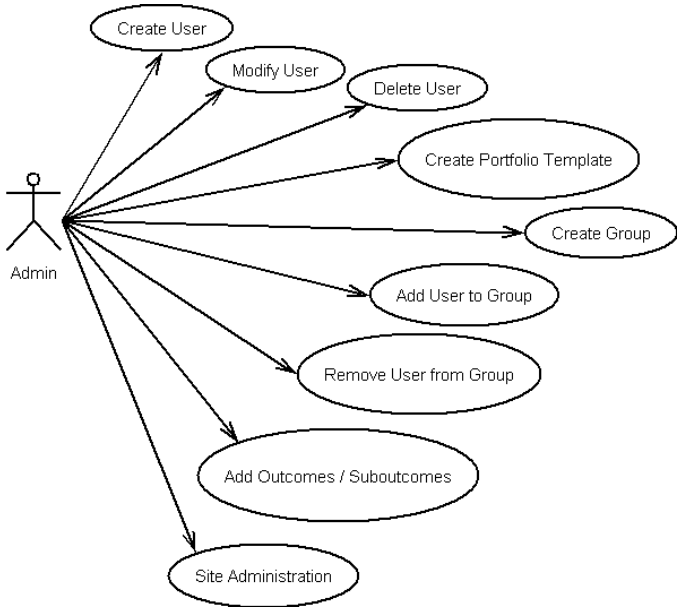


Fig. 3. View for admin user for Mahara

Chair view: The chair is the head of the committee and has access to assessment related information along with the other portfolio information. In addition to this, the chair also has access to other committee member's evaluations and an ability to approve or reject and request re-evaluation from any of the committee members. The view for chair can be depicted diagrammatically in Fig 4.

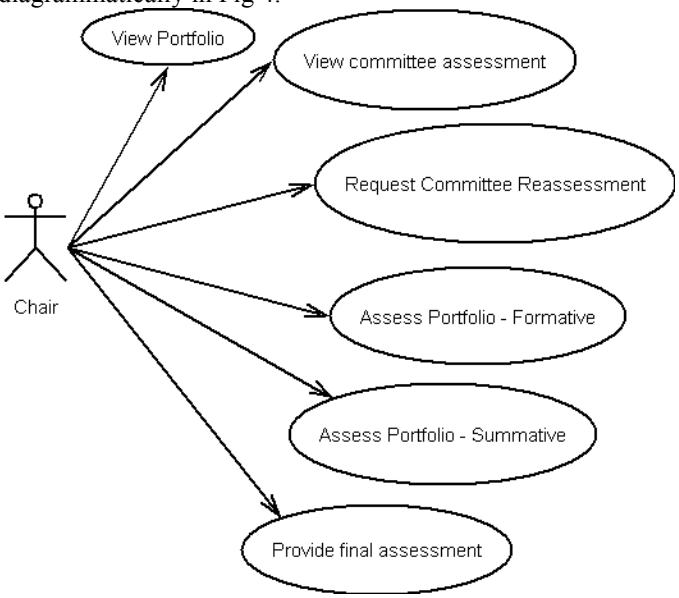


Fig. 4. View for chair user for Mahara

Faculty view: They include the other committee members of a group. They too have access to the assessment related screens but in this case it is restricted to their own assessments. They submit their evaluations which are then reviewed by the chair. The view faculty can be depicted diagrammatically in Fig 5.

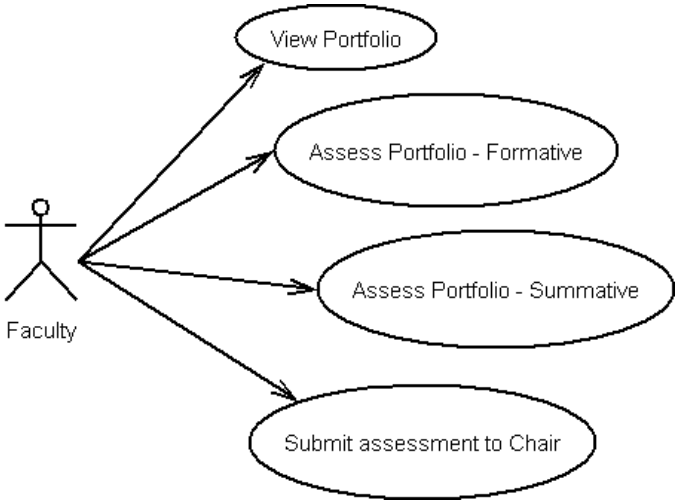


Fig. 5. View for faculty user for Mahara

Student view: The student is responsible for creating the portfolios and submitting them for formative and summative feedback. They have access only to the portfolio specific screens and once a portfolio has been evaluated, the outcome

results can be viewed. The view for faculty can be depicted diagrammatically in Fig 6.

