'Kaseya University' is a partnership between FIU IT-Scholars and Kaseya to provide clients with educational workshops and certification programs. During the rollout of this program, two separate instances of Moodle were created and maintained, one by IT-Scholars and the other by Kaseya. These two systems are not integrated and were created differently within different Moodle versions.

There are numerous issues with this design:

- Users have to create and manage two user accounts
- There is general confusion about how to navigate between the two systems
- Users have to login twice to get to their desired service, often with different password (due to different password requirements)
- Users have to sign up for workshops on one system, roam to the other site (IT-Scholars) and sign in with a different account, to then be able to actually enroll and participate in those workshops.

Kaseya’s Moodle is stock and has little to no modifications or external plugins. IT-Scholars’ Moodle is heavily modified and is running a number of plugins, some of which rely on external systems. Some of these external systems have specific database constraints. In light of this, most of the modifications for user data requirements (passwords & fields) would have to be changed on Kaseya’s Moodle in order to meet the requirements of the IT-Scholars system.

The two Moodle sites should communicate in the background so that when Kaseya users roam over to the IT-Scholars site, the external services are already configured and require no extra user interaction.

Two Virtual Machines were created and configured to emulate the environments that are hosting the two Moodle sites currently in production.

A bottom-up approach to testing was used for this project. Numerous unit tests and integration tests were manually conducted, until the system was complete, at which point system testing was conducted.

All implemented use cases were tested thoroughly to ensure they functioned as intended.

**Requirements**

- Create a single sign on, so users only have to log in once.
- Establish a single standard for passwords.
- Standardize extra profile fields between the two systems.
- Establish a backend so that user information and changes can be synchronized between the two systems.

**System Design**

Kaseya’s Moodle is stock and has little to no modifications or external plugins. IT-Scholars’ Moodle is heavily modified and is running a number of plugins, some of which rely on external systems. Some of these external systems have specific database constraints. In light of this, most of the modifications for user data requirements (passwords & fields) would have to be changed on Kaseya’s Moodle in order to meet the requirements of the IT-Scholars system.

The two Moodle sites should communicate in the background so that when Kaseya users roam over to the IT-Scholars site, the external services are already configured and require no extra user interaction.

Two Virtual Machines were created and configured to emulate the environments that are hosting the two Moodle sites currently in production.

A bottom-up approach to testing was used for this project. Numerous unit tests and integration tests were manually conducted, until the system was complete, at which point system testing was conducted.

All implemented use cases were tested thoroughly to ensure they functioned as intended.

**Current System**

- Users have to create and manage two user accounts
- There is general confusion about how to navigate between the two systems
- Users have to login twice to get to their desired service, often with different password (due to different password requirements)

**Object Design**

Class diagram of the user and profile information that is used to keep accounts synced across the two systems.

**Implementation**

Implemented Use Cases

After some research, it was discovered that there was an internal moodle plugin called 'Moodle Networking' which allows for two distinct moodle sites to communicate and share information. The functions that this plugin provided were invaluable, and provided much of the framework for the single signon.

Moodle to Moodle communications are conducted across the web, using cURL and XMLRPC as the protocol. This information is kept secure by the use of 128-bit public-key encryption.

**Verification**

Two Virtual Machines were created and configured to emulate the environments that are hosting the two Moodle sites currently in production.

A bottom-up approach to testing was used for this project. Numerous unit tests and integration tests were manually conducted, until the system was complete, at which point system testing was conducted.

All implemented use cases were tested thoroughly to ensure they functioned as intended.

**Screenshots**

Test Kaseya User is logged in

- The user is logged in to an external database (Kaseya Moodle)

Test IT-Scholars

- The user is logged in to an external database (IT-Scholars Moodle)

User is redirected and logged in automatically

**Summary**

The intention of this project was to provide consistency between the two Moodle systems while also allowing Kaseya clients to roam to the IT-Scholars site without having to log in twice. All these features have been implemented in such a way as to provide the most seamless transition possible.

As implemented, Kaseya’s customers log into Kaseya’s Moodle and click the link to the IT-Scholars site without having to log in twice. All these features have been implemented in such a way as to provide the most seamless transition possible.

Acknowledgement

The material presented in this poster is based upon the work supported by Prof. Masoud Sadjadi and Industry Advisor Juan Caraballo. I am thankful for the help that I received from my group member xxxxx xxxx.