

# LOT o' Learning

Valerie German, Cody Winters, Pierce Bringhurst

## **Abstract**

Our project continued the Learning Outcome Tracker. Administrators are now able to remove users on the role change page. Courses may be assigned multiple instructors and can be given a status and due date for their learning outcome progress. Instructors may view courses that have been archived to get a course history. Users of the Learning Outcome Tracker can now message each other. Instructors can now add or remove Evaluation metrics and the assignment files associated with them. These features allow users to complete metrics for learning outcomes and communicate better with other users on the status of each course's completion.

## **URLs:**

Valerie: <http://ec2-34-228-242-182.compute-1.amazonaws.com/>

Cody: <http://ec2-52-91-216-212.compute-1.amazonaws.com/>

## **Introduction**

Our project expanded the learning outcome tracker, providing some helpful and required functionality.

We finished the implied aspects of the project, including instructor assignment to courses and the ability to create, edit, view and delete courses, evaluation metrics and sample files. This also included file uploading and downloading capabilities. Instructors can associate an assignment file with evaluation metrics and sample files. Departments are also able to be created, viewed, edited and deleted, each course must belong to a department. Deleting departments shows an alert and prevents deletion if the department contains courses. Administrators are also now able to remove users.

An additional feature available is notifications and course history. Administrators may create a course with any status, typically courses will begin 'in-progress' meaning the instructor is working on the course. Each course can be assigned a due date for the instructor to complete the evaluation metrics and sample files. Once the instructor is finished they may submit the course for approval. This sets the course's status to 'awaiting approval' and notifies the department chairs. From here, the department chair may approve the course, setting its status to

‘complete’ or set the course to ‘in-review’. When a chair sets the ‘in-review’ status they may send a message along with the notification to the instructor that changes need to be made before the course is approved. Instructors may make changes then resubmit for approval. Additionally, a department chair may set courses to ‘archived’. Archived courses cannot be edited by any instructor, and they may be viewed by any instructor. From a ‘Course History’ link, instructors may view any archived courses as examples. Any changes to the course status involves a confirmation and success or failure alert. Course status updates on course views use ajax calls and update the web page to display the new status.

Users can now message each other using a small messaging box which appears in the bottom right-hand corner of the screen when the user is logged in. This was implemented using SignalR to provide real-time responses. A ‘Messages’ table was added into the database to accommodate this functionality. The messaging and contact list boxes can be hidden or slid down out of view. This will be useful for users as it will provide more real time and immediate cooperation between Administrators, Chairs, and Instructors while fulfilling learning outcome requirements.

Index views are now searchable by authorized users, and these views are also paginated and sortable. Pagination allows the users to view only a limited number of items at a time, instead of every viewable result. The user has the option to set how many items they would like to view per page of results. These changes were made to allow the user to navigate more quickly through their accessible items in the database.

Finally, the style of the project was updated to give it a more modern feel. Initially, navigation took place in a large part through text hyperlinks. Now, buttons are used where appropriate. Additionally, some elements were combined so as to fit with modern convention and to avoid confusion.

#### Feature Table

Feature Name	Scope	Primary Programmer	Time Spent (Hours)	File/Function	Lines of Code
--------------	-------	--------------------	-----------------------	---------------	---------------

Database Updates for new features	DB	Valerie	3	Models/DB	2000
CRUD for Courses, Multiple instructors for course	Back-end, UI	Valerie	4	Controllers/CourseController	1000
Department CRUD, Department code updates for Chairs	Back-end, UI	Valerie	4.5	Controllers/DepartmentController, Controllers/DeptManagerController, Views/Department/..., Views/DeptManager/...	1500
Sample Files CRUD, File uploading/download ing, Database File Seeding	Back-End, UI, DB	Valerie	4	Controllers/SampleFilesController, Views/Instructor/...	700
User Deleting	Back-End, UI	Valerie	1	Controllers/UserController, Views/User/Index	100
Notifications, Course Statuses, Course History	Back-End, UI	Valerie	14	Controllers/InstructorController, Controller/SampleFilesController, Controllers/DepartmentController, Models/Notification, Views/Home/Notifications, Views/Instructor/..., Views/Department/...	2000
Messaging	Back-End, UI	Pierce	25	Site.js Layout.cshtml Site.css Controllers/MessagesController	4000
CRUD Evaluation Metrics	Back-end, UI, DB	Pierce	5	Controllers/EvaluationMetricsController Views/Instructor/Course.cshtml Site.js File.js Views/Instructor/EvaluationMetric..	2385

Search	Back-end, UI	Cody	5	Controllers/CourseController Controllers/LearningOutcomeController Controllers/InstructorController Views/ Course/Index Views/Instructor/ArchivedCourses View/LearningOutcomes/Index	~2000
Pagination	Back-end, UI	Cody	6	PaginatedList Controllers/CourseController Controllers/LearningOutcomeController Controllers/InstructorController Views/ Course/Index Views/Instructor/ArchivedCourses View/LearningOutcomes/Index	~1200
Style	UI, back-end	Cody	15+	Made changes to almost every view, and several controllers. Had to modify what types of objects were being sent to the views, and what database elements were being selected.	~1200

### Individual Contribution

Team Member	Time Spent on Project	Lines of Code Committed
<i>Valerie German</i>	36 hours	7,356
<i>Cody Winters</i>	26 hours	~4400
<i>Pierce Bringham</i>	30 hours	6,385

### Summary

Our team performed at a Superior level, we each contributed useful features toward the project and dedicated time toward testing our and each other's code. Everyone worked on

separate branches in our repository and we regularly merged onto a shared development branch. Our project completed the necessary functionality for a usable learning outcome tracker and provided some helpful tools such as searching, messaging and notifications. We used many of the topics taught during this course, including CSS, JavaScript, AJAX, jQuery, Database Migrations, Model-View-Controller architecture and other functional/design concepts.