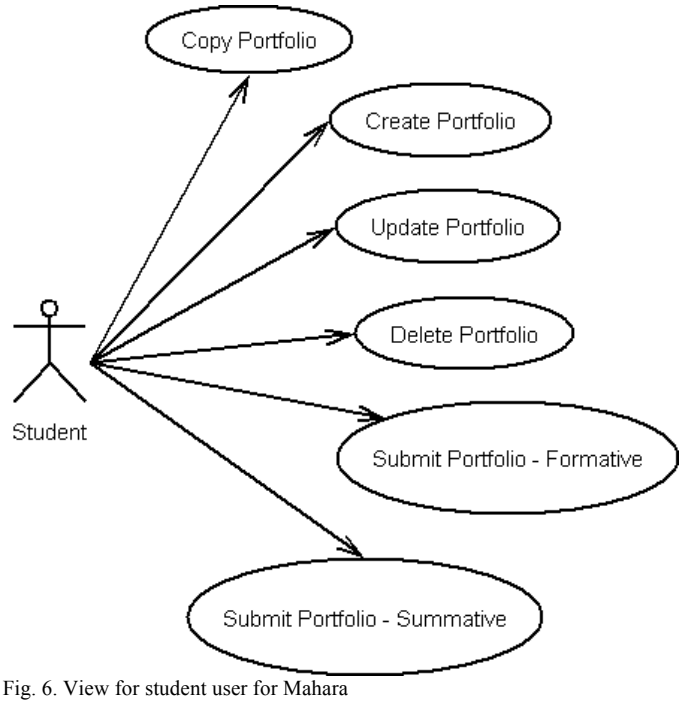


Fig. 6. View for student user for Mahara

Security: As there is no public access, all users are mandatorily required to login to the application. There are two types of users. All ASU users login to the application using their ASURITE whereas all non-ASU users have accounts setup in the E-portfolio database which stores encrypted passwords.
Authorized access: As the e-portfolio system employs several user access levels, it is imperative that each category of user only has access as per the roles they have defined in the system. This is achieved through access level information stored in the database.

B. Database Design

From a database perspective, the two applications, namely CISA and Mahara, have been synchronized such that all the outcome, level and rubric data is now housed and maintained in the CISA instance and the Mahara application refers to CISA tables to display its respective content. The changes are described below.

CISA:The CISA system uses a Postgres database to handle and store its data. Therefore, the database needs to be migrated to a MySQL instance to allow easier integration with Mahara for synchronization purposes. As the Postgres and MySQL database syntax is different, the dump file generated for CISA in Postgres is manually redesigned to support it for MySQL. This involves changing the CRUD queries as well as removing some keywords specific only to PostGres. As a result, once the migration is done, the NpgSQL adapter being used by CISA is no longer required. It is replaced by a MySQL adapter for all purposes.

The system uses a MySQL database instance and has a total of 23 tables out of which 15 are pre existing. However they have been migrated to MysQL. The new tables and mappings are added to synchronize the course and program outcomes with Mahara and can be summarized as follows:

- allcourseoutcomenames_view: It contains the unique course names from both CISA and Mahara
- allcourseoutcomes_view: It contains unique course information from both CISA and Mahara such as coursed, courseoutcomeno and courseoutcomename
- allprogramoutcomes: It contains the unique program names from both CISA and Mahara
- courseoutcomes_mahara: It contains course outcome information which have been fetched from Mahara for synchronization
- courseoutcome_levels: It contains course outcome level information (initially fetched from Mahara) which would be used henceforth for maintaining level information in Mahara
- courserubrics_mahara: It contains course outcome rubric information (initially fetched from Mahara) which would be used henceforth for maintaining rubric information in Mahara
- programoutcome_levels: It contains the levels for program outcomes which are going to be used by both CISA as well as Mahara

Mahara: The system uses a MySQL database instance and has a total of 128 tables. Some of the tables are obsolete now and not in use anymore as the schemas were merged and placed into the CISA database. The tables which are used to synchronize outcome data can be summarized as follows:

- group : It contains the group related information to which the students and committee belong. It is mapped to a single outcome. After moving the outcome data to CISA, the mapping of the column ‘outcome’ is changed to reflect the new outcome ids
- outcomes: This table stores course and program outcome data for Mahara. After synchronization, this table has become obsolete
- outcome_levels: This table stores course and program outcome level data for Mahara. After synchronization, this table has become obsolete
- rubrics: This table stores course and program outcome rubric data for Mahara. After synchronization, this table has become obsolete
- view: It contains student portfolio information. The view is tightly coupled with the group and outcome associated with that group. After moving the outcome data to CISA, the mapping of the column ‘submittedoutcome’ are changed to reflect the new outcome ids

C. Software Architecture

The CISA and Mahara systems both have a 3-tier architecture namely a Presentation Tier, a Logical Tier and the Data Tier. This is in line with following best practices for developing

web applications namely high cohesion, loose coupling and separation of concerns.

CISA system architecture:
Presentation Tier: This includes the ASP.NET web forms which fetch data from the logical tier and displays the results to the user. They have no business logic and their primary task is to deliver results to the user in a presentable manner. Therefore, they contain elements which are related to the view component of the application. They may also include some javascript to carry out basic validation but most of the validation in CISA is handled in the logical tier.
Logical Tier: This is the tier which is responsible for coordinating user requests and server responses. When a user makes a request through the web form, the request is then routed to the appropriate code-behind component (Business Logic Layer) which performs the task calling the correct model class (Data Access Layer). The model classes have access to the data tier through ADO.NET which is used to fetch related information from the data store. This information is returned to the code-behind class which in turn forwards it to the web forms at the presentation tier to finally display the server response to the user.
Data Tier: This is the tier which stores all the information for the application. In the case of CISA, it is a MySQL database which has 23 tables. All information requests are made to this single database instance.

Mahara system architecture:
Presentation Tier: This includes the PHP template files which are used to display results fetched from the logical tier to the user in a presentable manner. Like CISA, they also contain elements which are related to the view aspects of the system.
Logical Tier: This is the tier which is responsible for coordinating user requests and server responses. When a user makes a request through the presentation tier, the request is then routed to the appropriate PHP code-behind component which performs the task of calling the functions defined in the common PHP library scripts. These library scripts connect to the Mahara database using ADODB functions and fetch related information from the datastore. This information is returned to the PHP code-behind file which in turn forwards it to the template files at the presentation tier to finally display the server response to the user.
Data Tier: This is the tier which stores all the information for the application. In the case of Mahara as well, it is a MySQL database which has 128 tables. The Mahara application interacts with two databases. The first one is the Mahara instance which is used to fetch non-outcome related information such as user, group, portfolio or assessment related information. The second is the CISA database which is used to fetch course and program outcome information along with their respective rubric and level information.

The architectures of CISA and Mahara can be represented diagrammatically in Fig 7.

IV. METHODOLOGY

A. System Analysis for CISA:

Public view of CISA: This view does not require any authentication and the user can view course, program or ABET related information as follows:

- Course information: The user can view course related information such as course description, course outcome and course activity information by navigating the course menus. The course navigation menus are further optimized for access by categorizing the courses based on program and division.
- Program Information: The user can view program related information such as program objectives, student outcomes and curriculum specific to that program. The curriculum is further subdivided into 5 categories namely Overview, Description, Technical Outcomes, Continuous Improvement and Major Map.
- ABET Information: The user can view ABET outcome related information as well. The outcome information is categorized based on the programs and it has two subcategories namely General Criteria and Student Outcomes.

Secured View of CISA: This view is only applicable for registered users and requires authentication before access. Its primary function is to manage course, program and ABET outcome related information.

- Login: As this section has restricted access, it is mandatory for users to login before updating information. The user can login by clicking the 'CISA Login' link at the top of the page to be taken to the login screen. The user can then enter the login and password to get access to the secured version of the system. If the user cannot be authenticated, an appropriate message is displayed. Once logged in, the user can proceed with any one of the following tasks.
- Course Management: Under 'Course Management', the user can maintain course information.
    - The 'Modify Course' link is used to maintain courses
    - The 'Modify Course Outcomes' link is used to maintain course outcomes
    - The 'Modify Course Activities' link is used to maintain course activities
- Program Management: Under 'Program Management', the user can maintain program information.
    - The 'Modify Program link is used to maintain programs
    - The 'Modify Program Objectives' link is used to maintain program objectives
    - The 'Modify Student Outcomes' link is used to maintain student outcomes

- Curriculum Management: Is used to maintain curriculum information such as Overview, Description, Technical Outcomes, Continuous Improvement and Major map specific to a particular program.

- ABET Outcomes Management: Under 'Abet Outcomes Management', the user can maintain ABET outcomes information.
  - The 'Modify General Criteria' link is used to maintain the general criteria
  - The 'Modify ABET Student Criteria' link is used to maintain the student outcomes
- User Management: Under 'User Management', the user information can be maintained.
  - The 'Modify Users' link is used to maintain system users
  - The 'Change Password' link is used to maintain user passwords
- Logout: Once all changes have been done, the user can logoff from the system

B. *System Analysis for Mahara:*

Mahara is a private application in the sense that all users have to mandatorily login before accessing information. Mahara supports four classes of users namely Admin, Student, Faculty and Chair. Each of these users has different views and functionality which can be summarized as given below.

View for Admin: The admin is the super user of the application and is responsible for setup and content management.

- Outcome Management: The admin user can add, update and delete course and program outcome information.
- Group Management: The admin user can create new groups, map them to outcomes and add users to the group. This would complete the setup process for a group with respect to assessment
- User Management: The admin user has the ability to maintain user information such as user profile, username and password

View for Student: The student user has the ability to create portfolios for assessment. Once the portfolios are submitted, the assessment process starts. Following are the tasks that a student user can perform.

- Copy portfolio: The user can copy an existing template portfolio to create a new portfolio
- Create portfolio: The user can create a brand new portfolio

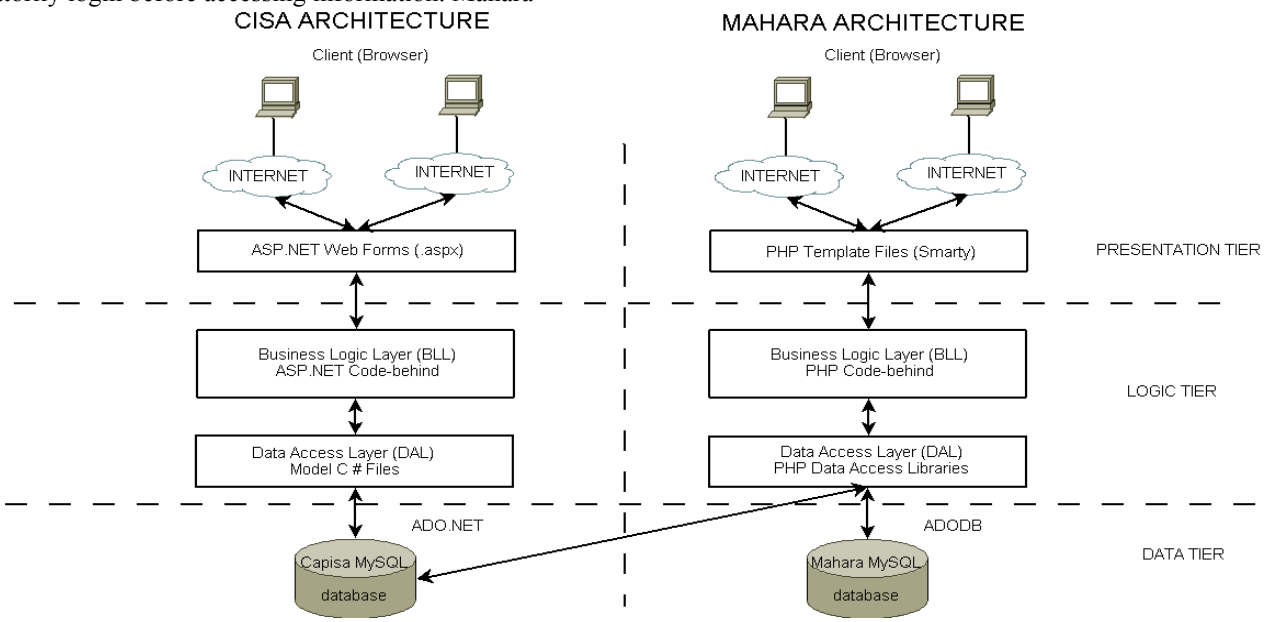


Fig 7. CISA – Mahara Architecture

- Submit portfolio for formative assessment: Once a portfolio is evaluated, the student can submit a portfolio for formative assessment. This process can be done any number of times before a summative assessment submission. The aim of this step is to enable the faculty to give informal feedback about the portfolio
- Submit portfolio for summative assessment: This is an irreversible step where the student submits a portfolio for evaluation. Once submitted, the assessment phase begins where the committee members evaluate the portfolio and publish the results

- View results: Once the results have been published, the student user can view them.
- Maintain profile: The user can maintain his/her profile and personal information

View for Faculty: The faculty is a member on the committee who is not the chair. The faculty is responsible for evaluating the portfolios, making an assessment and submitting the assessment to the chair for evaluation. Following are the tasks that a faculty user can perform.

- View portfolio: Open the portfolio and view its contents and artifacts
- Provide formative assessment: Based on the portfolio artifacts, the faculty may post an informal feedback of the portfolio back to the student
- Submit summative assessment: Once the faculty has viewed the portfolio, he/she would then proceed to evaluate the portfolio based on the rubrics. Once this assessment has been done, the faculty may submit it for review by the chair.
- Maintain profile: The user can maintain his/her profile and personal information

View for Chair: The chair is the head of the committee. The chair is responsible for evaluating the portfolios, making an assessment, reviewing the other committee members' assessment and making a final assessment. Once the final assessment has been made, the results are published for the student to view. Following are the tasks that a chair user can perform.

- View portfolio: Open the portfolio and view its contents and artifacts
- Provide formative assessment: Based on the portfolio artifacts, the chair may post an informal feedback of the portfolio back to the student
- Submit summative assessment: Once the chair has viewed the portfolio, he/she would then proceed to evaluate the portfolio based on the rubrics.
- Evaluate committee assessment: Once the chair has submitted his/her assessment, he/she could then evaluate the other assessments submitted by the committee members. If the chair feels that a certain committee member needs to reassess, he/she requests a reassessment
- Request reassessment: If the chair feels a committee member needs to reassess, a reassessment request is raised with appropriate comments on the justification for this request
- Make final assessments: Once all assessments have been reviewed, the chair makes a final decision on the assessment of the student for that portfolio
- Publish assessment: After final assessment, the results are published which can be viewed by the student

Maintain profile: The user can maintain his/her profile and personal information

C. Security

Security:
Since both applications involve non-public views, security is an important issue to be addressed. Let us take a look at the way security is implemented in both systems.

Security in CISA: CISA provides a secured view for administrators. It does so by validating the user against his/her username and password. In CISA, forms authentication is triggered by a configuration file called *Web.config* placed in the application deploy directory.
The usernames and passwords are stored securely in the database and not available to public. As a further precautionary measure, the passwords are encrypted and stored in the database to prevent stealing passwords by hacking the database. The SHA1 hashing algorithm is used to encrypt the passwords.
If the user forgets his/her password, the application generates a random password and updates the user entry. It then sends a mail to the user providing information about the password change.

Security in Mahara: As Mahara is a private application with no public views, security is a very critical aspect in its architecture. ASU users are authenticated through their ASURITE ids and passwords. Security in Mahara is a two step process. The first step is a pseudo authorization step performed on the Mahara side which ensures that the user's profile exists in the system. The second step is where the actual authorization takes place where a request is sent to the ASU Kerberos server to validate the user against his password in the ASU database. If the authentication was successful, the ASU server returns a token indicating success and this means the user has successfully managed to login to the application. Then, based on the user role, the welcome screen is displayed with the appropriate permissions.
Parameterized Queries: Both applications employ good practices which prevent a malicious user from entering queries into input fields which might cause unintended effects to the database. It does so by using parameterized queries which treats every input as a literal and is never appended to the SQL code [14] and thereby preventing harmful effects.

D. Deployment

The CISA and Mahara applications are deployed on a Linux (SUSE) server. Both run on an Apache 2 server supported by a MySQL database as a back-end. In addition to this, CISA has an ASP.NET engine running on the Mono platform.
CISA uses an ANT script to build its code and deploy the application into Apache. The CISA web application is deployed in Apache by creating a virtual directory in Apache and pointing the virtual directory to the CISA's *deploy* directory [2].

E. Testing of CISA and Mahara:

The CISA and Mahara applications are tested on the Windows, Linux and MacOSX platforms as well as tested for

cross browser compatibility. All screens in CISA and Mahara are individually tested and verified.
Since the two applications are synchronized, tests were also conducted to ensure that the data manipulated and read was correct which will be detailed in the validation section
For Mahara, in addition to copying over the code into Apache's htdocs directory, a data folder is created which houses the artifacts of the application. In addition to this, the verify utility provided by ASU is installed which handles communication with the ASU Kerberos server. Also, the services file at etc/services and the hosts file at etc/hosts was updated to register the port at which ASU communicates with the tool to achieve single sign on.

F. *Validation of CISA and Mahara:*

Since the two systems are integrated with respect to outcomes, changes made in one application must reflect correctly on the other as well. Following are the scenarios which were validated.
Courses: The functions of adding, updating and deleting a course in one application was tested and verified correctly on the other.
Course Outcomes: The functions of adding, updating and deleting a course outcome or course sub outcome in one application is tested and verified correctly on the other.
Program: The functions of adding, updating and deleting a program in one application is tested and verified correctly on the other.
Program Outcomes: The functions of adding, updating and deleting a program outcome in one application is tested and verified correctly on the other.

Moving outcome rubrics and levels to CISA from Mahara: Since the outcome rubric and level information, which were initially housed in the Mahara database, is now moved to CISA, tests are conducted to ensure that the Mahara application functions properly while manipulating and reading outcome rubric and level information.

G. *Installation*

To setup and run the CISA and Mahara systems on a Unix (Suse), we need to perform the following steps:

Install Apache2: Apache2 is an open source web server. You need to download and install the distribution from http://httpd.apache.org/download.cgi
Install Mono: Mono is a software platform designed to allow developers to easily create cross platform applications and is an open source implementation of Microsoft's .NET Framework [27]. You can download and install mono and mod-mono from http://download.mono-project.com/sources/
Install PHP: It is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML [5]. You can download and install it from http://php.net/downloads.php
Install MySQL: It is a very popular open source database server which is ideal for small and large scale applications [6]. You can download and install it from http://dev.mysql.com/downloads/mysql/
Setup CISA source: Copy the CISA source to any local directory on your machine. Thereafter, create a virtual directory and setup an alias in the httpd.conf file of apache2. Details are provided in the appendix
Setup Mahara Source: Copy the Mahara source into the htdocs directory of apache2. Copy the maharadata directory into a local folder. Specify the location of the maharadata directory in the config.php file of mahara (details provided in appendix).
Setup CISA and Mahara tables: Create two databases namely capisa and mahara in MySQL. Import the sql files into these databases to populate them with data. Details on this process are provided in the readme files attached in the appendix

Additional information for setting up the systems is provided in the appendix

V. SUMMARY, CONCLUSION AND FUTURE WORK

This project involves enhancing the existing functionalities of CISA and integrating it with the Mahara application to facilitate an integrated approach to continuous improvement and curriculum information management. For CISA, it requires converting the database from PostGres to MySQL and also adding several public and admin screens for curriculum management for programs. Also, the ABET section of CISA are redesigned to better support the format of ABET general criteria and student outcomes for different programs. The information layout hierarchy is also changed to enhance information retrieval.
Integrating Mahara and CISA involves several schema changes to existing tables of both Mahara and CISA as well as addition of new tables to move outcome related information from Mahara to CISA.
On the Mahara side, a separate database connection is required to access the CISA database instance and the application code and workflow is changed to route the application requests through CISA. This involves changing as well as addition of base helper methods for connecting and retrieving information.
A considerable amount of research and analysis is spent researching and analyzing the two systems to devise a seamless integration of the two systems with minimum overhead.

Future work:

- Mahara currently support identifying the strengths and weaknesses of students relative to program outcomes. There does not exist any traceability to program curricular changes that are made in response to these evaluations. The systems could provide that traceability and store it in the database for future reference.
- Support for student providing input to curriculum/course change can be included. That is, the student provides input to course changes that would better aid achievement of course outcomes.

- Neither application has support for pedagogy currently. It would be useful to include it as an aid to curriculum development.
- Both applications can be deployed over SSL to achieve better security [2]

VI. ACKNOWLEDGEMENT

I thank my committee chair Dr. Timothy Lindquist for giving his invaluable guidance and advice throughout this work. The various suggestions provided by the committee members Dr. Kevin Gary and Dr. Darryl Morrell have added success to this project. A special thanks is also extended to Dr. Ben Huey who offered his invaluable inputs while the scope was defined for this project. I also extend my thanks to Ramon Anguamea for his help in the deployment of this project in the department server.

VII. REFERENCES

[1] Gopalakrishnan, A,EPortfolio Management Software. Dec 2010
[2] Sampath Kumaran, V, Web-based Course Information System supporting Accreditation using ASP.NET, Dec 2006
[3] ASP.NET – Wikipedia. Available: http://en.wikipedia.org/wiki/ASP.NET
[4] C Sharp – Wikipedia. Available: http://en.wikipedia.org/wiki/C_Sharp_(programming_language)
[5] PHP Manual. Available: http://www.php.net/manual/en/intro-whatis.php
[6] PHP Introduction. Available: http://www.w3schools.com/php/php_intro.asp
[7] ADO.NET – Wikipedia. Available: http://en.wikipedia.org/wiki/ADO.NET
[8] Overview of ADO.NET. Available: http://msdn.microsoft.com/en-us/library/h43ks021(v=vs.71).aspx
[9] ADODB Database Abstraction Library for PHP. Available: http://adodb.sourceforge.net/
[10] ADODB Manual. Available: http://phplens.com/lens/adodb/docs-adodb.htm#intro
[11] McGourty, J., Shuman, L., Besterfield-Sacre, M., Hoare, R.,"Using Technology to enhance Outcomes Assessment in Engineering Education", presented at FIE Conference, Reno, NV, Oct10-13, 2001.
[12] Walcerz, D.B., " EnableOA: A Software-Driven Outcomes Assessment Process Consistent with the Principles of Good Practice for Assessing Student Learning", presented at ASEE Mid-Atlantic Conference, NJ, April 17, 1999.
[13] Outcomes Assessment Solutions, "TrueOutcomes". Available: http://www.trueoutcomes.com (Oct 16, 2006)
[14] Petrova, R., Tibrewal, A., Sobh, T., "An Electronic Web-Based Assessment System " , presented at ASEE Annual Conference, Salt Lake city, Utah, June 2004.
[15] CourseWork – Stanford. Available: https://coursework.stanford.edu/portal/
[16] Stellar – MIT. Available: http://stellar.mit.edu/
[17] Pallapu, S., "Automating Outcomes Based Assessment", presented at ACM conference, Island of Hawaii, Jan 2005.
[18] Alzubi, O., "BlackBoard and True Outcomes Integration Software", Arizona State University East, 2005.
[19] Desire2Learn – Wikipedia. Available: http://en.wikipedia.org/wiki/Desire2Learn
[20] Bring Courses to Life with Desire2Learn. Available: http://www.desire2learn.com/services/content/courseware/
[21] Digication e-portfolios: Highered. Available: http://www.digication.com/highered
[22] Foliotek – Wikipedia. Available: http://en.wikipedia.org/wiki/Foliotek_(software)
[23] MAPS – Wikipedia. Available: http://en.wikipedia.org/wiki/MAPS_(software)
[24] Pass-port – Wikipedia. Available: http://en.wikipedia.org/wiki/Pass-port
[25] PebblePAD – Wikipedia. Available: http://en.wikipedia.org/wiki/PebblePAD_(software)
[26] A Collaborative Learning Community. Available: http://www.rcampus.com/
[27] Mono Web Site. Available: http://www.mono-project.com/Main_Page

APPENDIX:

Creating a virtual directory and setting up an alias in httpd.conf for CISA .
Sample httpd.conf file content:

```
Include /etc/apache2/conf.d/mod_mono.conf

Alias /capisa "/yourlocaldirectory/CISA/deploy"

MonoApplications "/capisa:/yourlocaldirectory/CISA/deploy"

<Directory "/yourlocaldirectory/CISA/deploy">
    SetHandler monio
    Options Indexes FollowSymlinks
    Order allow,deny
    Allow from all
    DirectoryIndex default.htm default.html default.aspx
</Directory>
```

Setting up the CISA configuration in Web.Config

```
<appSettings>
   <add key="connectionString" value="Server=129.219.40.31; Database=capisa; User
ID=capisausername;Password=password;"/>
</appSettings>
```

Readme file for CISA installation (includes database setup)

```
Masters Project
Web-Based Course Information System Supporting Accreditation (CISA)
Author: Vidhya Sampath Kumaran, Shashank Balasubramanian
Version:1.0 Nov 2006
Chair of Committee: Dr. Timothy Lindquist

Software Used & their References:
*************************
1) Visual Studio .NET 2005 (from DCST, ASU)
          online MSDN documentation:http://msdn2.microsoft.com/en-
us/library/ms229284(vs.80).aspx

2) MySql data provider for MySQL
        (libmySQL.dll, MySql.Data.dll,Mono.Security.Protocol.Tss.dll )
  http://www.mysql.com/

3) MySQL server 5.1.36
  http://www.mysql.com/

4) Internet Information Server 6.0 (IIS)(with ASP.NET2.0 extensions installed)
         Deploying ASP.NET application in IIS

http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/2f45f2ea-
bcf8-4d23-9f15-5caf7ebdc2ef.mspx?mfr=true
   Registering ASP.NET on IIS after installing .net framework
         http://www.devx.com/vb2themax/Tip/18849

5) ASP.NET Development Server webdev 8.0 (From Visual Studio .net 2005)

6) Ant 1.6.5
```

http://ant.apache.org/

Instructions to compile and run the CISA application
****
I. Creating and Populating the Database

1.Create a user for this database.
CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';
The username is username and the password is password

2. Drop existing database :
drop DATABASE IF EXISTS capisa

3. Create a new database 'capisa'
CREATE DATABASE IF NOT EXISTS capisa

4. Connect to the database
mysql -u username -h localhost -p capisa
(password:password)

5.To populate the database
Run the dump database script placed in the root of the directory.
mysql -u username -p capisa < location of dump file

6.To Create a backup of the database
mysqldump -h hostname -u username --password=password capisa > capisadump.sql

Additional Information to populate the database
_____
To create database from scratch from database ddl file instead of using dump script follow the instructions below:

Go to database folder and execute the following script in order
mysqldump -h hostname -u username --password=password capisa < databaseddl.sql
mysqldump -h hostname -u username --password=password capisa < dept-users.sql

This will create the database tables and populate with users values.
Now you can login to the application (ex:user: admin@asu.edu) and add data.

If you want to populate the database from dml scripts with with sample data please see the
instructions given in databaseuser.txt in "database\additional dml scripts" folder.

II. Deploying the application

1.Replace User ID, Password, Database, Server for connecting to database in "Web.config" located in deploy directory
deploy\Web.config -(under appsettings key="connectionString")(Make sure that only one connectionString is uncommented at any time)

2. To compile and run the application in ASP.NET webdev server go to command prompt
ant deploy
This will deploy the webapplication in webdev server (please change the virtual path and physical of webdev server if default one is not correct)

3.Public View: http://localhost:8090/public/Default.aspx
Secured View: http://localhost:8090/Admin/AdminDefault.aspx
Example credentials To login: userid: admin@asu.edu

password: dcstadmin

4. To run the application in IIS.
   Go to command prompt: ant build (to compile the source files)
   Create a virtual directory (ex: cisa) in IIS pointing to the deploy directory
   Access the application:
   public view: http://hostname/cisa/public/Default.aspx
   Authenticated view: http://hostname/cisa/Admin/AdminDefault.aspx

Additional Information

_____

To deploy the application in a production environment like IIS Server running on Windows Server 2003,
simply create a virtual directory on server machine and copy the contents of CISA deploy directory to the server.
Then register the virtual directory in IIS.
(CISA application is currently deployed on ASU server using this method. You can access the deployed application
using the URL: http://technology.poly.asu.edu/capisa/public/Default.aspx)

5. To connect to the database running on ASU dcstserver instead of "localhost":
   Uncomment the following ConnectionString in web.config file:
   <add key="connectionString" value="Server=129.219.77.20;Database=capisa;User ID=xxx;Password=xxx"/>

Additional Requirements
****
1. To send email from the application you have to connect to the ASU network (wired or wireless) or use ASU VPN.
   You can download the ASU VPN software from
http://www.asu.edu/security/software/html/vpn/index.htm

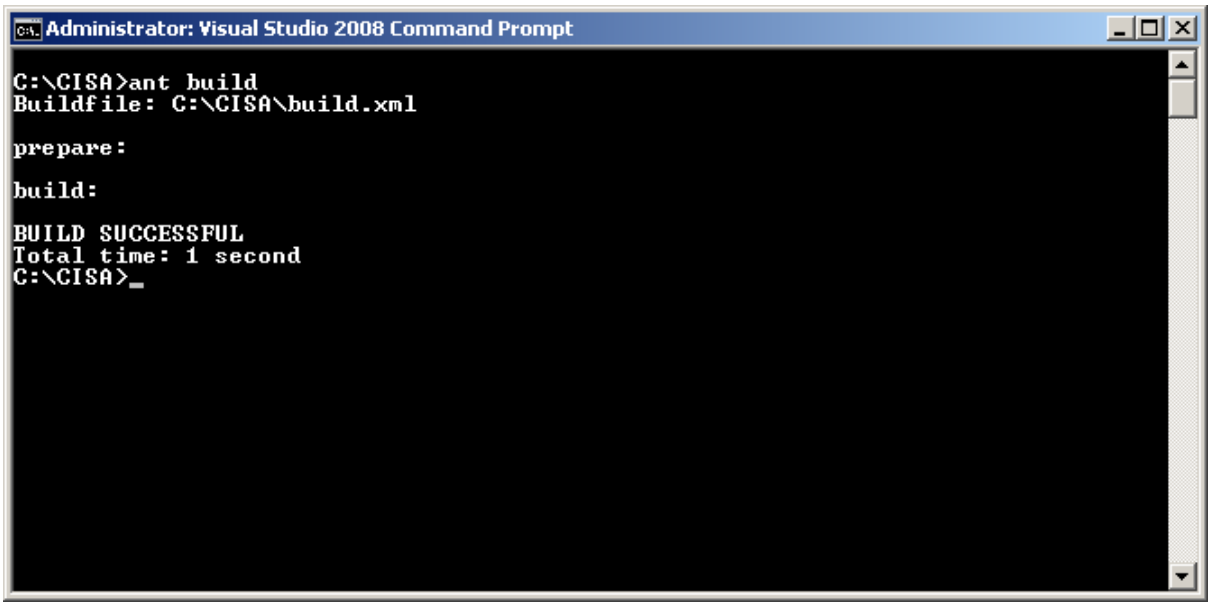
2. Please refer to CISA User Manual to see the instructions on entering data in Secured CISA

Possible Errors and Solutions
****
1. Application unable to send email.
   Solution: Try after connecting to ASU VPN. Also try disabling 'Virus Scan' running on your machine)

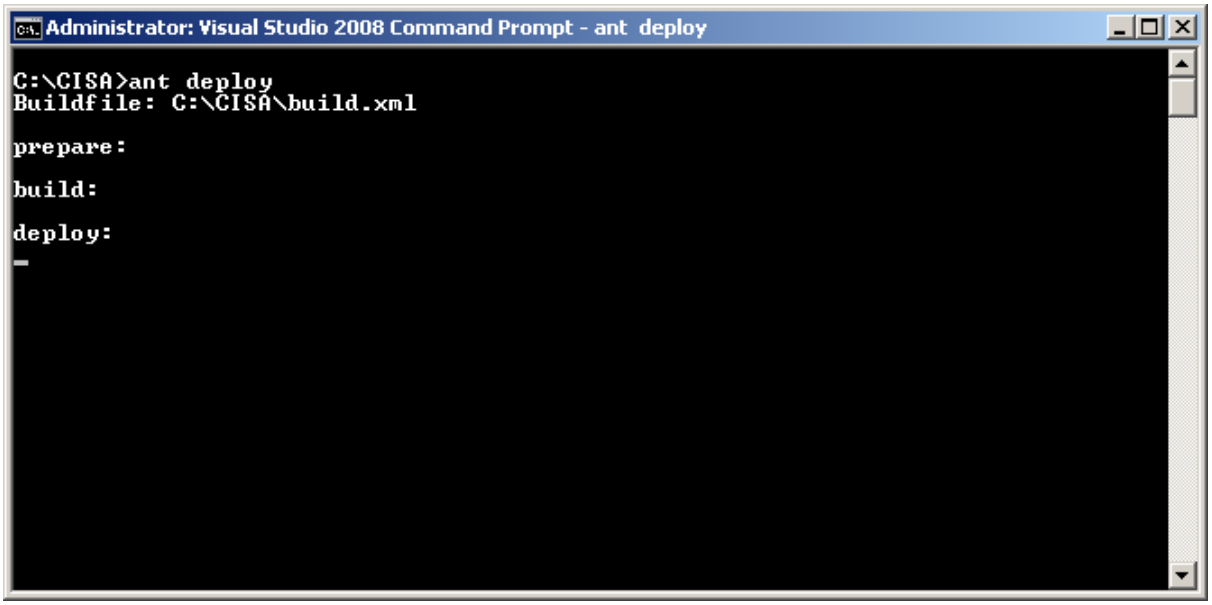
2. TreeView not displayed on public view or Unable to login to application even after entering correct credentials
   Solution: This problem occurs if the CISA application is unable to connect to the database (connection attempt failed)
   a. Check the "connectionString" in web.config for username, password, database and localhost.
      Ensure that only one "connectionstring" information is uncommented
   b.Check if the database server is accepting connections from outside. Ensure that Firewall port is open
     to accept connections (Postgresql port should be open in Firewall)

3. Unable to access the application in IIS
   Solution: Ensure that ASP.NET 2.0 extensions are properly installed in IIS (otherwise reinstall the extensions)
   Then refer to II-4 to deploy the application properly in IIS

Build the CISA application: CISA uses ANT to build its application. Open a terminal and navigate to the CISA home directory and fire the `ant build` *command to build the file.*



Deploy the CISA application: CISA uses ANT to deploy its application. Open a terminal and navigate to the CISA home directory and fire the `ant deploy` *command to build the file.*



Readme file for Mahara installation (includes database setup)

```
Software Used & their References:
**************************
1)  PHP
  http://www.php.net

2)  MySql data provider for MySQL (libmySQL.dll)
  http://www.mysql.com/

3)  MySQL server 5.1.36
  http://www.mysql.com/

4)  Apache2 server
  http://www.apache.org

Instructions for installing the e-portfolio system on a local setup

1. Copy the mahara directory under the root folder of apache

2. Copy the data folder to a local directory

3. Create a database mahara and import the SQL dump file maharadump.sql
   a.Create a user for this database.
   CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';

   b. Drop existing database :
      drop DATABASE IF EXISTS mahara

   c. Create a new database 'mahara'
      CREATE DATABASE IF NOT EXISTS mahara

   d. Connect to the database
    mysql -u username -h localhost -p mahara
      (password:password)

   e.To populate the database
     Run the dump database script placed in the root of the directory.
```

```
   mysql -u username -p mahara < location of dump file

   f.To Create a backup of the database
     mysqldump -h hostname -u username --password=password mahara > maharadump.sql
```

4. Change the config.php to reflect the username and password for your mysql instance and the location of the data directory

5. Once the database has been loaded, start apache and fire http://localhost/mahara

6. The system will prompt you for a username and password
   The credentials for the admin user are as follows:
   Username: admin
   Password: adminuser1

7. You can now login to the system and run the application

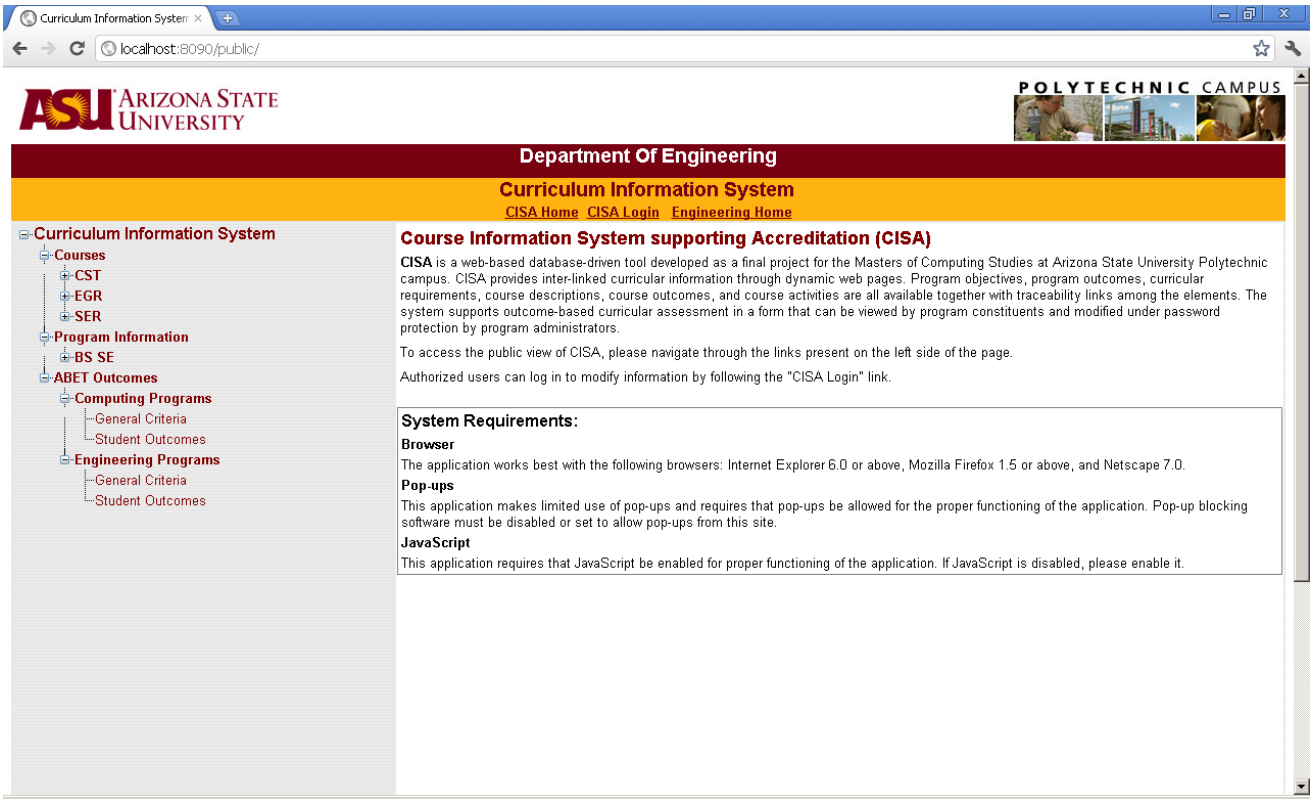
Setting up the mahara configuration in config.php

```
$cfg->dbtype   = 'mysql5';
$cfg->dbhost   = 'localhost';
$cfg->dbport   = null;
$cfg->dbname   = 'mahara';
$cfg->dbname_cap   = 'capisa';
$cfg->dbuser   = 'capisausername'; // you may choose one or two user accounts
$cfg->dbuser2   = 'maharausername';
$cfg->dbpass   = 'password'; // maybe same or different password

$cfg->wwwroot = 'location of the root folder';

$cfg->dataroot = 'location of the data folder';
```

Welcome Screen for CAPISA:



Welcome screen for MAHARA: